J. Symbolic Computation (1996) **11**, 1–000

Bi-rewrite Systems†

JORDI LEVY‡ AND JAUME AGUSTÍ§

‡*Departament de Llenguatges i Sistemes Informàtics*

Universitat Politècnica de Catalunya

§*Institut d'Investigació en Intel·ligència Artificial*

Consejo Superior de Investigaciones Científicas

(Received)

In this article we propose an extension of term rewriting techniques to automate the deduction in monotone pre-order theories. To prove an inclusion $a \subseteq b$ from a given set I of them, we generate from I, using a completion procedure, a *bi-rewrite system* $\langle R_\subseteq, R_\supseteq \rangle$, that is, a pair of rewrite relations $\xrightarrow[R_\subseteq]{}$ and $\xrightarrow[R_\supseteq]{}$, and seek a common term c such that $a \xrightarrow[R_\subseteq]{*} c$ and $b \xrightarrow[R_\supseteq]{*} c$. Each component of the bi-rewrite system $\xrightarrow[R_\subseteq]{}$ and $\xrightarrow[R_\supseteq]{}$ is allowed to be a subset of the corresponding inclusion relation $\subseteq$ or $\supseteq$ defined by the theory of I. In order to assure the decidability and completeness of such proof procedure we study the termination and commutation of $\xrightarrow[R_\subseteq]{}$ and $\xrightarrow[R_\supseteq]{}$. The proof of the commutation property is based on a critical pair lemma, using an *extended* definition of critical pair. We also extend the existing techniques of rewriting modulo equalities to bi-rewriting modulo a set of inclusions. Although we center our attention on the completion process à la Knuth-Bendix, the same notion of extended critical pair is suitable of being applied to the so called unfailing completion procedures. The completion process is illustrated by means of an example corresponding to the theory of the union operator. We show that confluence of *extended* critical pairs may be ensured adding *rule schemes*. Such rule schemes contain variables denoting schemes of expressions, instead of expressions. We propose the use of the *linear second-order typed λ-calculus* to *codify* these expression schemes. Although the general second-order unification problem is only semi-decidable, the second-order unification problems we need to solve during the completion process are decidable.

1. Introduction

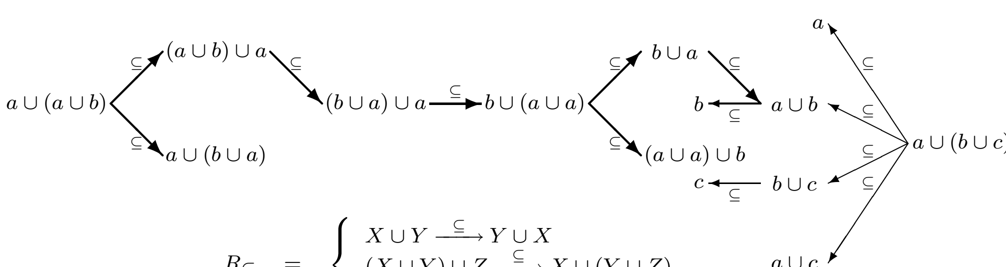
Rewrite systems are usually associated with rewriting on equivalence classes of terms, defined by a set of equations. However term rewriting techniques may be used to compute other relations than congruence. Particularly interesting are non-symmetric relations like pre-orders. In this article we will show the applicability of rewrite techniques to monotonic

† This work was partially supported by the project DISCOR (TIC 94-0847-C02-01) funded by the CICYT.

0747–7171/90/000000 + 00 $03.00/0

pre-order relations on terms, that is the deduction of inequalities —here we call them inclusions— from a given set of them.

The idea of applying rewrite techniques to the deduction of inclusions between terms, like $a \subseteq b$, is very simple. We compute by repeatedly replacing both 1) subterms of a by "bigger" terms using the axioms and 2) subterms of b by "smaller" terms using the same axioms until a path is found between a and b. Evidently there are many paths starting from a in the direction $\xrightarrow{\subseteq}$ and from b in the direction $\xrightarrow{\supseteq}$ (see figure 1). Many of them are blind alleys and others are not terminating. It is essential that the search procedure avoids infinite sequences of rewrite steps with infinitely many different terms (infinite paths due to cycles can be avoided if we control the introduction of repeated terms). Obviously infinitely many different rewrite steps would prevent the termination of the procedure. The solution to non-termination is, like in term rewriting systems, to orient the axioms using a well founded ordering on terms. Because the relation is non-symmetric, the orientation results in a pair of rewrite systems $\langle R_{\subseteq}, R_{\supseteq}\rangle$, i.e. we get what we call a bi-rewrite system. We introduce the definitions of Church-Rosser and quasi-terminating bi-rewrite system in order to assure the soundness, completeness and termination of the search procedure. That is, given a set of axioms, if we can orient and complete them obtaining a quasi-terminating and Church-Rosser bi-rewrite system, then we will have a decision algorithm to test $a \subseteq b$.



$$R_{\subseteq} \quad = \quad \begin{cases} X \cup Y \xrightarrow{\subseteq} Y \cup X \\ (X \cup Y) \cup Z \xrightarrow{\subseteq} X \cup (Y \cup Z) \\ X \cup X \xrightarrow{\subseteq} X \end{cases}$$

$$R_{\supseteq} \quad = \quad \begin{cases} X \cup Y \xrightarrow{\supseteq} X \\ X \cup Y \xrightarrow{\supseteq} Y \end{cases}$$

Figure 1. A graphical representation of the bi-rewrite algorithm

Most of the notions of rewriting developed for the equational case can be extended to bi-rewriting and the development of the article follows the same pattern as equational rewriting: the Church-Rosser property is proved by means of a critical pair lemma, and we use a completion process to ensure the confluence of the critical pairs (Knuth and Bendix, 1970; Huet, 1980; Klop, 1987; Dershowitz and Jouannaud, 1990). However there are also some differences. Equational rewriting is in essence a theory of normal forms, while bi-rewriting disregards this notion. Bi-rewriting can also be seen as a generalization of equational rewriting: equations can be translated to pairs of inclusions and then we can reproduce the equational case. One of the costs of this generalization is that bi-rewriting is based on a search procedure, which is avoided in canonical rewrite systems thanks to the existence of unique normal forms. Another cost is that now critical pairs must be computed considering variable overlapping, producing possibly infinitely many of them, which are represented as critical pair schemes.

This article proceeds as follows.

In section 2 we present a version of the critical pair lemma for bi-rewrite systems using an *extended* definition of critical pairs. We also give a counter-example that invalidates this lemma stated only in terms of *standard* critical pairs.

In section 3 we generalize the results of section 2 to bi-rewrite systems modulo a set of (non-orientable) inclusions. We have divided this section in two subsections, the first devoted to abstract bi-rewrite properties and the second to term dependent properties.

In section 4 we present an example of canonical bi-rewrite system for the theory of non-distributive lattices. We show that although in general extended critical pairs could be intractable, there exist for this theory, and possibly for others, practical ways to handle them.

We also show in section 5 some of the disadvantages of using equations to model inclusions in lattice theories.

Unfortunately, the set of extended critical pairs is in general infinite. Although there exists canonical bi-rewrite systems for many inclusion theories, the standard completion procedures are of little practical help to automatically complete a bi-rewrite system. In section 6 we show how these infinitely many extended critical pairs can be made confluent introducing rule schemes, where these rule schemes can be *implemented* using second-order rules. However, the use of the simply typed second-order λ-calculus for rewriting purposes introduces some problems, stated in subsection 6.1. Because of that, we define a restricted second-order language called *linear second-order λ-calculus*, which is described in section 7. There we also describe a unification procedure for the linear second-order typed λ-calculus.

Then the new critical pair lemma for second-order bi-rewrite systems is proved in section 8.

We illustrate how the Knuth-Bendix completion procedure could be implemented for second-order bi-rewrite systems by means of an example in section 9.

In section 10 we present related work and in section 11 we conclude summarizing present and further work.

2. Inclusions and Bi-rewrite Systems

If nothing is said, we follow the notation and the standard definitions used in (Huet, 1980; Klop, 1987; Dershowitz and Jouannaud, 1990). We are concerned with first-order terms $\mathcal{T}(\mathcal{F},\mathcal{X})$ over a nonempty *signature* $\mathcal{F} = \bigcup_{n\in\mathbb{N}}\mathcal{F}_n$ of function symbols, and a denumerable set $\mathcal{X}$ of variables.[†] The *set of variables* of a term t is denoted by $\mathcal{FV}(t)$. A *position* p is a sequence of positive integers. Given two positions, $p_1 \cdot p_2$ denotes their concatenation. We write $p_1 \prec p_2$ when p_1 is a prefix of p_2 and $p_1|p_2$ when they are disjoint. The *occurrence* of a subexpression at a position p of a term t is denoted by $t|_p$. The expression $t[u]_p$ denotes the result of replacing in t the occurrence of $t|_p$ by

[†] As we will see later, in most cases we also require the finiteness of $\mathcal{F}$. We suppose that $\mathcal{F}_n$ are disjoint sets. The set $\mathcal{T}(\mathcal{F},\mathcal{X})$ is defined as the smallest set containing $\mathcal{X}$ such that if $f\in\mathcal{F}_n$ and $t_i\in\mathcal{T}(\mathcal{F},\mathcal{X})$ for $i=1,\ldots,n$ then $f(t_1,\ldots,t_n)\in\mathcal{T}(\mathcal{F},\mathcal{X})$.

u. A *context* $F[\cdot]_p$ is an expression with a *hole* $[\cdot]$ at a distinguished position p.[‡] A *substitution* $\sigma = [X_1 \mapsto t_1, \ldots, X_n \mapsto t_n]$ is a mapping from a finite set $\{X_1, \ldots, X_n\} \subseteq \mathcal{X}$ of variables to $\mathcal{T}(\mathcal{F}, \mathcal{X})$, extended as a morphism to $\mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathcal{T}(\mathcal{F}, \mathcal{X})$. The set $\mathcal{D}om(\sigma) \stackrel{def}{=} \{X_1, \ldots, X_n\}$ is called the *domain* of the substitution.

We use the *relational logic* notation (de Kogel, 1992; Bäumer, 1992) to present the abstract bi-rewriting properties. The inverse of the relation R is denoted by R^{-1}, its reflexive-transitive closure by R^*, the transitive composition by $R_1 \circ R_2$, the union by $R_1 \cup R_2$, and the intersection by $R_1 \cap R_2$. Notation R^+ is a shorthand for $R \circ R^*$. A relation R is said to be *terminating* if R^+ is a well-founded ordering, *quasi-terminating* if the set $\{u \mid t\ R^*\ u\}$ is finite for any value t; and *finitely branching* if $\{u \mid t\ R\ u\}$ is finite for any t. A binary relation R on terms is said to be *closed under substitutions* if $t\ R\ u$ implies $\sigma(t)\ R\ \sigma(u)$, for any substitution σ and pair of terms t and u; *monotonic* if $t\ R\ u$ implies $F[t]_p\ R\ F[u]_p$, for any context $F[\cdot]_p$; and a *rewrite relation* if it is closed under substitutions and monotonic. We denote by $\xrightarrow[R]{}$ the rewrite relation defined by the set of rules R.[†] Notation $\xleftarrow[R]{}$ is a shorthand for $(\xrightarrow[R]{})^{-1}$.

An *inclusion* is a pair of terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ written $s \subseteq t$. Given a finite set of inclusions Ax and a pair of terms s and t, we say that $s \subseteq_{Ax} t$ iff $Ax \vdash_{POL} s \subseteq t$, where *POL* stands for *Pre-Order Logic* and $\vdash_{POL}$ is the entailment relation defined by the following inference rules

$$\frac{}{\Delta, s \subseteq t \vdash_{POL} s \subseteq t} \qquad \frac{}{\Delta \vdash_{POL} s \subseteq s} \qquad \frac{\Delta \vdash_{POL} s \subseteq t \quad \Delta \vdash_{POL} t \subseteq u}{\Delta \vdash_{POL} s \subseteq u}$$

$$\frac{\Delta \vdash_{POL} s \subseteq t}{\Delta \vdash_{POL} \sigma(s) \subseteq \sigma(t)} \qquad \frac{\Delta \vdash_{POL} s \subseteq t}{\Delta \vdash_{POL} u[s]_p \subseteq u[t]_p}$$

where σ is a substitution, p a position in u, i.e. $u[\cdot]_p$ is a context, and Δ is a finite set of inclusions.

Meseguer (1990), Meseguer (1992) has studied widely the logic of conditional inequalities, which he names *rewriting logic*, and its models.

The set of inclusions $s \subseteq t$ that can be inferred from Ax using $\vdash_{POL}$ forms an inclusion theory, noted by $Th(Ax)$. Notice that, in first-order logic, $Th(Ax)$ is a denumerable set and the deduction problem $Ax \vdash_{POL} s \subseteq t$ is semi-decidable. In the following we will propose sufficient conditions to have a decision algorithm for $Ax \vdash_{POL} s \subseteq t$ based on rewrite techniques.

Given an inclusion $s \subseteq t$ of Ax, we can orient it obtaining either a term rewriting rule $s \xrightarrow{\subseteq} t$ or a rule $t \xrightarrow{\supseteq} s$. Thus, the orientation, for rewriting purposes, of a finite set of inclusions Ax results in two sets of rewrite rules, $R_\subseteq$ with rules like $s \xrightarrow{\subseteq} t$ and $R_\supseteq$ with rules like $s \xrightarrow{\supseteq} t$. The pair $\langle R_\subseteq, R_\supseteq \rangle$ is called a bi-rewrite system.

DEFINITION 2.1. *A* **(term) bi-rewriting system** *is a pair* $\langle R_\subseteq, R_\supseteq \rangle$ *of finite sets of*

‡ We write $p_1 \prec p_2$ when there exists a sequence q such that $p_2 = p_1 \cdot q$, and $p_1 | p_2$ when $p_1 \not\prec p_2$ and $p_2 \not\prec p_1$. If p is an empty sequence then $t|_p$ is defined by $t|_{<>} \stackrel{def}{=} t$ otherwise it is defined inductively by $f(t_1, \ldots, t_n)|_{<i_1, i_2, \ldots, i_r>} \stackrel{def}{=} t_{i_1}|_{<i_2, \ldots, i_r>}$. If p is the empty sequence then $t[u]_{<>} \stackrel{def}{=} u$, otherwise $f(t_1, \ldots, t_n)[u]_{<i_1, \ldots, i_m>} \stackrel{def}{=} f(t_1, \ldots, t_{i_1}[u]_{<i_2, \ldots, i_m>}, \ldots, t_n)$.

† The minimal rewrite relation satisfying $s \xrightarrow[R]{} t$ for any rule $s \to t \in R$.

(term) rewriting rules

$$R_{\subseteq} = \{s_1 \xrightarrow{\subseteq} t_1, \ldots, s_n \xrightarrow{\subseteq} t_n\}$$
$$R_{\supseteq} = \{u_1 \xrightarrow{\supseteq} v_1, \ldots, u_m \xrightarrow{\supseteq} v_m\}$$

Given a bi-rewrite system $\langle R_{\subseteq}, R_{\supseteq}\rangle$*, its* **corresponding inclusion theory** *is defined by the set of axioms* $Ax = \{s \subseteq t \mid s \xrightarrow{\subseteq} t \in R_{\subseteq} \ \vee \ t \xrightarrow{\supseteq} s \in R_{\supseteq}\}$.

The orientation criteria is based, like in rewrite systems, on a well-founded ordering on terms (noted as $\succ$) (Dershowitz, 1987). In this section we suppose that each inclusion $s \subseteq t$ in Ax may be oriented, putting either $s \xrightarrow{\subseteq} t$ in $R_{\subseteq}$ if $s \succ t$, or $t \xrightarrow{\supseteq} s$ in $R_{\supseteq}$ if $t \succ s$. In the next section we will consider the case of inclusions which can not be oriented because $s \not\succ t$ and $t \not\succ s$. For example, inclusions defining the inclusion theory of the union may be oriented using a simplification ordering as it is shown in figure 2.

$$Ax = \begin{cases} X \cup X \subseteq X \\ X \subseteq X \cup Y \\ Y \subseteq X \cup Y \end{cases} \qquad R_{\subseteq} = \{\ r_1 : \quad X \cup X \xrightarrow{\subseteq} X \qquad R_{\supseteq} = \begin{cases} r_2 : & X \cup Y \xrightarrow{\supseteq} X \\ r_3 : & X \cup Y \xrightarrow{\supseteq} Y \end{cases}$$

Figure 2. Orientation of the inclusion theory of the union.

Given a bi-rewrite system $\langle R_{\subseteq}, R_{\supseteq}\rangle$ the monotonic and substitution closure of each one of its components $R_{\subseteq}$ and $R_{\supseteq}$ results in a pair of rewrite relations, noted by $\xrightarrow[R_{\subseteq}]{}$ and $\xrightarrow[R_{\supseteq}]{}$ respectively, defined as follows.

DEFINITION 2.2. *We say that* s R**-rewrites to** t*, written* $s \xrightarrow[R]{}_{[p,\sigma,l\to r]} t$*, or simply* $s \xrightarrow[R]{} t$ *when there is no confusion, if there exist a rule* $l \longrightarrow r \in R$*, a position* p *in* s*, and a substitution* σ*, such that* $s|_p = \sigma(l)$ *and* $t = s[\sigma(r)]_p$*.*

If $s|_p = \sigma(l)$ then we say that $s|_p$ and l match. Notice that if $\mathcal{FV}(r) \subseteq \mathcal{FV}(l)$ then the substitution σ in the previous definition, with its domain restricted to $\mathcal{D}om(\sigma) \subseteq \mathcal{FV}(l)$, is unique.

A variant of the theorem of Birkhoff (Birkhoff, 1935) allows to prove the following lemma.

LEMMA 2.3. *Given a bi-rewrite system* $\langle R_{\subseteq}, R_{\supseteq}\rangle$ *and its corresponding inclusion theory* Ax*, for any pair of terms* s, t *we have* $s\ (\xrightarrow[R_{\subseteq}]{} \cup \xleftarrow[R_{\supseteq}]{})^*\ t$ *if, and only if,* $Ax \vdash_{POL} s \subseteq t$*.*

However, the relation $(\xrightarrow[R_{\subseteq}]{} \cup \xleftarrow[R_{\supseteq}]{})^*$ is in general not computable, i.e. given two terms s and t there does not exist a decision algorithm for $s\ (\xrightarrow[R_{\subseteq}]{} \cup \xleftarrow[R_{\supseteq}]{})^*\ t$. We are interested in reducing the previous relation into the subrelation $\xrightarrow[R_{\subseteq}]{*} \circ \xleftarrow[R_{\supseteq}]{*}$, which we will show is computable.

Based on the bi-rewrite system $\langle R_{\subseteq}, R_{\supseteq}\rangle$ a deduction procedure for its corresponding inclusion theory $Th(Ax)$ can be easily defined (see figure 1). To prove $Ax \vdash_{POL} s \subseteq t$ the procedure enumerates recursively the nodes of two trees T_1 and T_2, defined by $\mathsf{root}_{T_1} = s$, $\mathsf{root}_{T_2} = t$, $\mathsf{branch}_{T_1}(s_1) = \{s_2 \mid s_1 \xrightarrow[R_{\subseteq}]{} s_2\}$ and $\mathsf{branch}_{T_2}(t_1) = \{t_2 \mid t_1 \xrightarrow[R_{\supseteq}]{} t_2\}$, avoiding repeated nodes. If the procedure finds a common node in both trees then it stops and answers *true*, otherwise if both sets of nodes are finite then it stops and answers *false* or else it does not stop.

Notice that the nodes of both trees are always recursively enumerable, although the trees may be infinitely branching. We say that a tree is infinitely branching if it contains a node with infinitely many branches.

The following definition states sufficient conditions for the soundness and completeness, and for the termination of this procedure. Notice that the soundness and completeness properties are based on the equivalence of the relation $\xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*}$ computed by the algorithm and the relation $(\xrightarrow[R_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^*$ implementing the inclusion relation defined by the theory. The termination property is based on the finiteness of both search trees.

DEFINITION 2.4. *A bi-rewrite system $\langle R_\subseteq, R_\supseteq \rangle$ is said to be*

(i) **terminating** *iff* $(\xrightarrow[R_\subseteq]{} \cup \xrightarrow[R_\supseteq]{})^*$ *is a well-founded ordering;*
(ii) **quasi-terminating** *or* **globally finite** *iff the sets* $\{u \mid t \xrightarrow[R_\subseteq]{*} u\}$ *and* $\{v \mid t \xrightarrow[R_\supseteq]{*} v\}$ *are both finite for any term t; and*
(iii) **Church-Rosser** *iff* $(\xrightarrow[R_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^* \subseteq \xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*}$.

In previous versions of this work (Levy and Agustí, 1993; Levy, 1994), a bi-rewrite system $\langle R_\subseteq, R_\supseteq \rangle$ is said to be terminating iff both $\xrightarrow[R_\subseteq]{*}$ and $\xrightarrow[R_\supseteq]{*}$ are well-founded orderings. This is a weaker condition and clearly it is not enough to prove later the equivalence between the Church-Rosser and the local bi-confluence properties. This error was communicated to the authors by Professor Harald Ganzinger.

We can prove the following results for the decision procedure based on a bi-rewrite system, and the $Ax \vdash_{POL} t \subseteq u$ deduction problem of its corresponding inclusion theory.

LEMMA 2.5. *If the bi-rewrite system $\langle R_\subseteq, R_\supseteq \rangle$ is Church-Rosser then the decision procedure based on it is sound and complete, i.e. $Ax \vdash_{POL} t \subseteq u$ holds if, and only if, the procedure terminates and answers* true.

If the bi-rewrite system is Church-Rosser and quasi-terminating then the decision procedure is sound, complete and terminates, therefore the satisfiability problem is decidable.

We only need to require the quasi-termination property of the bi-rewrite system — which is (strictly) weaker than the termination property— in order to prove the termination of the procedure; whereas in the equational case, the termination property of the rewrite system is needed to prove the termination of a procedure based on the computation of the normal form.

LEMMA 2.6. *Any terminating term bi-rewriting system is quasi-terminating.*

PROOF. If $(\xrightarrow[R_\subseteq]{} \cup \xrightarrow[R_\supseteq]{})^*$ is terminating then both $\xrightarrow[R_\subseteq]{*}$ and $\xrightarrow[R_\supseteq]{*}$ are terminating, and the problem is reduced to prove that any terminating term rewrite system is quasi-terminating.

First we prove that any terminating *term* rewriting relation is finitely branching. If $\xrightarrow[R]{}$ is terminating then any rewrite rule $l \longrightarrow r$ in R satisfies $\mathcal{FV}(r) \subseteq \mathcal{FV}(l)$. Now, to rewrite a term we have finitely many ways to select a rule $l \longrightarrow r$ and a subterm $t|_p$. Once we have fixed them, if it exists, there is a unique substitution satisfying $\mathcal{Dom}(\sigma) \subseteq \mathcal{FV}(l)$

and $t|_p = \sigma(l)$. Finally, if $\mathcal{FV}(r) \subseteq \mathcal{FV}(l)$, such substitution determines the result of the rewrite step. Second to prove that any finitely branching and terminating relation is quasi-terminating is a straightforward application of the Koenig's lemma. □

In order to test automatically the Church-Rosser property we extend the standard procedure used in term rewriting to bi-rewriting. So we reduce the Church-Rosser property to three simpler properties, namely *bi-confluence* (or commutativity), *local bi-confluence* and *critical pair bi-confluence*.

Definition 2.7. *A bi-rewrite system $\langle R_\subseteq, R_\supseteq \rangle$ is said to be*

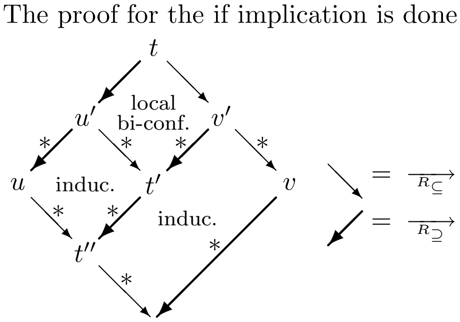
(i) **bi-confluent** *iff* $\xleftarrow[R_\supseteq]{*} \circ \xrightarrow[R_\subseteq]{*} \;\subseteq\; \xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*}$
(ii) **locally bi-confluent** *iff* $\xleftarrow[R_\supseteq]{} \circ \xrightarrow[R_\subseteq]{} \;\subseteq\; \xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*}$

A pair of terms $\langle s, t \rangle$ is said to be **bi-confluent** *iff* $s \xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*} t$.

A variant of the Newman's lemma (Newman, 1942; Huet, 1980) proves the following result for bi-rewrite systems. If fact the statement is implied by lemma 1.2 in (Bachmair and Dershowitz, 1986).

Lemma 2.8. *A terminating bi-rewrite system is Church-Rosser iff it is locally bi-confluent.*

Proof. The only if implication is trivially proved since $\xleftarrow[R_\supseteq]{} \circ \xrightarrow[R_\subseteq]{} \;\subseteq (\xrightarrow[R_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^*$. The proof for the if implication is done by noetherian induction.



We prove that property

$P(t) \stackrel{def}{=} \forall u, v \,.\, u \xleftarrow[R_\supseteq]{*} t \xrightarrow[R_\subseteq]{*} v \;\Rightarrow\; u \xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*} v$

holds for any term t by noetherian induction, using the well-founded ordering $(\xrightarrow[R_\supseteq]{} \cup \xrightarrow[R_\subseteq]{})^+$. The base cases $t = u$ or $t = v$ are trivially satisfied. The induction case follows directly from the induction hypothesis $P(u')$ and $P(v')$ using the diagram on the left.

□

Notice that in the previous lemma we require the union of both rewrite relations to be well-founded, and it is not sufficient if both relations are well-founded separately. The following counter-example was communicated to the authors by Professor Harald Ganzinger to show this fact. This counter-example invalidates the corresponding previous results in (Levy and Agustí, 1993; Levy, 1994). The bi-rewrite system defined by $R_\subseteq = \{b \xrightarrow{\subseteq} c, c \xrightarrow{\subseteq} d\}$ and $R_\supseteq = \{c \xrightarrow{\supseteq} b, b \xrightarrow{\supseteq} a\}$ is locally bi-confluent and both rewrite relations $\xrightarrow[R_\subseteq]{}$ and $\xrightarrow[R_\supseteq]{}$ are terminating, not their union. However, the bi-rewrite system is not Church-Rosser.

A simple adaptation of the standard critical pair definition (Knuth and Bendix, 1970) can be given for bi-rewrite systems. However, as we will see, it is not sufficient to prove the critical pair lemma. This simple definition of critical pair arises from the most general non-variable overlap between the left hand side of a rule in $R_{\subseteq}$ and a sub-term of the left hand side of a rule in $R_{\supseteq}$, (or vice versa). Given a pair of rules $l \xrightarrow{\subseteq} r$ and $s \xrightarrow{\supseteq} t$, a position p of a non-variable subterm of s, and the most general unifier σ of l and $s|_p$, the pair $\sigma(t) \subseteq \sigma(s[r]_p)$ is a (standard) critical pair between $R_{\subseteq}$ and $R_{\supseteq}$; and similarly for critical pairs between $R_{\supseteq}$ and $R_{\subseteq}$.

Unfortunately, in the presence of non-left-linear rules,† the critical pair lemma stated in terms of such standard critical pairs can not be proved because the confluence of variable overlaps is no longer possible. The same fact has already been discussed in (Bachmair, 1991). Here is a simple counter-example to the validity of this lemma.

COUNTER-EXAMPLE 2.9. The following bi-rewrite system

$$R_{\subseteq} = \{f(X, X) \xrightarrow{\subseteq} X\} \qquad R_{\supseteq} = \{a \xrightarrow{\supseteq} b\}$$

is terminating and has no standard critical pairs, however the divergence

$$f(a, b) \xleftarrow[R_{\supseteq}]{} f(a, a) \xrightarrow[R_{\subseteq}]{} a$$

does not satisfy the Church-Rosser property (the pair $f(a, b) \subseteq a$ is not bi-confluent). This problem would be avoided if $a \xrightarrow{\subseteq} b \in R_{\subseteq}$, but then the inclusion theory corresponding to the bi-rewrite system would be different.

Non-left-linear rules also invalidate the bi-rewrite parallel of Toyama's theorem (Toyama, 1987) as the following counter-example shows.

COUNTER-EXAMPLE 2.10. The following bi-rewrite system

$$R_{\subseteq} = \left\{ \begin{array}{l} X \cup X \xrightarrow{\subseteq} X \\ X \cup Y \xrightarrow{\subseteq} Y \cup X \\ X \cup (Y \cup Z) \xrightarrow{\subseteq} (X \cup Y) \cup Z \end{array} \right. \qquad R_{\supseteq} = \left\{ \begin{array}{l} X \cup Y \xrightarrow{\supseteq} X \\ X \cup Y \xrightarrow{\supseteq} Y \end{array} \right.$$

is Church-Rosser and quasi-terminating, if we consider a signature containing uniquely constants and the binary union operator, i.e. $\mathcal{F}_2 = \{\cup\}$ and $\mathcal{F}_i = \emptyset$ for $i \notin \{0, 2\}$. However, if we introduce a new 1-ary symbol in the signature $f \in \mathcal{F}_1$ then we have the following divergence which is not bi-confluent.

$$f(X) \cup f(Y) \xleftarrow[R_{\supseteq}]{} f(X) \cup f(X \cup Y) \xleftarrow[R_{\supseteq}]{} f(X \cup Y) \cup f(X \cup Y) \xrightarrow[R_{\subseteq}]{} f(X \cup Y)$$

This means that many properties of bi-rewrite systems depend not only on the axioms of the theory but also on the signature.

Using the standard definition of critical pairs, the critical pair lemma is only true for left-linear systems: a terminating and left-linear bi-rewrite system is Church-Rosser iff all standard critical pairs are bi-confluent. In order to keep this lemma for non-left-linear bi-rewrite systems, we have to enlarge the set of critical pairs to be considered as follows.

DEFINITION 2.11. *If $l \xrightarrow{\subseteq} r \in R_{\subseteq}$ and $s \xrightarrow{\supseteq} t \in R_{\supseteq}$ are two rewrite rules (with variables distinct) and p a position in s, then*

† A rule $l \longrightarrow r$ is left- (right-) linear iff any variable in l (in r) occurs at most once in l (in r).

(i) *if* $s|_p$ *is a non-variable subterm and* σ *is the most general unifier of* $s|_p$ *and* l *then*

$$\sigma(t) \subseteq \sigma(s[r]_p)$$

is a **(standard) critical pair** *of* $ECP(R_\subseteq, R_\supseteq)$

(ii) *if* $s|_p = x$ *is a repeated variable in* s, F *is a term not sharing variables with* $s \xrightarrow{\supseteq} t$ *such that* $F|_q = l$, *and* $l \xrightarrow[R_\supseteq]{*} r$ *does not hold,*† *then*

$$t[x \mapsto F] \subseteq \big(s[x \mapsto F]\big)[r]_{p\cdot q}$$

is an **(extended) critical pair** *of* $ECP(R_\subseteq, R_\supseteq)$.

Similarly for critical pairs between $R_\supseteq$ *and* $R_\subseteq$, *written* $ECP(R_\supseteq, R_\subseteq)$.

The set of (extended) critical pairs of the previous definition is in general infinite, $t[x \mapsto F] \subseteq \big(s[x \mapsto F]\big)[r]_{p\cdot q}$ is really a *critical pair scheme* because we do not impose any restriction on the *context* $F[\cdot]_q$ (notice that the only condition imposed to F is $F|_q = l$). In section 4 we will see an example where we use such kind of schemes. So the critical pair lemma even if true with this definition of critical pairs, will be of little practical help to test bi-confluence. Then the conditions of bi-confluence have to be studied in each case taking into account the particular shape of the non-left-linear rules. In section 6 we face the problem of testing bi-confluence automatically by codifying extended critical pairs using the linear second-order typed λ-calculus.

Theorem 2.12. (Extended Critical Pair Lemma) *A terminating bi-rewrite system* $\langle R_\subseteq, R_\supseteq\rangle$ *is Church-Rosser iff any (standard or extended) critical pair* $s \subseteq t$ *in* $ECP(R_\subseteq, R_\supseteq)$ *or* $s \supseteq t$ *in* $ECP(R_\supseteq, R_\subseteq)$ *is bi-confluent, i.e. it satisfies* $s \xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*} t$.

Proof. For the if part, see the proof of theorem 3.12, which states a more general result, taking $I_\subseteq = \emptyset$. For the only if part, extended critical pairs are sound deductions, therefore if $s \subseteq t$ is an extended critical pair, then $s(\xrightarrow[R_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^* t$ holds. Now, if the bi-rewrite system is Church-Rosser, then $s \xrightarrow[R_\subseteq]{*} \circ \xleftarrow[R_\supseteq]{*} t$. □

This theorem, lemma 3.9 and theorem 3.12 could be considered as instances of the *general critical pair theorem* proved by Geser in his thesis (Geser, 1990). Nevertheless, we think it is worthy to face the critical pair problem directly for our case.

The extended critical pair theorem generalizes the critical pair lemma (Knuth and Bendix, 1970) for bi-rewrite systems. However, we require the bi-confluence of not only the standard critical pairs, but also of the extended critical pairs. Nevertheless, if all rules come from the translation of an equational theory E, then any equation $a = b$ with $a \succ b$ results in two bi-rewrite rules $a \xrightarrow{\subseteq} b$ in $R_\subseteq$ and $a \xrightarrow{\supseteq} b$ in $R_\supseteq$ and both bi-rewrite relations $\xrightarrow[R_\subseteq]{} = \xrightarrow[R_\supseteq]{}$ are equal. Then we only obtain standard critical pairs because the condition $l \xrightarrow[R_\supseteq]{*} r$ in the definition 2.11 of extended critical pair is always satisfied. So we recover the old results for the equational case.

† If this condition is satisfied then we can make the pair resulting from the variable overlapping confluent like in the equational case.

3. Bi-rewriting Modulo a Set of Inclusions

Like in equational rewriting, in bi-rewriting it is not always possible to orient all inclusions of a theory presentation in two terminating rewrite relations, as was assumed in the previous section. Frequently enough, we must handle three rewrite relations, the terminating relations $\xrightarrow[R_\subseteq]{}$ and $\xrightarrow[R_\supseteq]{}$ resulting from the inclusions $R_\subseteq$ and $R_\supseteq$ oriented to the right and to the left respectively, and the non-terminating relation $\xleftrightarrow[I_\subseteq]{}$ resulting from the non-oriented inclusions I. Then we say to have a $\langle R_\subseteq, R_\supseteq\rangle$ bi-rewrite system modulo I. Although we use the word *modulo*, it does not mean that $\xleftrightarrow[I_\subseteq]{*}$ is a congruence, be aware it is a non-symmetric relation (monotonic pre-order). Figure 3 in section 4 shows an example of these bi-rewrite systems. The inverse of the relation $\xleftrightarrow[I_\subseteq]{}$ is noted $\xleftrightarrow[I_\supseteq]{}$. The Birkhoff's theorem is stated then as $Ax \vdash_{POL} t \subseteq u$ iff $t\ (\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^*\ u$.

3.1. From Church-Rosser to Local Bi-Confluence

The simplest way to have a complete and decidable proof procedure based on the $\langle R_\subseteq, R_\supseteq\rangle$ bi-rewrite system modulo I is reducing it to the bi-rewrite system $\langle R_\subseteq \cup I, R_\supseteq \cup I\rangle$ and, using the results of the previous section, requiring of it the following properties:

1 The relations $\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{}$ and $\xrightarrow[R_\supseteq]{} \cup \xleftrightarrow[I_\supseteq]{}$ are both quasi-terminating, and
2 they satisfy the (weak) Church-Rosser property

$$(\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^* \subseteq (\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{})^* \circ (\xleftarrow[R_\supseteq]{} \cup \xleftrightarrow[I_\subseteq]{})^*$$

However, as we have seen in the previous section the quasi-termination of $\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{}$ and $\xrightarrow[R_\supseteq]{} \cup \xleftrightarrow[I_\supseteq]{}$ is not enough to reduce the (weak) Church-Rosser property to the local bi-confluence property $(\xleftarrow[R_\supseteq]{} \cup \xleftrightarrow[I_\subseteq]{}) \circ (\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{}) \subseteq (\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{})^* \circ (\xleftarrow[R_\supseteq]{} \cup \xleftrightarrow[I_\subseteq]{})^*$ using lemma 2.8. To do this we would need the termination of $\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{} \cup \xrightarrow[R_\supseteq]{} \cup \xleftrightarrow[I_\supseteq]{}$, which of course never holds, because the relation $\xleftrightarrow[I_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{}$ is cycling. The solution to this problem comes from requiring the termination of $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{*} \circ \xrightarrow[R_\supseteq]{}$. Using this termination property, the weak Church-Rosser property can be reduced to a local bi-confluence property.

Lemma 3.1. *If the relation* $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{*} \circ \xrightarrow[R_\supseteq]{}$ *is terminating, then the following properties*

$$(\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^* \subseteq (\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{})^* \circ (\xleftarrow[R_\supseteq]{} \cup \xleftrightarrow[I_\subseteq]{})^* \quad \textit{(weak) Church-Rosser}$$

$$\xleftarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \subseteq (\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{})^* \circ \xleftrightarrow[I_\subseteq]{*} \circ (\xleftarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\subseteq]{*})^* \quad \textit{(weak) local bi-confluence}$$

are equivalent.

Proof. Using the equalities $(A \cup B)^* = (A^* \circ B)^* \circ A^* = A^* \circ (B \circ A^*)^*$ we prove that right hand sides of both inclusions are equal. Now $\xleftarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \subseteq (\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^*$ shows that Church-Rosser implies local bi-confluence.

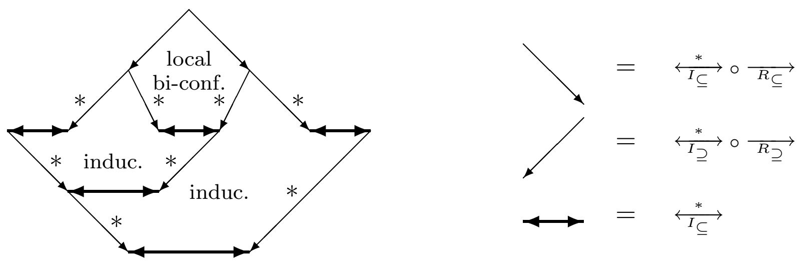
For the converse we use $(A \cup B)^* \subseteq A^* \circ B^* \Leftrightarrow B^* \circ A^* \subseteq A^* \circ B^*$ to prove the equivalence between the Church-Rosser property and the following one.

$$\xleftrightarrow[I_\subseteq]{*} \circ (\xleftarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\subseteq]{*})^* \circ (\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{})^* \circ \xleftrightarrow[I_\subseteq]{*} \subseteq (\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{})^* \circ \xleftrightarrow[I_\subseteq]{*} \circ (\xleftarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\subseteq]{*})^*$$

Now, if $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{*} \circ \xrightarrow[R_\supseteq]{}$ is terminating we can prove by noetherian induction that this property is equivalent to the local bi-confluence property.

The base cases $\xleftrightarrow[I_\subseteq]{*} \circ (\xleftarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\subseteq]{*})^n \circ (\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{})^m \circ \xleftrightarrow[I_\subseteq]{*}$ with $n = 0$ or $m = 0$ trivially hold.

The following diagram shows a sketch of the proof for $n > 0$ and $m > 0$.



□

If $\xleftrightarrow[I_\subseteq]{}$ is symmetric ($\xleftrightarrow[I_\subseteq]{} = \xleftrightarrow[I_\supseteq]{}$) the above termination property becomes similar to the termination property required in rewriting modulo a set of equations (Bachmair and Dershowitz, 1989). That is, $\xleftrightarrow[I_\subseteq]{}$ symmetric means we can define equivalence classes ($[s]_I \xrightarrow[R]{} [t]_I$ iff $s \xleftrightarrow[I]{*} \circ \xrightarrow[R]{} \circ \xleftrightarrow[I]{*} t$) and, the termination of $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{*} \circ \xrightarrow[R_\supseteq]{}$ is ensured by the existence of a well-founded I-compatible quasi-ordering, i.e. by the existence of a well-founded, reflexive and transitive relation $\succeq$ satisfying $\xrightarrow[R_\subseteq]{} \subseteq \succ$, $\xrightarrow[R_\supseteq]{} \subseteq \succ$ and $\xleftrightarrow[I_\subseteq]{} = \xleftrightarrow[I_\supseteq]{} \subseteq \approx$, where the equivalence relation $\approx$ is the intersection of $\succeq$ and $\preceq$ and the strict ordering $\succ$ is the difference of $\succeq$ and $\approx$. The quasi-termination property of $\xleftrightarrow[I_\subseteq]{}$ means that each $\xleftrightarrow[I_\subseteq]{}$-class of equivalence is finite.

However, like in the equational case, rewriting by $\xleftrightarrow[I]{*} \circ \xrightarrow[R]{}$ is inefficient, and the local commutativity of $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{}$ and $\xleftrightarrow[I_\supseteq]{*} \circ \xrightarrow[R_\supseteq]{}$ can not be reduced to the bi-confluence of a finite set of critical pairs. Therefore we will approximate them by two weaker, but more practical rewrite relations, named $I\backslash R_\subseteq$ and $I\backslash R_\supseteq$ respectively by similarity to the corresponding equational definitions. Notice that although we use the notation $\xrightarrow[I\backslash R]{}$, it does not means that this relation is the monotonic and substitution closure of a set of rules. In the following, we prove the abstract properties of these relations. We will suppose that they satisfy:

$$\xrightarrow[R_\subseteq]{} \subseteq \xrightarrow[I\backslash R_\subseteq]{} \subseteq \xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{}$$
$$\xrightarrow[R_\supseteq]{} \subseteq \xrightarrow[I\backslash R_\supseteq]{} \subseteq \xleftrightarrow[I_\supseteq]{*} \circ \xrightarrow[R_\supseteq]{}$$

leaving their definition for the next subsection.

We require these new rewrite relations to satisfy what we call a *strong Church-Rosser modulo I* property, defined as follows.

Definition 3.2. *The bi-rewrite system* $\langle R_\subseteq, R_\supseteq \rangle$ *modulo* I *is* **(strong) Church-Rosser** *iff*

$$(\xrightarrow[R_\subseteq]{} \cup \xleftrightarrow[I_\subseteq]{} \cup \xleftarrow[R_\supseteq]{})^* \subseteq \xrightarrow[I\backslash R_\subseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xleftarrow[I\backslash R_\supseteq]{*}$$

The following lemma states sufficient conditions to define a search decision procedure for $Ax \vdash_{POL} t \subseteq u$ based on the relations $I\backslash R_\subseteq$ and $I\backslash R_\supseteq$.

Lemma 3.3. *If the relations* $\xrightarrow[I\backslash R_\subseteq]{}$ *and* $\xrightarrow[I\backslash R_\supseteq]{}$ *are both computable*[†] *and quasi-terminating, the relation* $\underset{I_\subseteq}{\overset{*}{\longleftrightarrow}}$ *is decidable, and* $\langle R_\subseteq, R_\supseteq\rangle$ *is strong Church-Rosser modulo* I*, then there exists a decision procedure for the inclusion relation defined by these relations.*

Proof. Like in the simpler case of the previous section, given two terms s and t, the algorithm generates the sets $\{s' \mid s \xrightarrow[I\backslash R_\subseteq]{*} s'\}$ and $\{t' \mid t \xrightarrow[I\backslash R_\supseteq]{*} t'\}$ and seek for a term s' from the first set and a term t' from the second one such that $s' \underset{I_\subseteq}{\overset{*}{\longleftrightarrow}} t'$. If relations $\xrightarrow[I\backslash R_\subseteq]{*}$ and $\xrightarrow[I\backslash R_\supseteq]{*}$ satisfy the above inclusions and the Church-Rosser property then $(\xrightarrow[R_\subseteq]{} \cup \underset{I_\subseteq}{\longleftrightarrow} \cup \xleftarrow[R_\supseteq]{})^* = \xrightarrow[I\backslash R_\subseteq]{*} \circ \underset{I_\subseteq}{\overset{*}{\longleftrightarrow}} \circ \xleftarrow[I\backslash R_\supseteq]{*}$. Now, it is easy to prove that the algorithm is a decision procedure for the relation $\xrightarrow[I\backslash R_\subseteq]{*} \circ \underset{I_\subseteq}{\overset{*}{\longleftrightarrow}} \circ \xleftarrow[I\backslash R_\supseteq]{*}$ and $Ax \vdash_{POL} s \subseteq t$ is equivalent to $s(\xrightarrow[R_\subseteq]{} \cup \underset{I_\subseteq}{\longleftrightarrow} \cup \xleftarrow[R_\supseteq]{})^* t$. □

The solutions we propose of reducing the strong Church-Rosser property to a local bi-confluence property are inspired mainly by the solutions known for the equational case. In the following we consider how they can be adapted to bi-rewriting.

Huet (1980) proved that given a set of rules R and equations E such that $\underset{E}{\overset{*}{\longleftrightarrow}} \circ \xrightarrow[R]{}$ is terminating, R is strong Church-Rosser modulo E iff all *peaks* and *cliffs* are confluent: $\xleftarrow[R]{} \circ \xrightarrow[R]{} \subseteq \xrightarrow[R]{*} \circ \underset{E}{\overset{*}{\longleftrightarrow}} \circ \xleftarrow[R]{*}$ and $\underset{E}{\longleftrightarrow} \circ \xrightarrow[R]{} \subseteq \xrightarrow[R]{*} \circ \underset{E}{\overset{*}{\longleftrightarrow}} \circ \xleftarrow[R]{*}$. Notice that these are sufficient and, what is also important, necessary conditions. Besides, the finiteness of the E-equivalence classes is not required. However, these confluence properties are too strong and can not be reduced to the confluence of critical pairs unless the rules are left-linear.

To overcome this limitation for non-left-linear systems Peterson and Stickel (1981) propose the use of a new rewrite relation $E\backslash R$ satisfying $\xrightarrow[R]{} \subseteq \xrightarrow[E\backslash R]{} \subseteq \underset{E}{\overset{*}{\longleftrightarrow}} \circ \xrightarrow[R]{}$. They prove that when this relation is E-compatible, that is when $\underset{E}{\overset{*}{\longleftrightarrow}} \circ \xrightarrow[R]{} \subseteq \xrightarrow[E\backslash R]{} \circ \underset{E}{\overset{*}{\longleftrightarrow}} \circ (\xleftarrow[R]{} \circ \underset{E}{\overset{*}{\longleftrightarrow}})^*$, and terminating, then the Church-Rosser property becomes equivalent to the confluence of peaks of the form $\xleftarrow[E\backslash R]{} \circ \xrightarrow[E\backslash R]{} \subseteq \xrightarrow[E\backslash R]{*} \circ \underset{E}{\overset{*}{\longleftrightarrow}} \circ \xleftarrow[E\backslash R]{*}$. They also study how a rewrite relation R can be extended to obtain a E-compatible rewrite relation $E\backslash R$ when E is an associative and commutative theory. However, in this case the problem is that the set of critical pairs of the form $t \xleftarrow[E\backslash R]{} u \xrightarrow[E\backslash R]{} v$ is in general infinite.

Jouannaud and Kirchner (1986), theorem 5 and Kirchner (1985), theorem 4, chapter 2 generalize the Peterson and Stickel's concept of E-compatibility to coherence, and prove that when $\underset{E}{\overset{*}{\longleftrightarrow}} \circ \xrightarrow[R]{}$ is terminating (R is E-terminating) then the following three conditions are equivalent[†]

1 R is $E\backslash R$-Church-Rosser modulo E

$$(\xrightarrow[R]{} \cup \underset{E}{\longleftrightarrow} \cup \xleftarrow[R]{})^* \subseteq \xrightarrow[E\backslash R]{*} \circ \underset{E}{\overset{*}{\longleftrightarrow}} \circ \xleftarrow[E\backslash R]{*}$$

[†] We say that a relation $\xrightarrow[R]{}$ is computable iff for any term t, the set $\{u \mid t \xrightarrow[R]{} u\}$ is calculable. We say that a relation $\xrightarrow[R]{}$ is decidable iff for any pair of terms t and u, it is decidable when $t \xrightarrow[R]{} u$ holds or not.

[†] They define coherence as $\underset{E}{\overset{*}{\longleftrightarrow}} \circ \xrightarrow[E\backslash R]{} \subseteq \xrightarrow[E\backslash R]{+} \circ \underset{E}{\overset{*}{\longleftrightarrow}} \circ \xleftarrow[E\backslash R]{*}$, however, as they notice, if R is E terminating, both definitions are equivalent.

2 $E\backslash R$ is confluent and coherent modulo E

$$\xleftarrow[E\backslash R]{} \circ \xrightarrow[E\backslash R]{} \subseteq \xrightarrow[E\backslash R]{*} \circ \xleftrightarrow[E]{*} \circ \xleftarrow[E\backslash R]{*} \qquad \text{(global) peak}$$
$$\xleftrightarrow[E]{*} \circ \xrightarrow[E\backslash R]{} \subseteq \xrightarrow[E\backslash R]{*} \circ \xleftrightarrow[E]{*} \circ \xleftarrow[E\backslash R]{*} \qquad \text{(global) cliff}$$

3 $E\backslash R$ is locally confluent and locally coherent modulo E

$$\xleftarrow[R]{} \circ \xrightarrow[E\backslash R]{} \subseteq \xrightarrow[E\backslash R]{*} \circ \xleftrightarrow[E]{*} \circ \xleftarrow[E\backslash R]{*} \qquad \text{local peak}$$
$$\xleftrightarrow[E]{} \circ \xrightarrow[E\backslash R]{} \subseteq \xrightarrow[E\backslash R]{*} \circ \xleftrightarrow[E]{*} \circ \xleftarrow[E\backslash R]{*} \qquad \text{local cliff}$$

Then local confluence and coherence can be reduced to critical pair confluence and to extended rules respectively. Here we call these properties *confluence of peaks* and *confluence of cliffs* respectivelly, following the notation of Dershowitz and Jouannaud (1990).

Jouannaud and Kirchner also notice that this result is false if we require termination of $\xrightarrow[E\backslash R]{}$ instead of that for $\xleftrightarrow[E]{*} \circ \xrightarrow[R]{}$. As a counter-example† we can take the rewrite system $R = E\backslash R = \{b \longrightarrow a, a \longrightarrow d\}$ with $E = \{a = b, b = c\}$. It satisfies local confluence properties and termination of $\xrightarrow[E\backslash R]{}$, but it is not Church-Rosser. However, termination of $\xrightarrow[E\backslash R]{}$ is enough to prove the equivalence between Church-Rosser property and "global" confluence properties (see (Huet, 1980) for a similar proof). Evidently, this termination property is not enough to prove equivalence between local and global confluence properties. For our purposes, we are more interested on global confluence properties than on the local ones, therefore we use them in the following lemma.

LEMMA 3.4. *Let* $\xrightarrow[I\backslash R_{\subseteq}]{}$ *and* $\xrightarrow[I\backslash R_{\supseteq}]{}$ *be two rewrite relations satisfying* $\xrightarrow[R_{\subseteq}]{} \subseteq \xrightarrow[I\backslash R_{\subseteq}]{} \subseteq \xleftrightarrow[I_{\subseteq}]{*} \circ \xrightarrow[R_{\subseteq}]{}$ *and* $\xrightarrow[R_{\supseteq}]{} \subseteq \xrightarrow[I\backslash R_{\supseteq}]{} \subseteq \xleftrightarrow[I_{\supseteq}]{*} \circ \xrightarrow[R_{\supseteq}]{}$. *If their union* $\xrightarrow[I\backslash R_{\subseteq}]{} \cup \xrightarrow[I\backslash R_{\supseteq}]{}$ *is terminating then the following three global confluence properties*

$$\left.\begin{array}{l} \xleftrightarrow[I_{\subseteq}]{*} \circ \xrightarrow[I\backslash R_{\subseteq}]{} \subseteq \xrightarrow[I\backslash R_{\subseteq}]{*} \circ \xleftrightarrow[I_{\subseteq}]{*} \circ \xleftarrow[I\backslash R_{\supseteq}]{*} \\ \xleftarrow[I\backslash R_{\supseteq}]{} \circ \xleftrightarrow[I_{\subseteq}]{*} \subseteq \xrightarrow[I\backslash R_{\subseteq}]{*} \circ \xleftrightarrow[I_{\subseteq}]{*} \circ \xleftarrow[I\backslash R_{\supseteq}]{*} \end{array}\right\} \quad \textit{cliffs}$$

$$\xleftarrow[I\backslash R_{\supseteq}]{} \circ \xrightarrow[I\backslash R_{\subseteq}]{} \subseteq \xrightarrow[I\backslash R_{\subseteq}]{*} \circ \xleftrightarrow[I_{\subseteq}]{*} \circ \xleftarrow[I\backslash R_{\supseteq}]{*} \qquad \textit{peaks}$$

and the strong Church-Rosser property

$$\left(\xrightarrow[R_{\subseteq}]{} \cup \xleftrightarrow[I_{\subseteq}]{} \cup \xleftarrow[R_{\supseteq}]{} \right)^* \subseteq \xrightarrow[I\backslash R_{\subseteq}]{*} \circ \xleftrightarrow[I_{\subseteq}]{*} \circ \xleftarrow[I\backslash R_{\supseteq}]{*}$$

are equivalent.

PROOF. It is evident that the Church-Rosser property implies the three local bi-confluence properties, so we will prove the opposite implication. Such proof is based on the ideas of proof transformation and proof ordering proposed by Bachmair in his thesis (Bachmair, 1991) and in (Bachmair *et al.*, 1986).

Given a sequence of terms $\langle v_1, \dots v_n \rangle$, we say that *it is a proof of* $s \subseteq t$ iff $v_1 = s$, $v_n = t$, and for any $i \in [1..n-1]$ we have $v_i \xrightarrow[I\backslash R_{\subseteq}]{} v_{i+1}$ or $v_i \xleftarrow[I\backslash R_{\supseteq}]{} v_{i+1}$ or $v_i \xleftrightarrow[I_{\subseteq}]{+} v_{i+1}$. Notice that we allow to concentrate one or more $\xleftrightarrow[I_{\subseteq}]{}$ rewrite steps in a single proof step. Evidently, $t \subseteq u$ has a proof iff $t(\xrightarrow[R_{\subseteq}]{} \cup \xleftrightarrow[I_{\subseteq}]{} \cup \xleftarrow[R_{\supseteq}]{})^* u$.

† This counter-example is, in fact, equivalent to one given by Huet (1980).

In the following we define a set of transformations on the proofs of an inclusion. Given a proof transformation rule $\langle s, \overline{t}, u\rangle \Rightarrow \langle s, \overline{v}, u\rangle$, we can use it to transform $\langle \overline{w_1}, s, \overline{t}, u, \overline{w_2}\rangle \Rightarrow \langle \overline{w_1}, s, \overline{v}, u, \overline{w_2}\rangle$. To prove the termination of such transformation relation we associate a multiset $S(\langle v_1, \dots, v_n\rangle)$ of terms to each proof $\langle v_1, \dots, v_n\rangle$ defined as follows.

$$S(\langle v\rangle) = \emptyset$$

$$S(\langle v_1, \dots, v_n\rangle) = S(\langle v_1, \dots, v_{n-1}\rangle) \cup \begin{cases} \{v_{n-1}, v_n\} & \text{if } v_{n-1} \xrightarrow[I\setminus R_\subseteq]{} v_n \\ & \text{or } v_n \xrightarrow[I\setminus R_\supseteq]{} v_{n-1} \\ \{v_{n-1}^2, v_n^2\} & \text{if } v_{n-1} \xleftrightarrow[I_\subseteq]{+} v_n \end{cases}$$

where $\cup$ denotes the multiset union operator and superscripts denote the number of occurrences of an element in a multiset. We define a well-founded ordering $\succ$ on these term multisets as the multiset extension of the order relation $\xrightarrow[I\setminus R_\subseteq]{+} \cup \xrightarrow[I\setminus R_\supseteq]{+}$ which we have supposed terminating. This ordering on associated multisets defines a well-founded ordering on proofs. Notice that this ordering is monotonic, i.e. if $S(\langle s, \overline{t}, u\rangle) \succ S(\langle s, \overline{v}, u\rangle)$, then $S(\langle \overline{w_1}, s, \overline{t}, u, \overline{w_2}\rangle) \succ S(\langle \overline{w_1}, s, \overline{v}, u, \overline{w_2}\rangle)$. This is a key point to prove that if any proof transformation rule $\langle s, \overline{t}, u\rangle \Rightarrow \langle s, \overline{v}, u\rangle$ satisfies $S(\langle s, \overline{t}, u\rangle) \succ S(\langle s, \overline{v}, u\rangle)$ then the proof transformation relation is terminating.

If cliffs are bi-confluent, then for any cliff $s \xleftrightarrow[I_\subseteq]{+} t \xrightarrow[I\setminus R_\subseteq]{} u$ we have

$$s \xrightarrow[I\setminus R_\subseteq]{} v_1 \cdots v_{p-1} \xrightarrow[I\setminus R_\subseteq]{} v_p \xleftrightarrow[I_\subseteq]{*} w_q \xleftarrow[I\setminus R_\supseteq]{} w_{q-1} \cdots w_1 \xleftarrow[I\setminus R_\supseteq]{} u$$

and we can apply one of the following proof transformations rules to eliminate it

$$\begin{array}{ll} \langle s, t, u\rangle \Rightarrow \langle s, v_1, \dots, v_p, w_q, \dots, w_1, u\rangle & \text{if } v_p \xleftrightarrow[I_\subseteq]{+} w_q \\ \langle s, t, u\rangle \Rightarrow \langle s, v_1, \dots, v_{p-1}, w_q, \dots, w_1, u\rangle & \text{if } s \xrightarrow[I\setminus R_\subseteq]{+} v_p = w_q \\ \langle s, t, u\rangle \Rightarrow \langle s, w_{q-1}, \dots, w_1, u\rangle & \text{if } s = v_p = w_q \xleftarrow[I\setminus R_\supseteq]{+} u \\ \langle s, t, u\rangle \Rightarrow \langle s\rangle & \text{if } s = v_p = w_q = u \end{array}$$

where $p, q \geq 0$, except in the second rule where $p \geq 1$, and the third rule where $q \geq 1$. Now, taking into account that $s \succ v_1 \succ \cdots \succ v_p$ and $t \succ u \succ w_1 \succ \cdots \succ w_q$, we can prove that the multiset associated to the left part of the rules $S(\langle s, t, u\rangle) = \{s^2, t^3, u\}$ is strictly greater than the multisets associated to the right part of the rules, which are respectively:

$$\begin{array}{l} S(\langle s, v_1, \dots, v_p, w_q, \dots, w_1, u\rangle) = \{s, v_1^2, \dots, v_p^2, w_q^2, w_2^2, \dots, w_1^2, u\} \cup {}^{\dagger}\{v_p, w_q\} \\ S(\langle s, v_1, \dots, v_{p-1}, w_q, \dots, w_1, u\rangle) = \{s, v_1^2, \dots, v_{p-1}^2, w_q^2, \dots, w_1^2, u\} \\ S(\langle s, w_{q-1}, \dots, w_1, u\rangle) = \{s, w_{q-1}^2, \dots, w_1^2, u\} \\ S(\langle s\rangle) = \emptyset \end{array}$$

Similarly, if peaks are bi-confluent, then we can also apply the same proof transformations rule to any peak $s \xleftarrow[I\setminus R_\supseteq]{} t \xrightarrow[I\setminus R_\subseteq]{} u$. And, taking into account that now $t \succ s \succ v_1 \succ \cdots \succ v_p$ and $t \succ u \succ w_1 \succ \cdots \succ w_q$, we can also prove that the multiset associated to the left part of the rule, now $S(\langle s, t, u\rangle) = \{s, t^2, u\}$ is also strictly greater than the multisets associated to the corresponding right parts of the rules.

Evidently, if we iterate this process, the resulting canonical (normal) proof will not contain any cliffs nor peaks. Therefore it will be of the form $\xrightarrow[I\setminus R_\subseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xleftarrow[I\setminus R_\supseteq]{*}$. The

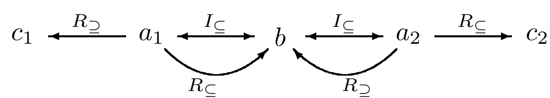
[†] Notice that in this case we can have $s = v_p$, $u = w_q$ or both together. With such union we capture four cases.

process can not be applied infinitely, because the transformation relation is terminating. We conclude that if $s \subseteq t$ has a proof, then it has a canonical proof of the form $s \xrightarrow[I\backslash R_{\subseteq}]{*} \circ \xleftrightarrow[I_{\subseteq}]{*} \circ \xleftarrow[I\backslash R_{\supseteq}]{*} t$. Therefore, the Church-Rosser property holds for these rewrite relations. $\square$

Now, the logical process would be to reduce the bi-confluence of peaks of the form $\xleftarrow[I\backslash R_{\supseteq}]{} \circ \xrightarrow[I\backslash R_{\subseteq}]{}$ to the bi-confluence of peaks of the form $\xleftarrow[I\backslash R_{\supseteq}]{} \circ \xrightarrow[R_{\subseteq}]{}$ or $\xleftarrow[R_{\supseteq}]{} \circ \xrightarrow[I\backslash R_{\subseteq}]{}$, as Jouannaud and Kirchner did for the equational case. However, as the following counter-example shows, not any definition of $\xrightarrow[I\backslash R]{}$ satisfying $\xrightarrow[R]{} \subseteq \xrightarrow[I\backslash R]{} \subseteq \xleftrightarrow[I]{*} \circ \xrightarrow[R]{}$ permits such reduction, unless we require termination of $(\xleftrightarrow[I_{\subseteq}]{} \cup \xleftrightarrow[I_{\supseteq}]{})^{*} \circ (\xrightarrow[R_{\subseteq}]{} \cup \xrightarrow[R_{\supseteq}]{})$.

Counter-example 3.5. Consider the rewrite relations defined by the following sets of rules.

$$\begin{array}{lcl} I_{\subseteq} & = & \{a_1 \xleftrightarrow{\subseteq} b, b \xleftrightarrow{\subseteq} a_2\} \\ R_{\subseteq} & = & \{a_1 \xrightarrow{\subseteq} b, a_2 \xrightarrow{\subseteq} c_2\} \\ R_{\supseteq} & = & \{a_2 \xrightarrow{\subseteq} b, a_1 \xrightarrow{\subseteq} c_1\} \end{array}$$



If we define $\xrightarrow[I\backslash R_{\subseteq}]{} \stackrel{def}{=} \xrightarrow[R_{\subseteq}]{} \cup \xleftrightarrow[I_{\subseteq}]{} \circ \xrightarrow[R_{\subseteq}]{}$ and $\xrightarrow[I\backslash R_{\supseteq}]{} \stackrel{def}{=} \xrightarrow[R_{\supseteq}]{} \cup \xleftrightarrow[I_{\supseteq}]{} \circ \xrightarrow[R_{\supseteq}]{}$, we will obtain two rewrite relations such that $\xrightarrow[I\backslash R_{\subseteq}]{} \cup \xrightarrow[I\backslash R_{\supseteq}]{}$ is terminating and the properties $\xrightarrow[R]{} \subseteq \xrightarrow[I\backslash R]{} \subseteq \xleftrightarrow[I]{*} \circ \xrightarrow[R]{}$ hold. However, although any peak of the form $\xleftarrow[I\backslash R_{\supseteq}]{} \circ \xrightarrow[R_{\subseteq}]{}$ or $\xleftarrow[R_{\supseteq}]{} \circ \xrightarrow[I\backslash R_{\subseteq}]{}$ and any cliff is bi-confluent, there is a peak $c_1 \xleftarrow[I\backslash R_{\supseteq}]{} b \xrightarrow[I\backslash R_{\subseteq}]{} c_2$ which is not bi-confluent. Notice that in this counter-example $\xleftrightarrow[I_{\subseteq}]{*} \circ \xrightarrow[R_{\subseteq}]{} \cup \xleftrightarrow[I_{\supseteq}]{*} \circ \xrightarrow[R_{\supseteq}]{}$ is also terminating, not so $(\xleftrightarrow[I_{\subseteq}]{} \cup \xleftrightarrow[I_{\supseteq}]{})^{*} \circ (\xrightarrow[R_{\subseteq}]{} \cup \xrightarrow[R_{\supseteq}]{})$.

Remark 3.6. The reader may prove that, when $(\xleftrightarrow[I_{\subseteq}]{} \cup \xleftrightarrow[I_{\supseteq}]{})^{*} \circ (\xrightarrow[R_{\subseteq}]{} \cup \xrightarrow[R_{\supseteq}]{})$ is terminating, then (strong) Church-Rosser property and "local" confluence properties are equivalent. This result is closer to the theorem proved by Jouannaud and Kirchner (1986), theorem 5, and its proof is left to the reader.

Alternately, if we only require the termination of $\xrightarrow[I\backslash R_{\subseteq}]{} \cup \xrightarrow[I\backslash R_{\supseteq}]{}$, then the method of *rule extensions* and the concrete definition of the relation $\xrightarrow[I\backslash R]{}$ ensures that, if inclusions in I are linear, then $\xleftrightarrow[I]{*}$ and $\xrightarrow[I\backslash R]{*}$ commute, i.e. $\xleftrightarrow[I]{*} \circ \xrightarrow[I\backslash R]{*} \subseteq \xrightarrow[I\backslash R]{*} \circ \xleftrightarrow[I]{*}$. This property is stronger than the confluence of cliffs, and permits the desired reduction. However, such method takes into account the structure of terms, so we will describe it in the next subsection.

3.2. From Local Bi-Confluence to (Extended) Critical Pairs

Till now, we have studied Church-Rosser, termination and bi-confluence properties in the framework of relational algebra (Bäumer, 1992). All proofs were done without any reference to the structure of terms. In the following, we will consider the term structure in order to reduce the bi-confluence properties to the bi-confluence of critical pairs and rule extensions.

We begin defining the rewrite relations $I\backslash R_{\subseteq}$ and $I\backslash R_{\supseteq}$ that were only axiomatically characterized by $\xrightarrow[R]{} \subseteq \xrightarrow[I\backslash R]{} \subseteq \xleftrightarrow[I]{*} \circ \xrightarrow[R]{}$ in the previous subsection. The choice of such definition is motivated, as in the equational case, by the fact that local bi-confluence of peaks $\xleftarrow[R_{\supseteq}]{} \circ \xrightarrow[I\backslash R_{\subseteq}]{}$ and $\xleftarrow[I\backslash R_{\supseteq}]{} \circ \xrightarrow[R_{\subseteq}]{}$ can be reduced to the bi-confluence of a selected set of critical pairs.

DEFINITION 3.7. *We say that s $R_\subseteq$*-**rewrites** *to t* **modulo** *$I_\subseteq$, written $s \xrightarrow[I\backslash R_\subseteq]{} t$, iff there exists a rule $l \longrightarrow r$ in $R_\subseteq$, an occurrence p in s, and a substitution σ such that $s|_p \xleftrightarrow[I_\subseteq]{*} \sigma(l)$ and $t = s[\sigma(r)]_p$. We write $s \xrightarrow[I\backslash R_\subseteq]{}{}_{[p,\sigma,l\to r]}\; t$ when we want to make explicit the position, substitution and rule involved in the rewrite step.*

Similarly for s $R_\supseteq$-rewrites to t modulo $I_\supseteq$, written $s \xrightarrow[I\backslash R_\supseteq]{} t$.

With this definition $\xrightarrow[I\backslash R_\subseteq]{}$ really verifies $\xrightarrow[R_\subseteq]{} \subseteq \xrightarrow[I\backslash R_\subseteq]{} \subseteq \xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{}$ although in general $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \not\subseteq \xrightarrow[I\backslash R_\subseteq]{}$. Notice that in a $\xrightarrow[I\backslash R]{}$ rewrite step, the $\xleftrightarrow[I]{}$ rewrite steps take place *bellow* the $\xrightarrow[R]{}$ rewrite step. We say that the $\xrightarrow[R]{}$ rewrite step *covers* such $\xleftrightarrow[I]{}$ rewrite steps.

We will use the notions of E-matching and E-unification from (Peterson and Stickel, 1981) but adapted to bi-rewriting. Given two terms s and t, we say that s I-matches t iff there exists a substitution σ such that $s \xleftrightarrow[I_\subseteq]{*} \sigma(t)$, and s I^{-1}-matches t iff there exists a substitution σ such that $s \xleftrightarrow[I_\supseteq]{*} \sigma(t)$. We say that s I-unifies with t iff there exists a substitution σ such that $\sigma(s) \xleftrightarrow[I_\subseteq]{*} \sigma(t)$. Notice that, since $\xleftrightarrow[I_\subseteq]{}$ is not necessarily symmetric, I-matching and I^{-1}-matching are in general different non-symmetric relations, and I-unification also is a non-symmetric relation. We will suppose in the following that I-unification and I and I^{-1}-matching are decidable. We have to be careful defining minimum unifiers since definition of critical pairs is based on them. Given two terms s and t, we say that $\mathcal{M}$ is a *complete set of minimum unifiers* iff for any I-unifier τ of s and t, there exists a minimum unifier $\sigma \in \mathcal{M}$ and a substitution ρ such that $\tau(x)(\xleftrightarrow[I_\subseteq]{*} \cap \xleftrightarrow[I_\supseteq]{*})\rho(\sigma(x))$ for any $x \in \mathcal{D}om(\tau)$. We will suppose in the following that a finite and complete set of minimum unifiers exists for our relation $\xleftrightarrow[I_\subseteq]{}$.

As in the equational case (to prove bi-confluence of cliffs or E-compatibility), we will prove the commutativity properties by means of the rule extension and the extensionally closed property defined as follows.

DEFINITION 3.8. *Given an inclusion $l \subseteq r$ in I, and a rule $s \xrightarrow{\subseteq} t$ in $R_\subseteq$, if $r|_p$ I-unifies with s, being σ a minimum unifier, and $r|_p$ is neither a variable nor equal to r, then we say that $\sigma(l) \xrightarrow{\subseteq} \sigma(r[t]_p)$ is a* **right-I-extended** *rule of $R_\subseteq$.*

Given a set of rules $R_\subseteq$ and inclusions I, $R_\subseteq$ is said to be **right-I-extensionally closed** *iff any right-I-extended rule $l \xrightarrow{\subseteq} r$ of $R_\subseteq$ satisfies $l \xrightarrow[I\backslash R_\subseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} r$.*

We define left-I-extended rule and left-I-extensionally closed similarly changing $\subseteq$ by $\supseteq$ and "$r|_p$ I-unifies with s" by "s I-unifies with $r|_p$".

Notice that in the previous definition, to consider a bi-rewrite system extensionally closed, we require any rule extension $l \xrightarrow{\subseteq} r$ to satisfy $l \xrightarrow[I\backslash R_\subseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} r$. It is not enough to require the pair $l \subseteq r$ to be bi-confluent.

Since $\xleftrightarrow[I_\subseteq]{}$ may be non-symmetric, we have had to distinguish between right- and left-extensionality in the previous definition. We will use a completion procedure to ensure that the final bi-rewrite system satisfies that $R_\subseteq$ is right-I-extensionally closed, and that $R_\supseteq$ is left-I-extensionally closed.

The following lemma states that, if all inclusions in I are linear, then the extensionally closed property ensures the commutativity of $\xleftrightarrow[I]{*}$ and $\xrightarrow[I\backslash R]{}$. Notice that this property is stronger than the bi-confluence of cliffs required in the previous subsection.

LEMMA 3.9. (CRITICAL CLIFF LEMMA) *If all inclusions in I are linear, and $R_\subseteq$*

is right-I-extensionally closed, then $\xleftrightarrow[I_\subseteq]{*}$ *and* $\xrightarrow[I\backslash R_\subseteq]{*}$ *commute, i.e. they satisfy* $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[I\backslash R_\subseteq]{*} \subseteq \xrightarrow[I\backslash R_\subseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*}$.

Moreover, we also have $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[I\backslash R_\subseteq]{} \subseteq \xrightarrow[I\backslash R_\subseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} \cup \xleftrightarrow[I_\subseteq]{*}$.

Similarly for $\xleftrightarrow[I_\supseteq]{*}$ *and* $\xrightarrow[I\backslash R_\supseteq]{*}$ *if the later is left-I-extensionally closed.*

Proof. The conclusion of the lemma is equivalent to $\xleftrightarrow[I]{} \circ \xrightarrow[I\backslash R]{} \subseteq \xrightarrow[I\backslash R]{} \circ \xleftrightarrow[I]{*} \cup \xleftrightarrow[I]{*}$. Suppose that

$$a \xleftrightarrow[I]{}_{[p_1,\sigma,s\subseteq t]} b \xrightarrow[I\backslash R]{}_{[p_2,\sigma,l\to r]} c$$

where p_1 and p_2 are positions, σ is a substitution (assume that $\mathcal{FV}(t)\cap\mathcal{FV}(l)=\emptyset$), $s\subseteq t$ is an inclusion of I and $l \xrightarrow{\subseteq} r$ a rule of R. We have to consider the following four cases.

case $p_1|p_2$ If p_1 and p_2 are disjoint occurrences then both rewrite steps trivially commute.
case $p_1 \prec p_2$ Let v satisfy $p_2 = p_1\cdot v$. We have $\sigma(t)|_v \xleftrightarrow[I]{*} \sigma(l)$. There are two possibilities:

variable overlapping There exist two occurrences v_1 and v_2 satisfying $p_1\cdot v_1\cdot v_2 = p_2$ and being $t|_{v_1} = x$ a variable. If all inclusions in I are right-linear then $t|_{v_1}$ is the only occurrence of x in t, moreover if all inclusions are left-linear then x occurs at most once in s. Let v_1' be this occurrence of x in s, if there is one. First, we have $a|_{p_1\cdot v_1'\cdot v_2} \xleftrightarrow[I]{*} \sigma(l)$ and therefore $a \xrightarrow[I\backslash R]{}_{[p_1\cdot v_1'\cdot v_2,\sigma,l\to r]} a[\sigma(r)]_{p_1\cdot v_1'\cdot v_2}$. Second, we prove that $a[\sigma(r)]_{p_1\cdot v_1'\cdot v_2} \xleftrightarrow[I]{}_{[p_1,\sigma',s\subseteq t]} c$ where σ' is defined as $\sigma'(y)=\sigma(y)$ for any $y\neq x$, and $\sigma'(x)=\sigma(x)[\sigma(r)]_{v_2}$. Otherwise, if x does not occur in s then we have $a \xleftrightarrow[I]{}_{[p_1,\sigma',s\subseteq t]}, c$, where σ' is defined as above.
Notice that it is in this case, with variable overlapping, when we have to require both left- and right-linearity of $s\subseteq t$.

strict overlapping If v is a position in t, and $t|_v$ is not a variable, we are in the conditions of definition 3.8, i.e. $t|_v$ I-unifies with l being τ a minimum unifier, and we can generate an extensional rule $l' \xrightarrow{\subseteq} r' \stackrel{def}{=} \tau(s) \xrightarrow{\subseteq} \tau(t[r]_v)$ between $s\subseteq t$ and $l \xrightarrow{\subseteq} r$. Now, using our concrete definition of minimum I-unifier, a variant of the E-critical pair lemma (Jouannaud, 1983) ensures that $a|_{p_1}(\xleftrightarrow[I_\subseteq]{*} \cap \xleftrightarrow[I_\supseteq]{*})\rho(l')$ and $c|_{p_1}(\xleftrightarrow[I_\subseteq]{*} \cap \xleftrightarrow[I_\supseteq]{*})\rho(r')$ where $\sigma=\rho\circ\tau$. In particular, we have $a|_{p_1} \xleftrightarrow[I_\subseteq]{*} \rho(l')$ and $\rho(r') \xleftrightarrow[I_\subseteq]{*} c|_{p_1}$. In the equational case (Jouannaud and Kirchner, 1986) we would need to require the termination of the subterm relation modulo I. However, the stronger condition required in the definition of extensional closure allows us to disregard this requirement. If R is I-extensionally closed, then $l' \xrightarrow[I\backslash R]{} \circ \xleftrightarrow[I]{*} r'$. The $a|_{p_1} \xleftrightarrow[I]{*} \rho(l')$ rewrite steps are "covered" by the $\xrightarrow[I\backslash R]{}$ rewrite step, obtaining $a|_{p_1} \xrightarrow[I\backslash R]{} \circ \xleftrightarrow[I]{*} \rho(r') \xleftrightarrow[I]{*} c|_{p_1}$.
Notice that the proof $a|_{p_1} \xrightarrow[I\backslash R]{} \circ \xleftrightarrow[I]{*} \rho(r') \xleftrightarrow[I]{*} c|_{p_1}$ is normal, whereas if we only require extended rules to be bi-confluent, we would obtain $a|_{p_1} \xrightarrow[I\backslash R]{*} \circ \xleftrightarrow[I]{*} \circ \xleftarrow[I\backslash R]{*} \rho(r') \xleftrightarrow[I]{*} c|_{p_1}$. The later of course is not a normal proof and we would need to require the well-foundness of the strict subterm modulo I relation to prove that it is *smaller* than the original proof.

case $p_1 \succeq p_2$ The $a \xleftrightarrow[I]{} b$ rewrite step is *covered* by the $\xrightarrow[I\backslash R]{}$ rewrite step. Let v be the occurrence such that $p_2\cdot v = p_1$. We prove $a|_{p_2} \xleftrightarrow[I]{}_{[v,\sigma,s\subseteq t]} b|_{p_2}$, so $a|_{p_2} \xleftrightarrow[I]{} b|_{p_2} \xleftrightarrow[I]{*} \sigma(l)$ and we have $a \xrightarrow[I\backslash R]{}_{[p_2,\sigma,l\to r]} c$.

□

Notice that like in (Peterson and Stickel, 1981), and differently from (Jouannaud and Kirchner, 1986), the inclusions in I are required to be (both left- and right-) linear. However, thanks to the stronger condition required to extended rules, we can disregard the well-founded condition on the strict subterm modulo I relation.

Remark 3.10. The attentive reader will notice that our assumptions differ from those assumed in the equational case by Jouannaud and Kirchner (1986). Everywhere we have tried to require the weakest termination condition. Another option, similar to the one developed by Struth (1996),† starts from requiring the termination of the strict subterm relation modulo $\xleftrightarrow[I_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{}$ and the termination of the relation $(\xleftrightarrow[I_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{})^* \circ (\xrightarrow[R_\subseteq]{} \cup \xrightarrow[R_\supseteq]{})$, (notice that these conditions are stronger than the termination of $\xrightarrow[I\setminus R_\subseteq]{} \cup \xrightarrow[I\setminus R_\supseteq]{}$ assumed in this paper). They would allow us to relax the condition on extended rules $l \subseteq r$, requiring them to be bi-confluence $l \xrightarrow[I\setminus R_\subseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xleftarrow[I\setminus R_\supseteq]{*} r$ instead of $l \xrightarrow[I\setminus R_\subseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} r$. Moreover, if we also applied the notion of extended critical pair to *extended critical cliff*, requiring them to be bi-confluent, then we could drop the requirement on the linearity of the inclusions defining $\xleftrightarrow[I_\subseteq]{}$. These assumptions would also allow us to reproduce the classical results when any inclusion comes from an equality, and we have $\xleftrightarrow[I_\subseteq]{} = \xleftrightarrow[I_\supseteq]{}$ and $\xrightarrow[R_\subseteq]{} = \xrightarrow[R_\supseteq]{}$. The reader is invited to reproduce the proofs of Jouannaud and Kirchner (1986) for bi-rewrite systems.

The conclusion of the critical cliff lemma, not only ensures the bi-confluence of cliffs, but also allows to reduce the bi-confluence of peaks of the form $\xleftarrow[I\setminus R_\supseteq]{} \circ \xrightarrow[I\setminus R_\subseteq]{}$ to the confluence of the peaks of the form $\xleftarrow[R_\supseteq]{} \circ \xrightarrow[I\setminus R_\subseteq]{}$ or $\xleftarrow[I\setminus R_\supseteq]{} \circ \xrightarrow[R_\subseteq]{}$ using the following sequence of inclusions

$$\begin{array}{ll}
\xleftarrow[I\setminus R_\supseteq]{} \circ \xrightarrow[I\setminus R_\subseteq]{} \subseteq \xleftarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[I\setminus R_\subseteq]{} \subseteq \dots & \\
\xleftarrow[R_\supseteq]{} \circ (\xrightarrow[I\setminus R_\subseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} \cup \xleftrightarrow[I_\subseteq]{*}) \subseteq \cdots & \text{if cliffs commute} \\
\xrightarrow[I\setminus R_\subseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xleftarrow[I\setminus R_\supseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*} \subseteq \cdots & \text{if peaks are bi-confluent} \\
\xrightarrow[I\setminus R_\subseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xleftarrow[I\setminus R_\supseteq]{*} & \text{if cliffs commute}
\end{array}$$

For the bi-confluence of peaks we use a definition of (extended) critical pairs similar to the one introduced in the previous section.

Definition 3.11. *If* $l \xrightarrow{\subseteq} r \in R_\subseteq$ *and* $s \xrightarrow{\supseteq} t \in R_\supseteq$ *are two rewrite rules normalized apart, and* p *is a position in* s*, then*

(i) *if* $s|_p$ *is not a variable and* $s|_p$ I*-unifies with* l *being* σ *the minimum* I*-unifier, then*

$$\sigma(t) \subseteq \sigma(s[r]_p)$$

is a **(standard) critical pair** *of* $ECP(I \setminus R_\subseteq, R_\supseteq)$

(ii) *if* $s|_p = x$ *is a repeated variable in* s*,* F *is a term not sharing variables with* $s \xrightarrow{\supseteq} t$ *such that* $F|_q \xleftrightarrow[I_\subseteq]{*} \sigma(l)$ *for some* σ*, and* $l \xrightarrow[I\setminus R_\supseteq]{*} \circ \xleftrightarrow[I_\supseteq]{*} r$ *does not hold, then*

$$t[x \mapsto F] \subseteq \big(s[x \mapsto F]\big)[\sigma(r)]_{p\cdot q}$$

† Struth (1996) requires the termination of $\xleftrightarrow[I_\subseteq]{*} \circ \xrightarrow[R_\subseteq]{} \circ \xleftrightarrow[I_\subseteq]{*} \cup \xleftrightarrow[I_\supseteq]{*} \circ \xrightarrow[R_\supseteq]{} \circ \xleftrightarrow[I_\supseteq]{*}$, which is a weaker condition than the termination of $(\xleftrightarrow[I_\subseteq]{} \cup \xleftrightarrow[I_\supseteq]{})^* \circ (\xrightarrow[R_\subseteq]{} \cup \xrightarrow[R_\supseteq]{})$. However, then he has to use a condition for the local peaks and cliffs stronger than bi-confluence.

is an **(extended) critical pair** *of* $ECP(I\backslash R_\subseteq, R_\supseteq)$.
Moreover, if $\xleftrightarrow[I_\subseteq]{*}$ *is symmetric, then we can restrict extended critical pairs to those which satisfy* $F|_q = l$*, like in definition 2.11.*

The set $ECP(I\backslash R_\supseteq, R_\subseteq)$ *can be defined similarly.*

Again we have had to introduce critical pair schemes which may generate infinitely many critical pairs. Unlike definition 2.11 of previous section, here, if $\xleftrightarrow[I_\subseteq]{}$ is non-symmetric, then the whole term F is undetermined, not only the context $F[\cdot]_q$. The only restriction on $F|_q$ is $F|_q \xleftrightarrow[I_\subseteq]{*} \sigma(l)$ for some substitution σ.

Using this extended definition of critical pairs and the definition of extensionally closed bi-rewrite system we can prove the following theorem which characterizes the strong Church-Rosser property of a $\langle R_\subseteq, R_\supseteq\rangle$ bi-rewrite system modulo I.

Theorem 3.12. (Critical Pair Theorem) *Given two sets of rules* $R_\subseteq$ *and* $R_\supseteq$ *and a set of inclusions* I*, if* $\xrightarrow[I\backslash R_\subseteq]{} \cup \xrightarrow[I\backslash R_\supseteq]{}$ *is terminating,* $\xrightarrow[I\backslash R_\subseteq]{}$ *is right-*I*-extensionally closed,* $\xrightarrow[I\backslash R_\supseteq]{}$ *is left-*I*-extensionally closed, all inclusions in* I *are linear, and all standard and extended critical pairs* $ECP(I\backslash R_\subseteq, R_\supseteq)$ *and* $ECP(R_\subseteq, I\backslash R_\supseteq)$ *are bi-confluent, then* $\langle R_\subseteq, R_\supseteq\rangle$ *is (strongly) Church-Rosser modulo* I*.*

Proof. We use lemma 3.4 of previous sub-section to prove the Church-Rosser property. We are in the conditions of lemma 3.9, therefore we can suppose that cliffs commute. As we have commented, this condition is stronger than the bi-confluence of global cliffs required by lemma 3.4. Let's concentrate on the bi-confluence of global peaks.

Assume that

$$a \xleftarrow[I\backslash R_\supseteq]{} {}_{[p_1,\sigma,s\to t]}\ b \xrightarrow[I\backslash R_\subseteq]{} {}_{[p_2,\sigma,l\to r]}\ c$$

If $p_1|p_2$, as in the commutativity case, both rewrite steps trivially commute and we can reduce a and c to the same term $b[\sigma(t)]_{p_1}[\sigma(r)]_{p_2} = b[\sigma(r)]_{p_2}[\sigma(t)]_{p_1}$.

Otherwise, we can suppose, without lose of generality that $p_1 \preceq p_2$. We have:

$$a \xleftarrow[R_\supseteq]{} {}_{[p_1,\sigma,s\to t]}\ a' \xleftrightarrow[I_\subseteq]{*} b \xrightarrow[I\backslash R_\subseteq]{} {}_{[p_2,\sigma,l\to r]}\ c$$

where all the $\xleftrightarrow[I_\subseteq]{}$ rewrite steps between a' and b takes place bellow the position p_1 (notice that they are covered by the $\xrightarrow[I\backslash R_\supseteq]{}$ rewrite step). Now, lemma 3.9 allow us to commute these $\xleftrightarrow[I_\subseteq]{}$ steps and the $\xrightarrow[I\backslash R_\subseteq]{}$ step. We will have either

$$a \xleftarrow[R_\supseteq]{} {}_{[p_1,\sigma,s\to t]}\ a' \xleftrightarrow[I_\subseteq]{*} c$$

or

$$a \xleftarrow[R_\supseteq]{} {}_{[p_1,\sigma,s\to t]}\ a' \xrightarrow[I\backslash R_\subseteq]{} {}_{[p_2',\sigma,l\to r]}\ c' \xleftrightarrow[I_\subseteq]{*} c$$

where p_2' is a position bellow or equal to p_1 (notice that the original steps were all them bellow p_1, therefore, after commuting them, the resulting steps will also be bellow p_1).

Taking into account that we can also commute $\xleftrightarrow[I_\subseteq]{}$ steps and $\xrightarrow[I\backslash R_\supseteq]{}$ steps, we only have to prove that local peaks of the form:

$$t_1 \xleftarrow[R_\supseteq]{} {}_{[p_1,\sigma,s\to t]}\ t_2 \xrightarrow[I\backslash R_\subseteq]{} {}_{[p_2,\sigma,l\to r]}\ t_3$$

where $p_1 \preceq p_2$, are bi-confluent.

Let v be the occurrence such that $p_2 = p_1 \cdot v$. There are three possibilities:

Strict overlapping Position v is a non-variable occurrence of s. This case works as case *strict overlapping* in the proof of critical cliff lemma 3.9. That is, there exists a standard critical pair $l' \subseteq r'$, obtained I-unifying $s|_v$ and l, such that $t_1|_{p_1} \xleftrightarrow[I_\subseteq]{*} \rho(l')$ and $\rho(r') \xleftrightarrow[I_\subseteq]{*} t_3|_{p_1}$. If standard critical pairs are confluent then $\rho(l') \xrightarrow[I\setminus R_\subseteq]{*} \circ \xleftrightarrow[I_\subseteq]{*} \circ \xleftarrow[I\setminus R_\supseteq]{*} \rho(r')$. Finally, if lemma 3.9 holds, then pair $t_1 \subseteq t_3$ is bi-confluent.

Non-repeated variable overlapping Subterm $s|_v$ is a —or is bellow a— non-repeated variable, i.e. there exist two positions such that $v = v_1 \cdot v_2$ and $s|_{v_1} = x$ is a non-repeated variable of s. This case works similarly to case *variable overlapping* of lemma 3.9. That is, we can rewrite t_1 and t_3 into a common term in the following way. We apply the rewrite step $\sigma(x) = \xrightarrow[I\setminus R_\subseteq]{} {}_{[v_2,\sigma,l\to r]}\ \sigma(x)[\sigma(r)]_{v_2}$ to any occurrence of x in t, i.e. to some sub-terms $\sigma(x)$ of t_1. On the other side, we apply the rule $s \xrightarrow{\supseteq} t$ to the position p_1 of t_3, but using the substitution σ' defined as $\sigma'(y) = \sigma(y)$ for any $y \neq x$ and $\sigma'(x) = \sigma(x)[\sigma(r)]_{v_2}$ instead of σ. It can be proved that in both cases we obtain the same result.

Repeated variable overlapping Subterm $s|_v$ is a —or is below a— repeated variable x of s, i.e. there exists a pair of positions $v_1 \cdot v_2 = v$ such that $s|_{v_1}$ is a repeated variable x of s.

In this case the divergence $t_1|_{p_1} \subseteq t_3|_{p_1}$ being studied is an instance of the extended critical pair $l' \subseteq r'$ of the form:

$$t[x \mapsto F] \subseteq \big(s[x \mapsto F]\big)[\tau(r)]_v$$

where $F = \sigma(x) = t_2|_{(p_1 \cdot v_1)}$ and $\tau(y) = \sigma(y)$ for any $y \in \mathcal{FV}(l)$. That is, we can prove that $t_1|_{p_1} = \rho(l')$ and $t_3|_{p_1} = \rho(r')$ where $\rho(y) = \sigma(y)$ for any $y \in \mathcal{FV}(s)\setminus\{x\}$. Now, if any extended critical pair of $ECP(I\setminus R_\subseteq, R_\supseteq)$ is bi-confluent, the corresponding proof instantiated by ρ will be a proof for $t_1|_{p_1} \subseteq t_3|_{p_1}$.

If $\xleftrightarrow[I_\subseteq]{*}$ is symmetric, then we can use the extended critical pair $l' \subseteq r'$ of the form:

$$t[x \mapsto F] \subseteq \big(s[x \mapsto F]\big)[r]_v$$

where $F = \sigma(x)[l]_q$.

In this case we can prove that $t_1|_{p_1} \xleftrightarrow[I_\subseteq]{*} \rho(l')$ and $t_3|_{p_1} \xleftrightarrow[I_\subseteq]{*} \rho(r')$, where $\rho(y) = \sigma(y)$ for any $y \in \mathcal{FV}(s) \cup \mathcal{FV}(l)\setminus\{x\}$. Now, if extended critical pairs are bi-confluent then $t_1|_{p_1} \xleftrightarrow[I_\subseteq]{*} \rho(l') \xrightarrow[I\setminus R_\subseteq]{*} \xleftrightarrow[I_\subseteq]{*} \xleftarrow[I\setminus R_\supseteq]{*} \rho(r') \xleftrightarrow[I_\supseteq]{*} t_3|_{p_1}$, and if $\xleftrightarrow[I_\subseteq]{*}$ is symmetric, and cliffs commute, then the pair t_1, t_3 is bi-confluent.

Finally, if $l \xrightarrow[I\setminus R_\supseteq]{*} \circ \xleftrightarrow[I_\supseteq]{*} r$ holds, and therefore the pairs $l' \subseteq r'$ are not extended critical pairs, then we can make pair $t_1 \subseteq t_3$ bi-confluent using the same technique as in the equational case.

□

4. An Example: Towards a Completion Procedure

As we said in the previous sections, bi-rewriting compared with equational rewriting, faces the extra difficulty of a possible infinite set of critical pairs. Non-left-linear rules may generate what we called critical pair schemes (see definitions 2.11 and 3.11). In this section instead of giving the completion procedure we sketch out the possibilities of completion à la Knuth-Bendix of an example of bi-rewrite system by means of rule schemes. Other completion methods, like unfailing completion (Bachmair *et al.*, 1989) are also suitable of being applied to automate the deduction in theories with monotonic order relations, using the same notion of extended critical pair.

The inclusions defining the theory of the union operator can be oriented following a simplification ordering as follows:

$$\begin{array}{ll} r_1 & X \cup X \xrightarrow{\subseteq} X \\ r_2 & X \cup Y \xrightarrow{\supseteq} X \\ r_3 & X \cup Y \xrightarrow{\supseteq} Y \end{array}$$

Although the standard critical pairs (*scp*) of this system are bi-confluent, the presence of the non-left-linear rule $X \cup X \xrightarrow{\subseteq} X$ also makes necessary the consideration of the extended critical pairs (*ecp*). We will do this in two steps. First, we consider scp and the finite subset of ecp of the particular form $\langle t[x \mapsto l], \big(s[x \mapsto l]\big)[r]_p)\rangle$ where $s|_p = x$ is a repeated variable in the non-left-linear rule $\langle s \xrightarrow{\subseteq} t\rangle \in R_{\subseteq}$ and $\langle l \xrightarrow{\supseteq} r\rangle \in R_{\supseteq}$ being the other rule. It corresponds to the general extended critical pair definition where the context $F[\cdot]_q$ is a hole $[\cdot]$ itself. Using the standard Knuth-Bendix completion procedure and a reduction ordering, we generate, among others, the following rules:

$$\begin{array}{lll} r_4 & Y \cup (X \cup Y) \xrightarrow{\subseteq} X \cup Y & \text{ecp from } r_1 \text{ and } r_3 \\ r_5 & Y \cup X \xleftrightarrow{\subseteq} X \cup Y & \text{scp from } r_2 \text{ and } r_4 \\ & & \\ r_6 & (X \cup Y) \cup Y \xrightarrow{\subseteq} X \cup Y & \text{ecp from } r_1 \text{ and } r_3 \\ r_7 & (X \cup Y) \cup (Y \cup Z) \xrightarrow{\subseteq} X \cup (Y \cup Z) & \text{ecp from } r_2 \text{ and } r_6 \\ r_8 & (X \cup Y) \cup Z \xleftrightarrow{\subseteq} X \cup (Y \cup Z) & \text{scp from } r_3 \text{ and } r_7 \end{array}$$

Rules r_5 and r_8, corresponding to the commutativity and associativity (AC) properties of the union, make redundant any other rule generated by the subset of ecp we are considering. It is well known that these rules can not be oriented in a reduction ordering. This fact makes necessary the use of $\langle\{r_1\}, \{r_2\}\rangle$ bi-rewriting modulo $I = \{r_5, r_8\}$. Notice that in this case the relation defined by non-orientable rules is symmetric, i.e. $\xleftrightarrow[I_{\subseteq}]{*} = \xleftrightarrow[I_{\supseteq}]{*}$, thus we can use the standard algorithms of AC-matching and AC-unification, as well as the flat notation for the infix operator $\cup$.

Let's consider now the general form of ecp when I is symmetric, i.e. $\langle t[x \mapsto F[l]_q], \big(s[x \mapsto F[l]_q]\big)[r]_{p \cdot q}\rangle$ where $F[\cdot]_q$ is a context. Using them we generate an extended critical pair which is made bi-confluent adding the following rule scheme:

$$r_9 \quad F[X] \cup F[X \cup Y] \xrightarrow{\subseteq} F[X \cup Y] \quad \text{ecp from } r_1 \text{ and } r_2$$

The orientation of this rule does not depend on the instance we take of the critical pair scheme, and it will be the same for any simplification ordering. This rule scheme generates the following scp:

$$F[X] \cup F[Y] \subseteq F[X \cup Y] \quad \text{scp from } r_2 \text{ and } r_9$$

Now, the orientation of this critical pair depends on the reduction ordering being used. If we use a lexicographic path ordering where $\cup$ is greater than any other symbol of the signature, then it will be oriented as follows for any instance of the critical pair.

$$r_{10} \quad F[X] \cup F[Y] \xrightarrow{\subseteq} F[X \cup Y] \quad \text{from } r_2 \text{ and } r_9$$

Now r_9 is subsumed by r_1 and r_{10}.

Notice that we are dealing with rule schemes instead of ordinary rules, thus we can not continue the completion process unless we have a critical pair definition for rule schemes.

The repeated context $F[\cdot]$ in the left hand side of the rule originates a problem similar to the one caused by non-left-linear rules. We can consider the following particular form of r_{10}, where we suppose that $F[\cdot]$ is a context containing $X' \cup Y'$ as a subexpression, i.e. $F[\cdot] \stackrel{def}{=} G[\cdot, X' \cup Y']$.

$$r_{11} \quad G[X, X' \cup Y'] \cup G[Y, X' \cup Y'] \xrightarrow{\subseteq} G[X \cup Y, X' \cup Y']$$

This instantiation of the rule scheme r_{10} generates new non-confluent critical pairs with r_1, which introduces the following rule schemes:

$$\begin{array}{l} G[X, X'] \cup G[Y, X' \cup Y'] \xrightarrow{\subseteq} G[X \cup Y, X' \cup Y'] \\ G[X, X'] \cup G[Y, Y'] \xrightarrow{\subseteq} G[X \cup Y, X' \cup Y'] \end{array}$$

It can be induced then that the completion process would introduce infinitely many rule schemes with the form:

$$r_{12} \quad G[X_1, \ldots, X_n] \cup G[Y_1, \ldots, Y_n] \xrightarrow{\subseteq} G[X_1 \cup Y_1, \ldots, X_n \cup Y_n]$$

for any $n > 0$.

If we are interested in an unfailing completion procedure, the fact that this set of rules would be infinite is not relevant, but we can not obtain a canonical bi-rewrite system (in the sense of Knuth-Bendix completion) in this way. However, in this case, if the signature $\mathcal{F}$ is finite, these (infinite) set of rule schemes will be subsumed by the following (finite) set of rules:

$$r_{13}^{(f)} \quad f(X_1, \ldots, X_n) \cup f(X'_1, \ldots, X'_n) \xrightarrow{\subseteq} f(X_1 \cup X'_1, \ldots, X_n \cup X'_n) \quad \text{for any } n > 0 \text{ and any } f \in \mathcal{F}_n$$

To prove this result we decompose an application of the rule scheme r_{12} into simple applications of the rules r_{13} using the following compositional property:

$$\begin{array}{rcl} F[G[X_1 \ldots X_n]] \cup F[G[Y_1 \ldots Y_n]] & \xrightarrow{\subseteq} & F[G[X_1 \ldots X_n] \cup G[Y_1 \ldots Y_n]] \\ & \xrightarrow{\subseteq} & F[G[X_1 \cup Y_1 \ldots X_n \cup Y_n]] \end{array}$$

Finally, using this *"manual"* completion process we obtain the canonical $\langle R_{\subseteq}, R_{\supseteq} \rangle$ bi-rewriting modulo I system shown in figure 3. Rules r_1^{ext} and r_{13}^{ext} are the I-extensions of the rules r_1 and r_{13}.

$$R_{\subseteq} = \left\{ \begin{array}{ll} r_1 & X \cup X \xrightarrow{\subseteq} X \\ r_1^{ext} & X \cup X \cup Y \xrightarrow{\subseteq} X \cup Y \\ \forall f \in \mathcal{F}^n & \\ r_{13} & f(X_1 \ldots X_n) \cup f(Y_1 \ldots Y_n) \xrightarrow{\subseteq} f(X_1 \cup Y_1 \ldots X_n \cup Y_n) \\ r_{13}^{ext} & f(X_1 \ldots X_n) \cup f(Y_1 \ldots Y_n) \cup Z \xrightarrow{\subseteq} f(X_1 \cup Y_1 \ldots X_n \cup Y_n) \cup Z \end{array} \right.$$

$$R_{\supseteq} = \left\{ \; r_2 \quad X \cup Y \xrightarrow{\supseteq} X \right.$$

$$I = \left\{ \begin{array}{ll} r_5 & Y \cup X \xleftrightarrow{\subseteq} X \cup Y \\ r_8 & (X \cup Y) \cup Z \xleftrightarrow{\subseteq} X \cup (Y \cup Z) \end{array} \right.$$

Figure 3. A canonical bi-rewrite system for the inclusion theory of the union.

As the reader may suppose, this canonical bi-rewrite system can be easily extended to automate the deduction in non-distributive free lattice theory. We only have to duplicate all rules interchanging $\subseteq$ by $\supseteq$, and changing $\cup$ by $\cap$.

5. Why Inclusions and not Equations

In section 4 we have seen the possibility of modeling the deduction in a non-distributive free lattice by a canonical bi-rewrite system. This represents an advantage of the inclusion theory over the equational theory of lattices because there is not a canonical rewrite system for the equational theory of lattices (Freese *et al.*, 1993). In general inclusions express weaker constraints between terms than equations. Even in the case of lattices where inclusions may be modeled by equations —the inclusion $a \subseteq b$ is modeled by $a \cup b = b$ or by $a \cap b = a$— inclusions are more natural and have some advantages. The transitivity and monotonicity of inclusions which are captured implicitly by bi-rewrite systems, must be "implemented" explicitly by equational rewrite rules. Let's consider an example. The inclusions $a \subseteq b$ and $b \subseteq c$ can be oriented like $a \xrightarrow{\subseteq} b$ and $b \xrightarrow{\subseteq} c$ and we can prove $a \subseteq c$ rewriting a into b and b into c. However, their translation into equations results in two rules $a \cup b \longrightarrow b$ and $b \cup c \longrightarrow c$. These rules generate non-confluent critical pairs with the other rules $X \cap (X \cup Y) \longrightarrow X$ and $X \cup (X \cap Y) \longrightarrow X$ defining the union and intersection, and the completion process leads to add the following rules $a \cap b \longrightarrow a$ and $b \cap c \longrightarrow b$. And, what is worse, it introduces the rules $a \cup c \longrightarrow c$ and $a \cap c \longrightarrow a$. It means that in general the completion of a theory where the sequence $a_1 \subseteq \ldots \subseteq a_n$ can be proved leads to add rules $a_i \cup a_j \longrightarrow a_j$ and $a_i \cap a_j \longrightarrow a_i$ for any $i < j$, during the completion process.

The transitivity of inclusions is not captured by the transitivity of the equality relation or by the transitivity of the rewrite relation $\xrightarrow{*}$, weakening in this way the power of rewrite systems, and loosing in most cases the possibility of having a canonical rewrite system for a theory.

Moreover, the stability (closure for congruence) of the rewrite relation captures the congruence property for $=$, but not the monotonicity property for $\subseteq$. This would make necessary to consider the inclusion $f(X) \subseteq f(X \cup Y)$ and the corresponding rule $f(X) \cup f(X \cup Y) \longrightarrow f(X \cup Y)$ for each symbol f in the signature if we use the implementation described below.

6. Codifying Rule Schemes by means of Second-Order Rules

We face now the problem of codifying rule schemes using a restricted form of second-order typed λ-calculus. From now on we will be concerned with the simply typed second-order λ-calculus. Thus, we will deal with a set of types $\mathcal{T} = \bigcup_{n \geq 1} \mathcal{T}^n$ built up over a set $\mathcal{T}^1$ of base —or first order— types; where, as usual, $\mathcal{T}^n$ is the set of n-ordered types defined inductively as the minimum set containing $\mathcal{T}^{n-1}$ and such that if $\tau \in \mathcal{T}^{n-1}$ and $\tau' \in \mathcal{T}^n$ then $\tau \rightarrow \tau' \in \mathcal{T}^n$. Terms of the simply typed second-order λ-calculus $\mathcal{T}(\mathcal{F}, \mathcal{X})$ are defined over a signature of third-order typed constants $\mathcal{F} = \bigcup_{\tau \in \mathcal{T}^3} \mathcal{F}_\tau$ and second-order typed variables $\mathcal{X} = \bigcup_{\tau \in \mathcal{T}^2} \mathcal{X}_\tau$. The typing relation $t : \tau$ is defined by the following set of inference rules

$$\frac{\{c \in \mathcal{F}_\tau\}}{c : \tau} \qquad \frac{\{x \in \mathcal{X}_\tau\}}{x : \tau} \qquad \frac{x : \tau \qquad t : \tau'}{\lambda x \,.\, t \;:\; \tau \rightarrow \tau'} \qquad \frac{t : \tau \rightarrow \tau' \qquad t' : \tau}{t(t') : \tau'}$$

The term t is said to be a well-formed n-order typed term, noted $t \in \mathcal{T}^n(\mathcal{F}, \mathcal{X})$, if $t : \tau$

can be inferred from the set of rules below and $\tau \in \mathcal{T}^n$. The set of *free variables* of a term (noted $\mathcal{FV}(t)$), replacement (noted $t[X \mapsto u]$), and other concepts commonly used in λ-calculus are defined as usual (Barendregt, 1981; Hindley and Seldin, 1986). We will note free variables with capital letters (by $X, Y, Z, \ldots$ when they are first-order typed and by $F, G, H, I, \ldots$ when they are second-order typed), bound variables and constants are noted using lower case letters.

A *substitution* $\sigma = [X_1 \mapsto t_1, \cdots, X_n \mapsto t_n]$ is a mapping from a finite set of variables $\{X_1, \ldots, X_n\} \subseteq \mathcal{X}$ to $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that X_i and t_i have the same type. This mapping is extended as a type-preserving mapping $\sigma : \mathcal{T}(\mathcal{F}, \mathcal{X}) \to \mathcal{T}(\mathcal{F}, \mathcal{X})$ defined by $\sigma(u) \stackrel{def}{=} (\lambda X_1 \ldots X_n \,.\, u)(t_1, \ldots, t_n) = \big(u[X_1 \mapsto t_1] \ldots\big)[X_n \mapsto t_n]$.[†] A substitution σ is said to be a *unifier* of two given expressions t and u if $\sigma(t) =_{\beta\eta} \sigma(u)$ and σ is idempotent.[‡] A partial order between the unifiers of two given terms and *minimum unifiers* can be defined as usual, i.e. we say that $\rho \preceq \sigma$ if there exists a substitution π such that $\sigma = \pi \circ \rho$, and we say that σ is a minimum unifier if for any other unifier ρ satisfying $\rho \preceq \sigma$ we have also $\sigma \preceq \rho$.

The inclusion theory of the union operator will be used throughout to motivate the definition of second-order bi-rewrite systems. In section 4 we proved the existence of a canonical first-order bi-rewrite system for such a theory (shown in figure 2). The same example is completed in section 9 for the second-order case. Our intention is to replace the set of rules of such example

$$f(X_1, \ldots, X_n) \cup f(Y_1, \ldots, Y_n) \xrightarrow{\subseteq} f(X_1 \cup Y_1, \ldots, X_n \cup Y_n)$$

by second-order rules. If we take up again the completion process described in section 4, we have that this set of rules is generated by the rule schema $F[X] \cup F[Y] \xrightarrow{\subseteq} F[X \cup Y]$, which results of making bi-confluent an extended critical pair, and where $F[\cdot]$ denotes a context. We will see now that we can *translate* this rule scheme into the second-order rule $G(X) \cup G(Y) \xrightarrow{\subseteq} G(X \cup Y)$, where now G denotes a second-order typed variable. Then, it is easy to see that this second-order rule subsumes the previous rule schema because the function variable G can be instantiated by $\lambda x \,.\, F[x]$. However, it does not subsume other rules like $f(X_1, \ldots, X_n) \cup f(Y_1, \ldots, Y_n) \xrightarrow{\subseteq} f(X_1 \cup Y_1, \ldots, X_n \cup Y_n)$ for $n \geq 2$. To obtain second-order rules subsuming them we must complete the system by generating all the critical pairs between $G(X) \cup G(Y) \xrightarrow{\subseteq} G(X \cup Y)$ and some other rules.

The simply typed second-order λ-calculus is enough to *model* an untyped first-order language with context variables, like the one described by Comon (1993). In such a model, we can suppose that there exists a unique first-order type $Term \in \mathcal{T}^1$. Any n-ary symbol f of the signature is interpreted as a unary second-order typed constant $f : Term \to \stackrel{n}{\ldots} \to Term \to Term$, any variable X as a first-order variable $X : Term$ and any context variable $F[\cdot]$ as a second-order variable $F : Term \to Term$.

† Notice that $\big(u[X_1 \mapsto t_1]\big)[X_2 \mapsto t_2] = u[X_1 \mapsto t_1, X_2 \mapsto t_2]$, but in general, $u[X_1 \mapsto t_1, X_2 \mapsto t_2] \neq u[X_2 \mapsto t_2, X_1 \mapsto t_1]$.

‡ The relation $=_{\beta\eta}$ is the congruence defined by the β and η rules of the λ-calculus. A sufficient condition for the idempotence of σ is $\mathcal{Dom}(\sigma) \cap \mathcal{FV}(\sigma) = \emptyset$. This restriction does not suppose a loss of generality.

6.1. Some Problems of Second-Order Rewrite Systems

The use of *full* simply typed second-order λ-calculus in rewrite systems is not free from problems. If we unify a term (pattern) with a ground term (a term without free variables), the resulting unifier(s) do not necessarily instantiate all the free variables of the pattern. For instance, if we unify the pattern $F(X)$ with the ground expression $f(a)$, a minimum unifier σ may assign $\sigma(F) = \lambda x \,.\, f(a)$ and leave X free. It means that, although all variables appearing in the right part of a rule would also appear in its left part, not all the instantiations of such rule will satisfy this property. Therefore, the use of this rule can introduce new free variables during the rewrite process. For instance, the rule $F(X) \longrightarrow X$ satisfies $\mathcal{FV}(X) = \{X\} \subseteq \{X, F\} = \mathcal{FV}(F(X))$, even so it introduces a fresh variable X when is used to rewrite a into X using the substitution $\sigma = [F \mapsto \lambda x.a]$. That problem prevents the orientation of the rules to obtain a terminating rewrite system. In the previous example, we can rewrite $a \longrightarrow a \longrightarrow a \longrightarrow \cdots$ using the rule $F(X) \longrightarrow X$ and the substitution $\sigma = [F \mapsto \lambda x \,.\, a][X \mapsto a]$. The first-order matching satisfies the following property: given a pair of terms t and u there exists at most one substitution σ such that $\mathcal{Dom}(\sigma) \subseteq \mathcal{FV}(t)$ and $\sigma(t) = u$. This result does not hold in general for second-order languages. It means that a second-order rewrite relation can be infinitely branching and many properties of term rewriting systems do not hold. In particular, a second-order terminating rewrite system is not necessarily quasi-terminating.

In the next section we define the *linear second-order typed λ-calculus* which avoids these problems (see lemma 7.3). The same kind of problems are studied by Nipkow (Nipkow, 1991; Nipkow, 1992) to justify the definition of *higher-order rewrite systems* based on *patterns*. (A term t in $\beta\eta$-normal form is said to be a *pattern* if every occurrence of a free variable F is in a subterm $F(\overline{u_n})$ such that $\overline{u_n}$ is a list of distinct bound variables (Nipkow, 1991, definition 3.1)). Our approach can be seen as a new notion of higher-order rewrite systems based on the linear second-order typed λ-calculus.

7. A Second-Order Unification Procedure

Like in the first-order case, to prove the completeness of a second-order bi-rewrite system we have to generate all the possible (extended) critical pairs between rules in $R_{\subseteq}$ and rules in $R_{\supseteq}$ and prove their bi-confluence. This process requires the use of a unification procedure.

The first sound and complete second-order unification procedure was described by Pietrzykowski (1973), and subsequently a modified version of this algorithm was proposed to solve the unification problem for the simply typed λ-calculus (Jensen and Pietrzykowski, 1976). Based on it, Huet (1975) proposed the computation of the so called independent pre-unifiers using a pre-unification procedure. This procedure does not try to solve the flexible-flexible pairs of a unification problem for which there always exist a unifier, thus a pre-unification procedure is enough if we only want to check if a unification problem is satisfiable. Unfortunately, the simply typed λ-calculus unification problem, and even the second-order unification problems are undecidable (Goldfarb, 1981).

Since then many decidable classes of higher-order unification problems have been described. Miller (Miller, 1991a; Miller, 1991b) in the context of logic programming and Nipkow (Nipkow, 1991; Nipkow, 1992) in the context of rewrite systems, propose a restricted higher-order language –which expressions they call *patterns*– preserving the good properties of the first-order logic. If there exists a minimum unifier of two patterns, then it is unique. They also define a unification algorithm (Nipkow, 1991, theorem 3.2) to find *this* most general pattern unifier and prove its termination. However, in our case we need a more expressive language. If we consider the rule $G(X) \cup G(Y) \xrightarrow{\subseteq} G(X \cup Y)$ for example, we realize that neither the left-part nor the right part of the rule is a pattern. In general, if we look at the particular form of extended critical pairs we will see that they always contain a subexpression $F(t)$ where F is a free variable and t is the right hand side of a rewrite rule, so we can not suppose that t is a list of distinct bounded variables, as the definition of patterns requires. Recently Prehofer has proved in his thesis (Prehofer, 1995) decidability results for some unification problems based on Nipkow's patterns. He proves, for instance, that unification of linear second-order systems is decidable (theorem 5.3.1). Unfortunately, *linear* refers here to the system, not to terms: a linear second-order system of equations is of the form $\overline{\lambda\overline{x_k}\,.\,X_n(\overline{t_{n_m}}) \stackrel{?}{=} \lambda\overline{x_k}\,.\,t_n}$ where $\overline{X_n}$ are distinct and not occurring elsewhere second-order variables and $\overline{\lambda\overline{x_k}\,.\,t_{n_m}}$ and $\overline{\lambda\overline{x_k}\,.\,t_n}$ are patterns. This decidable case neither covers our needs.

Comon (1993) describes a second-order language based on *context variables* (a second-order language without λ-abstractions and where second-order variables are restricted to be unary). He also proves a decidability result for the unification problem in this language and provides a unification algorithm. However, a rather strong condition is imposed: any occurrence of a free variable F is always applied to the same argument t. This restriction is also violated in our case: in the rule $G(X) \cup G(Y) \xrightarrow{\subseteq} G(X \cup Y)$, the second-order variable G is applied to two different terms, X and Y. Finally, Schmidt-Schauß (1995) proves that second-order unification of stratified terms, in the scope of *context unification* is decidable. Here stratified terms means that the string of second-order variables on the path from the root of a term to every occurrence of a given variable is always the same.

Other cases are currently being proposed, but none of them is adequate for the computation of extended critical pairs. The most specific unification problem subsuming ours is the general second-order case studied by Pietrzykowski. However our particular case turns up to be attractive as long as it enjoys better properties, as we shall see at the end of this section. On the other hand, our linear second-order unification problem generalizes the associative unification (Makanin, 1977) and the monadic second-order unification (Farmer, 1988) problems. These unification problems are known to be decidable, although such results are not as easy to prove as one may suppose at first glance. Thus, as far as we know, the decidability of our linear second-order λ-calculus unification problem is still an open question, and the procedure we give in this section is in general non-terminating.

We define now what we have called the *linear second-order unification problem.* As we will see, this is a generalization of the context unification problem where we can have (a kind of restricted) λ-abstractions, and free second-order variables are not restricted to be unary. The main idea is to define a second-order calculus where λ-abstractions always bind one and only one occurrence of a variable.

The inference rules to define well-typed terms $t : \tau$ are the following ones.

$$\frac{x \in \mathcal{X}_\tau}{x : \tau} \qquad \frac{c \in \mathcal{F}_\tau}{c : \tau} \qquad \frac{\begin{array}{cc} x : \tau_1 & t : \tau_2 \\ \multicolumn{2}{c}{\{x \text{ occurs once in } t\}} \end{array}}{\lambda x\,.\,t : \tau_1 \to \tau_2} \qquad \frac{t : \tau_1 \to \tau_2 \qquad u : \tau_1}{t(u) : \tau_2}$$

We also consider the β and η equations:

$$\begin{array}{l} (\lambda x\,.\,t)(u) =_\beta t[x \mapsto u] \\ \lambda x\,.\,t(x) =_\eta t \end{array}$$

Notice that the side condition $x \notin \mathcal{FV}(t)$ is not necessary in the η-rule because if $\lambda x \,.\, t(x)$ is well-typed then this condition is ensured. These equations used as rewrite rules:

$$
\begin{array}{ll}
(\lambda x \,.\, t)(u) \rightarrow_\beta t[x \mapsto u] & \\
t \rightarrow_\eta \lambda x \,.\, t(x) & \text{if it does not introduce a new } \beta\text{-redex and } x \notin \mathcal{FV}(t)
\end{array}
$$

constitute a normalizing rewrite system. Notice that these rules transform linear second-order λ-expressions into linear second-order λ-expressions of the same type. The normal form of a term t is denoted by $t\downarrow_{\beta\eta}$ and has the form $\lambda x_1 \ldots x_n \,.\, a(t_1, \ldots, t_m)$ where a can be either a bound variable, a free variable or a constant, $a(t_1, \ldots, t_m)$ is a first-order typed term, and $t_1 \ldots t_m$ are normalized terms.

Definition 7.1. *A* **linear second-order unification problem** *is a finite set of pairs of linear second-order terms* $\{t_1 \stackrel{?}{=} u_1, \ldots, t_n \stackrel{?}{=} u_n\}$ *such that* t_i *and* u_i *have the same type.*

Given a second-order substitution σ*, we say that it is* **lineal second-order substitution** *iff for any* $X \in \mathcal{Dom}(\sigma)$ *the term* $\sigma(X)$ *is lineal, and has the same second-order type as* X*.*

We can prove the following lemmas, which hold in the linear second-order λ-calculus, but not in the simply typed second-order λ-calculus.

Lemma 7.2. *For any pair of linear second-order terms* t *and* u*, if* $t =_{\beta\eta} u$ *then* $\mathcal{FV}(t) = \mathcal{FV}(u)$*.*

Proof. For the η-equation it is trivial because $\mathcal{FV}(\lambda x \,.\, t(x)) = \mathcal{FV}(t) \setminus \{x\}$, but since $x \notin \mathcal{FV}(t)$ we have $\mathcal{FV}(\lambda x \,.\, t(x)) = \mathcal{FV}(t)$. For the β-equation it is necessary to take linear terms. Thus, if $(\lambda x \,.\, t)(u)$ is a well-formed linear term then $x \in \mathcal{FV}(t)$. Therefore $\mathcal{FV}(t[x \mapsto u]) = \big(\mathcal{FV}(t) \setminus \{x\}\big) \cup \mathcal{FV}(u) = \mathcal{FV}\big((\lambda x \,.\, t)(u)\big)$. Notice that if $\lambda x \,.\, t$ is not linear then it is possible to have $x \notin \mathcal{FV}(t)$ and therefore $\mathcal{FV}(t[x \mapsto u]) = \mathcal{FV}(t) \setminus \{x\} \neq \mathcal{FV}((\lambda x \,.\, t)(u))$. $\Box$

Lemma 7.3. *Given a pair of linear second-order terms* t *and* u*, if* u *is ground* ($\mathcal{FV}(u) = \emptyset$)*, and the linear substitution* σ *satisfies* $\mathcal{Dom}(\sigma) \subseteq \mathcal{FV}(t)$*, and* $\sigma(t) =_{\beta\eta} u$*, then* σ *is a ground substitution (i.e. for any variable* $X \in \mathcal{Dom}(\sigma)$*,* $\mathcal{FV}(\sigma(X)) = \emptyset$*).*

Proof. The condition $\mathcal{Dom}(\sigma) \subseteq \mathcal{FV}(t)$ ensures that $\mathcal{FV}(\sigma(t)) = \big(\mathcal{FV}(t) \setminus \mathcal{Dom}(\sigma)\big) \cup \bigcup_{X \in \mathcal{Dom}(\sigma)} \mathcal{FV}(\sigma(X))$, therefore $\mathcal{FV}(\sigma(X)) \subseteq \mathcal{FV}(\sigma(t))$ for any $X \in \mathcal{Dom}(\sigma)$. Now, using the previous lemma, $\mathcal{FV}(\sigma(t)) = \mathcal{FV}(u) = \emptyset$. Concluding $\mathcal{FV}(\sigma(X)) = \emptyset$ for any $X \in \mathcal{Dom}(\sigma)$. $\Box$

Thanks to these lemmas we avoid the problems discussed in the previous section. So the linear second-order typed λ-calculus is adequate to model our critical pair schemes as pairs of linear second-order terms.

We describe now a sound and complete procedure to find minimum linear second-order unifiers. In the definition of the procedure we use a compact notation based on sets of indexes and indexed sets of indexes. For any set of indexes $P = \{p_1, \ldots, p_n\}$, the expression $a(\overline{b_P})$ denotes $a(b_{p_1}, \ldots, b_{p_n})$, and for any P-indexed set of indexes $Q_P = \{Q_{p_1}, \ldots, Q_{p_n}\} = \{\{q_1^1, \ldots, q_1^{m_1}\}, \ldots, \{q_n^1, \ldots, q_n^{m_n}\}\}$ the expression $a\big(\overline{b_P(\overline{c_{Q_P}})}\big)$ denotes

$a(b_{p_1}(c_{q_1^1} \dots c_{q_1^{m_1}}), \dots, b_{p_n}(c_{q_n^1} \dots c_{q_n^{m_n}}))$. Notice that capital letters denote set of indexes whereas lower case letters denote concrete indexes.

We use the notation on transformations introduced by Gallier and Snyder (Gallier and Snyder, 1990) for describing unification processes. Any state of the process is represented by a pair $\langle S, \sigma \rangle$ where $S = \{t_1 \stackrel{?}{=} u_1, \dots, t_n \stackrel{?}{=} u_n\}$ is the set of unification problems still to be solved and σ is the substitution leading from the initial problem to the actual one. The algorithm is described by means of transformation rules on states $\langle S, \sigma \rangle \Rightarrow \langle S', \sigma' \rangle$. The initial state is $\langle S_0, Id \rangle$. If it can be transformed into a state where the unification problem is empty $\langle S_0, Id \rangle \Rightarrow^* \langle \emptyset, \sigma \rangle$ then σ is a solution –unifier– of the unification problem S_0.

We suppose that the initial state $\langle S_0, Id \rangle$ is in normal form (i.e. any pair of terms $t \stackrel{?}{=} u \in S_0$ have been $\beta\eta$-normalized and their more externally bounded variables α-converted to assign them the same names), and that after applying any transformation rule the resulting unification problem is also normalized. Therefore, we can suppose that any pair $t \stackrel{?}{=} u \in S$ has the form $\lambda \overline{x_N} \,.\, a(\overline{t_P}) \stackrel{?}{=} \lambda \overline{x_N} \,.\, b(\overline{u_Q})$ where a, b may be either a constant, a bound or a free variable.

Definition 7.4. *The transformations rules of the unification procedure have the form:*

$$\langle \{t \stackrel{?}{=} u\} \cup S, \sigma \rangle \Rightarrow \langle \rho(R \cup S), \rho \circ \sigma \rangle$$

where the transformation $t \stackrel{?}{=} u \Rightarrow R$ and the linear substitution ρ are defined by cases as follows.

On these cases the P and Q indexed sets of indexes $\{N_i\}_{i \in P}$ and $\{N'_j\}_{j \in Q}$ are defined as follows.

$$\begin{aligned} N_i &\stackrel{def}{=} \{k \in N \mid x_k \in \mathcal{FV}(t_i)\} \\ N'_j &\stackrel{def}{=} \{k \in N \mid x_k \in \mathcal{FV}(u_j)\} \end{aligned}$$

(i) Instantiation. *If X does not occur free in t then*

$$\begin{aligned} &X \stackrel{?}{=} t \Rightarrow \emptyset \\ &\rho = [X \mapsto t] \end{aligned}$$

(ii) Equal terms.

$$\begin{aligned} &t \stackrel{?}{=} t \Rightarrow \emptyset \\ &\rho = Id \end{aligned}$$

(iii) Rigid-rigid. *If a is a constant or a bound variable, and for any $i \in P$ we have $N_i = N'_i$ then*

$$\begin{aligned} &\lambda \overline{x_N} \,.\, a(\overline{t_P}) \stackrel{?}{=} \lambda \overline{x_N} \,.\, a(\overline{u_P}) \Rightarrow \textstyle\bigcup_{i \in P} \{\lambda \overline{x_{N_i}} \,.\, t_i \stackrel{?}{=} \lambda \overline{x_{N_i}} \,.\, u_i\} \\ &\rho = Id \end{aligned}$$

(iv) Imitation. *If a is a constant, F is a free variable, and $\bigcup_{j \in Q} R_j = P$ is a Q-indexed family of disjoint sets of indexes satisfying $\bigcup_{i \in R_j} N_i = N'_j$ for any $j \in Q$ then*

$$\begin{aligned} &\lambda \overline{x_N} \,.\, F(\overline{t_P}) \stackrel{?}{=} \lambda \overline{x_N} \,.\, a(\overline{u_Q}) \Rightarrow \textstyle\bigcup_{j \in Q} \left\{\lambda \overline{x_{N'_j}} \,.\, H_j(\overline{t_{R_j}}) \stackrel{?}{=} \lambda \overline{x_{N'_j}} \,.\, u_j\right\} \\ &\rho = \left[F \mapsto \lambda \overline{y_P} \,.\, a\left(\overline{H_Q(\overline{y_{R_Q}})}\right)\right] \end{aligned}$$

where $\{H_j\}_{j \in Q}$ are fresh free variables of the appropiate second-order types (which may be deduced from their context).

(v) Projection. *If a is a constant or a bound variable, and F is a free variable with type $F : \tau \rightarrow \tau$, then*

$$\lambda\overline{x_N} \,.\, F(t) \stackrel{?}{=} \lambda\overline{x_N} \,.\, a(\overline{u_Q}) \Rightarrow \{\lambda\overline{x_N} \,.\, t \stackrel{?}{=} \lambda\overline{x_N} \,.\, a(\overline{u_Q})\}$$
$$\rho = [F \mapsto \lambda x \,.\, x]$$

(vi) Flexible-flexible with equal heads. *If F is a free variable, and for any $i \in P$ we have $N_i = N'_i$ then*

$$\lambda\overline{x_N} \,.\, F(\overline{t_P}) \stackrel{?}{=} \lambda\overline{x_N} \,.\, F(\overline{u_P}) \Rightarrow \bigcup_{i\in P}\{\lambda\overline{x_{N_i}} \,.\, t_i \stackrel{?}{=} \lambda\overline{x_{N_i}} \,.\, u_i\}$$
$$\rho = Id$$

(vii) Flexible-flexible with distinct heads. *If F and G are free variables with $F \neq G$; $P' \subseteq P$ and $Q' \subseteq Q$ are two sets of indexes and $\{Q_i\}$ and $\{P_j\}$ two P'- and Q'-indexed families of disjoint sets of indexes satisfying*

$$\bigcup_{j\in Q'} P_j \cup P' = P \qquad N_i = \bigcup_{j\in Q_i} N'_j \text{ for any } i \in P'$$
$$\bigcup_{i\in P'} Q_i \cup Q' = Q \qquad N'_j = \bigcup_{i\in P_j} N_i \text{ for any } j \in Q'$$

then

$$\lambda\overline{x_N} \,.\, F(\overline{t_P}) \stackrel{?}{=} \lambda\overline{x_N} \,.\, G(\overline{u_Q}) \Rightarrow \bigcup_{i\in P'} \{\lambda\overline{x_{N_i}} \,.\, t_i \stackrel{?}{=} \lambda\overline{x_{N_i}} \,.\, W_i(\overline{u_{Q_i}})\} \cup \bigcup_{j\in Q'} \{\lambda\overline{x_{N'_j}} \,.\, V_j(\overline{t_{P_j}}) \stackrel{?}{=} \lambda\overline{x_{N'_j}} \,.\, u_j\}$$

$$\rho = \left[F \mapsto \lambda\overline{y_P} \,.\, H\left(\overline{V_{Q'}(\overline{y_{P_{Q'}}})}, \overline{y_{P'}}\right)\right] \left[G \mapsto \lambda\overline{z_Q} \,.\, H\left(\overline{z_{Q'}}, \overline{W_{P'}(\overline{z_{Q_{P'}}})}\right)\right]$$

where H, $\{W_i\}_{i\in P'}$ and $\{V_j\}_{j\in Q'}$ are fresh free variables.

The following theorem states soundness and completeness conditions of the unification procedure based on the previous transformation rules. The proof of this theorem is out of the scope of this article and it can be found in (Levy, 1996).

THEOREM 7.5. **soundness and completeness.** *Substitution σ is a unifier of the unification problem S if, and only if, there exists a transformation sequence $\langle S, Id\rangle \Rightarrow^* \langle \emptyset, \sigma\rangle$.*

Compared with the general procedure (Jensen and Pietrzykowski, 1976), we avoid the use of the prolific *elimination* and *iteration* rules. These rules always compromise the termination of Jensen and Pietrzykowski's procedure. On the contrary, our procedure always finishes for a very useful case: if no free variable occurs more than twice in a unification problem. This fact is related with the termination of the *naive* string unification procedure when variables occur at most twice (Schulz, 1991), and is also proved in (Levy, 1996).

THEOREM 7.6. **termination.** *If no free variable occurs more than twice in a linear second-order unification problem, then this problem is decidable.*

Although the condition of this theorem may seem very restrictive, it is not so. Given an inclusion, or a critical pair, where a variable occurs more than twice in one of its sides, we can find a set of refutationally equivalent inclusions such that no variable occurs more than twice in one of its sides. Let us see an example.

$$\{a(F(X),F(Y),F(Z),F(T)) \subseteq b(X,Y,Z,T,\lambda x\,.\,F(x))\} \cup Ax \quad \text{is inconsistent}$$
$$\text{iff}$$
$$\begin{array}{l}\{a(F(X),F(Y),G(Z),G(T)) \subseteq equals(\lambda x\,.\,F(x),\lambda x\,.\,G(x)),\\ \;equals(\lambda x\,.\,H(x),\lambda x\,.\,H(x)) \subseteq b(X,Y,Z,T,\lambda x\,.\,H(x))\} \cup Ax\end{array} \quad \text{is inconsistent}$$

Where *equals* is supposed to be a fresh function symbol. A similar process can be applied for any number of occurrences of any free variable. The details of such kind of transformations are left for further work.

8. The Critical Pair Lemma for Second-Order Bi-rewrite Systems

Second-order bi-rewrite rules are defined, as usual, as pairs of linear second-order terms. However we have to impose two restrictions to second-order bi-rewrite systems.

DEFINITION 8.1. *Given two sets of second-order bi-rewrite rules $R_\subseteq$ and $R_\supseteq$, we say that $\langle R_\subseteq, R_\supseteq\rangle$ is a* **second-order bi-rewrite system** *if any rule $l \xrightarrow{\subseteq} r$ in $R_\subseteq$ and any rule $l \xrightarrow{\supseteq} r$ in $R_\supseteq$ satisfies $\mathcal{FV}(r) \subseteq \mathcal{FV}(l)$ and l and r have the same first-order type.*

The first restriction is imposed to avoid the infinitely branching problem. The second restriction is required to avoid the introduction of free variables with type order higher than two during the completion process, as it will be motivated later.

Rewrite relations are defined as usual.

DEFINITION 8.2. *We say that s* **rewrites to** *t using the set of bi-rewrite rules R, noted $s \xrightarrow[R]{} t$, if there exists an occurrence p in s, a rule $l \longrightarrow r \in R$, and a linear second-order substitution σ such that $s|_p =_{\beta\eta} \sigma(l)$ and $t = s[\sigma(r)]_p|_{\beta\eta}$.*

We can prove then the following result.

LEMMA 8.3. *For any second-order bi-rewrite system we have:*

(i) *If the terms s and t satisfy $s \xrightarrow[R]{} t$, then $\mathcal{FV}(t) \subseteq \mathcal{FV}(s)$.*
(ii) *For any term s there are finitely many terms t such that $s \xrightarrow[R]{} t$.*

PROOF. (i) If $s \xrightarrow[R]{} t$ then there exists a context $u[\ ]_p$, a rule $l \longrightarrow r \in R$ and a substitution σ such that $s =_{\beta\eta} u[\sigma(l)]_p$ and $t = u[\sigma(r)]_p|_{\beta\eta}$. Relation $=_{\beta\eta}$ preserves free variables in linear second-order λ-calculus (lemma 7.2), therefore we only need to prove $\mathcal{FV}(\sigma(r)) \subseteq \mathcal{FV}(\sigma(l))$. This may be concluded from $\mathcal{FV}(\sigma(s)) = (\mathcal{FV}(s)\backslash\mathcal{D}om(\sigma)) \cup \mathcal{FV}(\sigma)$ and $\mathcal{FV}(r) \subseteq \mathcal{FV}(l)$ which holds for any rule $l \longrightarrow r$.

(ii) We can apply finitely many rules $l \longrightarrow r$ on finitely many different positions p of a term s to rewrite it. We only have to consider substitutions σ satisfying $\mathcal{D}om(\sigma) \subseteq \mathcal{FV}(r)$ in order to instantiate r. Now, if $\mathcal{FV}(r) \subseteq \mathcal{FV}(l)$, we only obtain finitely many substitutions σ satisfying $s|_p =_{\beta\eta} \sigma(l)$ and $\mathcal{D}om(\sigma) \subseteq \mathcal{FV}(r) \subseteq \mathcal{FV}(l)$. By lemma 7.3 these substitutions instantiate any variable of $\mathcal{FV}(r)$, therefore any rule, position and substitution determine completely a term $t = s[\sigma(r)]_p|_{\beta\eta}$, thus we will obtain finitely many of them.

□

It means that no new variables are introduced during the rewrite process and it guarantees that the rewrite relation is finitely branching and, therefore any terminating bi-rewrite system is quasi-terminating.

The use of second-order terms simplifies the definition of critical pairs.

DEFINITION 8.4. *Let* $l \xrightarrow{\subseteq} r$ *in* $R_{\subseteq}$ *and* $s \xrightarrow{\supseteq} t$ *in* $R_{\supseteq}$ *be two second-order bi-rewrite rules (with distinct free variables). If* σ *belongs to the set of minimum linear second-order unifiers of* l *and* $F(s)$*, being* F *a fresh free variable, then*

$$\sigma(F(t)) \subseteq \sigma(r)$$

is a **(second-order) critical pair**. *Similarly for critical pairs between* $R_{\supseteq}$ *and* $R_{\subseteq}$*.*

Nipkow in (Nipkow, 1991) could not define critical pairs in this way because $F(s)$ violates his definition of *pattern*. In our case, we have to take into account that the variable F in $l \stackrel{?}{=} F(s)$ has to be second-order typed, therefore we have to require all rewrite rules to be first-order typed. If this condition is satisfied, then $\sigma(r)$ and $\sigma(F(t))$ will also be first-order typed, and if we have to introduce $\sigma(F(t)) \xrightarrow{\subseteq} \sigma(r)$ or $\sigma(r) \xrightarrow{\supseteq} \sigma(F(t))$ as new rewrite rules during the completion process, they will also be first-order typed. Moreover, if l and r have the same type, as well as s and t, then $F(t)$ and r will have also the same type.

We can prove then the following critical pair lemma.

THEOREM 8.5. *A terminating second-order bi-rewrite system* $\langle R_{\subseteq}, R_{\supseteq}\rangle$ *is Church-Rosser if all the second-order critical pairs are bi-confluent.*

PROOF. The most general way in which two expressions l and s (the left part of two rules) can overlap is given by $\sigma(F(s)) \subseteq \sigma(l)$. All these pairs are captured by the definition of second-order critical pair, from this and from the fact that when the two left parts of the rules do not overlap, the resulting pair is always bi-confluent, we can conclude that the system is locally bi-confluent iff all second-order critical pairs are bi-confluent. The Church-Rosser property is proved by noetherian induction in the usual way. □

As we have already said, the conditions for the termination of second-order bi-rewrite systems are not considered in this article. The decidability of the linear second-order unification problem when a free variable occurs more than twice also remains an open question, and it seems to be not easy to answer. These two issues are left as further research work.

9. An Example: Towards a Second-Order Completion Procedure

To conclude, we illustrate the use of a second-order completion method by means of the same example of section 4. The commutativity and associativity properties of the union operator make necessary to consider bi-rewriting modulo a set of inclusions. This theory was developed in section 3 for first-order bi-rewrite systems, and it will not be considered in detail in this example. We shall use a set of non-oriented rules I, and we shall suppose that the second-order unification algorithm can be extended to second-order unification modulo commutativity and associativity. We also have to notice that this example has been completed *by hand* (for the moment the unification and the completion procedures have not been implemented), therefore it is quite possible that it contains some errors.

We take the following rules as initial rules

$$\begin{array}{ll} r_1: & X \cup X \xrightarrow{\subseteq} X \\ r_1^{ext}: & X \cup X \cup Y \xrightarrow{\subseteq} X \cup Y \\ r_2: & X \cup Y \xrightarrow{\supseteq} X \end{array}$$

and we use linear second-order unification modulo commutativity and associativity of the union operator.

We can generate an extended critical pair unifying the left part of the rule r_2 with a subexpression of the left part of the rule r_1. The solutions σ of the unification problem $F(X \cup Y) \stackrel{?}{=} Z \cup Z$ are used to compute the critical pair $\sigma(F(X)) \subseteq \sigma(Z)$. This unification problem has two minimum unifiers (up to $\cup$ associativity and commutativity):

$$\begin{array}{lll} \sigma & = & [F \mapsto \lambda x \,.\, x][X \mapsto Z][Y \mapsto Z] \\ \sigma & = & [F \mapsto \lambda x \,.\, G(x) \cup G(X \cup Y)][Z \mapsto G(X \cup Y)] \end{array}$$

These two unifiers generate two critical pairs. The first one is bi-confluent. The second one makes necessary to introduce the following rule, which has been oriented using a reduction ordering.

$$r_3: \; G(X) \cup G(X \cup Y) \xrightarrow{\subseteq} G(X \cup Y)$$

This rule generates new critical pairs with r_2 (the only rule belonging to $R_{\supseteq}$) which makes necessary to introduce more rules. One of them is the following one, which subsumes r_3.

$$r_4: \; G(X) \cup G(Y) \xrightarrow{\subseteq} G(X \cup Y)$$

Contrary to the previous case, the orientation of the rule r_4 is not so clear, but we do not consider the problem of orienting second-order rules in this article.

Rule r_4 is not left-linear, thus it generates new critical pairs with r_2. The expressions $G(X') \cup G(Y')$, and $F(X \cup Y)$ unify by $[F \mapsto \lambda x \,.\, H(x, X') \cup H(X \cup Y, Y')][G \mapsto \lambda x \,.\, H(X \cup Y, x)]$ and make necessary to introduce the following rule.

$$r_5: \; H(X, X') \cup H(X \cup Y, Y') \xrightarrow{\subseteq} H(X \cup Y, X' \cup Y')$$

This rule generates a new critical pair with r_2 which is made bi-confluent introducing the following rule, which subsumes r_5.

$$r_6: \; H(X, X') \cup H(Y, Y') \xrightarrow{\subseteq} H(X \cup Y, X' \cup Y')$$

Repeating this process we obtain a canonical but infinite bi-rewrite system composed by the initial set of rules plus the following infinite set of rules.

$$r_7^n: \; H(X_1, \ldots, X_n) \cup H(Y_1, \ldots, Y_n) \xrightarrow{\subseteq} H(X_1 \cup Y_1, \ldots, X_n \cup Y_n) \quad \text{For any } n \in \mathbb{N}$$

The reader is invited to follow out the corresponding completion process when rule r_4 is oriented in the opposite direction. In this case the resulting canonical bi-rewrite system is also infinite.

A solution to prevent the non-termination of this completion process is using β-reduction explicitly. We use now three symbols in the signature:

$$\begin{array}{ll} \cup: & \tau \to \tau \to \tau \\ \texttt{lambda}: & (\tau \to \tau) \to \tau \\ \texttt{apply}: & \tau \to \tau \to \tau \end{array}$$

and the following initial set of rules:

$$\begin{array}{llll} r_1: & X \cup X \xrightarrow{\subseteq} X & & \\ r_1^{ext}: & X \cup X \cup Y \xrightarrow{\subseteq} X \cup Y & r_3: & X \cup Y \xrightarrow{\supseteq} X \\ r_2: & \texttt{apply}(\texttt{lambda}(F), X) \xrightarrow{\subseteq} F(X) & r_4: & \texttt{apply}(\texttt{lambda}(F), X) \xrightarrow{\supseteq} F(X) \end{array}$$

All standard critical pairs of this system are bi-confluent, thus we have to concentrate our attention on two cases, the critical pairs obtained by overlapping the repeated variable of rule r_1 (or of rule r_1^{ext}) with the left part of rule r_3 in the first case and with r_4 in the second case. In the second case, as far as the rule r_4 also appears in the other rewrite system (as rule r_2), all *extended* critical pairs generated by it will be trivially bi-confluent. Therefore, we only have to consider the extended critical pair generated by r_1 and r_3, i.e.:

$$r_5: \ F(X) \cup F(X \cup Y) \xrightarrow{\subseteq} F(X \cup Y)$$

As we know, this rule generates a new rule r_6 which properly oriented subsumes r_5.

$$r_6 \ F(X) \cup F(Y) \xrightarrow{\subseteq} F(X \cup Y)$$

This rule is non-left-linear and may initiate an infinite sequence of non-confluent critical pairs, as we have seen. However, it also generates a standard critical pair with r_4. It is interesting to see that, using second-order bi-rewrite systems, we can generate standard critical pairs between rules not sharing any symbol of the signature. The reader can figure out that the same happens dealing with equational second-order rewrite systems.

Let's concentrate our attention on this standard critical pair. It is obtained unifying $H(\texttt{apply}(\texttt{lambda}(F), X))$ and $G(Y) \cup G(Z)$ using:

$$\sigma = [H \mapsto \lambda x \,.\, H_1(x) \cup H_1(\texttt{apply}(Z, X))][G \mapsto \lambda x \,.\, H'(\texttt{apply}(x, X))][Y \mapsto \texttt{lambda}(F)]$$

The resulting rule is:

$$r_7: H_1(F(X)) \cup H_1(\texttt{apply}(Z, X)) \xrightarrow{\subseteq} H_1(\texttt{apply}(\texttt{lambda}(F) \cup Z, X))$$

This rule generates a new critical pair with r_4 which introduces r_8, and r_8 a critical pair with r_3 which introduces r_9, and finally r_9 a critical pair with r_3 which introduces r_{10}.

$$\begin{array}{ll} r_8: & H_1(F(X)) \cup H_1(G(X)) \xrightarrow{\subseteq} H_1(\texttt{apply}(\texttt{lambda}(F) \cup \texttt{lambda}(G), X)) \\ r_9: & H_1(F(H_2(X))) \cup H_1(G(H_2(X \cup Y)) \xrightarrow{\subseteq} H_1(\texttt{apply}(\texttt{lambda}(F) \cup \texttt{lambda}(G), \\ & \qquad H_2(X \cup Y))) \\ r_{10}: & H_1(F(H_2(X))) \cup H_1(G(H_2(Y))) \xrightarrow{\subseteq} H_1(\texttt{apply}(\texttt{lambda}(F) \cup \texttt{lambda}(G), \\ & \qquad H_2(X \cup Y))) \end{array}$$

It is easy to see that we only need the instance of r_{10} obtained by $[H_1 \mapsto \lambda x \,.\, x][H_2 \mapsto \lambda x \,.\, x]$ to subsume rule r_5 and to make bi-confluent all critical pairs obtained from it.

$$r'_{10}: \ F(X) \cup G(Y) \xrightarrow{\subseteq} \texttt{apply}(\texttt{lambda}(F) \cup \texttt{lambda}(G), X \cup Y)$$

However, this rule generates new critical pairs with r_4 which introduce the following rules.

$$\begin{array}{ll} r_{11}: & F(G(X)) \cup H(Y) \xrightarrow{\subseteq} \texttt{apply}(\ \texttt{lambda}(\lambda x \,.\, F(\texttt{apply}(x, X))) \cup \texttt{lambda}(H), \\ & \qquad \texttt{lambda}(G) \cup Y) \\ r_{12}: & F(G(X)) \cup H(I(Y)) \xrightarrow{\subseteq} \\ & \xrightarrow{\subseteq} \texttt{apply}(\ \texttt{lambda}(\lambda x \,.\, F(\texttt{apply}(x, X))) \cup \texttt{lambda}(\lambda x \,.\, H(\texttt{apply}(x, Y))), \\ & \qquad \texttt{lambda}(G) \cup \texttt{lambda}(I)) \end{array}$$

Rule r_{12} concludes the completion process which results in a finite canonical bi-rewrite system shown in figure 4.

$$R_{\subseteq} = \begin{cases} r_1 : & X \cup X \xrightarrow{\subseteq} X \\ r_2 : & \texttt{apply}(\texttt{lambda}(\lambda x \,.\, F(x)), X) \xrightarrow{\subseteq} F(X) \\ r_{10} : & F(X) \cup G(Y) \xrightarrow{\subseteq} \texttt{apply}(\texttt{lambda}(\lambda x \,.\, F(x)) \cup \texttt{lambda}(\lambda x \,.\, G(x)), X \cup Y) \\ r_{11} : & F(G(X)) \cup H(Y) \xrightarrow{\subseteq} \\ & \quad \xrightarrow{\subseteq} \texttt{apply}(\texttt{lambda}(\lambda x \,.\, F(\texttt{apply}(x, X))) \cup \texttt{lambda}(H), \texttt{lambda}(G) \cup Y) \\ r_{12} : & F(G(X)) \cup H(I(Y)) \xrightarrow{\subseteq} \\ & \quad \xrightarrow{\subseteq} \texttt{apply}(\texttt{lambda}(\lambda x \,.\, F(\texttt{apply}(x, X))) \cup \texttt{lambda}(\lambda x \,.\, H(\texttt{apply}(x, Y))), \\ & \qquad \texttt{lambda}(G) \cup \texttt{lambda}(I)) \end{cases}$$

$$R_{\supseteq} = \begin{cases} r_3 : & X \cup Y \xrightarrow{\supseteq} X \\ r_4 : & \texttt{apply}(\texttt{lambda}(\lambda x \,.\, F(x)), X) \xrightarrow{\supseteq} F(X) \end{cases}$$

$$I = \begin{cases} X \cup Y \xleftrightarrow{\subseteq} Y \cup X \\ (X \cup Y) \cup Z \xleftrightarrow{\subseteq} X \cup (Y \cup Z) \end{cases}$$

Figure 4. A presumably canonical higher-order bi-rewrite system for the inclusion theory of the union with β-reduction.

10. Related Work

In the context of automated theorem proving, resolution is not very effective in dealing with transitive relations. Special techniques have been devised for such relations, specially for equivalence relations which have attracted most of the attention. Slagle (1972) was the first to encode resolution with the transitivity axiom in a chaining system with paramodulation (Robinson and Wos, 1969) for theories with equality, orders and sets. Chaining into variables, which is needed for completeness, is too prolific, like our extended critical pairs or like variable instance pairs in (Bachmair *et al.*, 1986). For special order theories this problem can be avoided. For dense total orderings without endpoints, Bledsoe and Hines (1980) proposed techniques for eliminating certain occurrences of variables from formulas. Bledsoe *et al.* (1985) and Hines (1992) gave completeness results for these restricted chaining systems. Monotonicity or anti-monotonicity of functions with respect to special (transitive) relations led Manna and Waldinger (1986) to propose subterm chaining methods for general clauses but the proposed calculus was shown to be incomplete (Manna and Waldinger, 1992). In (Levy and Agustí, 1993) we were the first to apply rewrite techniques to non-symmetric and monotonic relations by means of bi-rewrite systems. Bachmair and Ganzinger (1993b) used the idea of bi-rewriting to give a refutationally complete inference system of ordered chaining for general clauses and general transitive relations. They studied the particular case of dense total orderings using this technique in (Bachmair and Ganzinger, 1993a).

11. Conclusions and Further Work

We have shown the adequacy of using a pair of rewrite systems and a bi-directional search procedure to automate the deduction with monotonic inclusions. Like in the equational case, a soundness, completeness and decidability theorem can be stated. However, in this case, it is based on an *extended* definition of critical pair which includes *schemes* of critical pairs. It means that, if we want to use a kind of Knuth-Bendix completion algorithm, then we have to face the problem of working with schemes of rules. We undertake this problem by means of second-order rules. The use of higher-order terms in rewrite systems introduces some problems. Because of that, there is not a unique proposal for higher-order rewriting in the literature. We have discussed some of them and we have also proposed a definition of second-order bi-rewrite systems based on the use of the linear second-order typed λ-calculus. This proposal can also be seen as a new notion of higher-order rewrite systems. We have described a new sound and complete second-order unification procedure for such restricted second-order language. This procedure avoids the use of the iteration and elimination transformation rules of the general second-order unification procedure described by Jensen and Pietrzykowski. These transformation rules, in the general case, always make the procedure non terminating. Unfortunately, the decidability of our unification problem when a free variable occurs more than twice is still an open question and the termination of the procedure we have described is not guaranteed in such case. However, we have shown that a transformation eliminating such multiple occurrences of free variables may be applied. Another problem left open for future research is the termination of second-order bi-rewrite systems and the search of well-founded orderings on these linear second-order terms.

Acknowledgments

We want to acknowledge H. Comon, F. Esteva, R. Nieuwenhuis, F. Orejas, A. Rubio and M. Schorlemmer their comments and support during the development of this work, and specially to H. Ganzinger and G. Struth the detection of some errors in an early version of this article.

References

Bachmair, L. (1991). *Canonical Equational Proofs*. Birkhäuser, Boston, Massachusetts.

Bachmair, L., Dershowitz, N. (1986). Commutation, transformation and termination. In Siekmann, J., editor, *8th Conference in Automated Deduction, CADE-8*, volume 230 of *Lecture Notes in Computer Science*, pages 5–20. Springer-Verlag.

Bachmair, L., Dershowitz, N. (1989). Completion for rewriting module a congruence. *Theoretical Computer Science*, **67**:173–201.

Bachmair, L., Dershowitz, N., Hsiang, J. (1986). Orderings for equational proofs. In *Proc. Symp. on Logic in Computer Science, LICS'86*, pages 346–357, Boston, Massachusetts.

Bachmair, L., Dershowitz, N., Plaisted, D. (1989). Completion without failure. In Aït-Kaci, H., Nivat, M., editors, *Resolution of Equations in Algebraic Structures*, volume 2: Rewriting Techniques, chapter 1, pages 1–30. Academic Press, New York.

Bachmair, L., Ganzinger, H. (1993a). Ordered chaining for total orderings. Technical Report MPI-I-93-250, Max-Planck-Institut für Informatik, Saarbrücken, Germany. Short version in Proceedings of CADE'94.

Bachmair, L., Ganzinger, H. (1993b). Rewrite techniques for transitive relations. Technical Report MPI-I-93-249, Max-Planck-Institut für Informatik, Saarbrücken, Germany. Short version in Proceedings of LICS'94.

Barendregt, H. (1981). *The Lambda Calculus: its syntax and semantics*. Studies in Logic and the Foundations of Mathematics. Elsevier Science Publishers B. V.

Bäumer, H. (1992). On the use of relation algebra in the theory of reduction systems. Technical report, Dept. Informatica, Univ. of Twente, Enschede, The Netherlands.

Birkhoff, G. (1935). On the structure of abstract algebras. *Proc. Cambridge Philos. Soc.*, **31**:433–454.

Bledsoe, W., Hines, L. M. (1980). Variable elimination and chaining in a resolution-based prover for inequalities. In Bibel, W., Kowalski, R., editors, *5th Conference in Automated Deduction, CADE-5*, volume 87 of *Lecture Notes in Computer Science*, pages 70–87, Les Arcs, France. Springer-Verlag.

Bledsoe, W., Kunen, K., Shostak, R. (1985). Completeness results for inequality provers. *Artificial Intelligence*, **27**:255–288.

Comon, H. (1993). Completion of rewrite systems with membership constraints. Technical report, CNRS and LRI, Université de Paris Sud.

de Kogel, E. (1992). Relational algebra and equational proofs. Technical report, Department of Philosophy, Tilburg University.

Dershowitz, N. (1987). Termination of rewriting. *J. of Symbolic Computation*, **3**:69–115.

Dershowitz, N., Jouannaud, J.-P. (1990). Rewrite systems. In Leeuwen, J. V., editor, *Handbook of Theoretical Computer Science*. Elsevier Science Publishers.

Farmer, W. M. (1988). A unification algorithm for second-order monadic terms. *Annals of Pure and Applied Logic*, **39**:131–174.

Freese, R., Jezek, J., Nation, J. B. (1993). Term rewrite systems for lattice theory. *J. of Symbolic Computation*, **16**:279–288.

Gallier, J. H., Snyder, W. (1990). Designing unification procedures using transformations: A survey. *Bulletin of the EATCS*, **40**:273–326.

Geser, A. (1990). *Relative Termination*. PhD thesis, Universität Passau.

Goldfarb, W. D. (1981). The undecidability of the second-order unification problem. *Theoretical Computer Science*, **13**:225–230.

Hindley, J. R., Seldin, J. P. (1986). *Introduction to Combinators and λ-Calculus*. London Mathematical Society Student Texts. Cambridge University Press.

Hines, L. M. (1992). Completeness of a prover for dense linear logics. *J. of Automated Reasoning*, **8**:45–75.

Huet, G. (1975). A unification algorithm for typed λ-calculus. *Theoretical Computer Science*, **1**:27–57.

Huet, G. (1980). Confluent reductions: Abstract properties and applications to term rewriting systems. *J. of the ACM*, **27**(4):797–821.

Jensen, D. C., Pietrzykowski, T. (1976). Mechanizing ω-order type theory through unification. *Theoretical Computer Science*, **3**:123–171.

Jouannaud, J.-P. (1983). Confluent and coherent equational term rewriting systems application to proofs in abstract data types. In Ausiello, G., Protasi, M., editors, *Proc. Eighth Colloquium on Trees in Algebra and Programming, CAAP'83*, volume 159 of *Lecture Notes in Computer Science*, L'Aquila. Springer-Verlag.

Jouannaud, J.-P., Kirchner, H. (1986). Completion on a set of rules modulo a set of equations. *SIAM J. on Computing*, **15**(1):1155–1194.

Kirchner, H. (1985). *Preuves par Completion dans les Varietes d'Algebres*. PhD thesis, Universite de Nancy I.

Klop, J. W. (1987). Term rewriting systems: A tutorial. *Bulletin of the EATCS*, **32**:143–183.

Knuth, D. E., Bendix, P. B. (1970). Simple word problems in universal algebras. In Leech, J., editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Elmsford, N. Y.

Levy, J. (1994). *The Calculus of Refinements, a formal specification model based on inclusions*. PhD thesis, Dept. Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

Levy, J. (1996). Linear second-order unification. In Ganzinger, H., editor, *Rewriting Techniques and Applications, RTA'96*, volume 1103 of *Lecture Notes in Computer Science*, pages 332–346, New Jersey, USA. Springer-Verlag.

Levy, J., Agustí, J. (1993). Bi-rewriting, a term rewriting technique for monotonic order relations. In Kirchner, C., editor, *Rewriting Techniques and Applications, RTA'93*, volume 690 of *Lecture Notes in Computer Science*, pages 17–31, Montreal, Canada. Springer-Verlag.

Makanin, G. S. (1977). The problem of solvability of equations in a free semigroup. *Math. USSR Sbornik*, **32**(2):129–198.

Manna, Z., Waldinger, R. (1986). Special relations in automated deduction. *J. of the ACM*, **33**:1–60.

Manna, Z., Waldinger, R. (1992). The special-relation rules are incomplete. In Kapur, D., editor, *11th Int. Conf. on Automated Deduction, CADE-11*, volume 607 of *Lecture Notes in Artificial Intelligence*, Saratoga Springs, New York. Springer-Verlag.

Meseguer, J. (1990). Rewriting as a unified model of concurrency. In *Concur'90*, Lecture Notes in Computer Science, Amsterdam, The Netherlands. Springer-Verlag.

Meseguer, J. (1992). Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, **96**:73–155.

Miller, D. (1991a). A logic programming language with lambda-abstraction, function variables, and simple unification. Technical Report ECS-LFCS-91-159, Laboratory for Foundations of Computer Science, Edinburgh, Great Britain.

Miller, D. (1991b). Unification of simply typed lambda-terms as logic programming. Technical Report ECS-LFCS-91-160, Laboratory for Foundations of Computer Science, Edinburgh, Great Britain.

Newman, M. H. A. (1942). On theories with a combinatorial definition of "equivalence". *Annals of Math.*, **43**(2):223–243.

Nipkow, T. (1991). Higher-order critical pairs. In *Proc. Symp. on Logic in Computer Science, LICS'91*, pages 342–349.

Nipkow, T. (1992). Functional unification of higher-order patterns. Technical report, Institut für Informatik, TU München.

Peterson, G. E., Stickel, M. E. (1981). Complete sets of reductions for some equational theories. *J. of the ACM*, **28**(2):233–264.

Pietrzykowski, T. (1973). A complete mechanization of second-order type theory. *J. of the ACM*, **20**:333–364.

Prehofer, C. (1995). *Solving Higher-Order Equations: From Logic to Programming.* PhD thesis, Technische Universität München.

Robinson, G. A., Wos, L. T. (1969). Paramodulation and theorem proving in first order theories with equality. *Machine Intelligence*, **4**:133–150.

Schmidt-Schauß, M. (1995). Unification of stratified second-order terms. *J. of Symbolic Computation.* (Submited).

Schulz, K. U. (1991). Makanin's algorithm, two improvements and a generalization. Technical Report CIS-Bericht-91-39, Centrum für Informations und Sprachverarbeitung, Universität München.

Slagle, J. R. (1972). Automatic theorem proving for theories with built-in theories including equality, partial orderings, and sets. *J. of the ACM*, **19**:120–135.

Struth, G. (1996). Rewriting for non-symmetric transitive relations modulo a congruence. Technical report, Max-Planck-Institut für Informatik. Accesible by WWW at `http://www.mpi-sb.mpg.de/~struth/papers/rta.ps`.

Toyama, Y. (1987). On the Church-Rosser property for the direct sum of term rewriting systems. *J. of the ACM*, **34**(1):128–143.