1992

F. van Raamsdonk

A simple proof of confluence for weakly orthogonal combinatory reduction systems

Computer Science/Department of Software Technology Report CS-R9234 August

CWI is het Centrum voor Wiskunde en Informatica van de Stichting Mathematisch Centrum
CWI is the Centre for Mathematics and Computer Science of the Mathematical Centre Foundation

CWI is the research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a non-profit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands organization for scientific research (NWO).

A Simple Proof of Confluence for Weakly Orthogonal Combinatory Reduction Systems

Femke van Raamsdonk
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands
email: `femke@cwi.nl`

Abstract

Combinatory reduction systems (CRSs for short) are term rewriting systems in which binding structures for variables are present. Metavariables ranging over the set of terms are equipped with a fixed arity. As a consequence of these extensions, various substitution mechanisms can be expressed: besides first-order term rewriting systems also λ-calculus, extensions of λ-calculus, typed λ-calculi and proof normalizations fit in the framework of CRSs. Confluence of orthogonal CRSs has been proved by Klop. In this paper we present a new, much shorter, proof of confluence for weakly orthogonal CRSs. The proof of confluence gives rise to an extended notion of development, called 'superdevelopment'. We prove superdevelopments to be finite for both orthogonal term rewriting systems and λ-calculus, thus generalizing the well-known Finite Developments Theorem.

1991 Mathematics Subject Classification: 68Q50
1991 CR Categories: F.4.1.1, F.3.3.3.
Key Words and Phrases: term rewriting systems, λ-calculus, combinatory reduction systems, orthogonality, confluence, finite developments.

Note: This research was performed in the framework of NWO/SION project 612-316-045 *Extensions of orthogonal rewrite systems - syntactic properties.*

1. Introduction

1.1. Origins

In the thirties, λ-calculus was developed by Church, Kleene and Rosser. The original aims of the study of λ-calculus (a standard reference is [2]) were to provide a foundation for logic and mathematics, and moreover to develop a general theory of functions. Whereas the first aspect didn't work out fully satisfactory, because of the appearance of paradoxes such as Curry's paradox, on the other hand the latter has turned out to be very fruitful. Being λ-definable has been shown to be equivalent to other notions of effective computability, like for instance Turing computability. Somewhat before the development of λ-calculus a theory related to it, called Combinatory Logic (CL), was developed by Schönfinkel [20] and rediscovered by Curry [6]. Curry's aim in studying CL was mainly to provide an alternative foundation for mathematics, by analysing substitution and the use of variables. The close connection between λ-calculus and CL was demonstrated by Rosser, by means of a canonical translation of λ-terms into CL-terms and vice versa.

Report CS-R9234
ISSN 0169-118X
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Apart from investigating fundamental notions like computability, λ-calculus is of special interest for theoretical computer science as it can be considered as the prototype of a functional programming language. Nowadays both λ-calculus and CL play a prominent role in the design and implementation of functional programming languages.
The study of λ-calculus and the foundations of functional programming has led to a great variety of classes of reduction or rewriting systems, an important one being first-order term rewriting, which has already been applied successfully in many areas of computer science. Besides term rewriting [15, 9], a non-exhaustive list contains typed λ-calculi [3, 6], λ-calculus with δ-rules [2, 17], $\lambda(a)$-reductions [8], recursive program schemes [4] and proof normalizations [7, 19].
The first attempt to provide a uniform framework for a number of extensions of λ-calculus was given by Hindley [8] in formulating $\lambda(a)$-reductions. Aczel [1] has given in his contraction schemes a much more extensive format of rewriting, in which also variable binding mechanisms other than the usual one of λ-calculus can be expressed. However, Aczel's format does not cover general first-order term rewriting. Klop [14] developed a very encompassing formalism of rewriting under the name combinatory reduction systems (CRSs), which is a combination of Aczel's contraction schemes and first-order term rewriting. This format also covers normalization of proofs in natural deduction [7], in which variable binding mechanisms occur that differ essentially from the one in λ-calculus.
The use of CRSs is that for all systems mentioned above a uniform proof of some syntactical property can be given by proving this property for (a certain class of) CRSs. In this way one avoids the necessity of having several proofs that are basically the same.

1.2. Goal of the Present Paper

An important syntactical property is confluence, meaning that for every two coinitial reduction sequences $s \twoheadrightarrow t$ and $s \twoheadrightarrow u$ a common reduct of t and u can be found. The main corollary of confluence is uniqueness of normal forms, i.e. every term has at most one normal form. Church and Rosser showed the consistency of λ-calculus by showing that every two convertible temrs have a common reduct. This result, known as the Church-Rosser Theorem, is an immediate consequence of confluence.
As it turns out, like in the case of λ-calculus, in order to prove confluence for orthogonal CRSs two strategies can be followed. One is based on labelling reductions and tracing what happens with reductions in a reduction diagram, as it originates from an attempt to prove confluence by tiling with elementary reduction diagrams. In this way Klop [14] has proved confluence for orthogonal combinatory reduction systems. Although this method yields useful information on the manner in which reductions proceed, its drawback is that it is rather tedious and results in an extremely long and complicated confluence proof.
Tait and Martin-Löf have proved confluence for λ-calculus in another way, namely by analysing the reduction relation in terms of a simpler relation denoted by $\mapsto_1$. This proof strategy has been used as well by Aczel [1] for proving confluence of his contraction schemes, that form a restricted class of CRSs. In the present paper we present a proof of confluence for the class of weakly orthogonal CRSs following this proof strategy. The confluence proof turns out to be elegant and manageable and might enhance the accessibility to the framework of CRSs. All complications in the present proof are a consequence of admitting weakly overlapping redexes; these parts may be skipped by readers only interested in orthogonal CRSs.
In this proof of confluence a relation on terms is used that yields as a side-result a concept which may be of some interest on its own: superdevelopments.
The relation $\mapsto_1$ of the confluence proof of Tait and Martin-Löf corresponds exactly to a well-

known concept in λ-calculus, namely that of 'development'. A development is a reduction sequence in which only descendants of redexes that were already present in the initial term may be contracted, whereas contraction of redexes that were created along the way is not allowed. A classical result in the theory of λ-calculus is the Finite Developments Theorem, stating that all developments must terminate eventually.

In the confluence proof of Aczel the reduction relation is analysed by means of a relation on terms, denoted by $\geq$, that is a generalization of the one used by Tait and Martin-Löf. The reduction sequences corresponding exactly to this relation are less restrictive than developments and will be called 'superdevelopments'.

It will be proved for orthogonal TRSs and λ-calculus that all superdevelopments are finite. The same theorem holds for the case of CRSs, but that is not worked out in this paper.

1.3. Related Work

Work on confluence criteria for higher-order rewriting has been done by Nipkow [18]. Jouannaud and Okada [10] have a general modularity result for polymorphically typed λ-calculus extended by first-order and higher-order rewrite rules. Kennaway [12] has proved that for a large class of reduction systems, also containing all weakly orthogonal CRSs (although his definition of weakly orthogonal is slightly more restrictive), a recursive normalizing one-step evaluation strategy exist. Kahrs [11] investigates an extension of λ-calculus with general first-order rewriting, and proves several syntactic properties. This class of reduction systems is contained in the one of CRSs. Another new confluence proof for a class of rewriting systems with bound variables is given by Khasidashvili [13]. This proof proceeds along the lines of the proof given by Klop in [14]. Breazu-Tannen and Gallier prove in [5] that adding simply typed λ-calculus to a set of confluent first-order rewrite rules preserves confluence.

1.4. Contents of Present Paper

The remainder of this paper is organized as follows. In section 2 a concise overview of first-order term rewriting is given. Then we focus attention on combinatory reduction systems for which the main definitions are given and illustrated by examples. We use functional notation, like in [12], instead of the applicative one originally used by Klop in [14]. Section 3 is devoted to a proof of confluence for weakly orthogonal CRSs. Like in the proof of confluence for λ-calculus by Tait and Martin-Löf, we define a relation, denoted as $\geq$, on terms such that the transitive closure of $\geq$ equals reduction. Then we prove that $\geq$ satisfies the diamond property, which yields confluence for weakly orthogonal CRSs. The relation $\geq$ gives rise to a special kind of reduction sequences that form a generalization of developments, and that we shall call 'superdevelopments'. In section 4 we define superdevelopments for orthogonal TRSs and λ-calculus and prove them all to be finite.

2. Preliminaries

2.1. Term Rewriting Systems

A term rewriting system (TRS) is given by a non-empty set of *function symbols* $\mathcal{F}$ and a set of *reduction rules*. To each function symbol a fixed arity is associated, denoting the number of arguments it should have. Function symbols of arity 0 are called *constant symbols*. Next to function symbols we shall need a countably infinite set of *metavariables*, written as $Z, Z_0, Z_1, \ldots$ ranging over the set of terms. The set of metavariables *Mvar* and $\mathcal{F}$ are supposed to be disjoint.

The set $Ter(\mathcal{F}, Mvar)$ of terms built from function symbols and metavariables is defined as the smallest set such that

- each metavariable $Z \in Mvar$ is a term,
- if $t_1, \ldots, t_n$ are terms and $F \in \mathcal{F}$ is a function symbol of arity n, then $F(t_1, \ldots, t_n)$ is a term.

If C is a constant symbol we write the term $C()$ as C. Identity on terms is denoted by $\equiv$. A *context* is a 'term' with one or more occurrences of a special symbol $\Box$, denoting an empty place. A context with exactly one occurrence of $\Box$ is written as $C[\ ]$; one with n occurrences of $\Box$ as $C[, \ldots ,]$. For a context $C[, \ldots ,]$ with n occurrences of $\Box$, $C[t_1, \ldots, t_n]$ is the result of replacing from left to right the occurrences of $\Box$ by $t_1, \ldots, t_n$. A term s is a subterm of t if a context $C[\ ]$ exist such that $t \equiv C[s]$. A reduction rule is a pair (l, r) of terms of $Ter(\mathcal{F}, Mvar)$, usually written as $l \rightarrow r$, satisfying the following two conditions:

- the left-hand side l is not a metavariable,
- metavariables occurring in the right-hand side r occur already in the left-hand side l.

A *valuation* is a map $\sigma : Mvar \rightarrow Ter(\mathcal{F}, Mvar)$ extended to a homomorphism $Ter(\mathcal{F}, Mvar) \rightarrow Ter(\mathcal{F}, Mvar)$ by defining $\sigma(F(t_1, \ldots, t_n)) \equiv F(\sigma(t_1), \ldots, \sigma(t_n))$. The map σ is subject to the restriction that $\sigma(Z) \not\equiv Z$ only for finitely many metavariables Z. A term $\sigma(t)$ is called an *instance* of t. An instance of the left-hand side of some reduction rule is called a *redex* (short for reducible expression).

The reduction rules of a TRS R induce a reduction relation $\rightarrow_R$ (written as $\rightarrow$ if it is clear which term rewriting system is meant) on $Ter(\mathcal{F}, Mvar)$ in the following way: $s \rightarrow_R t$ if there exists a reduction rule $l \rightarrow_R r$, a valuation σ and a context $C[\ ]$ such that $s \equiv C[\sigma(l)]$ and $t \equiv C[\sigma(r)]$. Then s is said to *reduce* to t by *contracting* the redex $\sigma(l)$. The subterm $\sigma(r)$ is called the *contractum* of the reduction step, and the term $t \equiv C[\sigma(r)]$ is called the *reduct*. The reflexive-transitive closure of $\rightarrow$ is written as $\twoheadrightarrow$.

2.2. Combinatory Reduction Systems

In a combinatory reduction system (CRS for short), term formation is more complex than in the case of TRSs, where terms only can be built from metavariables and function symbols. We distinguish two kinds of 'variables': ordinary variables used to build up terms, and metavariables ranging over the set of terms that are used in the reduction rules. An essential feature of CRSs is the presence of a binding mechanism for variables. A term t of which some variable x has been abstracted will be written as $[x]t$, with [..].. the abstraction operator; x is then called a bound variable. Metavariables have a fixed arity, denoting the number of arguments they must have. For an n-ary metavariable Z, $Z(x_1, \ldots, x_n)$ represents an arbitrary term t possibly but not necessarily containing the variables $x_1, \ldots, x_n$. Then $Z(s_1, \ldots, s_n)$ corresponds to this term t in which $x_1, \ldots, x_n$ have been replaced by $s_1, \ldots, s_n$, respectively.

By extending TRSs in this way, a framework is obtained in which various substitution mechanisms can be expressed. For instance, the rule

$$F([x]Z(x)) \rightarrow Z(I)$$

expresses that in an argument of F starting with an abstraction the variable bound by this abstraction may be replaced by I.

In λ-calculus, we have variables and a binding mechanism, λ-abstraction, for them, but metavariables are treated on an informal level. In the rule for β-reduction, usually written as

$$(\lambda x.M)N \rightarrow_\beta M[x := N] \quad \text{for } \lambda\text{-terms } M \text{ and } N$$

M and N stand for arbitrary terms and are in fact metavariables, although they are not present in the language as such. In rules of TRSs only nullary metavariables (unfortunately usually called 'variables') are present, and no binding structures.

We now proceed with the formal definition of CRSs.
The *alphabet* of a CRS consists of
(1) a countably infinite set *Var* of variables, written as $x, y, z, \ldots$,
(2) a countably infinite set *Mvar* of metavariables, written as $Z, Z_0, Z_1, \ldots$, where each metavariable has a fixed arity denoting the number of arguments it is supposed to have,
(3) a non-empty set $\mathcal{F}$ of function symbols, each with a fixed arity,
(4) improper symbols (,), [,].
As is customary, function symbols with arity 0 are called *constant symbols*. All sets of symbols are supposed to be mutually disjoint. Metaterms, only used in reduction rules, are distinguished from terms. Both are built from the alphabet given above.

DEFINITION 2.1. The set $Mter(\mathcal{F}, Mvar, Var)$ of *metaterms*, henceforth simply referred to by $Mter$, is the smallest set such that
(1) each variable $x \in Var$ is a metaterm,
(2) if t is a metaterm and x a variable, then $[x]t$ is a metaterm,
(3) if F is a function symbol with arity n and $t_1, \ldots, t_n$ are metaterms, then $F(t_1, \ldots, t_n)$ is a metaterm,
(4) if Z is a metavariable with arity n and $t_1, \ldots, t_n$ are metaterms, then $Z(t_1, \ldots, t_n)$ is a metaterm.
The set $Ter(\mathcal{F}, Mvar, Var)$ of terms, henceforth written as Ter, consists of all metaterms not containing any metavariable.

In the (meta)term $[x]t$, all occurrences of x in t are bound by $[x]$. Variables that are not bound are called *free* variables. A (meta)term in which all variable occurrences are bound is called a *closed* (meta)term. Next to this usual terminology some usual conventions are adopted. Instead of $[x_1](\ldots([x_n]t)\ldots)$ we write $[x_1 \ldots x_n]t$. (Meta)terms that are identical up to a renaming of bound variables are identified. This permits to adopt the convention that for each abstraction another variable is used.
In the (meta)term $[x]t$, although it is well-formed, the meaning of the abstraction $[x]$ is not expressed. The only information we have is the fact that an abstraction has taken place. The actual (operational) meaning of this binding, however, will be expressed by the function symbol taking $[x]t$ as an argument, together with the reduction rules 'defining' that function symbol.
Another possibility for representing metaterms and terms is using instead of the functional format chosen here an applicative one. In that case, all function symbols are supposed to be nullary and next to them a special binary operator 'application' is considered. Although the applicative style yields more subterms, both set-ups are entirely equivalent and it is only a matter of taste which format is preferred.
A reduction relation on the terms of a CRS is generated by instantiated versions of *reduction rules.*

DEFINITION 2.2. A *reduction rule* is a pair (α, β), written as $\alpha \to \beta$, satisfying the following constraints:
- α and β are closed metaterms,
- α is of the form $F(\alpha_1, \ldots, \alpha_n)$, with $\alpha_1, \ldots, \alpha_n$ metaterms,
- metavariables occurring in β occur also in α,

- metavariables in α occur only in the form $Z(x_1, \ldots, x_n)$, with n the arity of Z and $x_1, \ldots, x_n$ distinct variables.

A reduction rule $\alpha \to \beta$ acts as a scheme from which actual reduction steps can be obtained by instantiating, by means of a *valuation* σ, all metavariables by terms. Then a reduction step $C[\sigma(\alpha)] \to C[\sigma(\beta)]$ is obtained by putting an instantiated version of a reduction rule $\alpha \to \beta$ in a context. Some care should be taken in the instantiating process, in order to avoid name clashes between bound variables and in order to prevent free variables from being captured by abstractions. Before defining valuations we first introduce as a notational device the n-ary *meta-abstraction.*

DEFINITION 2.3. Let t be a term in some CRS R.
(1) For an n-tuple of distinct variables $(x_1, \ldots, x_n)$, $\underline{\lambda}(x_1, \ldots, x_n).t$ is an n-ary meta-abstraction.
(2) The variables $x_1, \ldots, x_n$ in $\underline{\lambda}(x_1, \ldots, x_n).t$ are considered to be bound by $\underline{\lambda}$ and may be renamed, provided that no name clashes occur.
(3) An n-ary meta-abstraction $\underline{\lambda}(x_1, \ldots, x_n).t$ can be applied to an n-tuple of terms $(s_1, \ldots, s_n)$. The result is the term t in which $s_1, \ldots, s_n$ have been substituted simultaneously for $x_1, \ldots, x_n$:

$$(\underline{\lambda}(x_1, \ldots, x_n).t)(s_1, \ldots, s_n) = t[x_1 := s_1, \ldots, x_n := s_n]$$

(4) The free variables of an n-ary meta-abstraction $\underline{\lambda}(x_1, \ldots, x_n).t$ are the variables in t that don't occur in $(x_1, \ldots, x_n)$

We proceed by defining valuations.

DEFINITION 2.4. A *valuation* is a map σ that assigns to an n-ary metavariable Z an n-ary meta-abstraction: $\sigma(Z) = \underline{\lambda}(x_1, \ldots, x_n).t$, and is extended to a homomorphism on metaterms in the following way:
(1) $\sigma(x) = x$ for any variable x,
(2) $\sigma([x]t) = [x]\sigma(t)$,
(3) $\sigma(F(t_1, \ldots, t_n)) = F(\sigma(t_1), \ldots, \sigma(t_n))$,
(4) $\sigma(Z(t_1, \ldots, t_n)) = \sigma(Z)(\sigma(t_1), \ldots, \sigma(t_n))$.

Note that in the second clause of the definition new occurrences of x might get introduced in $\sigma(t)$ that shouldn't be bound by $[x]$. This is avoided in the following way. Let $\alpha \to \beta$ be a reduction rule and σ a valuation. The reduction rule $\alpha \to \beta$ is said to be *safe for* σ if no free variable in the co-domain of σ occurs in $\alpha \to \beta$. By renaming bound variables in a reduction rule we can for every valuation σ always find a variant of the reduction rule that is safe for σ.

DEFINITION 2.5. Let $\alpha \to \beta$ be a reduction rule and σ a valuation such that $\alpha \to \beta$ is safe for σ. Then for every context $C[\ ]$, $C[\sigma(\alpha)] \to C[\sigma(\beta)]$ is a reduction step.

Some examples might be clarifying.

EXAMPLE 2.6. Suppose we would like to instantiate the rule $F([x]Z) \to F([y]Z)$ by the valuation σ defined by $\sigma(Z) = x$. Then first we have to rename bound variables in the reduction rule such that we obtain a variant of it that is safe for σ. Take e.g. $F([x']Z) \to F([y]Z)$. Then we have for instance the reduction step $\sigma(F([x']Z)) \to \sigma(F([y]Z))$, i.e. $F([x']x) \to F([y]x)$.

Furthermore λ-calculus can be represented in the CRS formalism.

Example 2.7. The alphabet of the CRS describing λ-calculus has two function symbols: @, a binary function symbol for application, and λ, a unary function symbol for λ-abstraction. The β-reduction rule is written in the CRS formalism as:

$$@(\lambda([x]Z_1(x)), Z_2) \to Z_1(Z_2)$$

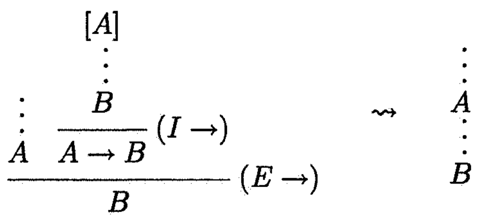
The reduction step $(\lambda x.xxy)z \to zzy$ can be written in the CRS formalism by taking the valuation σ defined by

$$\begin{aligned} \sigma(Z_1) &= \lambda x.xxy \\ \sigma(Z_2) &= z \end{aligned}$$

Then $(\lambda x.xxy)z \equiv \sigma(@(\lambda([x]Z_1(x)), Z_2)) \to \sigma(Z_1(Z_2)) \equiv xxy[x := z] \equiv zzy$.

The next example describes various proof normalizations in CRS format.

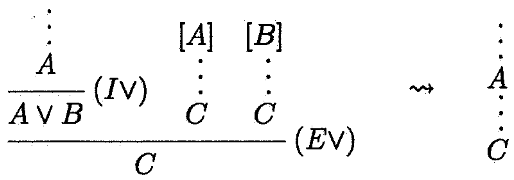
Example 2.8. In the following proof normalization a consecutive application of the rules introduction and respectively elimination arrow is eliminated.



This rule can be represented in the CRS formalism as follows :

$$\mathrm{El}(\mathrm{Int}([x]Z_1(x)), Z_2) \to Z_1(Z_2)$$

Note that this rule corresponds exactly to the one of β-reduction in λ-calculus!
Another example: a rule eliminating a consecutive use of introduction and elimination of $\vee$.



The representations of both proof normalization rules for $\vee$ in CRS format are given as follows:

$$\mathrm{El}(\mathrm{Int1}(Z_0), [x]Z_1(x), [y]Z_2(y)) \to Z_1(Z_0)$$

$$\mathrm{El}(\mathrm{Int2}(Z_0), [x]Z_1(x), [y]Z_2(y)) \to Z_2(Z_0)$$

A metaterm is called *linear* if no metavariable occurs in it more than once. A reduction rule is called *left-linear* if its left-hand side is linear. A CRS is said to be left-linear if all its reduction rules are so.

Two reduction rules $\alpha \to \beta$ and $\alpha' \to \beta'$ are said to *overlap* if there exist valuations σ and τ such that $\sigma(\alpha) \equiv \tau(\alpha'')$ with α'' a subterm of α' that is not of the form $Z(x_1, \ldots, x_n)$. Then $C[\sigma(\alpha)] \equiv \tau(\alpha')$ for some context $C[\ ]$, and for this term two possibilities to reduce exist: either $C[\sigma(\alpha)] \to C[\sigma(\beta)]$ or $\tau(\alpha') \to \tau(\beta')$. If $\alpha \to \beta$ and $\alpha' \to \beta'$ are identical, it is required that $\alpha' \not\equiv \alpha''$. Two reduction rules $\alpha \to \beta$ and $\alpha' \to \beta'$ *weakly overlap* if there exist valuations σ and τ such that $C[\sigma(\alpha)] \equiv \tau(\alpha')$, but it holds that $C[\sigma(\beta)] \equiv \tau(\beta')$.
A CRS is called *non-ambiguous* if no two reduction rules overlap. If all overlapping reduction rules only weakly overlap, a CRS is said to be *weakly non-ambiguous.* A CRS is called *orthogonal* if it is left-linear and non-ambiguous, and it is called *weakly orthogonal* if it is left-linear and weakly non-ambiguous. We shall consider only (weakly) orthogonal CRSs.

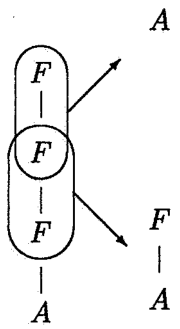
So far, no restrictions have been imposed on term formation. However, often it is the case that restrictions on term formation are present, for instance in λI-calculus or in typed λ-calculus. In order to make the CRS formalism also suitable for those cases, we define a *substructure* of a CRS as a subset of terms that is closed under reduction. If a CRS is orthogonal, then all its substructures are so. All proofs for CRSs in this paper carry over immediately to their substructures.

3. Confluence

In this section all weakly orthogonal CRSs are proved to be confluent. This means that for every two coinitial reduction sequences $s \twoheadrightarrow t$ and $s \twoheadrightarrow u$ a term v exists such that $t \twoheadrightarrow v$ and $u \twoheadrightarrow v$. This term v is called a *common reduct* of t and u. An important consequence of confluence is uniqueness of normal forms, where of course the existence of a normal form is not guaranteed. Before giving the proof of confluence for weakly orthogonal CRSs, we shall illustrate the importance of both aspects of weak orthogonality.
The TRS consisting only of the rule

$$F(F(Z)) \to A$$

is left-linear, but for instance the term $F(F(F(A)))$ contains two overlapping redexes. Both reducts A and $F(A)$ are normal forms; hence a common reduct cannot be found.



The following TRS given in Klop [15], is non-ambiguous but not left-linear.

$$\begin{array}{lll} D(Z,Z) & \to & E \\ C(Z) & \to & D(Z, C(Z)) \\ A & \to & C(A) \end{array}$$

For the divergent reductions $C(A) \twoheadrightarrow E$ and $C(A) \twoheadrightarrow C(E)$ no common reduct can be found.

The strategy of our proof of confluence is akin to the confluence proof for λ-calculus due to Tait and Martin-Löf; this proof can for instance be found in [2]. In a similar way, Aczel [1] proves confluence for his contraction schemes, that form a subclass of orthogonal CRSs.
The proof of confluence proceeds by defining a relation $\geq$ on the set of terms such that its reflexive-transitive closure equals reduction. Then it is proved that $\geq$ satisfies the diamond property, given in the following definition.

DEFINITION 3.1. A relation $\triangleright$ satisfies the *diamond property* if for each a, b and c such that $a \triangleright b$ and $a \triangleright c$, a d can be found such that both $b \triangleright d$ and $c \triangleright d$ are satisfied.

The diamond property of $\geq$ is easily seen to imply confluence of $\geq$, which in turn is equivalent to confluence of the reduction relation.
The difference between the proofs of Tait and Martin-Löf and Aczel lies not in the proof method, but in the relation for which the diamond property is proved: the relation of Tait and Martin-Löf is properly contained by the one defined by Aczel. The relation $\geq$ in our proof is similar to the one given by Aczel.

DEFINITION 3.2. The relation $\geq$ on *Ter* is defined as follows:
(1) $x \geq x$ for every $x \in Var$,
(2) if $s \geq t$, then $[x]s \geq [x]t$ for every variable x,
(3) if $s_1 \geq t_1, \ldots, s_n \geq t_n$, then $F(s_1, \ldots, s_n) \geq F(t_1, \ldots, t_n)$ for every function symbol F with arity n,
(4) for a reduction rule $\alpha \to \beta$, a valuation σ and a function symbol F of arity n: if $F(t_1, \ldots, t_n) \equiv \sigma(\alpha)$ and if $s_1 \geq t_1, \ldots, s_n \geq t_n$, then $F(s_1, \ldots, s_n) \geq \sigma(\beta)$.

The fourth clause can be visualized as follows:

$$\begin{array}{ccccc} & & F(s_1, \ldots, s_n) & & \\ & & \text{\rotatebox{-90}{$\geq$}} \quad \text{\rotatebox{-90}{$\geq$}} & \searrow_{\geq} & \\ \sigma(\alpha) & \equiv & F(t_1, \ldots, t_n) & \to & \sigma(\beta) \end{array}$$

The first three clauses of the definition state that $\geq$ is a reflexive relation that is closed under term formation. By addition of the fourth clause, we have that $s \geq t$ if the term s reduces to the term t by a parallel 'inside-out' reduction. The transitive closure of $\geq$ will be written as $\ggg$.

PROPOSITION 3.3. *The relation $\ggg$ equals reduction.*

PROOF. First, we prove with induction on the definition of $\geq$ that $s \twoheadrightarrow t$ if $s \geq t$. This yields immediately that $s \twoheadrightarrow t$ if $s \ggg t$.

- If $s \geq t$ is $x \geq x$ then indeed $s \twoheadrightarrow t$ since $x \twoheadrightarrow x$.
- If $s \geq t$ is $[x]s' \geq [x]t'$ with $s' \geq t'$, then by the induction hypothesis $s' \twoheadrightarrow t'$. Since reduction can be performed in a context, this yields $[x]s' \twoheadrightarrow [x]t'$.
- If $s \geq t$ is $F(s_1, \ldots, s_n) \geq F(t_1, \ldots, t_n)$ with $s_1 \geq t_1, \ldots, s_n \geq t_n$, then by the induction hypothesis $s_1 \twoheadrightarrow t_1, \ldots, s_n \twoheadrightarrow t_n$. Again the fact that reduction can be performed in a context is used, and we obtain that $F(s_1, \ldots, s_n) \twoheadrightarrow F(t_1, \ldots, t_n)$.
- In the last case $s \geq t$ is due to the fourth clause of the definition of $\geq$. Then we have $s \equiv F(s_1, \ldots, s_n)$ and $t \equiv \sigma(\beta)$ with $s_1 \geq t_1, \ldots, s_n \geq t_n$ and $F(t_1, \ldots, t_n) \equiv \sigma(\alpha)$, with $\alpha \to \beta$ a reduction rule. By the induction hypothesis, we have the reductions $s_1 \twoheadrightarrow t_1, \ldots, s_n \twoheadrightarrow t_n$. So we have $s \equiv F(s_1, \ldots, s_n) \twoheadrightarrow F(t_1, \ldots, t_n) \to \sigma(\beta) \equiv t$.

Conversely, let $s \to t$, i.e. $s \equiv C[\sigma(\alpha)]$ and $t \equiv C[\sigma(\beta)]$. Using the reflexivity of $\geq$ and the fourth clause of its definition, we obtain that $\sigma(\alpha) \geq \sigma(\beta)$ for a reduction rule $\alpha \to \beta$ and some valuation σ. Then the fact that $\geq$ is closed under term construction yields that $C[\sigma(\alpha)] \geq C[\sigma(\beta)]$. So if $s \twoheadrightarrow t$ then $s \ggg t$. □

We will introduce the concept 'coherence', which will be useful for proving that $\geq$ satisfies the diamond property. This notion is originally due to Aczel [1], but our definition of coherence, however, is more liberal (i.e. coherence in Aczel's sense implies coherence in our sense) because we also take weakly orthogonal CRSs into account.

DEFINITION 3.4. A relation $\triangleright$ on the set of terms of a CRS R is called *coherent* with respect to the reduction relation of R if the following holds: if $F(a_1, \ldots, a_n) \equiv \sigma(\alpha)$ for a reduction rule $\alpha \to \beta$ and some valuation σ and if $a_1 \triangleright b_1, \ldots, a_n \triangleright b_n$, then either $F(b_1, \ldots, b_n) \equiv \tau(\alpha)$ and $\sigma(\beta) \triangleright \tau(\beta)$, for a valuation τ, or $\sigma(\beta) \equiv F(a'_1, \ldots, a'_n)$ with $a'_1 \triangleright b_1, \ldots, a'_n \triangleright b_n$. The second case only arises if the redex $F(a_1, \ldots, a_n)$ overlaps with a redex in one of the a_i.

The notion of coherence is visualized in the following pictures; in the first case we have

$$\begin{array}{ccc} F(a_1, \ldots, a_n) & \to & a \\ \triangledown \quad \triangledown & & \triangledown \\ F(b_1, \ldots, b_n) & \to & b \end{array}$$

and the second case can be viewed as

$$\begin{array}{ccc} F(a_1, \ldots, a_n) & \to & F(a'_1, \ldots, a'_n) \\ \triangledown \quad \triangledown & & \triangledown \quad \triangledown \\ F(b_1, \ldots, b_n) & \equiv & F(b_1, \ldots, b_n) \end{array}$$

The next to be proved is that $\geq$ is coherent, which in turn will be used for proving that $\geq$ satisfies the diamond property. In order to prove coherence of $\geq$ we need two technical propositions.

PROPOSITION 3.5. *If $a \geq b$ and $s_1 \geq t_1, \ldots, s_n \geq t_n$, then*

$$a[x_1 := s_1, \ldots, x_n := s_n] \geq b[x_1 := t_1, \ldots, x_n := t_n]$$

PROOF. The proof proceeds by induction on the definition of $a \geq b$.

- In the case that $a \geq b$ is $x \geq x$ it is clear that the statement holds.
- If $a \geq b$ is $[x]a' \geq [x]b'$, then we have, if x is not among $x_1, \ldots, x_n$,

$$\begin{aligned} ([x]a')[x_1 := s_1, \ldots, x_n := s_n] &\equiv [x](a'[x_1 := s_1, \ldots, x_n := s_n]) \\ &\geq [x](b'[x_1 := t_1, \ldots, x_n := t_n]) \\ &\equiv ([x]b')[x_1 := t_1, \ldots, x_n := t_n] \end{aligned}$$

The second step follows by induction hypothesis. If in fact $x \equiv x_i$ for some $i \in \{1, \ldots, n\}$, then

$$([x]a')[x_1 := s_1, \ldots, x_n := s_n] = [x](a'[x_1 := s_1, \ldots, x_n := s_n])$$

with $x_i := s_i$ left out of the substitution, and reasoning along the same lines it follows that

$$([x]a')[x_1 := s_1, \ldots, x_n := s_n] = ([x]b')[x_1 := t_1, \ldots, x_n := t_n]$$

- If $a \geq b$ is $F(a_1, \ldots, a_n) \geq F(b_1, \ldots, b_n)$ with $a_1 \geq b_1, \ldots, a_n \geq b_n$, then again the statement is a direct consequence of the induction hypothesis.
- The most complicated case is when $a \geq b$ is $F(a_1, \ldots, a_n) \geq b$ with $a_i \geq b_i$ for $i = 1, \ldots, n$ and $F(b_1, \ldots, b_n) \rightarrow b$. In that case we have by definition $a[x_1 := t_1, \ldots, x_n := t_n] = F(a_1[\ldots], \ldots, a_n[\ldots])$. By the induction hypothesis, we have $a_1[\ldots] \geq b_1[\ldots], \ldots, a_n[\ldots] \geq b_n[\ldots]$ with $[\ldots]$ the appropriate substitution. The following that has to be proved is that $F(b_1[\ldots], \ldots, b_n[\ldots]) \rightarrow b[\ldots]$, and this is easily seen to be the case, since reduction rules do not contain any free variables.

□

PROPOSITION 3.6. *Let t be a metaterm containing only metavariables $Z_1, \ldots, Z_k$. Let σ and τ be valuations. If $\sigma(Z_i(x_1, \ldots, x_{p_i})) \geq \tau(Z_i(x_1, \ldots, x_{p_i}))$ for $i = 1, \ldots, k$, then $\sigma(t) \geq \tau(t)$.*

PROOF. The proof proceeds by induction on the structure of t.

- If $t \equiv x$ for some variable x, then $\sigma(t) \equiv x \geq x \equiv \tau(t)$.
- If $t \equiv [x]u$ for some metaterm u, then we have

$$\begin{aligned} \sigma(t) &\equiv \sigma([x]u) \\ &\equiv [x]\sigma(u) \\ &\geq [x]\tau(u) \\ &\equiv \tau([x]u) \\ &\equiv \tau(t) \end{aligned}$$

- If $t \equiv F(t_1, \ldots, t_n)$ for some n-ary function symbol F and metaterms $t_1, \ldots, t_n$, we have by induction hypothesis $\sigma(t_1) \geq \tau(t_1), \ldots, \sigma(t_n) \geq \tau(t_n)$. So we have

$$\begin{aligned} \sigma(t) &\equiv \sigma(F(t_1, \ldots, t_n)) \\ &\equiv F(\sigma(t_1), \ldots, \sigma(t_n)) \\ &\geq F(\tau(t_1), \ldots, \tau(t_n)) \\ &\equiv \tau(F(t_1, \ldots, t_n)) \\ &\equiv \tau(t) \end{aligned}$$

- The last case to be considered is when $t \equiv Z_i(t_1, \ldots, t_{p_i})$ for some $i \in \{1, \ldots, k\}$. Let $\sigma(Z_i(x_1, \ldots, x_{p_i})) \equiv a_i$ and $\tau(Z_i(x_1, \ldots, x_{p_i})) \equiv b_i$. In that case, we have $\sigma(t) \equiv a_i[x_1 := \sigma(t_1), \ldots, x_n := \sigma(t_n)]$ and $\tau(t) \equiv b_i[x_1 := \tau(t_1), \ldots, x_n := \tau(t_n)]$. By hypothesis respectively induction hypothesis, we have $a_i \geq b_i$ and $\sigma(t_1) \geq \tau(t_1), \ldots, \sigma(t_n) \geq \tau(t_n)$. Then, applying Proposition 3.5 indeed yields $\sigma(t) \geq \tau(t)$.

□

With the aid of these two propositions, we can now prove the main lemma.

LEMMA 3.7. *The relation $\geq$ is coherent.*

PROOF. Suppose $F(a_1, \ldots, a_n) \equiv \sigma(\alpha)$ for some reduction rule $\alpha \rightarrow \beta$ and denote by a its contractum $\sigma(\beta)$. Suppose furthermore that we have $a_1 \geq b_1, \ldots, a_n \geq b_n$. We shall prove that either $F(b_1, \ldots, b_n) \equiv \tau(\alpha) \rightarrow \tau(\beta)$ with $a \geq \tau(\beta)$ or $a \geq F(b_1, \ldots, b_n)$ by the third clause of the definition of $\geq$. By Proposition 3.3, we know that $a_1 \twoheadrightarrow b_1, \ldots, a_n \twoheadrightarrow b_n$. There may occur overlap between these reductions and the reduction $F(a_1, \ldots, a_n) \to a$, but by weak orthogonality every critical pair is trivial.

The proof proceeds by induction on the nesting of the overlaps. The basis of the induction is the case that no overlap at all occurs. Then the reductions $a_1 \twoheadrightarrow b_1, \ldots, a_n \twoheadrightarrow b_n$ don't affect the pattern of α and $F(b_1, \ldots, b_n)$ is still an instance of α, say $\tau(\alpha)$. So in this case it has to be proved that $\sigma(\beta) \geq \tau(\beta)$. The metaterm α is of the form $F(\alpha_1, \ldots, \alpha_n)$ with $\alpha_1, \ldots, \alpha_n$ metaterms. Since $a_1 \geq b_1, \ldots, a_n \geq b_n$, we have that $\sigma(\alpha_1) \geq \tau(\alpha_1), \ldots, \sigma(\alpha_n) \geq \tau(\alpha_n)$. Metavariables in α occur only in the form $Z_i(x_1, \ldots, x_{p_i})$, so this yields $\sigma(Z_i(x_1, \ldots, x_{p_i})) \geq \tau(Z_i(x_1, \ldots, x_{p_i}))$ for every metavariable Z_i occurring in α. Because $\alpha \to \beta$ is a reduction rule, β doesn't contain any metavariable that doesn't already occur in α, so we can apply Proposition 3.6 and obtain $a \equiv \sigma(\beta) \geq \tau(\beta)$.

In the induction step, we suppose that in the reduction sequences $a_1 \twoheadrightarrow b_1, \ldots, a_n \twoheadrightarrow b_n$ at least one overlap with α occurs. If overlap occurs on place i, then $a_i \equiv C_i[\rho_i(\alpha_i)]$. By weak orthogonality, we know that $a'_1, \ldots, a'_n$ exist such that $a \equiv F(a'_1, \ldots, a'_n)$ and such that $a'_i \equiv a_i$ if no overlap at place i occurs and $a'_i \equiv C_i[\rho_i(\beta_i)]$ if overlap does in fact occur. Now it has to be proved that $a'_i \geq b_i$ for $i = 1, \ldots, n$. For all i with $a'_i \equiv a_i$ this is immediate. The complicated case is when $a_i \equiv C_i[\rho_i(\alpha_i)]$, $a'_i \equiv C_i[\rho_i(\beta_i)]$ and $a_i \geq b_i$, by hypothesis, and it has to be proved that $a'_i \geq b_i$. This will be proved by induction on the structure of the context $C_i[\ ]$.

The first case is that the context is empty, so $a_i \equiv \rho_i(\alpha_i)$, $a'_i \equiv \rho_i(\beta_i)$ and we know by hypothesis that $a_i \geq b_i$. It has to be proved that $a'_i \geq b_i$. In the change from a_i to b_i, the redex $\rho_i(\alpha_i)$ has been contracted (that was the one causing overlap), so $a_i \geq b_i$ is due to the fourth clause of the definition of $\geq$. So $a_i \equiv \rho_i(\alpha_i) \equiv G(r_1, \ldots, r_m)$ and $s_1, \ldots, s_m$ exist such that $r_1 \geq s_1, \ldots, r_m \geq s_m$ and $G(s_1, \ldots, s_m) \to b_i$. In $r_i \geq s_i$ the nesting of the overlap is strictly less than in $a_i \geq b_i$, so we can apply the (main) induction hypothesis and obtain $a'_i \equiv \rho_i(\beta_i) \geq b_i$. Note that the reduction $G(s_1, \ldots, s_m) \to b_i$ in fact takes place in one step (not zero).

In the case that the context is $[x]..$, we have $a_i \equiv [x]\rho_i(\alpha_i)$, $a'_i \equiv [x]\rho_i(\beta_i)$ and we know that $a_i \geq b_i$. Then $b_i \equiv [x]b'_i$ with $\rho_i(\alpha_i) \geq b'_i$. With the sub-induction hypothesis, it follows that $\rho_i(\beta_i) \geq b'_i$, so $a_i \equiv [x]\rho_i(\beta_i) \geq [x]b'_i \equiv b_i$.

The last possibility is that $a_i \equiv H(\ldots, \rho_i(\alpha_i), \ldots)$, $a'_i \equiv H(\ldots, \rho_i(\beta_i), \ldots)$ and $a_i \geq b_i$. We distinguish two possibilities. First we consider the case that $a_i \geq b_i$ is due to the third clause of the definition of $\geq$. Then $b_i \equiv H(\ldots)$. The sub-induction hypothesis yields that we have in that case that $a'_i \geq b_i$ as a consequence of the third clause. The other possibility is that $a_i \geq b_i$ is due to the fourth clause. Again by the sub-induction hypothesis, then we have $a'_i \geq b_i$ as a consequence of the fourth clause. □

THEOREM 3.8. *The relation $\geq$ satisfies the diamond property.*

PROOF. We shall prove that for any a, b and c such that $a \geq b$ and $a \geq c$ there exists a d such that $b \geq d$ and $c \geq d$. The proof proceeds by induction on the definition of $a \geq b$.

- If $a \geq b$ is $x \geq x$, then take $d := c$.
- If $a \geq b$ is $[x]a' \geq [x]b'$ with $a' \geq b'$, then $a \geq c$ is necessarily of the form $[x]a' \geq [x]c'$ with $a' \geq c'$. By induction hypothesis, a d' exists such that $b' \geq d'$ and $c' \geq d'$. So, by defining $d := [x]d'$, both $b \geq d$ and $c \geq d$ are satisfied.

- In the case that $a \geq b$ is $F(a_1, \ldots, a_n) \geq F(b_1, \ldots, b_n)$ with $a_1 \geq b_1, \ldots, a_n \geq b_n$, $a \geq c$ can either be due to the third or to the fourth clause of the definition of $\geq$.

  If $a \geq c$ is $F(a_1, \ldots, a_n) \geq F(c_1, \ldots, c_n)$ with $a_1 \geq c_1, \ldots, a_n \geq c_n$; by induction hypothesis there exist $d_1, \ldots, d_n$ such that $b_i \geq d_i$ and $c_i \geq d_i$ for $i = 1, \ldots, n$. Then taking $d := F(d_1, \ldots, d_n)$ yields the desired result.

  In the case that $a \geq c$ is due to the fourth clause of the definition of $\geq$, we have $a \equiv F(a_1, \ldots, a_n)$ and $c_1, \ldots, c_n$ such that $a_1 \geq c_1, \ldots, a_n \geq c_n$, $F(c_1, \ldots, c_n) \equiv \sigma(\alpha)$ and $c \equiv \sigma(\beta)$ with $\alpha \to \beta$ a reduction rule. By induction hypothesis, $d_1, \ldots, d_n$ exist such that $b_i \geq d_i$ and $c_i \geq d_i$ for $i = 1, \ldots, n$. Since $\geq$ is coherent, either $F(d_1, \ldots, d_n) \equiv \tau(\alpha)$ and $\sigma(\beta) \geq \tau(\beta)$ or $c \equiv \sigma(\beta) \equiv F(c'_1, \ldots, c'_n)$ and $c'_1 \geq d_1, \ldots, c'_n \geq d_n$. In the first case, define $d := \tau(\beta)$. Then $b \equiv F(b_1, \ldots, b_n) \geq d$ by the fourth clause of the definition of $\geq$ and $c \equiv \sigma(\beta) \geq d$ by coherence. In the second case, define $d := F(d_1, \ldots, d_n)$. Then $b \equiv F(b_1, \ldots, b_n) \geq d$ by the third clause of the definition of $\geq$ and $c \geq d$ by coherence.
- The last case to be considered is when $a \geq b$ is a consequence of the fourth clause of the definition. In that case, $a \equiv F(a_1, \ldots, a_n)$ and b is the contractum of $F(b_1, \ldots, b_n)$ with $a_1 \geq b_1, \ldots, a_n \geq b_n$.

  For reasons of symmetry, the case that $a \geq c$ is due to the third clause of the definition of $\geq$ has already been treated.

  The other possibility is that $a \geq c$ is a consequence of the fourth clause of the definition. That is, we have $F(b_1, \ldots, b_n) \equiv \sigma_1(\alpha_1)$ with $b \equiv \sigma_1(\beta_1)$ and $F(c_1, \ldots, c_n) \equiv \sigma_2(\alpha_2)$ with $c \equiv \sigma_2(\beta_2)$ for valuations σ_1, σ_2 and reduction rules $\alpha_1 \to \beta_1$ and $\alpha_2 \to \beta_2$. By induction hypothesis, $d_1, \ldots, d_n$ exist such that $b_i \geq d_i$ and $c_i \geq d_i$ for $i = 1, \ldots, n$. Coherence of $\geq$ yields that on the one hand $F(d_1, \ldots, d_n) \equiv \tau_1(\alpha_1)$ with $\sigma_1(\beta_1) \geq \tau_1(\beta_1)$ or $\sigma_1(\beta_1) \equiv F(b'_1, \ldots, b'_n)$ with $b'_1 \geq d_1, \ldots, b'_n \geq d_n$ and on the other hand $F(d_1, \ldots, d_n) \equiv \tau_2(\alpha_2)$ with $\sigma_2(\beta_2) \geq \tau_2(\beta_2)$ or $\sigma_2(\beta_2) \equiv F(c'_1, \ldots, c'_n)$ with $c'_1 \geq d_1, \ldots, c'_n \geq d_n$. We consider all the different combinations.

  (1) $F(d_1, \ldots, d_n) \equiv \tau_1(\alpha_1)$ and $F(d_1, \ldots, d_n) \equiv \tau_2(\alpha_2)$. By weak orthogonality, $\tau_1(\beta_1) \equiv \tau_2(\beta_2)$. Define $d := \tau_1(\beta_1)$. Then coherence yields both $b \geq d$ and $c \geq d$.
  (2) $F(d_1, \ldots, d_n) \equiv \tau_1(\alpha_1)$ and $c \equiv F(c'_1, \ldots, c'_n)$ with $c'_1 \geq d_1, \ldots, c'_n \geq d_n$. Define $d := \tau_1(\beta_1)$. Then $b \geq d$ since $\sigma_1(\beta_1) \geq \tau_1(\beta_1)$ and $c \geq d$ by the fourth clause of the definition of $\geq$.
  (3) $b \equiv F(b'_1, \ldots, b'_n)$ with $b'_1 \geq d_1, \ldots, b'_n \geq d_n$ and $F(d_1, \ldots, d_n) \equiv \tau_2(\alpha_2)$. Define $d := \tau_2(\beta_2)$. Then $b \geq d$ by the fourth clause of the definition of $\geq$ and $\sigma_2(\beta_2) \geq \tau_2(\beta_2)$ or $c \geq d$.
  (4) $b \equiv F(b'_1, \ldots, b'_n)$ with $b'_1 \geq d_1, \ldots, b'_n \geq d_n$ and $c \equiv F(c'_1, \ldots, c'_n)$ with $c'_1 \geq d_1, \ldots, c'_n \geq d_n$. Define $d := F(d_1, \ldots, d_n)$, then $b \geq d$ and $c \geq d$ both by the third clause of the definition of $\geq$.

□

COROLLARY 3.9. *The relation $\geq$ is confluent.*

The main result now is a direct consequence of this corollary.

COROLLARY 3.10. *All weakly orthogonal CRSs are confluent.*

4. Superdevelopments

A development of a term t is a reduction sequence in which only redexes that descend from redexes in the initial term t may be contracted, whereas redexes that are created during the reduction are not allowed to be contracted. The Finite Developments Theorem, stating that all developments are finite, has been proved for λ-calculus [2], term rewriting systems and combinatory reduction systems [14]. Using this theorem, an alternative proof of confluence can be obtained.
Let $\mapsto_1$ be the relation fulfilling the same role in the confluence proof by Tait and Martin-Löf for λ-calculus as the relation $\geq$ in the proof presented here. Between $\mapsto_1$ and developments there exists the following relation: $s \mapsto_1 t$ if and only if there exists a development $s \twoheadrightarrow t$. If we consider only λ-calculus, it turns out that the relation $\mapsto_1$ is properly contained by $\geq$, that is, $s \geq t$ if $s \mapsto_1 t$ but not necessarily vice versa. So terms s and t can be found such that $s \geq t$ but s cannot reduce to t by a development. In this section we shall characterize the reduction sequences corresponding exactly to the relation $\geq$. The necessary notion will not surprisingly turn out to be a generalization of a development and will therefore be called a *superdevelopment.*
Intuitively, a superdevelopment is a reduction sequence in which besides redexes that descend from the initial term also those redexes may be contracted that have been created by means of reductions in their proper subterms. So, if we think of a term as a tree, 'upwards created' redexes are allowed to be contracted. The notion superdevelopment indeed is more liberal than the one of a development, since in a development the whole redex pattern must descend from the initial term.
In section 4.1 it will be proved for the case of orthogonal TRSs that $s \geq t$ iff s reduces to t by a superdevelopment. Furthermore, all superdevelopments are proved to be finite. In section 4.2 the same will be done for the case of λ-calculus.

4.1. Superdevelopments for orthogonal TRSs

In order to characterize reduction sequences in an orthogonal TRS R corresponding exactly to the relation $\geq$, terms of R are labelled using only the symbol '$*$'. On the set of labelled terms Ter^* a reduction relation $\rightarrow_*$ is defined. If a term t, labelled in some way, reduces to a normal form with respect to $\rightarrow_*$, then this reduction sequence is, after having erased all labels, a superdevelopment. It will be proved that all superdevelopments are finite.
When only orthogonal TRSs are considered, the relation $\geq$ on terms can be simplified.

DEFINITION 4.1. For an orthogonal TRS R, the relation $\geq$ on $Ter(\mathcal{F}, Mvar)$ is defined as follows:
(1) $Z \geq Z$ for every $Z \in Mvar$,
(2) if $s_1 \geq t_1, \ldots, s_n \geq t_n$, then $F(s_1, \ldots, s_n) \geq F(t_1, \ldots, t_n)$ for every function symbol with arity n,
(3) for a reduction rule $\alpha \rightarrow \beta$, a valuation σ and a function symbol F of arity n: if $F(t_1, \ldots, t_n) \equiv \sigma(\alpha)$ and if $s_1 \geq t_1, \ldots, s_n \geq t_n$, then $F(s_1, \ldots, s_n) \geq \sigma(\beta)$.

We proceed by defining the set of labelled terms and the reduction relation $\rightarrow_*$ on them.

DEFINITION 4.2. Define for an orthogonal TRS R with a set of function symbols $\mathcal{F}$ a set of labelled function symbols $\mathcal{F}^*$ as $\mathcal{F}^* = \{F^* | F \in \mathcal{F}\}$. The set of labelled terms is given by $Ter^*(\mathcal{F} \cup \mathcal{F}^*, Mvar)$ and is mostly written as Ter^*.

The function $E : Ter^* \to Ter$ that erases all labels is defined inductively as follows:

$$\begin{aligned} E(Z) &\equiv Z \text{ for a metavariable } Z \\ E(F(t_1, \ldots, t_n)) &\equiv F(E(t_1), \ldots, E(t_n)) \\ E(F^*(t_1, \ldots, t_n)) &\equiv F(E(t_1), \ldots, E(t_n)) \end{aligned}$$

A term $t \in Ter$ can be labelled by a partial function L from occurrences of symbols of t to $\{*\}$. This partial function is called a labelling for t, and the result of applying L to the symbols of t is written as t^L.

DEFINITION 4.3. Reduction $\to_*$ on Ter^* is defined as follows: $s \to_* t$ if for some reduction rule $\alpha \to \beta$ and some valuation $\sigma : Mter \to Ter$ it holds that $E(s) \equiv C[\sigma(\alpha)]$ and $E(t) \equiv C[\sigma(\beta)]$. Moreover, there are two conditions on the labellings: in s the head-symbol of α is labelled, and in t none of the function symbols of β is labelled. The reflexive-transitive closure of $\to_*$ is written as $\twoheadrightarrow_*$.

DEFINITION 4.4. A reduction sequence $s \twoheadrightarrow t$ is a *superdevelopment* if it can be obtained from a $\to_*$-reduction sequence to normal form by erasing all labels.

Consider for a simple example of a superdevelopment in which one redex is created upwards the TRS with reduction rules

$$\begin{aligned} F(B) &\to C \\ A &\to B \end{aligned}$$

Since $A \geq B$ and $F(B) \to C$, we have that $F(A) \geq C$. It is easily seen that no development from $F(A)$ to C exist, since the redex pattern $F(B)$, necessary to obtain C, is not yet present in $F(A)$. On the other hand, a superdevelopment from $F(A)$ to C does exist:

$$F(A) \to F(B) \to C$$

The redex $F(B)$ is created by contraction of A, and the head-symbol F of this created redex was already present in the very beginning in the term $F(A)$. The redex $F(B)$ is said to be 'created upwards', because reducing the argument has made F to be the head-symbol of a redex pattern.

THEOREM 4.5. *If $s \geq t$, then there exists a labelling L for s such that for some labelling L' $s^L \twoheadrightarrow_* t^{L'}$ is a reduction to $\to_*$-normal form.*

PROOF. Suppose $s \geq t$. By induction on the definition of $\geq$ it will be proved that a labelling L exists such that $s^L \twoheadrightarrow_* t^{L'}$ is, for some labelling L', a reduction to normal form.

- If $s \geq t$ is $Z \geq Z$ for a metavariable Z, then the empty reduction sequence $Z \twoheadrightarrow_* Z$ indeed is a reduction to normal form.
- If $s \geq t$ is $F(s_1, \ldots, s_n) \geq F(t_1, \ldots, t_n)$ with $s_1 \geq t_1, \ldots, s_n \geq t_n$, then by induction hypothesis labellings $L_1, \ldots, L_n$ exist such that $s_i^{L_i} \twoheadrightarrow_* t_i^{L'_i}$ is a reduction to normal form for $i = 1, \ldots, n$. Let L be the union of $L_1, \ldots, L_n$. Then $s^L \twoheadrightarrow_* t^{L'}$ is a reduction to normal form, with L' the union of $L'_1, \ldots, L'_n$.
- If $s \geq t$ is due to the last clause of the definition of $\geq$, then $s \equiv F(s_1, \ldots, s_n)$ with $s_1 \geq t_1, \ldots, s_n \geq t_n$, $F(t_1, \ldots, t_n) \equiv \sigma(\alpha)$ and $t \equiv \sigma(\beta)$ for some reduction rule $\alpha \to \beta$ of R. By induction hypothesis, labellings $L_1, \ldots, L_n$ exist such that $s_i^{L_i} \twoheadrightarrow_* t_i^{L'_i}$ is a reduction to normal form for $i = 1, \ldots, n$. Let L be the union of $L_1, \ldots, L_n$, extended by assigning

$*$ to the head-symbol F. Then we have the following reduction: $F^*(s_1^{L_1}, \ldots, s_n^{L_n}) \twoheadrightarrow_* F^*(t_1^{L'_1}, \ldots, t_n^{L'_n}) \rightarrow_* t^{L'}$. The term $t^{L'}$ is in normal form since $t_1^{L'_1}, \ldots, t_n^{L'_n}$ are normal forms and the last reduction step cannot create any new redexes, by definition of labelled reduction.

□

THEOREM 4.6. *If $s \twoheadrightarrow_* t$ is a reduction to normal form with respect to labelled reduction, then $E(s) \geq E(t)$.*

PROOF. Let $s \twoheadrightarrow_* t$ be a $\rightarrow_*$-reduction to normal form. It will be proved by induction on the structure of $s \in Ter^*$ that $E(s) \geq E(t)$.

- If $s \equiv Z$ for some metavariable Z, then the only possible superdevelopment is $Z \twoheadrightarrow_* Z$. Indeed $E(Z) \equiv Z \geq Z \equiv E(Z)$.
- If $s \equiv F(s_1, \ldots, s_n)$ then superdevelopments from s to t are of the form $F(s_1, \ldots s_n) \twoheadrightarrow_* F(t_1, \ldots, t_n)$ with $s_i \twoheadrightarrow_* t_i$ a superdevelopment for $i = 1, \ldots, n$. By induction hypothesis, $E(s_1) \geq E(t_1), \ldots, E(s_n) \geq E(t_n)$. So

$$\begin{aligned} E(s) &\equiv F(E(s_1), \ldots, E(s_n)) \\ &\geq F(E(t_1), \ldots, E(t_n)) \\ &\equiv E(t) \end{aligned}$$

- If $s \equiv F^*(s_1, \ldots, s_n)$ and in a superdevelopment $s \twoheadrightarrow_* t$ the head-symbol F never becomes a redex, then this case is entirely similar to the previous one. If, however, s itself becomes a redex, then the superdevelopment is of the form

$$\begin{aligned} F^*(s_1, \ldots, s_n) &\twoheadrightarrow_* F^*(s'_1, \ldots, s'_n) \\ &\rightarrow_* G(t_1, \ldots, t_m) \\ &\twoheadrightarrow_* G(t'_1, \ldots, t'_m) \\ &\equiv t \end{aligned}$$

By the definition of labelled reduction, no redexes are created 'downwards', so all redexes in the term $G(t_1, \ldots, t_m)$ descend from redexes in $F^*(s'_1, \ldots, s'_n)$. By orthogonality, $F^*(s'_1, \ldots, s'_n)$ remains a redex if first the arguments are reduced to normal form. So the outermost contraction can be postponed until the last step, and the superdevelopment from s to t can be written as

$$\begin{aligned} s &\twoheadrightarrow_* F^*(s'_1, \ldots, s'_n) \\ &\twoheadrightarrow_* F^*(s''_1, \ldots, s''_n) \\ &\rightarrow_* t \end{aligned}$$

with s''_i the $\rightarrow_*$-normal form of s_i for $i = 1, \ldots, n$.

Then, by induction hypothesis, $E(s_i) \geq E(s''_i)$ for $i = 1, \ldots, n$. Moreover, for some reduction rule $\alpha \rightarrow \beta$ and some valuation σ, $E(F^*(s''_1, \ldots, s''_n)) \equiv \sigma(\alpha)$ and $E(t) \equiv \sigma(\beta)$. So we have by the third clause of the definition of $\geq$ that $E(s) \geq E(t)$.

□

COROLLARY 4.7. *$s \geq t$ if and only if there exists a superdevelopment $s \twoheadrightarrow t$.*

Now we shall prove all superdevelopments to be finite. The proof of finiteness of superdevelopments is very similar to that of finite developments, and also uses the following well-known fact.

THEOREM 4.8. *A relation $>$ on a set S is well-founded if and only if the multiset extension $>>$ of $>$ is well-founded on $M(S)$.*

THEOREM 4.9. *If $t \in Ter^*$ then all its $\rightarrow_*$-reduction sequences are finite.*

PROOF. Let t be in Ter^*. Assign to t a weight $W(t)$ in the following way:

$$W(t) = \begin{cases} 0 & \text{if } t \text{ is a metavariable} \\ \max\{W(t_1), \ldots, W(t_n)\} & \text{if } t \equiv F(t_1, \ldots, t_n) \\ 1 + \max\{W(t_1), \ldots, W(t_n)\} & \text{if } t \equiv F^*(t_1, \ldots, t_n) \end{cases}$$

Associate to t the multiset to which each symbol of t contributes the weight of the subterm of which it is the head-symbol. Contracting a redex wipes off at least one $*$, namely the one of the head-symbol of that contracted redex, and could possibly multiply subterms with a smaller weight. So performing a reduction step yields a decrease of the multiset associated to t. Well-foundedness of the multiset ordering yields that all reduction sequences of t with respect to $\rightarrow_*$ must terminate. □

COROLLARY 4.10. *All superdevelopments in orthogonal TRSs are finite.*

4.2. Superdevelopments for λ-calculus

In this section we shall characterize the reduction sequences corresponding exactly to the relation $\geq$ on λ-terms. In order to do so, a set of labelled λ-terms Λ_l and labelled β-reduction $\rightarrow_{\beta_l}$ on them will be defined. Lambda's will be labelled by a label from a countably infinite set of labels I, and application nodes will be labelled by a subset of I. If the labelling of a λ-term M satisfies certain conditions, then its β_l-reduction to normal form is, after having erased all labels, a superdevelopment. Furthermore, all β_l-reductions are proved to be finite.
In [16] Lévy analyses the different ways in which β-redexes can be created. The following possibilities are distinguished:
(1) $((\lambda x.\lambda y.M)N)P \rightarrow_\beta (\lambda y.M[x := N])P$
(2) $(\lambda x.x)(\lambda y.M)N \rightarrow_\beta (\lambda y.M)N$
(3) $(\lambda x.C[xM])(\lambda y.N) \rightarrow_\beta C[(\lambda y.N)M]$
The first two created redexes are 'innocent' and may be contracted in a superdevelopment. Note that, if we think of a λ-term as a tree built from application- and λ-nodes, the redexes in the first two cases are 'created upwards'. In the last case, on the other hand, the redex isn't created upwards, and may not be contracted in a superdevelopment. The result that all superdevelopments are finite illustrates that all infinite β-reduction sequences in λ-calculus are due to the third way of redex creation; indeed redex creation e.g. in the reduction sequence of $(\lambda x.xx)(\lambda x.xx)$ happens in this way.
In the following, we shall write the application nodes explicitly, but abstraction terms as usual. Further, the relation $\geq$ when only used on λ-terms can be simplified a bit.

DEFINITION 4.11. The relation $\geq$ on λ-terms is defined in the following way:
(1) $x \geq x$ for each variable x,

(2) if $M \geq M'$ then $\lambda x.M \geq \lambda x.M'$ for a λ-term M,
(3) if $M \geq M'$ and $N \geq N'$ then $@(M,N) \geq @(M',N')$ for λ-terms M and N,
(4) if $M \geq \lambda x.M'$ and $N \geq N'$, then $@(M,N) \geq M'[x := N']$ for λ-terms M and N.

We proceed by defining the set of labelled λ-terms.

DEFINITION 4.12. The set Λ_l of labelled λ-terms is defined as the smallest set such that
(1) $x \in \Lambda_l$ for every variable x,
(2) if $M \in \Lambda_l$ and $i \in I$, then $\lambda_i x.M \in \Lambda_l$,
(3) if $M, N \in \Lambda_l$ and $X \subset I$, then $@^X(M,N) \in \Lambda_l$.

Erasing all labels of a term $M \in \Lambda_l$ is done by a function $E : \Lambda_l \to \Lambda$ that is defined inductively as follows:

$$\begin{aligned} E(x) &= x \\ E(\lambda_i x.M) &= \lambda x.E(M) \\ E(@^X(M_1,M_2)) &= @(E(M_1),E(M_2)) \end{aligned}$$

A labelling L for a λ-term M is a partial function from the symbols of M to $I \cup \wp(I)$, where $\wp(I)$ is the set containing all subsets of I, called the powerset of I. The resulting term of Λ_l is written as M^L.
The reduction rule β_l on Λ_l is defined as

$$@^X(\lambda_i x.M, N) \to_{\beta_l} M[x := N] \quad \text{if } i \in X$$

where the substitution $[x := N]$ is defined as usual.

DEFINITION 4.13. A term $M \in \Lambda_l$ is called *good* if no label X of an application node contains the index i of a λ occurring outside the scope of this application node. A labelling L for a term $M \in \Lambda$ is called *good* if $M^L \in \Lambda_l$ is a good term. A labelling L for $M \in \Lambda$ is an *initial labelling* if it is good and all λ's have a unique label.

For example, $@^{\{2\}}(@^{\{1\}}(\lambda_1 x.\lambda_2 y.xy, z), u)$ is a good term but $@^{\{1\}}(\lambda_1 x.@^{\{2\}}(x,y), \lambda_2 y.y)$ isn't. The property 'good' is preserved under reduction, i.e. β_l-reduction cannot push a λ outside the scope of an application node in which it occurred originally. This is proved in the following proposition, that will be used implicitly.

PROPOSITION 4.14. *If $M \in \Lambda_l$ is a good term and $M \to_{\beta_l} N$, then N is a good term.*

PROOF. We shall prove that if a redex $@^Y(\lambda_i x.P, Q)$ is good, then its reduct $P[x := Q]$ is good. The proof proceeds by induction on the structure of P. It is obvious that all subterms of a good term are good, so in particular P and Q are good terms.

- If $P \equiv x$, then $P[x := Q] \equiv Q$ is a good term.
- If $P \equiv y \not\equiv x$, then $P[x := Q] \equiv y$ is a good term.
- If $P \equiv \lambda_j y.P_1$, then $P[x := Q] \equiv \lambda_j y.P_1[x := Q]$. Since $@^Y(\lambda_i x.\lambda_j y.P_1, Q)$ is a good term, $@^Y(\lambda_i x.P_1, Q)$ is good. By induction hypothesis, $P_1[x := Q]$ is a good term, so $\lambda_j y.P_1[x := Q] \equiv P[x := Q]$ is a good term.
- If $P \equiv @^X(P_1,P_2)$, then $P[x := Q] \equiv @^X(P_1[x := Q], P_2[x := Q])$. Since the term $@^Y(\lambda_i x.@^X(P_1,P_2), Q)$ is by hypothesis good, $@^Y(\lambda_i x.P_1, Q)$ is also good. By induction hypothesis, $P_1[x := Q]$ is a good term. In the same way we obtain that $P_2[x := Q]$ is a good term. Since there are no λ's at all outside the scope of the the outside application

node, $P[x := Q] \equiv @^X(P_1[x := Q], P_2[x := Q])$ is a good term.
□

Definition 4.15. A reduction sequence $M \twoheadrightarrow_\beta N$ is a *superdevelopment* if for some initial labelling L, $M^L \twoheadrightarrow_{\beta_l} N^{L'}$ is a β_l-reduction sequence to normal form.

The following proposition states that no β_l-redexes are created by substitution.

Proposition 4.16. *If $@^X(\lambda_i x.P, Q) \in \Lambda_l$ is a good term, then all patterns of β_l-redexes in $P[x := Q]$ descend either totally from P or totally from Q.*

Proof. Suppose $@^X(\lambda_i x.P, Q)$ is a good term and we have in $P[x := Q]$ a subterm of the form $@^Y(\lambda_j y.R, S)$. If the symbol $@^Y$ originates from P and λ_j from Q, then $j \notin Y$, because $@(\lambda_i x.P, Q)$ is a good term. So in that case $@^Y(\lambda_j y.R, S)$ is not a β_l-redex. It is impossible to have in $P[x := Q]$ a subterm $@^Y(\lambda_j y.R, S)$ with $@^Y$ originating from Q and λ_j from P. So if $@^Y(\lambda_j y.R, S)$ is a β_l-redex in $P[x := Q]$, then $@^Y$ and λ_j originate either both from P or both from Q. □

Proposition 4.17. *If $P \twoheadrightarrow_{\beta_l} P'$ and $Q \twoheadrightarrow_{\beta_l} Q'$ are β_l-reductions to normal form, and P and Q have no labels in common, then $P[x := Q] \twoheadrightarrow_{\beta_l} P'[x := Q']$ and $P'[x := Q']$ is a β_l-normal form.*

Proof. The proof proceeds by induction on the structure of P.

- If $P \equiv x$, then its reduction to normal form consists necessarily of zero steps. We have $P[x := Q] \equiv Q$ and $Q \twoheadrightarrow_{\beta_l} Q' \equiv x[x := Q']$, and moreover by hypothesis Q' is a β_l-normal form.
- If $P \equiv y \not\equiv x$, then the only possible reduction sequence is $y \twoheadrightarrow_{\beta_l} y$. We have $y[x := Q] \equiv y \twoheadrightarrow_{\beta_l} y \equiv y[x := Q']$, and y is clearly a β_l-normal form.
- If $P \equiv \lambda_i x.P_1$, then a β_l-reduction sequence of P to its normal form P' is of the form $\lambda_i x.P_1 \twoheadrightarrow_{\beta_l} \lambda_i x.P_1'$ with $P_1 \twoheadrightarrow_{\beta_l} P_1'$ a β_l-reduction to normal form. We have $P[x := Q] \equiv \lambda_i x.P_1[x := Q]$. By induction hypothesis, $P_1[x := Q] \twoheadrightarrow_{\beta_l} P_1'[x := Q']$, and hence $\lambda_i x.P_1[x := Q] \twoheadrightarrow_{\beta_l} \lambda_i x.P_1'[x := Q']$. So $P[x := Q] \twoheadrightarrow_{\beta_l} P'[x := Q']$. By induction hypothesis, $P_1'[x := Q']$ is a β_l-normal form, so $P'[x := Q']$ is in β_l-normal form too.
- If $P \equiv @^X(P_1, P_2)$, then let P_1' and P_2' be the β_l-normal forms of P_1 and P_2 respectively. If $@^X(P_1', P_2')$ is a normal form, then this is the normal form P' of P, and it follows by induction that $P[x := Q] \twoheadrightarrow_{\beta_l} P'[x := Q']$. By Proposition 4.16 we know that patterns of eventually occurring redexes in $P'[x := Q']$ stem either totally from P' or totally from Q', and since these terms are both in normal form, $P'[x := Q']$ is the normal form of $P[x := Q]$.

  If $@^X(P_1', P_2')$ isn't a normal form, then P_1' is of the form $\lambda_i y.P_{11}'$ with $i \in X$. By Proposition 4.16, we know that $P_{11}'[y := P_2']$ only contains redexes that are entirely in P_{11}' or in P_2'. Since these are both normal forms, $P_{11}'[y := P_2']$ contains no redexes and hence is the normal form of P. By induction hypothesis, $P_1[x := Q] \twoheadrightarrow_{\beta_l} \lambda_i y.P_{11}'[x := Q']$ and $P_2[x := Q] \twoheadrightarrow_{\beta_l} P_2'[x := Q']$. So we have $@^X(P_1, P_2)[x := Q] \twoheadrightarrow_{\beta_l} @^X(\lambda_i y.P_{11}'[x := Q'], P_2'[x := Q'])$. This term reduces to $P_{11}'[x := Q'][y := P_2'[x := Q']]$, which equals $P_{11}'[y := P_2'][x := Q']$, and by Proposition 4.16 this term is a normal form.

□

This proposition yields that if $@^X(\lambda_i x.P, Q)$ is a good term and its reduct $P[x := Q]$ reduces to a β_l-normal form M, then M is of the form $P'[x := Q']$, with P' and Q' the normal forms of P and Q respectively.

THEOREM 4.18. *If $M \geq M'$, then there exists an initial labelling L such that $M^L \twoheadrightarrow_{\beta_l} M'^{L'}$ and $M'^{L'}$ is a β_l-normal form for some labelling L'.*

PROOF. The proof proceeds by induction on the definition of $\geq$.

- If $M \geq M'$ is $x \geq x$, then indeed $x \twoheadrightarrow_{\beta_l} x$ without need of any labels.
- If $M \geq M'$ is $\lambda x.M_1 \geq \lambda x.M_1'$ with $M_1 \geq M_1'$, then by induction hypothesis, an initial labelling L exists such that $M_1^L \twoheadrightarrow_{\beta_l} M_1'^{L'}$ is a reduction to normal form. Let i be a fresh label. Then L extended by assigning i to the first λ is an initial labelling for M, and $\lambda_i x.M_1^L \twoheadrightarrow_{\beta_l} \lambda_i x.M_1'^{L'}$ is a reduction to β_l-normal form.
- If $M \geq M'$ is $@(M_1, M_2) \geq @(M_1', M_2')$ with $M_1 \geq M_1'$ and $M_2 \geq M_2'$, then by induction hypothesis initial labellings L_1 and L_2 for M_1 respectively M_2 exist such that $M_1^{L_1} \twoheadrightarrow_{\beta_l} M_1'^{L_1'}$ and $M_2^{L_2} \twoheadrightarrow_{\beta_l} M_2'^{L_2'}$ are reductions to β_l-normal form with $E(M_1'^{L_1'}) \equiv M_1'$ and $E(M_2'^{L_2'}) \equiv M_2'$. Without losing generality we can suppose all labels of λ's in $M_1^{L_1}$ to be different from those of λ's in $M_2^{L_2}$. Then the labelling L defined as the union of L_1 and L_2, extended by assigning $\emptyset \subset I$ to the head-symbol $@$, is an initial labelling for M, and $M^L \twoheadrightarrow_{\beta_l} M'^{L'}$ with $E(M'^{L'}) \equiv M'$ is a reduction to normal form.
- If $M \geq M'$ is due to the fourth clause of the definition of $\geq$, then $M \equiv @(M_1, M_2)$ with $M_1 \geq \lambda x.M_1'$, $M_2 \geq M_2'$ and $M' \equiv M_1'[x := M_2']$. By induction hypothesis, there exist initial labellings L_1 and L_2 for M_1 and M_2 respectively, such that $M_1^{L_1} \twoheadrightarrow_{\beta_l} \lambda_i x.M_1'^{L_1'}$ and $M_2^{L_2} \twoheadrightarrow_{\beta_l} M_2'^{L_2'}$ are reduction sequences to normal form with $E(\lambda_i x.M_1'^{L_1'}) \equiv \lambda x.M_1'$ and $E(M_2'^{L_2'}) \equiv M_2'$. Again we can without loss of generality suppose that all labels of λ's in M_1 are different from those of λ's in M_2. Then the labelling L defined as the union of L_1 and L_2 extended by assigning $\{i\}$ to the first application node is an initial labelling for M. Moreover, we have the following reduction sequence: $@^{\{i\}}(M_1^{L_1}, M_2^{L_2}) \twoheadrightarrow_{\beta_l} @^{\{i\}}(\lambda_i x.M_1'^{L_1'}, M_2'^{L_2'}) \rightarrow_{\beta_l} M_1'^{L_1'}[x := M_2'^{L_2'}]$. Since $M_1'^{L_1'}$ and $M_2'^{L_2'}$ are both in normal form, we have by Proposition 4.17 that $M_1'^{L_1'}[x := M_2'^{L_2'}]$ is a normal form.

□

THEOREM 4.19. *If $M \in \Lambda_l$ is a good term and $M \twoheadrightarrow_{\beta_l} M'$ is a β_l-reduction sequence to normal form, then $E(M) \geq E(M')$.*

PROOF. The proof proceeds by induction on the structure of M.

- If $M \equiv x$, then the only possible β_l-reduction sequence is $x \twoheadrightarrow_{\beta_l} x$, and indeed $x \geq x$.
- If $M \equiv \lambda_i x.M_1$, then reduction sequences of M are of the form $\lambda_i x.M_1 \twoheadrightarrow_{\beta_l} \lambda_i x.M_1'$ with $M_1 \twoheadrightarrow_{\beta_l} M_1'$ a β_l-reduction to normal form. By induction hypothesis, $E(M_1) \geq E(M_1')$. This yields $E(M) \equiv \lambda x.E(M_1) \geq \lambda x.E(M_1') \equiv E(\lambda_i x.M_1')$.
- If $M \equiv @^X(M_1, M_2)$, then we distinguish two possibilities. The first possibility is $M' \equiv @^X(M_1', M_2')$, with $M_1 \twoheadrightarrow_{\beta_l} M_1'$ and $M_2 \twoheadrightarrow_{\beta_l} M_2'$ β_l-reduction sequences to normal form. By induction hypothesis, $E(M_1) \geq E(M_1')$ and $E(M_2) \geq E(M_2')$. This yields $E(M) \equiv @(E(M_1), E(M_2)) \geq @(E(M_1'), E(M_2')) \equiv E(M')$. The other possibility is that M becomes a redex, i.e. $M \equiv @^X(M_1, M_2) \twoheadrightarrow_{\beta_l} @^X(\lambda_i x.P, Q)$ and this term is a β_l-redex. Then the reduction proceeds in the following way: $@^X(\lambda_i x.P, Q) \rightarrow_{\beta_l} P[x := Q] \twoheadrightarrow_{\beta_l} M'$. In this case, $M_1 \twoheadrightarrow_{\beta_l} \lambda_i x.P$ and $M_2 \twoheadrightarrow_{\beta_l} Q$. Let P' and Q' be the normal forms of P and Q respectively. Then $M_1 \twoheadrightarrow_{\beta_l} \lambda_i x.P'$ and $M_2 \twoheadrightarrow_{\beta_l} Q'$ are β_l-reductions to normal form. By induction hypothesis, $E(M_1) \geq E(\lambda_i x.P')$ and $E(M_2) \geq E(Q')$. By Proposition 4.16, we have $M' \equiv P'[x := Q']$. Applying the fourth clause of the definition of $\geq$ yields

$$E(M) \geq E(P')[x := E(Q')] \equiv E(P'[x := Q']) \equiv E(M')$$

□

Corollary 4.20. *$M \geq N$ if and only if there exists a superdevelopment $M \twoheadrightarrow N$.*

Theorem 4.21. *If a λ-term M is labelled by an initial labelling L then all its β_l-reductions are finite.*

Proof. Suppose infinite β_l-reduction sequences exist, and let M be a minimal (w.r.t. the number of symbols) λ-term, labelled by an initial labelling, that admits an infinite β_l-reduction sequence. By minimality M has to be an application, so M is of the form $@^X(M_1, M_2)$. The infinite β_l-reduction sequence starting with M then must be of the form

$$@^X(M_1, M_2) \twoheadrightarrow_{\beta_l} @^X(\lambda_i x.M_1', M_2') \rightarrow_{\beta_l} M_1'[x := M_2'] \rightarrow_{\beta_l} \dots$$

In this reduction sequence, we have $M_1 \twoheadrightarrow_{\beta_l} \lambda x_i.M_1'$ and $M_2 \twoheadrightarrow_{\beta_l} M_2'$, and moreover $i \in X$. Now we claim that all reducts of this sequence are of the form $M_1''[M_{21}'', \dots, M_{2n}'']$ with $M_1'[, \dots,] \twoheadrightarrow_{\beta_l} M_1''[, \dots,]$ and $M_2' \twoheadrightarrow_{\beta_l} M_{2i}''$ for $i = 1, \dots, n$. So all reductions take place either in descendants of $M_1'[, \dots,]$ or in descendants of M_2'. The claim follows from proposition 4.16 and the observation that nothing can be substituted into a descendant of M_2'. From the claim it follows immediately that either M_1 or M_2 admits an infinite reduction sequence, contradicting the minimality of M.
□

Corollary 4.22. *All superdevelopments in λ-calculus are finite.*

Concluding Remarks

Although the proof of confluence presented in this paper applies to a very large class of reduction systems, it is quite simple and elegant. The study of superdevelopments sheds some light on the different possibilities of creating redexes, and classifies certain of them as harmless in the sense that they do not cause infinite reduction sequences.
As a possibility for further research one might think of adding associative and commutative operators, thus providing a link with concurrent calculi, like for instance π-calculus. Probably confluence can be obtained for the deterministic version.

Acknowledgements

I am very grateful to my supervisor Jan Willem Klop for introducing me to the subject of CRSs and for dropping in to explain his new ideas. I would like to thank Aart Middeldorp and Vincent van Oostrom for their remarks on earlier versions of this paper.

References

[1] P. Aczel. A general Church-Rosser theorem. Technical report, University of Manchester, 1978.

[2] H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics.* North Holland, second edition, 1984.

[3] H.P. Barendregt. Typed lambda-calculi. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science, Volume I.* Oxford University Press, 1992.

[4] G. Berry and J.-J. Lévy. Minimal and optimal computations of recursive programs. *JACM*, 26(1):27–38, 1979.

[5] V. Breazu-Tannen and J. Gallier. Polymorphic rewriting conserves algebraic strong normalization and confluence. In *Proceedings of the 16th international colloquium on automata, languages and programming*, pages 137–150, 1989. Lecture Notes in Computer Science 372.

[6] H.B. Curry and R. Feys. *Combinatory Logic*, volume I. North-Holland, Amsterdam, 1958.

[7] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[8] R. Hindley. The equivalence of complete reductions. *Transactions of the American Mathematical Society*, 229:227–248, 1977.

[9] G. Huet and J.-J. Lévy. Computations in orthogonal rewrite systems I and II. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic, Essays in Honor of Alan Robinson*, pages 395–444. MIT Press, 1991.

[10] J.-P. Jouannaud and M. Okada. A computational model for executable higher-order algebraic specification languages. In *Proceedings of the 6th annual IEEE Symposium on Logic in Computer Science*, pages 350–361, 1991.

[11] S. Kahrs. *λ-rewriting.* PhD thesis, Universität Bremen, 1991.

[12] J.R. Kennaway. Sequential evaluation strategies for parallel-or and related systems. *Annals of Pure and Applied Logic*, 43:31–56, 1989.

[13] Z. Khasidashvili. Church-Rosser Theorem in Orthogonal Combinatory Reduction Systems. INRIA Rocquencourt, 1992.

[14] J.W. Klop. *Combinatory Reduction Systems.* Mathematical Centre Tracts Nr. 127. CWI, Amsterdam, 1980. PhD Thesis.

[15] J.W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science, Volume II.* Oxford University Press, 1992.

[16] J.-J. Lévy. An algebraic interpretation of the $\lambda\beta$K-calculus and a labelled λ-calculus. In C. Böhm, editor, *Proceedings Rome Conference 1975*, pages 147–165. Springer Verlag, 1975. Lecture Notes in Computer Science 37.

[17] G. Mitschke. λ-Kalkül, δ-Konversion und axiomatische Rekursionstheorie. Preprint 274, Technische Hochschule Darmstadt, 1976.

[18] T. Nipkow. Higher-order critical pairs. In *Proceedings of the 6th annual IEEE Symposium on Logic in Computer Science*, pages 342–349, 1991.

[19] D. Prawitz. Ideas and results in proof theory. In J.E. Fenstad, editor, *Proc. 2nd Scandinavian Logic Symposium.* North-Holland, 1971.

[20] M. Schönfinkel. On the building blocks of mathematical logic. In J. van Heyenoort, editor, *From Frege to Gödel.* Harvard University Press, 1967.