Outermost-Fair Rewriting

Femke van Raamsdonk

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract. A rewrite sequence is said to be outermost-fair if every outermost redex occurrence is eventually eliminated. O'Donnell has shown that outermost-fair rewriting is normalising for almost orthogonal first-order term rewriting systems. In this paper we extend this result to the higher-order case.

1 Introduction

It may occur that a term can be rewritten to normal form but is also the starting point of an infinite rewrite sequence. In that case it is important to know how to rewrite the term such that eventually a normal form is obtained. The question of how to rewrite a term can be answered by a strategy, which selects one or more redex occurrences in every term that is not in normal form. If repeatedly contracting the redex occurrences that are selected by the strategy yields a normal form whenever the initial term has one, the strategy is said to be *normalising*.

A classical result for λ-calculus with β-reduction is that the strategy selecting the leftmost redex occurrence is normalising. This is proved in [CFC58]. For orthogonal first-order term rewriting systems, O'Donnell has shown in [O'D77] that the parallel-outermost strategy, which selects all redex occurrences that are outermost to be contracted simultaneously, is normalising. This result is a consequence of a stronger result which is also proved in [O'D77], namely that every outermost-fair rewrite sequence eventually ends in a normal form whenever the initial term has one. A rewrite sequence is said to be outermost-fair if every outermost redex occurrence is eventually eliminated.

This paper is concerned with the question of how to find a normal form in a higher-order rewriting system, in which rewriting is defined modulo simply typed λ-calculus. We extend the result by O'Donnell to the higher-order case: we show that outermost-fair rewriting is normalising for almost orthogonal higher-order rewriting systems, that satisfy some condition on the bound variables. This condition is called full extendedness. As in the first-order case, an immediate corollary of the main result is that the parallel-outermost strategy is normalising for orthogonal higher-order rewriting systems that are fully extended.

Our result extends and corrects a result by Bergstra and Klop, proved in the appendix of [BK86], which states that outermost-fair rewriting is normalising for orthogonal Combinatory Reduction Systems. Unfortunately, the proof presented in [BK86] is not entirely correct.

The remainder of this paper is organised as follows. The next section is concerned with the preliminaries. In Section 3 the notion of outermost-fair rewriting

is explained. In Section 4 the main result of this paper is proved, namely that outermost-fair rewriting is normalising for the class of almost orthogonal and fully extended higher-order rewriting systems. The present paper is rather concise in nature; for a detailed account the interested reader is referred to [Raa96].

2 Preliminaries

In this section we recall the definition of higher-order rewriting systems [Nip91, MN94], following the presentation in [Oos94, Raa96]. We further give the definitions of almost orthogonality and full extendedness. The reader is supposed to be familiar with simply typed λ-calculus with β-reduction (denoted by $\rightarrow_\beta$) and restricted $\overline{\eta}$-expansion (denoted by $\rightarrow_{\overline{\eta}}$); see for instance [Bar92, Aka93]. *Simple types*, written as $A, B, C, \ldots$ are built from base types and the binary type constructor $\rightarrow$. We suppose that for every type A there are infinitely many *variables of type* A, written as $x^A, y^A, z^A, \ldots$.

Higher-Order Rewriting Systems. The meta-language of higher-order rewriting systems, which we call the *substitution calculus* as in [Oos94, OR94, Raa96], is simply typed λ-calculus.

A *higher-order rewriting system* is specified by a pair $(\mathcal{A}, \mathcal{R})$ consisting of a rewrite alphabet and a set of rewrite rules over $\mathcal{A}$.

A *rewrite alphabet* is a set $\mathcal{A}$ consisting of simply typed *function symbols*. A *preterm* of type A over $\mathcal{A}$ is a simply typed λ-term of type A over $\mathcal{A}$. Preterms are denoted by $s, t, \ldots$. Instead of $\lambda x^A.s$ we will write $x^A.s$. We will often omit the superscript denoting the type of a variable. Preterms are considered modulo the equivalence relation generated by α, β and η. Every $\alpha\beta\eta$-equivalence class contains a $\beta\overline{\eta}$-normal form that is unique up to α-conversion. Such a representative is called a *term*. Terms are the objects that are rewritten in a higher-order rewriting system.

In the following, all preterms are supposed to be in $\overline{\eta}$-normal form. Note that the set of $\overline{\eta}$-normal forms is closed under β-reduction.

For the definition of a rewrite rule we first need to introduce the notion of rule-pattern, which is an adaptation of the notion of pattern due to Miller [Mil91]. A *rule-pattern* is a closed term of the form $x_1.\ldots.x_m.fs_1\ldots s_n$ such that every $y \in \{x_1, \ldots, x_m\}$ occurs in $fs_1\ldots s_n$, and it occurs only in subterms of the form $yz_1\ldots z_p$ with $z_1, \ldots, z_p$ the $\overline{\eta}$-normal forms of different bound variables not among $x_1, \ldots, x_m$. The function symbol f is called the *head-symbol* of the rule-pattern.

A *rewrite rule* over a rewrite alphabet $\mathcal{A}$ is defined as a pair of closed terms over $\mathcal{A}$ of the same type of the form $x_1.\ldots.x_m.s \rightarrow x_1.\ldots.x_m.t$, with $x_1.\ldots.x_m.s$ a rule-pattern. The head-symbol of a rewrite rule is the head-symbol of its left-hand side. Rewrite rules are denoted by $\mathtt{R}, \mathtt{R}', \ldots$.

The next thing to define is the rewrite relation of a higher-order rewriting system $(\mathcal{A}, \mathcal{R})$. To that end we need to introduce the notion of context. We use the symbol $\Box^A$ to denote a variable of type A that is supposed to be free. A

context of type A is a term with one occurrence of $\Box^A$. A context of type A is denoted by $C\Box^A$ and the preterm obtained by replacing $\Box^A$ by a term s of type A is denoted by $C[s]$.

Now the *rewrite relation* of a higher-order rewriting system $(\mathcal{A}, \mathcal{R})$, denoted by $\rightarrow$, is defined as follows : we have $s \rightarrow t$ if there is a context $C\Box^A$ and a rewrite rule $l \rightarrow r$ in $\mathcal{R}$ with l and r of type A such that s is the β-normal form of $C[l]$ and t is the β-normal form of $C[r]$. That is, such a rewrite step is decomposed as

$$s \,{}^{!}_{\beta}\!\!\leftarrow C[l] \rightarrow C[r] \twoheadrightarrow^{!}_{\beta} t$$

where $\twoheadrightarrow^{!}_{\beta}$ denotes a β-reduction to β-normal form. Note that the rewrite relation is defined on terms, not on preterms. The rewrite relation is decidable because the left-hand sides of rewrite rules are required to be rule-patterns.

As an example, we consider untyped lambda-calculus with beta-reduction and eta-reduction in the format of higher-order rewriting systems. The rewrite alphabet consists of the function symbols $\mathsf{app} : 0 \rightarrow 0 \rightarrow 0$ and $\mathsf{abs} : (0 \rightarrow 0) \rightarrow 0$ with 0 the only base type. The rewrite rules are given as follows:

$$z.z'.\mathsf{app}(\mathsf{abs}\, x.zx)z' \rightarrow_{\mathsf{beta}} z.z'.zz'$$
$$z.\mathsf{abs}(x.\mathsf{app}zx) \rightarrow_{\mathsf{eta}} z.z$$

The rewrite step $\mathsf{app}(\mathsf{abs}\, x.x)y \rightarrow_{\mathsf{beta}} y$ is obtained as follows:

$$\mathsf{app}(\mathsf{abs}\, x.x)y \,{}^{!}_{\beta}\!\!\leftarrow (z.z'.\mathsf{app}(\mathsf{abs}\, x.zx)z')(x.x)y \rightarrow_{\mathsf{beta}} (z.z'.zz')(x.x)y \twoheadrightarrow^{!}_{\beta} y.$$

In the sequel, types won't be mentioned explicitly.

Residuals. In the remainder of this paper the notions of redex occurrence and residual will be important. For their definitions we need two auxiliary notions we suppose the reader is familiar with.

The first one is the notion of position, and an ordering $\preceq$ on positions. A *position* is a finite sequence over $\{0, 1\}$. We write positions as $\phi, \chi, \psi, \ldots$. There is an operator for concatenating positions that is denoted by juxtaposition and is supposed to be associative. The neutral element for concatenation of positions is the empty sequence denoted by ϵ.

The set of positions of a (pre)term, and the sub(pre)term of a (pre)term at position ϕ are defined as in λ-calculus. For instance, the set of positions of the term $f(x.x)a$ is $\{\epsilon, 0, 00, 01, 010, 1\}$. The subterm of $f(x.x)a$ at position 01 is $x.x$ and the subterm of $f(x.x)a$ at position 1 is a.

The ordering $\preceq$ on the set of positions is defined as follows: we have $\phi \preceq \chi$ if there exists a position ϕ' such that $\phi\,\phi' = \chi$. The strict variant of this order is obtained by requiring in addition that $\phi' \neq \epsilon$.

The second auxiliary notion needed for the definition of the residual relation of a higher-order rewriting system, is a *descendant relation* tracing positions along β-reductions. We illustrate this notion by an example. Consider the preterm $f((x.y.xx)ab)$ and its reduction to β-normal form $f((x.y.xx)ab) \twoheadrightarrow^{!}_{\beta} f(aa)$. The descendant of the position 0 in $f((x.y.xx)ab)$ is the position 0 in

$f(aa)$, the descendants of the position 101 in $f((x.y.xx)ab)$ are the positions 10 and 11 in $f(aa)$, and the descendant of the position 10000 in $f((x.y.xx)ab)$ is the position 1 in $f(aa)$.

A *redex occurrence* in a term s is a pair $(\phi, l \to r)$ consisting of a position in s and a rewrite rule such that ϕ is the descendant of the position of the head-symbol of l along the reduction $C[l] \twoheadrightarrow^!_\beta s$. A redex occurrence $(\phi, l \to r)$ as above is said to be *contracted* in the rewrite step $s \to t$ with $s \,{}^!_\beta\!\!\leftarrow C[l]$ and $C[r] \twoheadrightarrow^!_\beta t$. Redex occurrences are denoted by $u, v, w, \ldots$.

In the remainder of the paper it will often be essential to know which redex occurrence is contracted in a rewrite step. This is made explicit by writing $u : s \to t$ or $s \xrightarrow{u} t$, if the redex occurrence u is contracted in the rewrite step $s \to t$.

The ordering on positions induces an ordering on redex occurrences as follows. Let $(\phi, l \to r)$ and $(\phi', l' \to r')$ be redex occurrences in a term t. Suppose that $l = x_1.\ldots.x_m.fs_1.\ldots.s_n$ and $l' = x_1.\ldots.x_{m'}.f's'_1.\ldots.s'_{n'}$. Then $\phi = \phi_0\, 0^m$ and $\phi' = \phi'_0\, 0^{m'}$ for some ϕ_0 and ϕ'_0. Now an ordering on redex occurrences, also denoted by $\preceq$, is defined as follows: $(\phi, l \to r) \preceq (\phi', l' \to r')$ if $\phi_0 \preceq \phi'_0$. For example, if $x.fx \to_{\mathrm{R}_1} x.gx$ and $a \to_{\mathrm{R}_2} b$ are rewrite rules, then we have $(0, \mathrm{R}_1) \preceq (1, \mathrm{R}_2)$ in the term fa. For the intuition behind the ordering on redex occurrences, it might be helpful to think of $fs_1 \ldots s_m$ as $f(s_1, \ldots, s_m)$.

The *descendant relation* for a higher-order rewriting system is defined using the descendant relation for β as follows. Let $s \,{}^!_\beta\!\!\leftarrow C[l] \to C[r] \twoheadrightarrow^!_\beta s'$ be the decomposition of a rewrite step $s \to s'$. A position ϕ in s descends to a position ϕ' in s' if there is a position χ in the $C\Box$-part of $C[l]$, which is hence also a position in $C[r]$, such that χ descends to ϕ along $C[l] \twoheadrightarrow^!_\beta s$ and χ descends to ϕ' along $C[r] \twoheadrightarrow^!_\beta s'$. Note that a position in s which descends from a position in the l-part of $C[l]$ along $C[l] \twoheadrightarrow_\beta s$ does not have a descendant in s'.

As an example, we consider the higher-order rewriting system defined by the rewrite rule

$$x.fx \to x.gxx.$$

We have the rewrite step $h(fa) \to h(gaa)$ which is obtained as follows:

$$h(fa) \,{}^!_\beta\!\!\leftarrow h((x.fx)a) \to h((x.gxx)a) \twoheadrightarrow^!_\beta h(gaa).$$

The descendant of the position 0 in $h(fa)$ is the position 0 in $h(gaa)$, the descendant of the position 11 in $h(fa)$ are the positions 101 and 11 in $h(gaa)$ and the position 10 in $h(fa)$ doesn't have a descendant in $h(gaa)$.

In this paper we will be concerned with higher-order rewriting systems that have the following property: if $(\phi, l \to r)$ is a redex occurrence in s, and ϕ descends to ϕ' along the rewrite step $u : s \to s'$, then $(\phi', l \to r)$ is a redex occurrence in s'. The redex occurrence $(\phi', l \to r)$ is then said to be a *residual* of the redex occurrence $(\phi, l \to r)$. If u and v are redex occurrences in a term s, then the set of residuals of v after performing the rewrite step $u : s \to t$ is denoted by $\mathrm{Res}(s, u)(v)$. The residual relation is extended in a straightforward way to rewrite sequences consisting possibly of more than one step and sets of redex occurrences.

Almost Orthogonality and Full Extendedness. The main result of this paper is concerned with higher-order rewriting systems that are *almost orthogonal* and *fully extended.* We will now explain the notions of almost orthogonality and full extendedness.

For the definition of almost orthogonality we need the notion of left-linearity. A rule-pattern $x_1.\ldots.x_m.fs_1\ldots s_n$ is *linear* if every $y \in \{x_1,\ldots,x_m\}$ occurs at most once (and hence, by the definition of a rule-pattern, exactly once) in $fs_1\ldots s_n$. A rewrite rule is said to be *left-linear* if its left-hand side is linear, and a higher-order rewriting system is said to be *left-linear* if all its rewrite rules are left-linear. For example, the rewrite rule $x.fx \to x.gx$ is left-linear, but the rewrite rule $x.fxx \to x.gx$ is not.

A higher-order rewriting system is said to be *orthogonal* if it is left-linear and has no critical pairs. A higher-order rewriting system is said to be *weakly orthogonal* if it is left-linear and all its critical pairs are trivial. Almost orthogonality lies in between orthogonality and weak orthogonality. A higher-order rewriting system is said to be *almost orthogonal* if it is weakly orthogonal with the additional property that overlap between redex occurrences occurs only at the root of the redex occurrences. The notion of overlapping redex occurrences is defined below. The notion of critical pair is the usual one, as defined in [Hue80]. The formal definition for higher-order rewriting systems is not given in the present paper.

Definition 1. Let $u = (\phi\, 0^n, l \to r)$ and $u' = (\phi'\, 0^{n'}, l' \to r')$ with $l = x_1.\ldots.x_m.fs_1\ldots s_n$ and $l' = x_1.\ldots.x_{m'}.f's'_1\ldots s'_{n'}$ be redex occurrences in a term s.

1. The redex occurrence u *nests* the redex occurrence u' if $\phi' = \phi\,\chi\,\psi$ such that the subterm of $fs_1\ldots s_n$ at position $\chi\, 0^i$ is a variable $y \in \{x_1,\ldots,x_m\}$, and ψ is an arbitrary position.
2. The redex occurrences u and u' are *overlapping* if
   (a) they are different,
   (b) $\phi \preceq \phi'$ and u does not nest u', or $\phi' \preceq \phi$ and u' does not nest u.

Definition 2. 1. A higher-order rewriting system is said to be *weakly head-ambiguous* if for every term s and for every two overlapping redex occurrences $u = (\phi\, 0^n, l \to r)$ and $u' = (\phi'\, 0^{n'}, l' \to r')$ with $l = x_1.\ldots.x_m.fs_1\ldots s_n$ and $l' = x_1.\ldots.x_{m'}.f's'_1\ldots s'_{n'}$ in s, we have $u : s \to t$ and $u' : s \to t$, that is, u and u' define the same rewrite step, and moreover $\phi = \phi'$, and hence necessarily $n = n'$.
2. A higher-order rewriting system is said to be *almost orthogonal* if it is left-linear and weakly head-ambiguous.

The rewriting system defined by the rewrite rules

$$x.fx \to x.gx$$
$$a \to b$$

is orthogonal. The rewriting system defined by the rewrite rules

$$fa \to fb$$
$$a \to b$$

is weakly orthogonal but not almost orthogonal. The rewriting system for parallel or defined by the rewrite rules

$$x.fax \to x.a$$
$$x.fxa \to x.a$$

is almost orthogonal but not orthogonal.

We will use the following notation. By $u \sharp v$ we denote that the redex occurrences u and v are overlapping. Two overlapping redex occurrences are by definition different, so the relation $\sharp$ is not reflexive. In an almost orthogonal rewriting system, we have the following: if $u \sharp v$ and $v \sharp w$, then $u \sharp w$ or $u = w$. This implication does not hold in a weakly orthogonal rewriting system. Consider for instance the weakly orthogonal higher-order rewriting system defined by the following rewrite rules:

$$x.f(gx) \to_{\mathrm{R}_1} x.f(gb)$$
$$ga \to_{\mathrm{R}_2} gb$$
$$a \to_{\mathrm{R}_3} b$$

In the term $f(ga)$, we have $(0, \mathrm{R}_1) \sharp (10, \mathrm{R}_2)$ and $(10, \mathrm{R}_2) \sharp (11, \mathrm{R}_3)$ but not $(0, \mathrm{R}_1) \sharp (11, \mathrm{R}_3)$.

We denote by $u \parallel v$ that the redex occurrences u and v are not overlapping. The relation $\parallel$ is reflexive. It is extended in the obvious way to denote that a redex occurrence is not overlapping with a set of redex occurrences.

Finally, the notion of full extendedness will be needed. The definition is given in [HP96, Oos96a].

Definition 3. A rewrite rule $x_1 \ldots . x_m.s \to x_1 \ldots . x_m.t$ is said to be *fully extended* if every occurrence of $y \in \{x_1, \ldots, x_m\}$ in s has the $\overline{\eta}$-normal form of every bound variable in which scope it occurs as argument.

A higher-order rewriting system is *fully extended* if every rewrite rule of it is.

An example of a rewrite rule that is fully extended is the rule for **beta** in the higher-order rewriting system representing lambda-calculus. The rewrite rule for **eta** in the same system is not fully extended, because in $\mathsf{abs}(x.\mathsf{app}zx)$, the variable z does not have the variable x as an argument. Note that the traditional version of the eta-reduction rule contains a side-condition concerning the bound variable.

In Section 4 it will be explained which restrictions imposed on higher-order rewriting systems are necessary for outermost-fair rewriting to be normalising, and which are mainly there to make the proof work.

The Weakly Orthogonal Projection. The weakly orthogonal projection is defined by van Oostrom in [Oos94, p.49]. It is used to prove confluence of weakly orthogonal higher-order rewriting systems by developments. Here we recall the construction of the weakly orthogonal projection which, as the terminology indicates, is defined for all weakly orthogonal higher-order rewriting system. We will use it for the smaller class consisting of all almost orthogonal and fully extended higher-order rewriting systems.

The reader is supposed to be familiar with complete developments and the projection of a rewrite sequence over a rewrite step in the orthogonal case. A complete development of a set of (non-overlapping) redex occurrences $\mathcal{U}$ is denoted by $\mathcal{U} : s \multimap t$ or by $s \overset{\mathcal{U}}{\multimap} t$. If v is a redex occurrence in s, then the set of residuals of v after performing a complete development of $\mathcal{U}$ is denoted by $\mathsf{Res}(s,\mathcal{U})(v)$. A *development rewrite sequence* is obtained by concatenating complete developments.

Let $\mathcal{H}$ be a weakly orthogonal higher-order rewriting system. Consider a finite or infinite rewrite sequence

$$\tilde{\sigma} : s_0 \overset{\tilde{u}_0}{\to} s_1 \overset{\tilde{u}_1}{\to} s_2 \overset{\tilde{u}_2}{\to} s_3 \overset{\tilde{u}_3}{\to} \dots$$

and a rewrite step

$$v : s_0 \to t_0.$$

Let $\mathcal{V}_0 = \{v\}$ and define for $m \geq 0$ the following:

$$u_m = \begin{cases} \tilde{u}_m \text{ if } \tilde{u}_m \parallel \mathcal{V}_m, \\ v_m \text{ if } \tilde{u}_m \sharp v_m \text{ for some } v_m \in \mathcal{V}_m, \end{cases}$$
$$\mathcal{V}_{m+1} = \mathsf{Res}(s_m, u_m)(\mathcal{V}_m).$$

Since the rewriting system is weakly orthogonal, we have for every $m \geq 0$ that $u_m : s_m \to s_{m+1}$ if $\tilde{u}_m : s_m \to s_{m+1}$. Let σ be the rewrite sequence

$$\sigma : s_0 \overset{u_0}{\to} s_1 \overset{u_1}{\to} s_2 \overset{u_2}{\to} s_3 \overset{u_3}{\to} \dots.$$

By construction, we have for every $m \geq 0$ that $u_m \parallel \mathcal{V}_m$. Note that $\mathcal{V}_m$ is the set of residuals of v in s_m. For every $m \geq 0$ we define

$$\mathcal{U}_m = \mathsf{Res}(s_m, \mathcal{V}_m)(u_m).$$

Let τ be the development rewrite sequence

$$\tau : t_0 \overset{\mathcal{U}_0}{\multimap} t_1 \overset{\mathcal{U}_1}{\multimap} t_2 \overset{\mathcal{U}_2}{\multimap} t_3 \overset{\mathcal{U}_3}{\multimap} \dots.$$

In a diagram, the situation is depicted as follows:

$$\begin{array}{lccccccccc}
\tilde{\sigma}: & s_0 & \xrightarrow{\tilde{u}_0} & s_1 & \xrightarrow{\tilde{u}_1} & s_2 & \xrightarrow{\tilde{u}_2} & s_3 & \xrightarrow{\tilde{u}_3} & \cdots \\
\sigma: & s_0 & \xrightarrow{u_0} & s_1 & \xrightarrow{u_1} & s_2 & \xrightarrow{u_2} & s_3 & \xrightarrow{u_3} & \cdots \\
& \mathcal{V}_0 \downarrow & & \mathcal{V}_1 \downarrow & & \mathcal{V}_2 \downarrow & & \mathcal{V}_3 \downarrow & & \\
\tau: & t_0 & \xrightarrow{\mathcal{U}_0} & t_1 & \xrightarrow{\mathcal{U}_1} & t_2 & \xrightarrow{\mathcal{U}_2} & t_3 & \xrightarrow{\mathcal{U}_3} & \cdots
\end{array}$$

We use the following terminology. The rewrite sequence σ is said to be a *simulation* of the rewrite sequence $\tilde{\sigma}$. The development rewrite sequence τ is said to be the *orthogonal projection* of the rewrite sequence σ over the rewrite step $v : s_0 \to t_0$. The development rewrite sequence τ is said to be a *weakly orthogonal projection* of the rewrite sequence $\tilde{\sigma}$ over the rewrite step $v : s_0 \to t_0$.

In almost orthogonal higher-order rewriting systems, a simulation of a rewrite sequence constructed for the weakly orthogonal projection is unique. This is the case since if we have $u_m \sharp v_m$ and $u_m \sharp v'_m$ in the notation as above, then by almost orthogonality $v_m \sharp v'_m$ or $v_m = v'_m$. Since $\mathcal{V}_m$ consists of non-overlapping redex occurrences, we have $v_m = v'_m$. In a weakly orthogonal higher-order rewriting system, a simulation of a rewrite sequence constructed for the weakly orthogonal projection is not necessarily unique.

We conclude this section with an example of the weakly orthogonal projection.

Example 1. Consider the rewriting system defined by the following rules:

$$\begin{aligned} x.fax &\to_{\mathrm{R}_1} x.a \\ x.fxa &\to_{\mathrm{R}_2} x.a \\ x.gx &\to_{\mathrm{R}_3} x.hxx \end{aligned}$$

It is almost orthogonal but not orthogonal. We construct the weakly orthogonal projection of the rewrite sequence

$$\tilde{\sigma} : g(faa) \overset{\tilde{u}_0}{\to} h(faa)(faa) \overset{\tilde{u}_1}{\to} ha(faa)$$

with $\tilde{u}_0 = (0, \mathrm{R}_3)$ and $\tilde{u}_1 = (0100, \mathrm{R}_2)$ over the rewrite step $v : g(faa) \to ga$ with $v = (100, \mathrm{R}_1)$. This yields the following:

$$\tilde{\sigma} : \quad g(faa) \xrightarrow{\tilde{u}_0} h(faa)(faa) \xrightarrow{\tilde{u}_1} ha(faa)$$

$$\begin{array}{lccccc} \sigma : & g(faa) & \xrightarrow{u_0} & h(faa)(faa) & \xrightarrow{u_1} & ha(faa) \\ & \mathcal{V}_0 \downarrow & & \mathcal{V}_1 \downarrow & & \mathcal{V}_2 \downarrow \\ \tau : & ga & \xrightarrow{\mathcal{U}_0} & haa & \xrightarrow{\mathcal{U}_1} & haa \end{array}$$

with

$$\begin{aligned} u_0 &= (0, \mathrm{R}_3) \quad & \mathcal{U}_0 &= \{(0, \mathrm{R}_3)\} \\ u_1 &= (0100, \mathrm{R}_1) & \mathcal{U}_1 &= \emptyset \end{aligned}$$

and

$$\mathcal{V}_0 = \{(100, \mathrm{R}_1)\}, \mathcal{V}_1 = \{(0100, \mathrm{R}_1), (100, \mathrm{R}_1)\}, \mathcal{V}_2 = \{(100, \mathrm{R}_1)\}.$$

3 Outermost-Fair Rewriting

A rewrite sequence is said to be outermost-fair, or in the terminology of [O'D77], eventually outermost, if every outermost redex occurrence is eventually eliminated. So a rewrite sequence is outermost-fair either if it ends in a normal form or if it is impossible to trace infinitely long an outermost redex occurrence. In order to formalise the notion of an outermost-fair rewrite sequence, the definition of an infinite outermost chain is given. First the definition of an outermost redex occurrence is presented.

Definition 4. A redex occurrence $(\phi, l \to r)$ in a term s is said to be *outermost* if for every redex occurrence $(\phi', l' \to r')$ in s we have the following: if $(\phi', l' \to r') \preceq (\phi, l \to r)$ then $\phi' = \phi$.

The term $g(faa)$ in the rewriting system of Example 1 contains one outermost redex occurrence, namely $(0, \mathrm{R}_3)$. The term faa in the same rewriting system contains two outermost redex occurrences, namely $(00, \mathrm{R}_1)$ and $(00, \mathrm{R}_2)$. This shows that, in an almost orthogonal higher-order rewriting system, outermost redex occurrences may be overlapping.

Definition 5. Let $\sigma : s_0 \xrightarrow{u_0} s_1 \xrightarrow{u_1} s_2 \xrightarrow{u_2} \ldots$ be an infinite rewrite sequence. An *infinite outermost chain in* σ is an infinite sequence of redex occurrences $w_m, w_{m+1}, \ldots$ such that

1. w_p is an outermost redex occurrence in s_p for every $p \geq m$,
2. w_p is a residual of w_{p-1} for every $p > m$.

Note that in the previous definition $w_p \parallel u_p$ for every $p \geq m$. Now the key definition of this paper can be given.

Definition 6. A rewrite sequence is said to be *outermost-fair* either if it ends in a normal form or if it is infinite and does not contain an infinite outermost chain.

The definition of an outermost-fair rewrite sequence is illustrated by the following example.

Example 2. Consider the higher-order rewriting system defined by the following rewrite rules:

$$x.fcx \to_{\mathrm{R}_1} x.fbx$$
$$b \to_{\mathrm{R}_2} c$$
$$a \to_{\mathrm{R}_3} a$$

1. The rewrite sequence

$$\sigma : fca \xrightarrow{u_0} fba \xrightarrow{u_1} fca \xrightarrow{u_2} \ldots$$

with $u_{2m} = (00, \mathrm{R}_1)$ and $u_{2m+1} = (01, \mathrm{R}_2)$ for every $m \geq 0$ is outermost-fair. Note that there is an infinite sequence of residuals starting in the first term of σ, namely $(1, \mathrm{R}_3), (1, \mathrm{R}_3), (1, \mathrm{R}_3), \ldots$. This infinite sequence of residuals is however not an infinite outermost chain, although infinitely many residuals in it are outermost.

2. The rewrite sequence

$$\sigma : faa \xrightarrow{u_0} faa \xrightarrow{u_1} faa \xrightarrow{u_2} \ldots$$

with $u_m = (01, \mathrm{R}_3)$ for every $m \geq 0$ is not outermost-fair since we have an infinite outermost chain, namely $(1, \mathrm{R}_3), (1, \mathrm{R}_3), (1, \mathrm{R}_3), \ldots$.

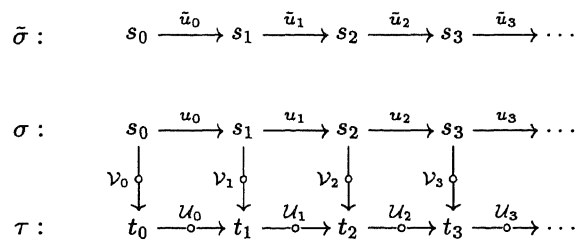
4 Outermost-Fair Rewriting is Normalising

In this section we present the proof of the main result of this paper, namely that outermost-fair rewriting is normalising for almost orthogonal and fully extended higher-order rewriting systems.

The Structure of the Proof. The structure of the proof is as follows. Suppose that the finite or infinite rewrite sequence

$$\tilde{\sigma} : s_0 \xrightarrow{\tilde{u}_0} s_1 \xrightarrow{\tilde{u}_1} s_2 \xrightarrow{\tilde{u}_2} s_3 \xrightarrow{\tilde{u}_3} \ldots$$

is outermost-fair. Let $v : s_0 \to t_0$ be a rewrite step and construct the weakly orthogonal projection of $\tilde{\sigma}$ over v as explained in Section 2. In a diagram:



The proof consists of the following steps:

1. If $\tilde{\sigma}$ is outermost-fair, then σ is outermost-fair (Proposition 7).
2. If σ is outermost-fair, then τ is outermost-fair (Proposition 8).
3. If σ is outermost-fair and τ ends in a normal form t, then σ ends in t (Proposition 9).
4. If a term s has a normal form t then every outermost-fair rewrite sequence starting in s eventually ends in t (Theorem 10).

The notation in the proof will be such that the diagram above applies.

Some Observations. Three restrictions are imposed on the higher-order rewriting systems that we consider: rewrite rules must be left-linear, all critical pairs are trivial and overlap occurs only at the root of redex occurrences, and finally rewrite rules must be fully extended. Before we embark on the proof, we first analyse the rôle of these restrictions.

The restriction to left-linear systems is necessary, since outermost-fair rewriting may not be normalising in a higher-order rewriting system that is not left-linear. This is illustrated by the following example:

$$\begin{aligned} x.fxx &\to_{\mathtt{R}_1} x.b \\ x.gx &\to_{\mathtt{R}_2} x.gx \\ a &\to_{\mathtt{R}_3} b \end{aligned}$$

The rewrite sequence $f(ga)(gb) \to f(ga)(gb) \to \ldots$ in which alternately the outermost redex occurrence $(010, \mathtt{R}_2)$ and the outermost redex occurrence $(10, \mathtt{R}_2)$ are contracted, is outermost-fair but does not end in the normal form b, although we have $f(ga)(gb) \to f(gb)(gb) \to b$.

The restriction to fully extended systems is also necessary. This was pointed out to me by Vincent van Oostrom [Oos96b], who gave the following example. Consider the higher-order rewriting system defined by the following rewrite rules:

$$\begin{aligned} z.f(x.z) &\to_{\mathtt{R}_1} z.a \\ z.gz &\to_{\mathtt{R}_2} z.a \\ z.hz &\to_{\mathtt{R}_3} z.hz \end{aligned}$$

It is not fully extended because of the rewrite rule $\mathtt{R}_1$: the variable z in $f(x.z)$ doesn't have the variable x as an argument. The rewrite sequence $f(x.h(gx)) \to f(x.h(gx)) \to \ldots$ in which in every step the outermost redex occurrence $(100, \mathtt{R}_3)$ is contracted, is outermost-fair but does not end in the normal form a, although we have $f(x.h(gx)) \to f(x.ha) \to a$. Note that $(0, \mathtt{R}_1)$ is *not* a redex occurrence in $f(x.h(gx))$ because of the occurrence of x at position 1011.

In both cases the problem is that an outermost-redex occurrence can be created by contracting a redex occurrence that is not outermost. In a higher-order rewriting system that is almost orthogonal and fully extended, an outermost redex occurrence can only be created by contracting a redex occurrence that is outermost itself.

Another important point is the elimination of outermost redex occurrences. In a higher-order rewriting system that is almost orthogonal and fully extended, an outermost redex occurrence can only be eliminated by contracting a redex occurrence that is also outermost itself. To be more precise, an outermost redex occurrence w can be eliminated in one of the following three ways:

1. by contracting w,
2. by contracting a redex occurrence that is overlapping with w,
3. by contracting a redex occurrence that creates a new outermost redex occurrence such that the residual of w is not outermost anymore.

It is quite easy to see that if a system is not left-linear, it can happen that an outermost redex occurrence is eliminated by contracting a redex occurrence that is not outermost.

Finally we discuss the restriction to higher-order rewriting systems that are weakly head-ambiguous. In a system that is weakly orthogonal but not almost orthogonal, it can happen that an outermost redex occurrence is created by contracting a redex occurrence that is not outermost, and it can also happen that an outermost redex occurrence is eliminated by contracting a redex occurrence that is not outermost. In both cases, the redex occurrence creating or eliminating an outermost redex occurrence is not necessarily outermost itself, but it is overlapping with an outermost redex occurrence. Consider for instance the higher-order rewriting system defined by the following rewrite rules:

$$\begin{aligned} fa &\to_{\mathrm{R}_1} fb \\ a &\to_{\mathrm{R}_2} b \\ b &\to_{\mathrm{R}_3} c \end{aligned}$$

In the rewrite step $(1, \mathrm{R}_2) : fa \to fb$ the outermost redex occurrence $(0, \mathrm{R}_1)$ is eliminated although the redex occurrence $(1, \mathrm{R}_2)$ is not outermost itself. Moreover, in the same rewrite step the outermost redex occurrence $(1, \mathrm{R}_3)$ is created.

We conclude that the restrictions of left-linearity and fully extendedness are necessary for the result to hold; the restriction to critical pairs that are trivial and that have overlap only at the root of the redex occurrences is mainly there to make the proof work. It is imaginable that a proof can be given for the larger class of weakly orthogonal and fully extended higher-order rewriting systems. In fact, in an earlier version we erroneously claimed to prove normalisation of outermost-fair rewriting for weakly orthogonal systems. The restriction on critical pairs can probably not be relaxed much more: for left-linear higher-order rewriting systems that are *parallel-closed* as defined by Huet in Section 3.3 of [Hue80], outermost-fair rewriting is not normalising. This is illustrated by the higher-order rewriting system defined by the following rewrite rules:

$$\begin{aligned} a &\to_{\mathrm{R}_1} b \\ x.hx &\to_{\mathrm{R}_2} x.hx \\ g(hb) &\to_{\mathrm{R}_3} b \end{aligned}$$

It is easy to see that it is left-linear and parallel-closed. The rewrite sequence $g(ha) \to g(ha) \to g(ha) \to \ldots$ in which in each rewrite step the redex occurrence $(10, \mathrm{R}_2)$ is contracted, is outermost-fair but does not end in a normal form, although we have $g(ha) \to g(hb) \to b$.

The Proof. In the following, we consider an almost orthogonal and fully extended higher-order rewriting system $\mathcal{H}$.

Proposition 7. *Let $\tilde{\sigma}$ be an outermost-fair rewrite sequence issuing from s_0 and let $v : s_0 \to t_0$ be a rewrite step. Let σ be the simulation of $\tilde{\sigma}$ constructed for the weakly orthogonal projection of $\tilde{\sigma}$ over $v : s_0 \to t_0$. Then σ is outermost-fair.*

Proof. Let $\tilde{\sigma} : s_0 \xrightarrow{\tilde{u}_1} s_1 \xrightarrow{\tilde{u}_2} s_2 \xrightarrow{\tilde{u}_3} \ldots$ be an outermost-fair rewrite sequence and let σ be the simulation of $\tilde{\sigma}$ for the weakly orthogonal projection of $\tilde{\sigma}$ over some rewrite step $v : s_0 \to t_0$.

If $\tilde{\sigma}$ ends in a normal form, then σ also ends in a normal form, so we restrict attention to the case that $\tilde{\sigma}$ is infinite and does not contain an infinite outermost chain. It is sufficient to show the following: if an outermost redex occurrence w_m in s_m is eliminated in a rewrite step $\tilde{u}_m : s_m \to s_{m+1}$, then w_m is also eliminated in the rewrite step $u_m : s_m \to s_{m+1}$. There are two possibilities: either $u_m = \tilde{u}_m$ or $u_m \sharp \tilde{u}_m$. In the first case, the outermost redex occurrence w_m is clearly eliminated in the rewrite step $u_m : s_m \to s_{m+1}$. For the case that $u_m \sharp \tilde{u}_m$, we consider the different possibilities in which the redex occurrences w_m is eliminated in the rewrite step $\tilde{u}_m : s_m \to s_{m+1}$.

1. If $\tilde{u}_m = w_m$, then we have $u_m \sharp w_m$, and hence the outermost redex occurrence w_m is eliminated in the rewrite step $u_m : s_m \to s_{m+1}$.
2. If $\tilde{u}_m \sharp w_m$, then we have either $u_m = w_m$ or $u_m \sharp w_m$, and in both cases the outermost redex occurrence w_m is eliminated in the rewrite step $u_m : s_m \to s_{m+1}$.
3. Finally, the outermost redex occurrence w_m can be eliminated in the rewrite step $\tilde{u}_m : s_m \to s_{m+1}$ because an outermost redex occurrence is created above the residual of w_m. The same outermost redex occurrence is created by contracting u_m, so also in that case the outermost redex occurrence w_m is eliminated in the rewrite step $u_m : s_m \to s_{m+1}$. □

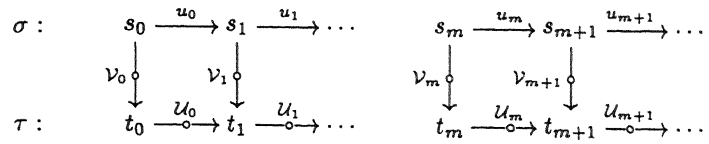
The proof of the following proposition requires some auxiliary definitions and results and is for lack of space omitted. The interested reader is referred to Proposition 6.2.11 in [Raa96]. Some of the ideas of the proof are also present in the proof of Proposition 9.

Proposition 8. *Let $\sigma : s_0 \xrightarrow{u_0} s_1 \xrightarrow{u_1} s_2 \xrightarrow{u_2} \ldots$ be an infinite rewrite sequence. Let $v : s_0 \to t_0$ be a rewrite step. Let $\tau : t_0 \xrightarrow{\mathcal{U}_0\!\!\circ} t_1 \xrightarrow{\mathcal{U}_1\!\!\circ} t_2 \xrightarrow{\mathcal{U}_2\!\!\circ} \ldots$ be the orthogonal projection of the rewrite sequence σ over the rewrite step $v : s_0 \to t_0$. If τ contains an infinite outermost chain, then σ contains an infinite outermost chain.*

A direct consequence of Proposition 8 is that if a rewrite sequence is outermost-fair, then its orthogonal projection over some rewrite step is also outermost-fair.

Proposition 9. *Let $\sigma : s_0 \xrightarrow{u_0} s_1 \xrightarrow{u_1} s_2 \xrightarrow{u_2} \ldots$ be an outermost-fair rewrite sequence. Let $\tau : t_0 \xrightarrow{\mathcal{U}_0\!\!\circ} t_1 \xrightarrow{\mathcal{U}_1\!\!\circ} t_2 \xrightarrow{\mathcal{U}_2\!\!\circ} \ldots$ be the orthogonal projection of σ over the rewrite step $v : s_0 \to t_0$. If τ ends in a normal form t then σ ends also in t.*

Proof. Let $\sigma : s_0 \xrightarrow{u_0} s_1 \xrightarrow{u_1} \ldots$ be an outermost-fair rewrite sequence. Let $\tau : t_0 \xrightarrow{\mathcal{U}_0\!\!\circ} t_1 \xrightarrow{\mathcal{U}_1\!\!\circ} \ldots$ be the orthogonal projection of σ over the rewrite step $v : s_0 \to t_0$. Suppose that τ ends in a normal form $t = t_m$. In a picture:



We have $\emptyset = \mathcal{U}_m = \mathcal{U}_{m+1} = \ldots$, so $t = t_m = t_{m+1} = \ldots$. Now we define the following:

1. $t_m^0 = s_m$,
2. $\mathcal{W}_m^0 = \mathcal{V}_m$,
3. $\mathcal{V}_m^0$ is the set of all outermost redex occurrences of $\mathcal{W}_m^0$ in t_m^0.

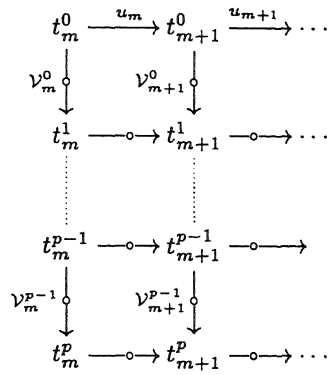
For $i > 0$ we define:

1. t_m^{i+1} is the term obtained by performing a complete development of $\mathcal{V}_m^i$ in t_m^i,
2. $\mathcal{W}_m^{i+1} = \mathsf{Res}(t_m^i, \mathcal{V}_m^i)(\mathcal{W}_m^i)$,
3. $\mathcal{V}_m^{i+1}$ is the set of all outermost redex occurrences of $\mathcal{W}_m^{i+1}$ in t_m^{i+1}.

Let moreover p be the smallest natural number such that $\mathcal{V}_m^p = \emptyset$. We consider the following stepwise development of $\mathcal{V}_m$:

$$s_m = t_m^0 \xrightarrow{\mathcal{V}_m^0} t_m^1 \xrightarrow{\mathcal{V}_m^1} \ldots \xrightarrow{\mathcal{V}_m^{p-1}} t_m^p = t_m.$$

Note that indeed $t_m^p = t_m$ since t_m is a normal form. Projecting σ over this rewrite sequence yields the following:



We have that $\mathcal{V}_j^i$ is a set of outermost redex occurrences in t_j^i for every $i \in \{0, \ldots, p-1\}$ and $j \geq m$.

Now we show that there exists an n such that $\mathcal{V}_n^0 = \ldots = \mathcal{V}_n^{p-1} = \emptyset$. That is, eventually the rewrite sequences σ and τ coincide and σ also ends in t.

Let $j \geq m$ arbitrary. We define $f(j)$ to be the smallest number such that $\mathcal{V}_j^{f(j)} \neq \emptyset$. Then $\mathcal{V}_j^{f(j)}$ consists of outermost redex occurrences in s_j $(= t_j^0)$. Since the rewrite sequence σ is outermost-fair, a redex occurrence w_j in $\mathcal{V}_j^{f(j)}$ will eventually be eliminated. This elimination can happen in one of the following two ways:

1. because w_j is contracted,
2. because a redex occurrence overlapping with w_j is contracted.

It cannot happen that w_j is eliminated because a redex occurrence is contracted that creates a redex occurrence above the residual of w_j because otherwise t_j^p would not be a normal form, which is a contradiction since $t_j^p = t_j = t$. Hence there exists a j' such that $\mathcal{V}_{j'}^{f(j)} = \emptyset$.

This shows that eventually σ coincides with τ and hence σ ends in the normal form t. □

Finally we present the proof of the main result.

Theorem 10. *Let s_0 be a weakly normalising term. Every outermost-fair rewrite sequence starting in s_0 eventually ends in a normal form.*

Proof. Let s_0 be a weakly normalising term and consider an outermost-fair rewrite sequence $\tilde{\sigma} : s_0 \xrightarrow{\tilde{u}_0} s_1 \xrightarrow{\tilde{u}_1} s_2 \xrightarrow{\tilde{u}_2} \dots$ starting in s_0. Let t be the normal form of s_0. We fix a rewrite sequence $\rho : s_0 \twoheadrightarrow t$ from s_0 to its normal form consisting of m rewrite steps. Now we prove by induction on m that $\tilde{\sigma}$ eventually ends in the normal form t.

If $m = 0$, then $s_0 = t$ and hence the statement trivially holds.

If $m > 0$, then the rewrite sequence ρ is of the form $s_0 \xrightarrow{v} t_0 \twoheadrightarrow t$. We construct the weakly orthogonal projection of the rewrite sequence $\tilde{\sigma}$ over the rewrite step $v : s_0 \to t_0$. Let σ be the simulation of $\tilde{\sigma}$ and let τ be the orthogonal projection of σ over v. By hypothesis $\tilde{\sigma}$ is outermost-fair. Hence by Proposition 7 the rewrite sequence σ is outermost-fair. By Proposition 8 this yields that τ is outermost-fair. By the induction hypothesis, τ ends in the normal form t. Applying now Proposition 9 yields that the rewrite sequence σ also ends in the normal form t. Since σ is the simulation of $\tilde{\sigma}$, we have that $\tilde{\sigma}$ also ends in the normal form t. This completes the proof. □

An immediate consequence of the main result of this paper is that the parallel-outermost strategy, which selects all outermost redex occurrences to be contracted simultaneously, is normalising for higher-order rewriting systems that are orthogonal and fully extended.

Acknowledgements. I thank Vincent van Oostrom for corrections and inspiring discussions. I further wish to thank Pierre-Louis Curien, Zurab Khasidashvili, Jan Willem Klop, Aart Middeldorp and the anonymous referees for comments and suggestions. The diagrams are made using the package Xy-pic of Kristoffer H. Rose.

References

[Aka93] Yohji Akama. On Mints' reduction for ccc-calculus. In M. Bezem and J.F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications (TLCA '93)*, pages 1–12, Utrecht, The Netherlands, March 1993. Volume 664 of Lecture Notes in Computer Science.

[Bar92] H.P. Barendregt. Lambda calculi with types. In S. Abramsky, Dov M. Gabbay, and T.S.E Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–310. Oxford University Press, New York, 1992.

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

[CFC58] Haskell B. Curry, Robert Feys, and William Craig. *Combinatory Logic, volume I.* Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, Amsterdam, 1958.

[HP96] Michael Hanus and Christian Prehofer. Higher-order narrowing with definitional trees. In Harald Ganzinger, editor, *Proceedings of the 7th International Conference on Rewriting Techniques and Applications (RTA '96)*, volume 1103 of *Lecture Notes in Computer Science*, pages 138–152, New Brunswick, USA, 1996.

[Hue80] Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the Association for Computing Machinery*, 27(4):797–821, October 1980.

[Mil91] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. *Journal of Logic and Computation*, 1(4):497–536, 1991.

[MN94] Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. Technical Report TUM-I9433, Technische Universität München, August 1994.

[Nip91] Tobias Nipkow. Higher-order critical pairs. In *Proceedings of the sixth annual IEEE Symposium on Logic in Computer Science (LICS '91)*, pages 342–349, Amsterdam, The Netherlands, July 1991.

[O'D77] Michael J. O'Donnell. *Computing in Systems Described by Equations*, volume 58 of *Lecture Notes in Computer Science*. Springer Verlag, 1977.

[Oos94] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting.* PhD thesis, Vrije Universiteit, Amsterdam, March 1994. Available at `http://www.cs.vu.nl/~oostrom`.

[Oos96a] Vincent van Oostrom. Higher-order families. In Harald Ganzinger, editor, *Proceedings of the 7th International Conference on Rewriting Techniques and Applications (RTA '96)*, volume 1103 of *Lecture Notes in Computer Science*, pages 392–407, New Brunswick, USA, 1996.

[Oos96b] Vincent van Oostrom. Personal communication, 1996.

[OR94] Vincent van Oostrom and Femke van Raamsdonk. Weak orthogonality implies confluence: the higher-order case. In A. Nerode and Yu.V. Matiyasevich, editors, *Proceedings of the Third International Symposium on Logical Foundations of Computer Science (LFCS '94)*, volume 813 of *Lecture Notes in Computer Science*, pages 379–392, St. Petersburg, July 1994.

[Raa96] Femke van Raamsdonk. *Confluence and Normalisation for Higher-Order Rewriting.* PhD thesis, Vrije Universiteit, Amsterdam, May 1996. Available at `http://www.cwi.nl/~femke`.