Confluence of Orthogonal Higher-Order Rewrite Systems: Proof by Parallel Moves

Toshiyuki Yamada

Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305–8573, Japan

toshi@score.is.tsukuba.ac.jp

ISE–TR–00–169

Abstract

In this paper we propose higher-order rewrite systems without bound variables. In order to prove their confluence under the assumption of orthogonality, we study a simple proof method which employs a characterization of the diamond property of a parallel reduction. By an application of the proof method, we obtain a new confluence result for orthogonal higher-order conditional rewrite systems.

Confluence of Orthogonal Higher-Order Rewrite Systems: Proof by Parallel Moves

Toshiyuki Yamada

Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305–8573, Japan

toshi@score.is.tsukuba.ac.jp

February 1, 2000

Abstract

In this paper we propose higher-order rewrite systems without bound variables. In order to prove their confluence under the assumption of orthogonality, we study a simple proof method which employs a characterization of the diamond property of a parallel reduction. By an application of the proof method, we obtain a new confluence result for orthogonal higher-order conditional rewrite systems.

1 Introduction

Higher-order rewriting is a computation model which deals with higher-order terms. Higher-order functions and bound variables are usually used for constructing the set of higher-order terms. The use of bound variables enriches the descriptive power of higher-order rewrite systems. However, it makes the computation mechanism more complicated. For example, consider the following two specifications of the addition function in Peano arithmetic. The rewrite rules on the left is of first-order rewriting and on the right is of higher-order rewriting.

$$\begin{array}{rcl} 0 + x & \to & x \\ \mathsf{S}(x) + y & \to & \mathsf{S}(x + y) \end{array} \qquad \begin{array}{rcl} \lambda x \,.\, 0 + x & \to & \lambda x \,.\, x \\ \lambda xy \,.\, \mathsf{S}(x) + y & \to & \lambda xy \,.\, \mathsf{S}(x + y) \end{array}$$

The presentation of the higher-order rewrite system in this introduction is based on [vR99] with a slight change in notation. Examples of computations

in two systems are:

$$\mathsf{S}(\underline{\mathsf{S}(0)+0}) \to \mathsf{S}(\mathsf{S}(0+0)) \qquad \mathsf{S}(\underline{\mathsf{S}(0)+0}) \to \mathsf{S}((\lambda xy\,.\,\mathsf{S}(x)+y)00) \to \mathsf{S}(\underline{(\lambda xy\,.\,\mathsf{S}(x+y))00}) \to \mathsf{S}(\mathsf{S}(0+0))$$

Here, the underlined subterm is computed in each step. The computational mechanism in first-order rewriting is simple: we replace a subterm which matches the left-hand side of a rule to the corresponding right-hand side. On the other hand, additional steps for dealing with bound variables (beta expansion and reduction) are required in higher-order rewriting.

In this paper, we propose higher-order rewrite systems whose computational behaviour is the same as first-order rewriting. Based on the new definition of higher-order rewrite systems (given in the next section) we investigate the confluence property of the rewrite systems. We first study a method for proving confluence using parallel reduction (in Section 3). Based on the proof method introduced, we prove the confluence of orthogonal higher-order rewrite systems (in Section 4). This confluence result is further extended to the case of conditional higher-order rewriting (in Section 5).

2 Higher-Order TRS without Bound Variables

We assume the reader is familiar with abstract rewrite systems (ARSs) and (first-order) term rewrite systems (TRSs). In this section we propose a simple extension of first-order rewrite systems (cf. [DJ90], [Klo92], [BN98]) to the higher-order case. As opposed to previous works (e.g. Combinatory Reduction Systems [Klo80], Higher-Order Rewrite Systems [NP98]), our higher-order extension dispenses with bound variables.

Arities are used for the purpose of constructing well-formed terms. In the first-order setting, an arity is just a natural number denoting the number of arguments of a function. We generalize the notion of arity in order to construct higher-order terms.

Definition 1 (arity)
The set of *arities* is the smallest set A which contains the *base arity* 0 and satisfies the property that $(\alpha_1 \cdots \alpha_n \alpha) \in \mathsf{A}$ whenever $\alpha_1, \ldots, \alpha_n, \alpha \in \mathsf{A}$ with $n \geq 1$. We call a non-base arity a *function arity*. The outermost parentheses of an arity may be omitted when no confusion can arise.

Example 2 (arity)
The following table contains some examples of arities and their intuitive meanings.

arity	meaning
0	a value
00	a function with one argument
000	a function with two arguments
(00)0	a function which takes a function as an argument
0(00)	a function which returns a function

Definition 3 (term)
Let $V^{[\alpha]}$ be a set of *variable symbols* of arity α and $C^{[\alpha]}$ be a set of *constant symbols* of arity α, for every arity α. The set $\mathsf{T}(V,C)^{[\alpha]}$ of *(higher-order) terms* of arity α is the smallest set satisfying the following two properties:
(1) If $t \in V^{[\alpha]} \cup C^{[\alpha]}$ then $t \in \mathsf{T}(V,C)^{[\alpha]}$,
(2) If $n \geq 1$, $t_0 \in \mathsf{T}(V,C)^{[\alpha_1 \cdots \alpha_n \alpha]}$, and $t_i \in \mathsf{T}(V,C)^{[\alpha_i]}$ for $i = 1, \ldots, n$, then $(t_0\, t_1 \cdots t_n) \in \mathsf{T}(V,C)^{[\alpha]}$.
We also define

$$V = \bigcup_{\alpha \in \mathsf{A}} V^{[\alpha]}, \quad C = \bigcup_{\alpha \in \mathsf{A}} C^{[\alpha]}, \quad \text{and} \quad \mathsf{T}(V,C) = \bigcup_{\alpha \in \mathsf{A}} \mathsf{T}(V,C)^{[\alpha]}.$$

In order to be consistent with the standard definition of first-order terms, we use $t_0(t_1, \ldots, t_n)$ as an alternative notation for $(t_0\, t_1 \cdots t_n)$. The outermost parentheses of a term can be omitted. To enhance readability, infix notation is allowed. We use the notation $t^{[\alpha]}$ to make the arity α of a term t explicit.

Note that we do not confuse non-variable symbols with function symbols as in the first-order case, because a variable symbol of non-base arity express a function. The term t_0 in a term of the form $(t_0\, t_1 \cdots t_n)$ expresses a function which is applied to the arguments $t_1, \cdots, t_n$. In the higher-order case we allow arbitrary terms, including variables, at the position of t_0, while only constant symbols are allowed in the first-order case. Thus a set of first-order terms is obtained as a subset of higher-order terms which satisfies both $V^{[\alpha]} = \varnothing$ whenever $\alpha \neq 0$, and $C^{[\alpha_1 \cdots \alpha_n \alpha]} = \varnothing$ whenever $\alpha_i \neq 0$ for some i or $\alpha \neq 0$.

Definition 4 (head symbol and variable set)
Let t be a term in $\mathsf{T}(V,C)$. The head symbol $\mathsf{head}(t)$ of t is defined as follows:

$$\mathsf{head}(t) = \begin{cases} t & \text{if } t \in V \cup C, \\ \mathsf{head}(t_0) & \text{if } t = (t_0\, t_1 \cdots t_n). \end{cases}$$

The set of variable symbols contained in t is denoted by $\mathsf{Var}(t)$.

The term t_0 in a term $(t_0\, t_1 \cdots t_n)$, which expresses a function, may be rewritten in higher-order rewriting. This fact should be reflected to the definitions of position and substitution.

Definition 5 (position)
A *position* is a sequence of natural numbers. The empty sequence ϵ is called the *root position*. Positions are partially ordered by $\leq$ as follows: $p \leq q$ if there exists a position r such that $pr = q$. The set of positions in a term t is denoted by $\mathsf{Pos}(t)$. The *subterm* $t_{|p}$ of t at position p is defined as follows:

$$t_{|p} = \begin{cases} t & \text{if } t \in V \cup C \text{ and } p = \epsilon, \\ t_{i|q} & \text{if } t = (t_0\, t_1 \cdots t_n) \text{ and } p = iq. \end{cases}$$

The term obtained from a term s by replacing its subterm at position p with a term t is denoted by $s[t]_p$. The set $\mathsf{Pos}(t)$ is divided into three parts. We say $p \in \mathsf{Pos}(t)$ is at a *variable position* of t if $t_{|p} \in V$, at a *constant position* if $t_{|p} \in C$, otherwise p is at an *application position*. We denote the set of variable positions, constant positions, and application positions in a term t by $\mathsf{Pos}_{\mathrm{v}}(t)$, $\mathsf{Pos}_{\mathrm{c}}(t)$, and $\mathsf{Pos}_{\mathrm{a}}(t)$, respectively.

Definition 6 (substitution)
A *substitution* σ is a function from V to $\mathsf{T}(V, C)$ such that its *domain*, defined as the set $\{\, x \in V \mid \sigma(x) \neq x \,\}$, is finite and $\sigma(x) \in \mathsf{T}(V, C)^{[\alpha]}$ whenever $x \in V^{[\alpha]}$. A substitution $\sigma : V \to \mathsf{T}(V, C)$ is extended to the function $\overline{\sigma} : \mathsf{T}(V, C) \to \mathsf{T}(V, C)$ as follows:

$$\overline{\sigma}(t) = \begin{cases} \sigma(t) & \text{if } t \in V \cup C, \\ (\overline{\sigma}(t_0)\, \overline{\sigma}(t_1) \cdots \overline{\sigma}(t_n)) & \text{if } t = (t_0\, t_1 \cdots t_n). \end{cases}$$

We will write $t\sigma$ instead of $\overline{\sigma}(t)$. A *renaming* is a bijective substitution from V to V. Two terms t and s are *unifiable* by a *unifier* σ if $s\sigma = t\sigma$.

Definition 7 (rewrite rule)
Let $\mathsf{T}(V, C)$ be a set of terms. A *rewrite rule* is a pair of terms, written as $l \to r$, such that

- $\mathsf{Var}(r) \subseteq \mathsf{Var}(l)$,
- $\mathsf{head}(l) \in C$, and
- l and r are of the same arity.

The terms l and r are called the *left-hand side* and the *right-hand side* of the rewrite rule, respectively. Let R be a set of rewrite rules. We call $\mathcal{R} = (R, V, C)$ a *higher-order term rewrite system (without bound variables)*. In this paper, we simply call $\mathcal{R}$ a TRS.

Definition 8 (rewrite relation)
Let $\mathcal{R} = (R, V, C)$ be a TRS. We say a term s *rewrites to* t, and write $s \to_{\mathcal{R}} t$, if there exists a rewrite rule $l \to r \in R$, a position $p \in \mathsf{Pos}(s)$ and a substitution σ such that $s_{|p} = l\sigma$ and $t = s[r\sigma]_p$. We call the subterm $s_{|p}$ a *redex* of s. In order to make the position p of a rewrite step explicit, we also use the notation $s \xrightarrow{p}_{\mathcal{R}} t$. Especially, a rewrite step at root position is denoted by $\xrightarrow{\epsilon}_{\mathcal{R}}$ and a rewrite step at non-root position is denoted by $\xrightarrow{>\epsilon}_{\mathcal{R}}$. When the underlying TRS $\mathcal{R}$ is clear from the context, we may omit the subscript $\mathcal{R}$ in $\to_{\mathcal{R}}$.

It is easy to see that every variable symbol is in normal form because of the second restriction imposed on the rewrite rules. The following example shows that a simple functional programming language with pattern matching can be modeled by higher-order term rewrite systems without bound variables.

Example 9 (higher-order rewriting)
Let $C = \{\mathsf{0}^{[0]}, \mathsf{S}^{[00]}, []^{[0]}, :^{[000]}, \mathsf{map}^{[(00)00]}, \circ^{[(00)(00)(00)]}, \mathsf{twice}^{[(00)(00)]}\}$, where $:$ and $\circ$ are infix constant symbols, and the set of variable symbols V contains $x^{[0]}$, $xs^{[0]}$, $F^{[00]}$, and $G^{[00]}$. We define the set of rewrite rules R as follows:

$$R = \left\{ \begin{array}{lcl} \mathsf{map}\ F\ [] & \to & [] \\ \mathsf{map}\ F\ (x : xs) & \to & F\ x\ :\ \mathsf{map}\ F\ xs \\ (F \circ G)\ x & \to & F\ (G\ x) \\ \mathsf{twice}\ F & \to & F \circ F \end{array} \right\}.$$

Examples of rewrite sequences of the TRS $\mathcal{R} = (R, V, C)$ are:

$$\begin{array}{lcl} \mathsf{map}\ \underline{(\mathsf{twice}\ \mathsf{S})}\ (0 : []) & \to_{\mathcal{R}} & \underline{\mathsf{map}\ (\mathsf{S} \circ \mathsf{S})\ (0 : [])} \\ & \to_{\mathcal{R}} & \underline{(\mathsf{S} \circ \mathsf{S})\ 0}\ :\ \mathsf{map}\ (\mathsf{S} \circ \mathsf{S})\ [] \\ & \to_{\mathcal{R}} & \mathsf{S}(\mathsf{S}0)\ :\ \underline{\mathsf{map}\ (\mathsf{S} \circ \mathsf{S})\ []} \\ & \to_{\mathcal{R}} & \mathsf{S}(\mathsf{S}0)\ :\ [] \\ \underline{\mathsf{map}\ (\mathsf{twice}\ \mathsf{S})\ (0 : [])} & \to_{\mathcal{R}} & \underline{(\mathsf{twice}\ \mathsf{S})\ 0}\ :\ \mathsf{map}\ (\mathsf{twice}\ \mathsf{S})\ [] \\ & \to_{\mathcal{R}} & \underline{(\mathsf{S} \circ \mathsf{S})\ 0}\ :\ \mathsf{map}\ (\mathsf{twice}\ \mathsf{S})\ [] \\ & \to_{\mathcal{R}} & \mathsf{S}(\mathsf{S}0)\ :\ \underline{\mathsf{map}\ (\mathsf{twice}\ \mathsf{S})\ []} \\ & \to_{\mathcal{R}} & \mathsf{S}(\mathsf{S}0)\ :\ [] \end{array}$$

where underlined redexes are rewritten.

3 Confluence by Parallel Moves

In this section we develop a method for proving confluence using parallel reduction. We assume the reader is familiar with the basic notions of abstract rewrite systems (ARSs). For more detailed descriptions on abstract rewriting, see, for example, Klop's survey [Klo92].

Definition 10 (properties of an ARS)
Let $\mathcal{A} = (A, \to)$ be an ARS. We use the following abbreviations:

property	definition	abbreviation
$\mathcal{A}$ has the *diamond property*	$\leftarrow \cdot \to \subseteq \to \cdot \leftarrow$	$\Diamond(\to)$
$\mathcal{A}$ is *confluent*	${}^*\!\leftarrow \cdot \to^* \subseteq \to^* \cdot {}^*\!\leftarrow$	$\mathrm{CR}(\to)$

Lemma 11 (confluence by simultaneous reduction)
Let $(A, \to)$ and $(A, \hookrightarrow)$ be ARSs.
(1) $\Diamond(\to) \Longrightarrow \mathrm{CR}(\to)$.
(2) If $\hookrightarrow^* = \to^*$ then $\mathrm{CR}(\hookrightarrow) \Longleftrightarrow \mathrm{CR}(\to)$.

Proof. Straightforward.

Definition 12 (parallel reduction)
Let $\mathcal{R} = (R, V, C)$ be a TRS. The *parallel reduction relation* induced by $\mathcal{R}$ is the smallest relation $\Vdash\!\!\!\rightarrow_{\mathcal{R}}$ such that
(1) if $t \in V \cup C$ then $t \Vdash\!\!\!\rightarrow_{\mathcal{R}} t$,
(2) if $s \xrightarrow{\epsilon}_{\mathcal{R}} t$ then $s \Vdash\!\!\!\rightarrow_{\mathcal{R}} t$, and
(3) if $n \geq 1$ and $s_i \Vdash\!\!\!\rightarrow_{\mathcal{R}} t_i$ for $i = 0, \ldots, n$, then $(s_0\, s_1 \cdots s_n) \Vdash\!\!\!\rightarrow_{\mathcal{R}} (t_0\, t_1 \cdots t_n)$.
We may omit the underlying TRS $\mathcal{R}$ in $\Vdash\!\!\!\rightarrow_{\mathcal{R}}$ if it is not important.

One can easily verify that every parallel reduction relation is reflexive. Note also that $\to \subseteq \Vdash\!\!\!\rightarrow \subseteq \to^*$, hence $\Vdash\!\!\!\rightarrow^* = \to^*$. From Lemma 11 we know that the diamond property of a parallel reduction relation is a sufficient condition for the confluence of the underlying TRS: $\Diamond(\Vdash\!\!\!\rightarrow) \Longrightarrow \mathrm{CR}(\Vdash\!\!\!\rightarrow) \Longleftrightarrow \mathrm{CR}(\to)$. The following lemma gives a characterization of the diamond property of a parallel reduction, which is inspired by Gramlich's characterization of the strong confluence of a parallel reduction [Gra96].

Lemma 13 (parallel moves)
We have $\Vdash\!\!\!\leftarrow \cdot \xrightarrow{\epsilon} \subseteq \Vdash\!\!\!\rightarrow \cdot \Vdash\!\!\!\leftarrow \Longleftrightarrow \Vdash\!\!\!\leftarrow \cdot \Vdash\!\!\!\rightarrow \subseteq \Vdash\!\!\!\rightarrow \cdot \Vdash\!\!\!\leftarrow$.

Proof. The implication from right to left is obvious by $\xrightarrow{\epsilon} \subseteq \Vdash\!\!\!\rightarrow$. For the reverse implication, suppose $\Vdash\!\!\!\leftarrow \cdot \xrightarrow{\epsilon} \subseteq \Vdash\!\!\!\rightarrow \cdot \Vdash\!\!\!\leftarrow$. We show that if $t \Vdash\!\!\!\leftarrow s \Vdash\!\!\!\rightarrow u$ then there exists a term v such that $t \Vdash\!\!\!\rightarrow v \Vdash\!\!\!\leftarrow u$, for all terms s, t, and u.

The proof is by induction on the structure of s. We distinguish three cases according to the parallel reduction $s \Vdash t$. If $s = t \in V \cup C$ then $t \Vdash v = u$ by taking $v = u$. If $s \xrightarrow{\epsilon} t$ or $s \xrightarrow{\epsilon} u$ then we use the assumption. Otherwise, we have $s = (s_0\, s_1 \cdots s_n)$, $t = (t_0\, t_1 \cdots t_n)$, and $u = (u_0\, u_1 \ldots u_n)$ for some $n \geq 1$ with $t_i \Vdash s_i \Vdash u_i$ $(i = 0, \ldots, n)$. By the induction hypothesis, we know the existence of terms v_i $(i = 0, \ldots, n)$ such that $t_i \Vdash v_i \Vdash u_i$. Let $v = (v_0\, v_1 \ldots v_n)$. Then we have $t \Vdash v \Vdash u$ by definition.

This lemma allows us to partially localize the test for the diamond property of a parallel reduction, though the complete localization is impossible as shown in the following example.

Example 14 (complete localization of parallel moves)
In this example we show that the implication $\leftarrow \cdot \xrightarrow{\epsilon} \subseteq \Vdash \cdot \Vdash \implies \Vdash \cdot \xrightarrow{\epsilon} \subseteq \Vdash \cdot \Vdash$ does not hold in general. Let $V = \varnothing$ and C be the set consisting of the constant symbol f of arity 000 and constant symbols $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}$ of the base arity. Consider the set R of rewrite rules defined by

$$R = \left\{ \begin{array}{llllll} \mathsf{f\,a\,a} & \to & \mathsf{c} & \mathsf{a} & \to & \mathsf{b} \\ \mathsf{f\,a\,b} & \to & \mathsf{d} & \mathsf{c} & \to & \mathsf{d} \\ \mathsf{f\,b\,a} & \to & \mathsf{d} & \mathsf{d} & \to & \mathsf{e} \\ \mathsf{f\,b\,b} & \to & \mathsf{e} & & & \end{array} \right\}.$$

It is easy to see that the inclusion $\leftarrow \cdot \xrightarrow{\epsilon} \subseteq \Vdash \cdot \Vdash$ holds. We have $\mathsf{f\,b\,b} \Vdash \mathsf{f\,a\,a} \xrightarrow{\epsilon} \mathsf{c}$ but $\mathsf{f\,b\,b} \Vdash \cdot \Vdash \mathsf{c}$ is not satisfied.

Definition 15 (parallel moves property)
We say a TRS satisfies the *parallel moves property*, and write PM($\to$), if the inclusion $\Vdash \cdot \xrightarrow{\epsilon} \subseteq \Vdash \cdot \Vdash$ holds.

Lemma 13 states that the parallel moves property is equivalent to the diamond property of a parallel reduction. The parallel moves property is a useful sufficient condition for proving the confluence of orthogonal rewrite systems.

Lemma 16 (confluence by parallel moves)
Every TRS with the parallel moves property is confluent.

Proof. We have PM($\to$) $\iff$ $\Diamond(\Vdash)$ $\implies$ CR($\to$) by Lemmata 13 and 11 with $\Vdash^* = \to^*$, see Fig.1.

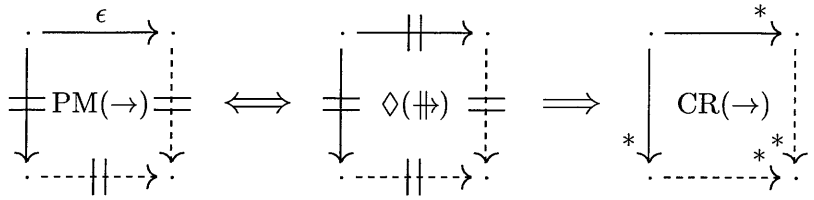

Figure 1: confluence by parallel moves

4 Confluence of Orthogonal Higher-Order TRSs

In this section, we give a simple proof of the confluence of orthogonal TRSs based on the parallel moves property. For the definition of orthogonality, we need the notions of left-linearity and overlap.

Definition 17 (left-linearity)
A TRS is *left-linear* if none of left-hand sides of rewrite rules contain multiple occurrences of a variable symbol.

Definition 18 (overlap)
A TRS is *overlapping* if there exist rewrite rules $l \to r$ and $l' \to r'$ without common variable symbols (after renaming) and a non-variable position $p \in \mathsf{Pos_c}(l) \cup \mathsf{Pos_a}(l)$ such that

- $l_{|p}$ and l' are unifiable, and
- if $p = \epsilon$ then $l \to r$ is not obtained from $l' \to r'$ by renaming variable symbols.

Example 19 (overlapping TRS)
Let C be the set consisting of constant symbols $\mathsf{f}, \mathsf{g}, \mathsf{h}$ of arity 00 and a, b of arity 0. Let V contain variable symbols F of arity 00 and x of arity 0. We define

$$R = \left\{ \begin{array}{rcl} \mathsf{f}\ (F\ x) & \to & F\ \mathsf{b} \\ \mathsf{g}\ \mathsf{a} & \to & \mathsf{h}\ \mathsf{b} \end{array} \right\}.$$

The TRS (R, C, V) is overlapping because the left-hand sides of the rewrite rules are unifiable at an application-position. In this TRS the term $\mathsf{f}\ (\mathsf{g}\ \mathsf{a})$ has two different normal forms: $\mathsf{g}\ \mathsf{b} \,{}_{\mathcal{R}}\!\leftarrow \mathsf{f}\ (\mathsf{g}\ \mathsf{a}) \to_{\mathcal{R}} \mathsf{f}\ (\mathsf{h}\ \mathsf{b}) \to_{\mathcal{R}} \mathsf{h}\ \mathsf{b}$. Note that the redex $(\mathsf{g}\ \mathsf{a})$ in the initial term is destroyed by the application of the first rewrite rule.

Definition 20 (orthogonality)
A TRS is *orthogonal* if it is left-linear and non-overlapping.

Now we are ready to give a proof that orthogonal TRSs are confluent. We first extend the use of parallel reduction to substitutions.

Definition 21 (parallel reduction of a substitution)
Let σ and τ be substitutions and X be a set of variable symbols. We write $\sigma \Vdash_{[X]} \tau$ if $\sigma(x) \Vdash \tau(x)$ for all $x \in X$.

Lemma 22 (parallel reduction of a substitution)
Let σ and τ be substitutions and t be a term. If $\sigma \Vdash_{[X]} \tau$ and $\mathsf{Var}(t) \subseteq X$ then $t\sigma \Vdash t\tau$.

Proof. An easy induction on the structure of t.

Lemma 23 (key properties for confluence)
Let $\mathcal{R} = (R, V, C)$ be an orthogonal TRS.
(1) $\xleftarrow{\epsilon} \cdot \xrightarrow{\epsilon} \subseteq =$.
(2) For all rewrite rules $l \to r \in R$, substitutions σ, and terms t, if $t \Vdash l\sigma \xrightarrow{\epsilon} r\sigma$ and not $l\sigma \xrightarrow{\epsilon} t$ then there exists a substitution τ such that $t = l\tau$ with $\sigma \Vdash_{[\mathsf{Var}(l)]} \tau$.

Proof.
(1) Since $\mathcal{R}$ is non-overlapping, we can only use (two renamed versions of) the same rewrite rule in R for rewriting a term at the same position. Hence we always obtain the same term.
(2) Since $\mathcal{R}$ is non-overlapping, there is no term s' with $l_{|p}\sigma \xrightarrow{\epsilon} s'$, for all non-variable position p in l. Hence we have $l_{|p} \Vdash t_{|p}$ for all variable positions p in l. Define the substitution τ by $\tau(x) = t_{|p}$ if $l_{|p} = x$ and $\tau(x) = x$ otherwise. This substitution is well-defined because there are no multiple ocuurences of x in l by left-linearity. It is easy to see that $\sigma \Vdash_{[\mathsf{Var}(l)]} \tau$ and $t = l\tau$ by construction.

Lemma 24 (Parallel Moves Lemma)
Every orthogonal TRS $\mathcal{R} = (R, V, C)$ has the parallel moves property, i.e., $\Vdash \cdot \xrightarrow{\epsilon} \subseteq \Vdash \cdot \Vdash$.

Proof. Suppose $t \Vdash s \xrightarrow{\epsilon} u$. We show $t \Vdash \cdot \Vdash u$. If $s \xrightarrow{\epsilon} t$ then the desired result follows from Lemma 23(1). Otherwise, we do not have $s \xrightarrow{\epsilon} t$. Since there exists a rewrite rule $l \to r \in R$ and a substitution σ such that $s = l\sigma$ and $u = r\sigma$, we know the existence of a substitution τ such that $t = l\tau$ with $\sigma \Vdash_{[\mathsf{Var}(l)]} \tau$ by Lemma 23(2). Therefore, $t = l\tau \xrightarrow{\epsilon} r\tau \Vdash r\sigma = u$ by Lemma 22 and $\mathsf{Var}(r) \subseteq \mathsf{Var}(l)$. Note that the case $s = t \in V$ is impossible because every variable symbol is in normal form and that the case $s = t \in C$ is contained in the case $s \xrightarrow{\epsilon} u$.

Note that the Parallel Moves Lemma does not hold for orthogonal higher-order rewrite systems with bound variables as observed in the literature, see [vO97] and [vR99]. A proof of confluence in such rewrite systems can be found in [MN98]. Now we conclude this section by the main result of this section and an example of its application.

Theorem 25 (confluence by orthogonality)
Every orthogonal TRS is confluent.

Proof. By Lemma 16 and the Parallel Moves Lemma (Lemma 24).

Example 26 (confluence by orthogonality)
The example TRS given in Example 9 is confluent because it is orthogonal.

5 Confluence of Orthogonal Higher-Order CTRSs

In this section, we generalize the confluence result presented in the previous section to the case of conditional rewriting. Bergstra and Klop proved the confluence of first-order orthogonal CTRSs in [BK86]. Their proof depends on the notion of development and the fact that every development is finite. Our result in this section generalizes their result to the higher-order case and also simplifies their confluence proof, based on the parallel moves property.

Definition 27 (conditional rewrite rule)
Let $\mathsf{T}(V, C)$ be a set of terms. A *conditional rewrite rule* $l \to r \Leftarrow c$ consists of a rewrite rule $l \to r$ and the *conditional part* c. Here c is a possibly empty finite sequence $c = l_1 \approx r_1, \ldots, l_n \approx r_n$ of equations such that every pair of terms l_i and r_i are of the same arity and $\mathsf{Var}(r) \subseteq \mathsf{Var}(c)$. If the conditional part is empty, we may simply write $l \to r$. Let R be a set of conditional rewrite rules. We call $\mathcal{R} = (R, V, C)$ a *(higher-order) conditional term rewrite system*, or simply a CTRS.

A variable in the right-hand side or in the conditional part of a rewrite rule which does not appear in the corresponding left-hand side is called an extra variable. In this paper, we allow extra variables only in the conditional part but not in the right-hand sides of rewrite rules.

Definition 28 (rewrite relation)
The rewrite relation $\to_{\mathcal{R}}$ of a CTRS $\mathcal{R} = (R, V, C)$ is defined as follows: $s \to_{\mathcal{R}} t$ if and only if $s \to_{\mathcal{R}_k} t$ for some $k \geq 0$. The minimum such k is

called the *level* of the rewrite step. Here the relations $\to_{\mathcal{R}_k}$ are inductively defined:

$$\begin{array}{lcl} \to_{\mathcal{R}_0} & = & \varnothing, \\ \to_{\mathcal{R}_{k+1}} & = & \{\, (t[l\sigma]_p, t[r\sigma]_p) \mid l \to r \Leftarrow c \in R,\ c\sigma \subseteq \to^*_{\mathcal{R}_k} \,\}. \end{array}$$

Here $c\sigma$ denotes the set $\{\, l'\sigma \approx r'\sigma \mid l' \approx r' \text{ belongs to } c \,\}$. Therefore $c\sigma \subseteq \to^*_{\mathcal{R}_k}$ with $c = l_1 \approx r_1, \ldots, l_n \approx r_n$ is a shorthand for $l_1 \to^*_{\mathcal{R}_k} r_1, \ldots, l_n \to^*_{\mathcal{R}_k} r_n$. We may abbreviate $\to_{\mathcal{R}_k}$ to $\to_k$ if there is no need to make the underlying CTRS explicit.

Properties of CTRSs are often proved by induction on the level of a rewrite step. So, it is useful for proving the confluence of orthogonal CTRSs to introduce the parallel reduction relations which are indexed by levels.

Definition 29 (parallel reduction relations indexed by levels)
Let $\mathcal{R}$ be a CTRS. We define $\Vdash_{\mathcal{R}_k}$ as the smallest relation such that
(1) $t \Vdash_{\mathcal{R}_k} t$ for all terms t,
(2) if $s \xrightarrow{\epsilon}_{\mathcal{R}_k} t$ then $s \Vdash_{\mathcal{R}_k} t$, and
(3) if $k \geq j_i$ and $s_i \Vdash_{\mathcal{R}_{j_i}} t_i$ for $i = 0, \ldots, n$, then $(s_0\, s_1 \cdots s_n) \Vdash_{\mathcal{R}_k} (t_0\, t_1 \cdots t_n)$.
We may abbreviate $\Vdash_{\mathcal{R}_k}$ to $\Vdash_k$ when no confusion can arise.

Observe that $s \Vdash_{\mathcal{R}} t$ if and only if $s \Vdash_{\mathcal{R}_k} t$ for some level $k \geq 0$. It is also easy to verify that $\to_{\mathcal{R}_k} \subseteq \Vdash_{\mathcal{R}_k} \subseteq \to^*_{\mathcal{R}_k}$ for all levels $k \geq 0$.

Definition 30 (properties of an ARS with indexes)
Let $\mathcal{A} = (A, \bigcup_{i\in I} \to_i)$ be an ARS whose rewrite relations are indexed. We use the following abbreviations:

definition	abbreviation
${}_j{\leftarrow}\cdot\to_k \subseteq \to_k \cdot {}_j{\leftarrow}$	$\Diamond^j_k(\to)$
${}^*_j{\leftarrow}\cdot\to^*_k \subseteq \to^*_k \cdot {}^*_j{\leftarrow}$	$\mathrm{CR}^j_k(\to)$

Lemma 31 (confluence by simultaneous reduction)
Let $(A, \bigcup_{i\in I} \to_i)$ and $(A, \bigcup_{i\in I} \hookrightarrow_i)$ be ARSs such that $\hookrightarrow^*_i = \to^*_i$ for all $i \in I$. We have $\Diamond^j_k(\hookrightarrow) \implies {}_j{\hookleftarrow}\cdot\to^*_k \subseteq \to^*_k \cdot {}_j{\hookleftarrow} \implies \mathrm{CR}^j_k(\to)$.

Proof. Straightforward.

Definition 32 (parallel moves property for CTRSs)
We say a CTRS satisfies the *parallel moves property* with respect to levels j and k, and write $\mathrm{PM}^j_k(\to)$, if the inclusion ${}_j{\Vdash}\cdot\xrightarrow{\epsilon}_k \subseteq \Vdash_k \cdot {}_j{\Vdash}$ holds.

Lemma 33 (parallel moves for CTRSs)
The following two statements are equivalent, for all $m \geq 0$.
(1) $\mathrm{PM}_k^j(\to)$ for all j, k with $j + k \leq m$.
(2) $\Diamond_k^j(\Vdash)$ for all j, k with $j + k \leq m$.

Proof. The implication (2) $\Rightarrow$ (1) is obvious because $\xrightarrow{\epsilon}_k \subseteq \Vdash_k$ by definition. For the proof of the implication (1) $\Rightarrow$ (2), suppose statement (1) holds. We show that if $t \,{}_j\!\lhd\!\Vdash\, s \Vdash_k u$ and $j + k \leq m$ then there exists a term v such that $t \Vdash_k v \,{}_j\!\lhd\!\Vdash\, u$, for all terms s, t, u and levels j, k. The proof is by induction on the structure of s. We distinguish three cases according to the parallel reduction $s \Vdash_j t$. If $s = t$ then $t \Vdash_k v = u$ by taking $v = u$. If $s \xrightarrow{\epsilon}_j t$ or $s \xrightarrow{\epsilon}_k u$ then we can use the assumption (1) in both cases because $j + k \leq m$. Otherwise, we have $s = (s_0\, s_1 \cdots s_n)$, $t = (t_0\, t_1 \cdots t_n)$, and $u = (u_0\, u_1 \ldots u_n)$ for some $n \geq 1$ with $t_i \,{}_{j_i}\!\lhd\!\Vdash\, s_i \Vdash_{k_i} u_i$ $(i = 0, \ldots, n)$. Since $j_i + k_i \leq j + k \leq m$, the induction hypothesis yields the existence of terms v_i such that $t_i \Vdash_{k_i} v_i \,{}_{j_i}\!\lhd\!\Vdash\, u_i$, for $i = 0, \ldots, n$. Let $v = (v_0\, v_1 \ldots v_n)$. Then we have $t \Vdash_k v \,{}_j\!\lhd\!\Vdash\, u$ by definition.

The following lemma gives a sufficient condition for the confluence of CTRSs.

Lemma 34 (confluence by parallel moves)
If a CTRS satisfies $\mathrm{PM}_k^j(\to)$ for all levels j and k then $\mathrm{CR}_k^j(\to)$ holds for all j, k, hence it is confluent.

Proof. By Lemmata 33 and 31 with $\Vdash_i^* = \to_i^*$ for all levels i, see Fig.2.

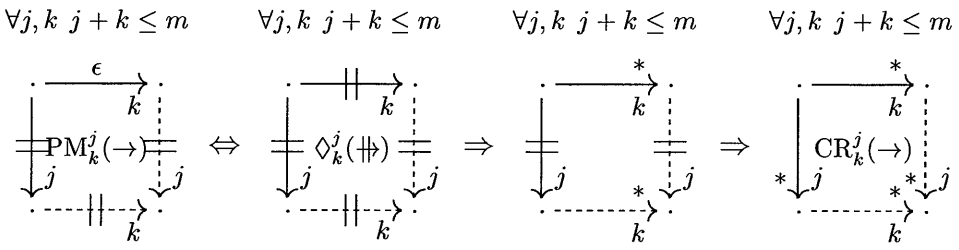

Figure 2: confluence of CTRSs by parallel moves

Imposing restrictions on reducibility of the right-hand sides of the conditions is important for ensuring the confluence of CTRSs.

Definition 35 (normal CTRS)
Let $\mathcal{R}$ be a CTRS. A term t is called *normal* if it contains no variables and is a normal form with respect to the unconditional version of $\mathcal{R}$. Here the unconditional version is obtained from a CTRS by dropping all conditions. A CTRS is called *normal* if every right-hand side of an equation in the conditional part of a rewrite rule is normal.

We extend the indexed version of a parallel reduction $\Vdash_k$ to the relation $\Vdash_{k[X]}$ on substitutions as in the unconditional case (Definition 21). It is easy to verify that the level version of Lemma 22 also holds. Now we are ready for proving the Parallel Moves Lemma for CTRSs. In the conditional case, we must confirm that the conditions are satisfied after the change of the substitution.

Lemma 36 (Parallel Moves Lemma for CTRSs)
Every orthogonal normal CTRS $\mathcal{R} = (R, V, C)$ satisfies $\mathrm{PM}_k^j(\to)$, i.e., ${}_j\Vdash \cdot \xrightarrow{\epsilon}_k \subseteq \Vdash_k \cdot {}_j\Vdash$, for all levels j and k.

Proof. We show that if $t\ {}_j\Vdash\ s \xrightarrow{\epsilon}_k u$ then there exists a term v such that $t \Vdash_k v\ {}_j\Vdash\ u$. The proof is by induction on $j + k$. The case $j + k = 0$ is trivial because $\xrightarrow{\epsilon}_0 = \varnothing$. Suppose $j + k > 0$. We distinguish two cases. If $s \xrightarrow{\epsilon}_j t$ then we have $s = u$ because $\mathcal{R}$ is non-overlapping and has no extra variable in the right hand sides of R. Hence we can take $v = s = u$ Consider the case that $s \xrightarrow{\epsilon}_j t$ does not hold. From $s \xrightarrow{\epsilon}_k u$ we know that there exists a conditional rewrite rule $l \to r \Leftarrow c \in R$ and a substitution σ such that $s = l\sigma$, $u = r\sigma$, and $c\sigma \subseteq \to_{k-1}^*$. Since $\mathcal{R}$ is non-overlapping, there is no term s' with $l_{|p}\sigma \xrightarrow{\epsilon}_j s'$ for all non-variable positions p in l. Define the substitution τ by $\tau(x) = t_{|p}$ if $l_{|p} = x$ and $\tau(x) = \sigma(x)$ otherwise. This substitution is well-defined by the left-linearity of $\mathcal{R}$. We have $\sigma \Vdash_{j[\mathsf{Var}(l,c)]} \tau$ and $t = l\tau$ by the definition of τ. From $\mathsf{Var}(r) \subseteq \mathsf{Var}(l)$ and the level version of Lemma 22 we obtain $r\sigma \Vdash_j r\tau$. It remains to show that $t = l\tau \Vdash_k r\tau$. So, we will prove $c\tau \subseteq \to_{k-1}^*$. Let $l' \approx r'$ be an arbitrary condition in c. We have to prove that $l'\tau \to_{k-1}^* r'\tau$. Since $c\sigma \subseteq \to_{k-1}^*$, we have $l'\sigma \to_{k-1}^* r'\sigma$. Moreover, $\sigma \Vdash_{j[\mathsf{Var}(l,c)]} \tau$, $\mathsf{Var}(c) \subseteq \mathsf{Var}(l, c)$, and the level version of Lemma 22 yields that $l'\sigma \Vdash_j l'\tau$. Hence $l'\tau\ {}_j\Vdash\ l'\sigma \to_{k-1}^* r'\sigma$. From the induction hypothesis and Lemmata 36 and 31, we know the existence of a term v such that $l'\tau \to_{k-1}^* v\ {}_j\Vdash\ r'\sigma$. Because $\mathcal{R}$ is normal, $r'\sigma = r' = r'\tau = v$. Hence $l'\tau \to_{k-1}^* r'\tau$. Therefore $l\tau \Vdash_k r\tau$.

Theorem 37 (confluence by orthogonality)
Every orthogonal normal CTRS is confluent.

Proof. By Lemma 34 and the Parallel Moves Lemma for CTRSs (Lemma 36).

6 Concluding Remarks

We have proposed higher-order rewrite systems without bound variables, which is close to the format of functional programming languages with pattern matching. For proving the confluence of orthogonal rewrite systems, we introduced the parallel moves property, which is a useful sufficient condition obtained by localizing the test for the diamond property of a parallel reduction. We proved the confluence of orthogonal higher-order TRSs and orthogonal normal higher-order CTRSs.

Since the class of higher-order (C)TRSs without bound variables is a proper extension of the first-order case, all known results for the first-order TRSs can be applied to the subclass of our (C)TRSs. We can also expect that many known results for the first-order TRSs can be lifted to the higher-order case without difficulty, because the behaviour of our higher-order extension is very close to that of the first-order (C)TRSs.

Suzuki et al. gave a sufficient condition for the confluence of orthogonal first-order CTRSs possibly with extra variables in the right-hand sides of the rewrite rules [SMI95]. The author conjectures that their result can be extended to the higher-order case.

Acknowledgments

The author is grateful to Aart Middeldorp, Femke van Raamsdonk, and Fer-Jan de Vries for their comments to the preliminary version of this paper.

References

[BK86] J.A. Bergstra and J.W. Klop. Conditional rewrite rules: Confluence and termination. *Journal of Computer and System Science*, 32:323–362, 1986.

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 6, pages 243–320. The MIT Press, 1990.

[Gra96] B. Gramlich. Confluence without termination via parallel critical pairs. In *Proceedings of the 21st International Colloquium on Trees in Algebra and Programming (CAAP'96)*, 1996. Lecture Notes in Computer Science 1059, pp. 211-225.

[Klo80] J.W. Klop. *Combinatory Reduction Systems.* PhD thesis, Rijksuniversiteit, Utrecht, 1980.

[Klo92] J.W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, chapter 1, pages 1–116. Oxford University Press, 1992.

[MN98] R. Mayr and T. Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192:3–29, 1998.

[NP98] T. Nipkow and C. Prehofer. *Higher-Order Rewriting and Equational Reasoning*, volume I, pages 399–430. Kluwer, 1998.

[SMI95] T. Suzuki, A. Middeldorp, and T. Ida. Level-confluence of conditional rewrite systems with extra variables in right-hand sides. In *Proceedings of the 6th International Conference on Rewriting Techniques and Applications*, 1995. Lecture Notes in Computer Science 914, pp. 179–193.

[vO97] V. van Oostrom. Developing developments. *Theoretical Computer Science*, 175:159–181, 1997.

[vR99] Femke van Raamsdonk. Higher-order rewriting. In *Proceedings of the 10th International Conference on Rewriting Techniques and Applications (RTA '99)*, 1999. Lecture Notes in Computer Science 1631, pp. 220–239.