# About the Confluence of Equational Pattern Rewrite Systems*

Alexandre Boudet and Evelyne Contejean

LRI, CNRS URA 410
Bt. 490, Universit Paris-Sud, Centre d'Orsay
91405 Orsay Cedex, France

**Abstract.** We study the confluence of higher-order pattern rewrite systems modulo an equational theory $E$. This problem has been investigated by Mayr and Nipkow [13], for the case of rewriting modulo a congruence *à la* Huet [8], (in particular, the equations of $E$ can be applied *above* the position where a rewrite rule is applied). The case we address here is rewriting using matching modulo $E$ as done in the first-order case by Jouannaud and Kirchner [10].
The theory is then applied to the case of $AC$-theories, for which we provided a complete unification algorithm in [1]. It happens that the $AC$-unifiers may have to be constrained by some flexible-flexible equations of the form $\lambda x_1 \cdots \lambda x_n.F(x_1, \ldots, x_n) = \lambda x_1 \cdots \lambda x_n F(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$, where $F$ is a free variable and $\sigma$ a permutation. This situation requires a slight technical adaptation of the theory.

## Introduction

Using equations as a programming language is very tempting because very natural. A theory of term rewriting systems (TRSs) has been developed so as to make this paradigm effective by orienting the equations into rewrite rules (in order to gain efficiency) while restoring the completeness by a *completion* process [11, 9]. Completion relies on a *critical pair lemma* which shows that whenever a term $t$ can be rewritten using rules $l_1 \to r_1$ and $l_2 \to r_2$ in two different terms $s_1$ and $s_2$, then either $s_1$ and $s_2$ rewrite (maybe in several steps) to a common reduct, or there exists a *critical pair* between $l_1 \to r_1$ and $l_2 \to r_2$. The critical pairs are obtained by unifying a left-hand side of a rule with a subterm of a left-hand side of a rule.

Some equations (like commutativity) cannot be oriented into terminating rewrite rules, hence the need of rewriting *modulo* an equational theory. The confluence of a terminating rewrite system $R$ modulo an equational theory $E$ (denoted by $R/E$) has been studied by Huet [8] who allows to choose an arbitrary term in the $E$-equivalence class of a term before to rewrite it. Jouannaud and Kirchner use a different reduction (denoted hereafter by $R^E$), in which $E$-equality

* This research was supported in part by the EWG CCL, the HCM Network CONSOLE, and the "GDR de programmation du CNRS".

steps can be performed only *under* the position to be rewritten [10]. *Matching modulo E* is used for rewriting instead of standard matching. On the other hand, a weaker notion of confluence suffices and the rewrite rules do not need to be left-linear.

A major drawback of using equations as a programming language is that they lack the expressiveness for providing higher-order features. Hence the idea of *higher-order rewrite systems* (HRSs) which allow to define functions by the means of rewrite rules over terms of the simply-typed lambda calculus. Actually HRSs are difficult to handle since the decidability of higher-order matching is still an open problem [4, 18] and higher-order unification is known to be undecidable [7, 5]. Matching and unification being crucial for reduction and deduction respectively, we need to restrict our attention to *pattern rewrite systems*, which are HRSs where the left-hand sides of the rules are *patterns*. Patterns are the largest subset of the terms of the simply-typed lambda calculus for which higher-order unification is known to be decidable [15]. Nipkow [17] has proved a critical pair lemma for pattern rewrite systems (PRSs) and Mayr and Nipkow [13] have lifted the theory of rewriting modulo an equational theory of Huet [8] to the case of PRSs [13]. On the other hand, Boudet and Contejean have provided an *AC*-unification algorithm for higher-order patterns [1]. The purpose of the present work is to lift the theory of Jouannaud and Kirchner [10] to the higher-order case, and to apply it to pattern rewrite systems modulo *AC*, completing the following table:

| | $R$ | $R/E$ | $R^E$ |
|---|---|---|---|
| First-order TRSs | Knuth–Bendix 70<br>Huet 81 | Huet 80 | Jouannaud–Kirchner 86 |
| Higher-order PRSs | Nipkow 91 | Mayr–Nipkow 97 | *This paper* |

We give now a simple motivating example. For this, we have to use some terminology that will be defined later, but a little background about term rewriting should suffice to follow the example.

*Example 1.* We consider the two base types `elt` and `mset`, having in mind to define a `map` function on multisets of `elt`. We will use the following constant function symbols:

```
empty   : mset
mk-mset : elt→ mset
union   : mset→ mset→ mset
map     : (elt→ elt)→ mset→ mset
```

and we consider the following rewrite rules:

1. $\texttt{map(}\lambda x.F(x)\texttt{,empty)} \rightarrow \texttt{empty}$
2. $\texttt{map(}\lambda x.F(x)\texttt{,union (mk-mset(}E\texttt{),}M\texttt{))} \rightarrow \texttt{union(map(}F\texttt{,}M\texttt{),}$ $\text{mk-mset}(F(E)))$
3. $\texttt{map(}\lambda x.x\texttt{,}M\texttt{)} \rightarrow M$

The third rule is just an optimization for avoiding to apply identity to all the elements of a multiset. Consider now the term $t$ (obtained by superposing the left-hand size of rule 3. in the left-hand size of rule 2.):

$$t \equiv \texttt{map(}\lambda x.x\texttt{,union(mk-mset(}E\texttt{),}M\texttt{))}$$

The term $t$ rewrites by rule 3. to

$$s_1 \equiv \texttt{union (mk-mset(}E\texttt{),}M\texttt{)}$$

which is no longer reducible. On the other hand, rule 2. applies to $t$ yielding the term

$$s_2' \equiv \texttt{union(map(}\lambda(x).x\texttt{,}M\texttt{),mk-mset(}\lambda x.x(E)\texttt{))}$$

After a $\beta$-reduction and an application of the rule 3., the term $s_2'$ reduces to the irreducible term

$$s_2 \equiv \texttt{union(}M\texttt{, mk-mset(}E\texttt{))}$$

Hence, the term $t$ can be reduced to two different irreducible terms $s_1$ and $s_2$.

But $s_1$ and $s_2$ are in fact equal modulo the associativity and commutativity of `union`.

## 1 Preliminaries

We assume the reader is familiar with simply-typed lambda-calculus, and term rewriting. Given a set $\mathcal{B}$ of *base types*, the set $\mathcal{T}$ of all *types* is the closure of $\mathcal{B}$ under the (right-associative) function space constructor $\rightarrow$. The terms of the *simply-typed lambda-terms* is generated from a set $\biguplus_{\tau\in\mathcal{T}} V_\tau$ of *typed variables* and a set $\biguplus_{\tau\in\mathcal{T}} C_\tau$ of *typed constants* using the following construction rules:

$$\frac{x \in V_\tau}{x \,:\, \tau} \qquad \frac{c \in C_\tau}{c \,:\, \tau} \qquad \frac{s \,:\, \tau \rightarrow \tau' \quad t \,:\, \tau}{(s\ t) \,:\, \tau'} \qquad \frac{x \,:\, \tau \quad s \,:\, \tau'}{(\lambda x.s) \,:\, \tau \rightarrow \tau'}$$

Some background is available in *e.g.* [6, 3] for lambda-calculus and term rewriting systems. We shall use the following notations: $\lambda x_1 \cdots \lambda x_n.s$ will be written $\lambda\overline{x_n}.s$, or even $\lambda\overline{x}.s$ if $n$ is not relevant. If in a same expression $\overline{x}$ appears several times it denotes the same sequence of variables. In addition, we will use the notation $t(u_1,\ldots,u_n)$ or $t(\overline{u_n})$ for $(\cdots(t\ u_1)\cdots)u_n$. If $\pi$ is a permutation of $(1,\ldots,n)$, $\overline{x_n}^\pi$ stands for the sequence $x_{\pi(1)},\ldots,x_{\pi(n)}$. The free (resp. bound) variables of a term $t$ are denoted by $fv(t)$ (resp. $bv(t)$). The *positions* of a term $t$ are words over $\{1,2\}$, $\Lambda$ is the empty word and $t|_p$ is the subterm of $t$ at position $p$. More precisely, $t|_\Lambda = t$, $(t_1\ t_2)|_i = t_i$ for $i \in \{1,2\}$, and $(\lambda x.t)|_1 = t$. The notation $t[u]_p$ stands for a term $t$ with a subterm $u$ at position $p$. $\mathcal{P}\text{os}(t)$ is the set of positions of a term $t$. We shall write $p \le q$ if $q = p \cdot p'$ for some $p'$, and $p||q$ if neither $p \le q$ nor $q \le p$ hold. $bv(t,p)$ is the set of variables that are bound in $t$ above position $p$.

Unless otherwise stated, we assume all terms to be in $\eta$-expanded, $\beta$-normal form, the $\eta$-expanded, $\beta$-normal form of a term $t$ being denoted by $t \updownarrow_\beta^\eta$.

A *substitution* $\sigma$ is a mapping from a finite set of variables to terms of the same type. If $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, the *domain* of $\sigma$ is $Dom(\sigma) = \{x_1, \ldots, x_n\}$ and the *set of variables introduced by* $\sigma$ is $VCod(\sigma) = fv(t_1) \cup \cdots \cup fv(t_n)$. When applying a substitution $\sigma$ to a term $t$, we will always assume that $VCod(\sigma) \cap bv(t) = \emptyset$ in order to avoid variable captures. In this case, $t\sigma$ denotes the term $\lambda \overline{x_n}.t(t_1, \ldots, t_n) \updownarrow_\beta^\eta$. We define $\theta_1 + \theta_2$ by $x(\theta_1 + \theta_2) = x\theta_2$ if $x \in \mathcal{D}\mathrm{om}(\theta_2)$, $x\theta_1$ otherwise.

## 2 Equational pattern rewrite systems

**Definition 1.** *A pattern is a term of the simply-typed $\lambda$-calculus in $\beta$-normal form in which the arguments of a free variable are $\eta$-equivalent to distinct bound variables.*

For instance, $\lambda xyz.f(H(x, y), H(x, z))$ and $\lambda x.F(\lambda z.x(z))$ are patterns while $\lambda xy.G(x, x, y)$, $\lambda xy.H(x, f(y))$ and $\lambda xy.H(F(x), y)$ are not patterns.

**Lemma 1.** *Let $s$ be a pattern, $p$ a position of $s$ and $\theta$ a substitution such that $bv(s, p) \cap \mathcal{D}\mathrm{om}(\theta) = \emptyset$. Then $(s|_p)\theta = s\theta|_p$.*

Let $E = \{l_1 \simeq r_1, \ldots, l_n \simeq r_n\}$ a set of *axioms* such that $l_i$ and $r_i$ are terms of a same base type, for $1 \leq i \leq n$. The *equational theory* $=_E$ generated by $E$ is the least congruence[1] containing all the instances of the axioms of $E$ (in the sequel, we shall not distinguish between $E$ and $=_E$). For instance, the associative-commutative ($AC$) theory of $+$ is the equational theory generated by $AC(+) = \{(x + y) + z \simeq x + (y + z), x + y \simeq y + x\}$. A substitution $\sigma$ is an *$E$-unifier* of $s$ and $t$ if $s\sigma$ and $t\sigma$ are equivalent modulo $\eta\beta$-equivalence and the theory $=_E$, which we write $s\sigma =_{\beta\eta E} t\sigma$. A *complete set of $E$-unifiers* of $s$ and $t$ is a set $\Sigma$ of substitutions such that every $\sigma$ in $\Sigma$ is an $E$-unifier of $s$ and $t$ and for every $E$-unifier $\theta$ of $s$ and $t$, there exist $\sigma \in \Sigma$ and $\rho$ such that $\theta =_{\beta\eta E} \sigma\rho$. The relation $=_E$ coincides with the reflexive, symmetric, transitive closure $\overset{*}{\leftrightarrow}_E$ of the relation $\rightarrow_E$ defined by $s \rightarrow_E t$ if there exist a position $p$ of $s$, an equation $l \simeq r \in E$ and a substitution $\theta$ such that $s|_p = l\theta$ and $t = s[r\theta]_p$.

Since we consider only terms in $\eta$-long $\beta$-normal form, the following result from Mayr and Nipkow will allow us to restrict our attention to $=_E$ instead of $=_{\beta\eta E}$.

**Theorem 1 ([14]).** *For any terms $u$ and $v$, $u =_{\beta\eta E} v$ if and only if $u \updownarrow_\beta^\eta =_E v \updownarrow_\beta^\eta$.*

The above theorem extends a result by Tannen where the equational theory is assumed to be defined by first-order equations [2].

[1] *i.e.*, compatible *also* with application and abstraction, in our context.

**Definition 2.** *A* rewrite rule *is a pair $< l, r >$ of terms, denoted by $l \to r$ such that $l$ is not a free variable, $l$ and $r$ are of the same base type and $fv(r) \subseteq fv(l)$. If $l$ is a pattern, then $l \to r$ is a* pattern rewrite rule. *A set of rewrite rules is called a* higher-order rewrite system *(HRS). A set of pattern rewrite rules is called a* pattern rewrite system *(PRS). A HRS $R$ induces a* rewriting relation $\to_R$ *on terms defined by $s \to_R t$ iff there exist $l \to r \in R$, $p \in \mathcal{P}\mathrm{os}(s)$ such that $s|_p = l\theta$ and $t = s[r\theta]_p$ for some substitution $\theta$. If necessary, we shall use one or more of the subscripts $s \to_R t$, $s \to_{l \to r} t$, $s \to_p t$, $s \to_\theta t$ to specify the rewrite system, the rule, the position and the substitution used in the reduction. The subscript $s \to_{\geq p} t$ means that the reduction occurred under position $p$. Where $\sigma$ and $\theta$ are two substitutions, we shall write $\sigma \to_R \theta$ if for any variable $x \in \mathcal{D}\mathrm{om}(\sigma) \cup \mathcal{D}\mathrm{om}(\theta)$ $x\sigma \to_R x\theta$.*

For any binary relation $\to$, $\stackrel{=}{\to}$ will denote its reflexive closure, $\stackrel{*}{\to}$ its reflexive transitive closure.

**Definition 3.** *Let $R$ be a pattern rewrite system and $E$ an equational theory whose axioms have patterns as left-hand and right-hand sides. $< R, E >$ is called an* equational PRS. *We write $s \to_{R^E} t$ if $s|_p =_E u$ for some $u$ and $s[u]_p \longrightarrow_{R,p} s[v]_p = t$. In other words, $R^E$ denotes the relation $\stackrel{*}{\longleftrightarrow}_{E,\geq p}\longrightarrow_{R,p}$.*

**Lemma 2.** *Let $< R, E >$ be an equational PRS such that the left-hand sides and right-hand sides of the equations of $E$ are patterns and $E$ is collapse-free. Let $s$, $s'$ be two terms in $\eta$-long, $\beta$-normal form, and $\theta$, $\theta'$ two substitutions such that $s \stackrel{*}{\to}_{R^E} s'$ and $\theta \stackrel{*}{\to}_{R^E} \theta'$. Then $s\theta \stackrel{*}{\to}_{R^E} s'\theta'$.*

*Proof.* By induction on (i) the order of $\theta$ and (ii) the number of steps in the proof $s \stackrel{*}{\to}_{R^E} s'$.

1. When $n = 0$ ($s = s'$), we proceed by induction on the structure of $s$. The term $s$ is in $\beta$-normal form: $s = \lambda\overline{x_m}.a(\overline{s_k})$ and by induction hypothesis $s_i\theta \stackrel{*}{\to}_{R^E} s_i\theta'$.
   (a) If $a \notin \mathcal{D}\mathrm{om}(\theta)$, then $a \notin \mathcal{D}\mathrm{om}(\theta')$. Indeed, $a\theta \stackrel{*}{\to}_{R^E} a\theta'$ and $a\theta = a$. But the left-hand sides of the rules are not free variables and $E$ is collapse-free, hence the case $a =_E l\sigma$ for some rule $l \to r$ is impossible and $a\theta = a\theta' = a$. We have $s\theta = \lambda\overline{x_m}.a(\overline{s_k\theta}) \stackrel{*}{\to}_{R^E} \lambda\overline{x_m}.a(\overline{s_k\theta'}) = s'\theta'$.
   (b) If $a \in \mathcal{D}\mathrm{om}(\theta)$, let $a\theta = \lambda\overline{y_k}.t$ (we know that $a$ requires $k$ arguments) and $a\theta = \lambda\overline{y_k}.t \to_{R^E} a\sigma \stackrel{*}{\to}_{R^E} a\theta'$. There exist $l \to r \in R$, $p \in \mathcal{P}\mathrm{os}(a\theta)$ and $\theta''$ such that $a\theta|_p =_E l\theta''$ and $a\sigma = a\theta[r\theta'']_p$. But $l\theta''$ is of a base type and the rewriting takes place below $\lambda\overline{y_k}$ in $\lambda\overline{y_k}.t$. Hence $a\sigma = \lambda\overline{y_k}.t''$ and $t'' \stackrel{*}{\to}_{R^E} t'$. Let $\delta = \{\overline{y_k \mapsto s_k\theta}\}$ and $\delta' = \{\overline{y_k \mapsto s_k\theta'}\}$. By induction hypothesis, $s_i\theta \stackrel{*}{\to}_{R^E} s_i\theta'$, hence $\delta \stackrel{*}{\to}_{R^E} \delta'$. Now, $a$ has type $\overline{\tau_k} \to \tau$ whose order is strictly greater than the order of each $\tau_i$, hence, the order of $\theta$ is strictly greater than that of $\delta$ and $\delta'$. We have $t \stackrel{*}{\to}_{R^E} t'$ and $\delta \stackrel{*}{\to}_{R^E} \delta'$ and by the induction hypothesis, $t\delta \stackrel{*}{\to}_{R^E} t'\delta'$. Finally,
   $$s\theta = \lambda\overline{x_m}.(a\theta)(\overline{s_k\theta}) \downarrow_\beta = \lambda\overline{x_m}.t\delta \stackrel{*}{\to}_{R^E} \lambda\overline{x_m}.t'\delta' = s\theta'$$

2. Assume now that $s \xrightarrow{*}_{R^E} s' \rightarrow_{R^E} s''$. By the induction hypothesis, $s\theta \xrightarrow{*}_{R^E} s'\theta'$. We are left to show that $s'\theta' \xrightarrow{*}_{R^E} s''\theta'$. Since $s' \rightarrow_{R^E} s''$, there exist $l \rightarrow r \in R$, $\theta''$ and $p \in \mathcal{D}\mathrm{om}(s')$ such that $s'|_p = l\theta''$ and $s'' = s'[r\theta'']_p$. We proceed by induction on the length of position $p$.
   (a) If $p = \Lambda$, then $s' = l\theta''$, $s'' = r\theta''$, hence $s'\theta' =_E l\theta''\theta'$ and $s''\theta' = r\theta''\theta$. We have $s'\theta' \xrightarrow{\Lambda}_{R_E} s''\theta'$.
   (b) If $|p| \geq 1$ and $s' = \lambda x.t'$ and $s'' = \lambda x.t''$ with $t' \rightarrow_{R_E} t''$, by induction hypothesis, $t'\theta' \xrightarrow{*}_{R^E} t''\theta'$ and $s'\theta' = \lambda x.t'\theta' \xrightarrow{*}_{R^E} \lambda x.t''\theta' = s''\theta'$.
   Finally if $|p| \geq 1$ and $s' = a(\overline{s'_k})$ and $s'' = a(\overline{s''_k})$, there exists an $s'_i$ such that $s'_i \rightarrow_{R^E} s''_i$ (for $j \neq i$, $s'_j = s''_j$). By the induction hypothesis, $s'_i\theta' \rightarrow_{R^E} s''_i\theta'$. If $a \notin \mathcal{D}\mathrm{om}(\theta')$, $s'\theta' = a(\overline{s'_k\theta'}) \xrightarrow{*}_{R^E} a(\overline{s''_k\theta'}) = s''\theta'$. If $a \in \mathcal{D}\mathrm{om}(\theta')$, $a\theta' = \lambda\overline{y_k}.t$, define $\delta = \{\overline{y_k \mapsto s'_k\theta'}\}$ and $\delta' = \{\overline{y_k \mapsto s''_k\theta'}\}$. Again, the order of $\delta'$ and $\delta''$ is strictly smaller than that of $\theta$ and $\theta'$. By induction hypothesis, $t\delta \xrightarrow{*}_{R^E} t\delta'$. Hence, $s'\theta' = (a\theta')(\overline{s'_k\theta'}) \downarrow_\beta = t\delta \xrightarrow{*}_{R^E} t\delta' = s''\theta'$.

The following definition is borrowed from Mayr and Nipkow [14]. It is useful for keeping track of the bound variables above a position $p$ when considering a subterm at position $p$ and for avoiding to have non-disjoint variable sets when superposing left-hand sides of rewrite rules.

**Definition 4.** *An $\overline{x_k}$-*lifter *of a term $t$ away from a set $W$ of variables is a substitution $\sigma = \{F \mapsto F\rho(\overline{x_k}) \mid F \in fv(t)\}$, where $\rho$ is a renaming such that $Dom(\rho) = fv(t)$, $VCod(\rho) \cap W = \emptyset$ and $F\rho$ has type $\tau_1 \rightarrow \cdots \rightarrow \tau_k \rightarrow \tau$ if $x_i$ has type $\tau_i$ for $1 \leq i \leq k$ and $F$ has type $\tau$.*

**Lemma 3 (adapted from [14]).** *Consider two patterns $l_1$ and $l_2$, $p$ a non-variable position of $l_1$. Let $\{\overline{x_k}\} = bv(l_1, p)$ and $\sigma$ an $\overline{x_k}$-lifter of $l_2$ away from $fv(l_1)$. Then $\lambda\overline{x_k}.(l_1|_p)$ and $\lambda\overline{x_k}.l_2\sigma$ are $E$-unifiable iff there exist two substitutions $\theta_1$ and $\theta_2$ such that $l_1|_p\theta_1 =_E l_2 theta_2$ and $\{\overline{x_k}\} \cap VCod(\theta_1) = \emptyset$.*

We close this section by making precise the assumptions we make in the rest of the paper. They are similar to those of Jouannaud and Kirchner in the first-order case.

**Assumptions** In the sequel $< R, E >$ denotes an equational pattern rewrite system. Moreover $E$ is assumed to be a *simple equational theory* that is a theory such that there is no proof $s =_E t$ where $s$ and $t$ are in $\eta$-long $\beta$-normal form and $t$ is a strict subterm of $s$. This implies in particular that the sets of free variables of the left-hand and right-hand sides of the axioms of $E$ are the same and are not reduced to a variable. We also assume that the relation $R/E := (=_E \rightarrow_R =_E)$ is terminating. This implies that $R/E \cup sst$ is terminating, where $sst$ is the strict subterm relation.

## 3 Critical pairs and coherence pairs

Mayr and Nipkow [13] do not need to consider coherence pairs since they deal with $R/E$ which allows $E$-equalities *above* the position to be rewritten. Similarly as in the work of Jouannaud and Kirchner [10], we need to consider not only critical pairs, but also *coherence pairs* to take into account the interactions of $R$ and $E$. Standard completion relies upon the fact that the local confluence amounts to the joinability of critical pairs. Then, confluence is obtained via Newman's lemma [16] with the additional termination assumption. In our case, as in Jouannaud and Kirchner's paper, we need to put together the assumptions of joinability of the critical pairs, and of the coherence pairs, plus the termination of $R/E$ to get the confluence. In this section, we give two technical lemmas (lemmas 5 and 6), which correspond to the "interesting" peaks. The next section is devoted to the confluence.

**Definition 5 (Critical pairs of $R^E$).** *Let $l_1 \to r_1$ and $l_2 \to r_2$ be two rewrite rules in an equational PRS $< R, E >$, and $p \in \mathcal{P}\mathrm{os}(l_1)$ such that*

- *$fv(l_1) \cap bv(l_1) = \emptyset$,*
- *the symbol at position $p$ in $l_1$ is not a free variable,*
- *$\lambda\overline{x_k}.l_1|_p\theta =_E \lambda\overline{x_k}.l_2\sigma\theta$, where $\{\overline{x_k}\} = bv(l_1, p)$ and $\sigma$ is an $\overline{x_k}$-lifter of $l_2$ away from $fv(l_1)$ and $\theta \in CSU_E(\lambda\overline{x_k}.l_1|_p, \lambda\overline{x_k}.l_2\sigma)$ such that $\mathcal{D}\mathrm{om}(\theta) \cup \{\overline{x_k}\} = \emptyset$.*

*Then $< r_1\theta, l_1[r_2\sigma]_p\theta >$ is an $E$-*critical pair *of $l_2 \to r_2$ on $l_1 \to r_1$ at position $p$. The set of all $E$-critical pairs of $< R, E >$ is denoted by $CP(R^E)$.*

The following lemma is straightforward in the first-order case, but requires a proof here, due to the presence of $\lambda$-binders. It states that when there is a critical pair, then there exists a corresponding critical peak.

**Lemma 4.** *If $< u_1, u_2 > \in CP(R^E)$ then there exists a term $s$ such that $s \to_{R,\Lambda} u_1$ and $s \to_{R^E} u_2$.*

*Proof.* $< u_1, u_2 > \in CP(R^E)$ implies by definition that $u_1 = r_1\theta$ and $u_2 = l_1[r_2\sigma]_p\theta$, where $\overline{x_k} = bv(l_1, p)$ and $\sigma$ is an $\overline{x_k}$-lifter of $l_2$ away from $fv(l_1)$ and $\theta$ is an $E$-unifier of $\lambda\overline{x_k}.l_1|_p$ and $\lambda\overline{x_k}.l_2\sigma$. Let $s \equiv l_1\theta$. Then $s \to_{R,\Lambda} r_1\theta \equiv u_1$ with the rule $l_1 \to r_1 \in R$. On the other hand, we have

$$\begin{array}{lll} \lambda\overline{x_k}.(l_1|_p\theta) & \equiv (\lambda\overline{x_k}.l_1|_p)\theta & \text{using } \alpha\text{-conversion if necessary} \\ & =_E (\lambda\overline{x_k}.l_2\sigma)\theta & \text{by definition of critical pairs} \\ & \equiv \lambda\overline{x_k}.(l_2\sigma\theta) & (Dom(\theta) \cup Cod(\theta)) \cap \overline{x_k} = \emptyset \end{array}$$

hence $l_1|_p\theta =_E l_2\sigma\theta$.
But $l_1|_p$ is a pattern and $bv(l_1, p) = \{\overline{x_k}\} \cap \mathcal{D}\mathrm{om}(\theta) = \emptyset$: by lemma 1, $l_1|_p\theta \equiv l_1\theta|_p$. Hence $l_1\theta|_p \equiv l_1|_p\theta =_E l_2\sigma\theta$. Finally

$$s \equiv l_1\theta \equiv l_1\theta[l_1\theta|_p]_p \overset{*}{\leftrightarrow}_{E,\geq p} l_1\theta[l_2\sigma\theta]_p \to_R l_1\theta[r_2\sigma\theta]_p \equiv u_2$$

**Definition 6 (Coherence pairs of $R$ on $E$).** *A* coherence pair *of $R$ on $E$ is a critical pair of $R$ on $E \cup E^{-1}$. The set of coherence pairs of $R$ on $E$ is denoted by $CHP(R, E)$.*

**Definition 7.** *Two terms $s$ and $t$ are* $R^E$-joinable *if there exist two terms $s'$ and $t'$ such that $s \xrightarrow{*}_{R^E} s'$ and $t \xrightarrow{*}_{R^E} t'$ and $s' \xleftrightarrow{*}_E t'$.*

**Definition 8.**

- $R^E$ *is* confluent *if whenever there exist $s$, $s_1$, $s_2$ such that $s \xrightarrow{*}_{R^E} s_1$ and $s \xrightarrow{*}_{R^E} s_2$, then $s_1$ and $s_2$ are $R^E$-joinable.*
- $R^E$ *is* coherent *if whenever there exist $s$, $s_1$, $s_2$ such that $s \overset{*}{\leftrightarrow}_E s_1$ and $s \xrightarrow{*}_{R^E} s_2$, then $s_1$ and $s_2$ are $R^E$-joinable.*

**Lemma 5.** *Assume that for all critical pairs $< u_1, u_2 > \in CP(R^E)$, $u_1$ and $u_2$ are $R^E$-joinable. Consider a peak of the form:*

$$\begin{tikzcd} & s \arrow[dl, "{R,p}"'] \arrow[dr, "{R^E, p\cdot q}"] & \\ s_1 & & s_2 \end{tikzcd}$$

*Then $s_1$ and $s_2$ are joinable in the following way:*

$$\begin{tikzcd} & & s \arrow[dl, "{R,p}"'] \arrow[dr, "{R^E, p\cdot q}"] & & \\ {} \arrow[r, leftrightarrow, "*", "E"'] \arrow[dr, "*", "R^E"'] & s_1 & & s_2 \arrow[r, leftrightarrow, "*", "E"'] & {} \arrow[dl, "*"', "R^E"] \\ & {} \arrow[rr, leftrightarrow, "*", "E"'] & & {} & \end{tikzcd}$$

*Proof.* Two cases are to be considered:
1. The redexes do not overlap.
$s|_p = l_1\theta_1$, $s_1 \equiv s[r_1\theta_1]_p$, $q = q_1 \cdot q_2$ with $l_1|_{q_1} = F(\overline{x_k})$ where $F$ is a free variable. $F\theta_1$ is of the form $\lambda\overline{x_k}.t$. Let $\theta'_1 = \theta_1 + \{F \mapsto \lambda\overline{x_k}.t[r_2\theta_2]_{q_2}\}$. Now $t|_{q_2} = (F(\overline{x_k})\theta_1)|_{q_2} = (l_1|_{q_1}\theta_1)|_{q_2} = (l_1\theta_1)|_{q_1}|_{q_2}$ (because $l_1$ is a pattern and applying $\theta_1$ creates no new redex).
$F\theta_1 = \lambda\overline{x_k}.t \rightarrow_{R^E} \lambda\overline{x_k}.t[r_2\theta_2]_{q_2} = F\theta'_1$. Hence $\theta_1 \overset{\equiv}{\rightarrow}_{R^E} \theta'_1$. By lemma 2, $r_1\theta_1 \xrightarrow{*}_{R^E} r_1\theta'_1$.

Let $H$ be a new variable and $l_0 = l_1[H(\overline{x_k})]_{q_1}$. let $\theta_0 = \theta_1 \cup \{H \mapsto F\theta'_1\}$ and $\theta'_0 = \theta_1 + \{F \mapsto F\theta'_1\}$. We have $\theta_1 \overset{\equiv}{\rightarrow}_{R^E} \theta'_1$, hence $\theta_0 \overset{\equiv}{\rightarrow}_{R^E} \theta'_0$. Now,

$$\begin{aligned} s_2|_p &= (l_1\theta_1)[r_2\theta_2]_q \\ &= (l_1\theta_1)[F(\overline{x_k})\theta'_1]_{q_1} \\ &= l_0\theta_0 \xrightarrow{*}_{R^E} l_0\theta'_0 \\ &= l_1\theta'_1 \rightarrow_R r_1\theta'_1 \end{aligned}$$

We are in the following situation:

$$\begin{tikzcd}
& & s \arrow[dll, "R{,}p"'] \arrow[drr, "R^E{,}p\cdot q"] & & \\
s[r_1\theta_1]_p \arrow[dr, "*", "R^E"'] & & & & s[r_2\theta_2]_{p\cdot q} \arrow[dl, "*"', "R^E"] \\
& s[r_1\theta'_1]_p & & s[l_1\theta'_1]_p \arrow[ll, "R"'] &
\end{tikzcd}$$

2. The two redexes overlap.
$s|_p = l_1\theta_1$ and $l_1|_q$ is defined and is not a free variable. $s|_{p\cdot q} =_E l_2\theta_2$.

$$\begin{aligned} l_1\theta_1|_q &= l_1|_q\theta_1 && \text{by lemma 1} \\ &= s|_{p\cdot q} =_E l_2\theta_2 \end{aligned}$$

Define $\theta_0$ as $\theta_0 = \theta_1 \cup \theta'_2$ with $\theta'_2 = \{F\rho \mapsto \lambda\overline{x_k}.F\theta_2 \mid F \in fv(l_2)\}$. We meet the hypotheses of lemma 3: let $\{\overline{x_k}\} = bv(l_1, q)$ and let $\sigma$ be an $\overline{x_k}$-lifter of $l_2$ away from $fv(l_1)$, and $\rho$ the renaming associated with $\sigma$. $\lambda\overline{x_k}.l_1|_q$ and $\lambda\overline{x_k}.l_2\sigma$ are $E$-unifiable by $\theta_0$: $(\lambda\overline{x_k}.l_1|_q)\theta_0 =_E (\lambda\overline{x_k}.l_2\sigma)\theta_0$.

There exists $\theta \in CSU(\lambda\overline{x_k}.l_1|_q, \lambda\overline{x_k}.l_2\sigma)$ such that $\theta_0 =_E \theta\delta$, for some substitution $\delta$. Hence there exist a critical pair $< r_1\theta, (l_1[r_2\sigma]_q)\theta >$ and the corresponding critical peak, which is joinable by hypothesis:

$$\begin{tikzcd}
& & l_1\theta \arrow[dll, "R{,}\Lambda"'] \arrow[drr, "R^E{,}q"] & & \\
r_1\theta \arrow[dr, "*", "R^E"'] & & & & (l_1[r_2\sigma]_q)\theta \arrow[dl, "R^E"', "*"] \\
& v_1 \arrow[rr, leftrightarrow, "*", "E"'] & & v_2 &
\end{tikzcd}$$

We apply $\delta$ to this diagram (remember that $\theta_0 =_E \theta\delta$).

$$\begin{tikzcd}
& & l_1\theta_0 \arrow[dl, "R{,}\Lambda"'] \arrow[dr, "R^E{,}q"] & & \\
r_1\theta\delta \arrow[r, leftrightarrow, "*", "E"'] \arrow[dr, "*", "R^E"'] & r_1\theta_0 & & l_1[r_2\sigma]_q)\theta_0 \arrow[r, leftrightarrow, "*", "E"'] & l_1[r_2\sigma]_q)\theta\delta \arrow[dl, "*"', "R^E"] \\
& v_1\delta \arrow[rr, leftrightarrow, "*", "E"'] & & v_2\delta &
\end{tikzcd}$$

Now, $l_1\theta_0 = l_1(\theta_1 \cup \theta'_2) = l_1\theta_1$ because $fv(l_1) \cap \mathcal{D}\mathrm{om}(\theta'_2) = \emptyset$.
$r_1\theta_0 = r_1(\theta_1 \cup \theta'_2) = l_1\theta_1$ because $fv(r_1) \subseteq fv(r_1)$.
We have $(l_1[r_2\sigma]_q)\theta_0 = (l_1\theta_0)[r_2\sigma\theta_0]_q = l_1\theta_1[r_2\theta_2]_q$. It is now sufficient to plug the whole diagram in the context $s[\,\cdot\,]_p$ to get the result.

Note that if in the above proof $r_1$ is not a variable, then the positions of the $E$-equality steps in $r_1\theta_0 \longleftrightarrow^*_E r_1\theta\delta$ are not $\Lambda$. Under our assumption that $E$ has no variables as left-hand sides or right-hand sides of its axioms, a slightly stronger result holds when an $E$-equality step is applied above an $R^E$- step:

**Lemma 6.** *Assume that for all coherence pairs* $< u_1, u_2 > \in CP(R^E, E)$, $u_1$ *and* $u_2$ *are* $R^E$*-joinable. Consider a proof of the form:*

$$\begin{tikzcd}
& s \arrow[dl, leftrightarrow, "{E,p}"'] \arrow[dr, "{R^E,p\cdot q}"] & \\
s_1 & & s_2
\end{tikzcd}$$

*Then* $s_1$ *and* $s_2$ *are joinable in the following way:*

$$\begin{tikzcd}
& & s \arrow[dl, leftrightarrow, "{E,p}"'] \arrow[dr, "{R^E,p\cdot q}"] & & \\
{} \arrow[r, leftrightarrow, "{E,>p}", "*"'] \arrow[dr, "*", "R^E"'] & s_1 & & s_2 \arrow[r, leftrightarrow, "E", "*"'] & {} \arrow[dl, "*"', "R^E"] \\
& {} \arrow[rr, leftrightarrow, "*", "E"'] & & {} &
\end{tikzcd}$$

## 4 Confluence

We are now ready to state our main result.

**Theorem 2.** *Assume that for all critical pair* $< u_1, u_2 > \in CP(R^E)$ $u_1$ *and* $u_2$ *are* $R^E$*-joinable and that for every coherence pair* $< u_1, u_2 > \in CHP(R, E)$ $u_1$ *and* $u_2$ *are* $R^E$*-joinable. Assume in addition that the relation* $R/E = (\leftrightarrow_E \rightarrow_R \leftrightarrow_E)$ *is terminating. Then* $R^E$ *is confluent (and coherent).*

We actually prove a little more than the confluence of $R^E$. We define a *general peak* as a proof of the following form:

$$\begin{tikzcd}
& & \bullet \arrow[dl, "*"', "R^E"] \arrow[dr, "*", "R^E"'] & & \\
\bullet \arrow[r, leftrightarrow, "*", "E"'] & \bullet & & \bullet \arrow[r, leftrightarrow, "*", "E"'] & \bullet
\end{tikzcd}$$

and we show that the extremes of every general peak are $R^E$-joinable.

There are 4 types of general peaks, and for each type, we define a measure. The first component of our measure is a term. The first components will be compared using the union of the strict subterm relation and the relation $R/E$. The second component is 1 if the peak really has $R^E$-steps in both directions, 0 otherwise. The third component is the number of $E$-equality steps at the top of the peak. The last two components are compared using the usual ordering on naturals.

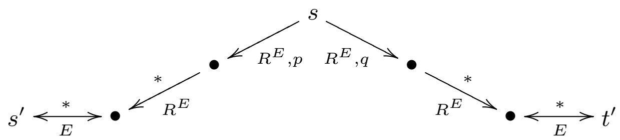
- type 1: There are $R^E$-steps in both directions and the first $R^E$ steps of the peak occur at comparable positions $p$ and $p \cdot q$.

measure $= (s, 1, n)$

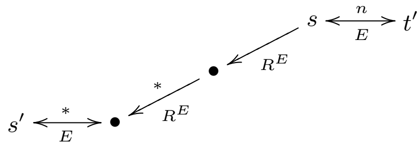
$$\begin{tikzcd}
& & & s \arrow[r, leftrightarrow, "n", "{E,\geq p}"'] \arrow[dl, "{R,p}"'] & t \arrow[dr, "{R^E,p\cdot q}"] & & & \\
& & \bullet \arrow[dl, "*", "R^E"'] & & & \bullet \arrow[dr, "*", "R^E"'] & & \\
s' \arrow[r, leftrightarrow, "*", "E"'] & \bullet & & & & & \bullet \arrow[r, leftrightarrow, "*", "E"'] & t'
\end{tikzcd}$$

– type 2: There are $R^E$-steps in both directions and the first $R^E$ steps of the peak occur at some parallel positions $p$ and $q$.

measure $= (s, 1, 0)$



– type 3: There is no $R^E$-step on one side of the peak.



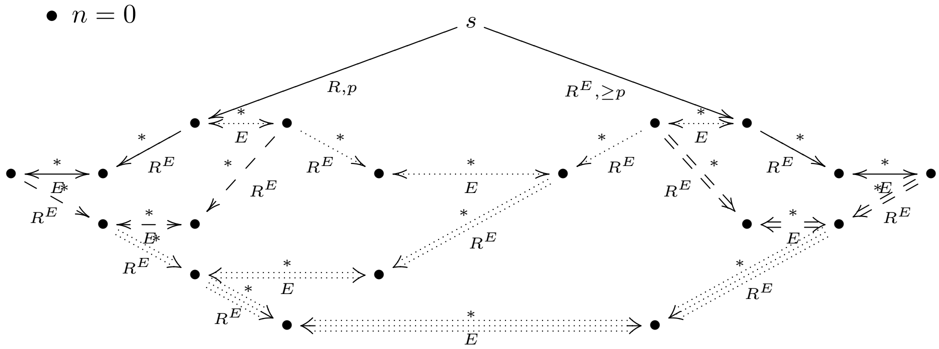
measure $= (s, 0, n)$

– type 4: There is no $R^E$-step in the peak.

$$s' \xleftrightarrow[E]{*} t' \quad \text{measure} = (s', 0, 0)$$

We investigate all the possible general peaks and we show the joinability of their extremes. The original peaks are drawn with plain arrows, and the various dashed arrows show how the peaks are joinable.

– When the general peak is of type 1, (with $(s, 1, n)$), we have to consider two distinct subcases, $n = 0$ and $n > 0$.
  - $n = 0$



Lemma 5

$- - -$ Induction hypothesis (1rst comp.)

$= = =$ Induction hypothesis (1rst comp.)

Induction hypothesis (1rst comp.)

Induction hypothesis (1rst comp.)

## 5 Application to $AC$ theories

In [1], we presented an $AC$-unification algorithm for higher-order patterns. It happens that $AC$-unification problems do not have minimal complete set of unifiers. Indeed, the equations of the form $\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^{\pi})$, while trivially solvable, have an infinite complete set of unifiers $\{\sigma_1, \sigma_2, \cdots\}$ such that $\sigma_{n+1}$ is strictly more general than $\sigma_n$. This was noticed by Qian and Wang [19] who give the following example:

*Example 2 ([19]).* Consider the equation $e \equiv \lambda xy.F(x,y) = \lambda xy.F(y,x)$ in the $AC$-theory of $+$. For $m \geq 0$, the substitution

$$\sigma_m = \{F \mapsto \lambda xy.G_m(H_1(x,y) + H_1(y,x), \ldots, H_m(x,y) + H_m(y,x))\}$$

is an $AC$-unifier of $e$. Every solution of $e$ is an instance of some $\sigma_i$ and $\sigma_{n+1}$ is strictly more general than $\sigma_n$.

On the other hand, the algorithm presented in [1] computes a finite complete set of *constrained $AC$-unifiers $\Sigma$*. A constrained $AC$-unifier is $\sigma|C$ where $\sigma$ is a substitution and $C$ a conjunction of flexible-flexible equations of the form $\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^\pi)$. Every $AC$-unifier of an equation $e$ is then an instance of $\sigma$ satisfying $C$ for some $\sigma|C \in \Sigma$.

$AC$-critical pairs will hence be represented by $< r_1\theta, l_1[r_2\sigma]_p\theta > |C$, where $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ are two rewrite rules and $\theta|C$ is a constrained $AC$-unifier of $\lambda\overline{x}^k.l_1|_p$ and $\lambda\overline{x}^k.l_2\sigma$ as given in definition 5. As usual, $< u_1, u_2 > |C$ represents all the equations $u_1\delta = u_2\delta$ such that $\delta$ is a solution of $C$. Checking the $R^{AC}$-joinability of $< u_1, u_2 > |C$ requires to check that for all the above mentioned $u_1\delta = u_2\delta$, $u_1\delta$ and $u_2\delta$ are $R^{AC}$-joinable.

We briefly sketch how to check the $R^{AC}$-joinability of the constrained $R^{AC}$-critical pairs. First, we assume that for each $F$ appearing in equations of the form $\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^\pi)$, the sequences of $\lambda$-bound variables above each occurrence of $F$ in the critical pair are the same. This can be acheived by using $\alpha$-conversion if necessary. Second, we assume that the constraint is *saturated* in the following sense : the set of permutations related to each $F$ is saturated wrt composition, yielding a subgroup $G_F$ of the permutation group of the arguments of $F$. Once the $G_F$s have been computed, we can assume that $F(\overline{t})$ and $F(\overline{t}^\pi)$ do not both occur in the critical pair if $\pi \in G_F$. The computation of a canonical representation of the argument sequence of the arguments of $F$ can be computed starting from the leaves of the terms of the critical pair. Now, it is clear that if an unconstrained critical pair $< u_1, u_2 >$ is $R^{AC}$-joinable, so is the constrained critical pair $< u_1, u_2 > |C$. The difficulty is to show that if the unconstrained critical pair is irreducible, then there exists a solution $\delta$ of $C$ such that $< u_1\delta, u_2\delta >$ is irreducible. This is the purpose of the following proposition:

**Proposition 1.** *Let $< R, AC >$ be an $AC$ pattern rewrite system. Let $t$ be a term, and $C$ a conjunction of equations of the form $\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^\pi)$. Let $G_F$ be the subgroup of permutations generated by the permutations of the equations involving $F$ in $C$. We assume that $t$ has not both occurrences of $F(\overline{t})$ and $F(\overline{t}^\pi)$ for $\pi \in G_F$. If $t$ is not $R^{AC}$-reducible, then there exists an instance of $t$ by a solution of $C$ which is irreducible.*

*Proof (Sketched).* The result is straightforward if $R$ is left-linear. The only case when applying a solution of $C$ to an irreducible term $t$ turns $t$ into a reducible term is when this makes two non $AC$-equal subterms $F(\overline{t_n})$ and $F(\overline{t'_n})$ $AC$-equal. Assume that $\{\overline{t_n}\}$ contains a term $t_i$ which does not appear in $\{\overline{t'_n}\}$ (or the converse). The substitution $\delta = \{F \mapsto \lambda\overline{x} \cdot \Sigma_{\pi \in G_F} H(x_{\pi(i)})\}$ is a solution of $C$ but $F(\overline{t_n})\delta$ contains an occurrence of $H(t_i)$ while $F(\overline{t'_n})\delta$ does not, hence the two terms cannot be $AC$-equal.

We assume now that the sets of terms occurring in $\{\overline{t_n}\}$ and $\{\overline{t'_n}\}$ are the same. The sequences $\overline{t_n}$ and $\overline{t'_n}$ can be completed in $\overline{t_m}$ and $\overline{t'_m}$ respectively in such a way to obtain the same associated multisets. Now, there exists a permutation $\pi$ such that $\overline{t'_m} = \overline{t_m}^{\pi}$. If $\pi \in G_F$, it operates only on the $n$ first elements, hence $\overline{t'_n} = \overline{t_n}^{\pi}$. But both $F(\overline{t_n})$ and $F(\overline{t'_n}) = F(\overline{t_n}^{\pi})$ appear in $t$, a contradiction.

We are left to consider the case when $\pi \notin G_F$. If for every $\pi' \in G_F$, $\overline{t'_n} = (t_{\pi(1)}, \ldots, t_{\pi(n)}) \neq \overline{t_n}^{\pi'}$, then the substitution $\delta = \{F \mapsto \lambda\overline{x_n}.\Sigma_{\pi' \in G_F} H(\overline{x_n}^{\pi'})\}$ is a solution of $C$. But $F(\overline{t_n})\delta$ has no occurrence of $H(t_{\pi(1)}, \ldots, t_{\pi(n)})$ while $F(\overline{t'_n})\delta$ has one. The two terms are not $AC$-equal. Finally, if there exists $\pi' \in G_F$ such that $(t_{\pi(1)}, \ldots, t_{\pi(n)}) = \overline{t_n}^{\pi'}$, then $\pi$ operates only on the $n$ first elements, hence $n = m$. Now $\overline{t_n}^{\pi} = \overline{t_n}^{\pi'}$ and $\pi \neq \pi'$ because $\pi \notin G_F$. Hence $\pi = \theta\pi'$ for some $\theta$ which permutes only identical $t_i$s (this means that $\overline{t_n} = \overline{t_n}^{\theta}$). We have $F(\overline{t'_n}) = F(\overline{t_n}^{\theta^{-1}\pi}) = F(\overline{t_n}^{\pi'})$ with $\pi' \in G_F$. Again, both $F(\overline{t_n})$ and $F(\overline{t'_n}) = F(\overline{t_n}^{\pi'})$ appear in $t$, a contradiction.

## 6 Conclusion

We have proposed a theory of pattern rewrite systems modulo an equational theory. The assumptions we make on both the rewrite system and the equational theory are very similar to those considered by Jouannaud and Kirchner in the first-order case. In particular, while $AC$ meets the assumptions on $E$, our work will need to be significantly extended for dealing with non-simple theories (the first that comes to mind being $ACU$). We will investigate the possibility to extend Marché's theory of $S$-normalized rewriting [12] to PRSs. For this, it will also be necessary to design unification algorithms for other theories than $AC$. The $AC$-unification algorithm proposed in [1] should extend to the usual extensions of $AC$.

## References


1. Alexandre Boudet and Evelyne Contejean. AC-unification of higher-order patterns. In Gert Smolka, editor, *Principles and Practice of Constraint Programming*, volume 1330 of *Lecture Notes in Computer Science*, pages 267–281, Linz, Austria, October 1997. Springer-Verlag.
2. Val Breazu-Tannen. Combining algebra and higher-order types. In *Proc. 3rd IEEE Symp. Logic in Computer Science, Edinburgh*, July 1988.
3. Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. North-Holland, 1990.

4. Gilles Dowek. Third order matching is decidable. In *Proc. 7th IEEE Symp. Logic in Computer Science, Santa Cruz*, pages 2–10. IEEE Comp. Society Press, 1992.
5. Warren D. Goldfarb. Note on the undecidability of the second-order unification problem. *Theoretical Computer Science*, 13:225–230, 1981.
6. R. Hindley and J. Seldin. *Introduction to Combinators and λ-calculus.* Cambridge University Press, 1986.
7. Gérard Huet. *Résolution d'équations dans les langages d'ordre* $1, 2, \ldots \omega$. Thèse d'Etat, Univ. Paris 7, 1976.
8. Gérard Huet. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, October 1980.
9. Gérard Huet. A complete proof of correctness of the Knuth-Bendix completion algorithm. *Journal of Computer and System Sciences*, 23:11–21, 1981.
10. Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4), November 1986.
11. Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.
12. Claude Marché. Normalized rewriting: an alternative to rewriting modulo a set of equations. *Journal of Symbolic Computation*, 21(3):253–288, 1996.
13. R. Mayr and T. Nipkow. Higher-order rewrite ststems and their confluence. To appear in TCS.
14. Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192(1):3–29, February 1998.
15. D. Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. In P. Schroeder-Heister, editor, *Extensions of Logic Programming*. LNCS 475, Springer Verlag, 1991.
16. M. H. A. Newman. On theories with a combinatorial definition of 'equivalence'. *Ann. Math.*, 43(2):223–243, 1942.
17. T. Nipkow. Higher order critical pairs. In *Proc. IEEE Symp. on Logic in Comp. Science*, Amsterdam, 1991.
18. Vincent Padovani. *Filtrage d'ordre supérieur*. PhD thesis, Université de Paris VII, 1996.
19. Zhenyu Qian and Kang Wang. Modular AC-Unification of Higher-Order Patterns. In Jean-Pierre Jouannaud, editor, *First International Conference on Constraints in Computational Logics*, volume 845 of *Lecture Notes in Computer Science*, pages 105–120, München, Germany, September 1994. Springer-Verlag.