Development Closed Critical Pairs

Vincent van Oostrom[⋆]

Information Processing Principles Research Group
3–1 Morinosato Wakamiya, Atsugi-Shi
Kanagawa Prefecture, 243-01, Japan

Abstract. The class of orthogonal rewriting systems (rewriting systems where rewrite steps cannot depend on one another) is the main class of not-necessarily-terminating rewriting systems for which confluence is known to hold. Huet and Toyama have shown that for left-linear first-order term rewriting systems (TRSs) the orthogonality restriction can be relaxed somewhat by allowing *critical pairs* (arising from maximally general ways of dependence between steps), but requiring them to be parallel closed. We extend these results by replacing the parallel closed condition by a *development closed* condition. This also permits to generalise them to higher-order term rewriting, yielding a confluence criterion for Klop's combinatory reduction systems (CRSs), Khasidashvili's expression reduction systems (ERSs), and Nipkow's higher-order pattern rewriting systems (PRSs).

1 Introduction

This paper is concerned with a method to prove confluence of rewriting systems. It's an extension of some confluence results in [CR36, Hue80, Toy88, Klo80, Kha92, Raa93, Tak, MN94, Oos94, ORb] and we refer the reader to these papers and to the handbook chapters [DJ, Klo] for motivation and for standard definitions as well. Here we will mainly be concerned with proving our result:

Left-linear development closed PRSs are confluent.

Let's explain the terminology used. A rewrite system for which the rewrite rules do not depend on one another is called *orthogonal*. Formalising this notion can be quite involved depending on the rewrite formalism it is applied to ([Hue80, Klo80, HL, GLM, MN94, Oos94]), but the intuition to be captured is always the same: an application of a rule replaces some substructure by another one, and in orthogonal systems we moreover have that if two distinct substructures can be replaced then these substructures are independent. Some (non)examples are:

1. The rules $\mathtt{F} \to \mathtt{G}$ and $\mathtt{G} \to \mathtt{H}$ are orthogonal (their left-hand sides $\mathtt{F}$ and $\mathtt{G}$ are independent) hence confluent.

⋆ Partially supported by de Vrije Universiteit, Amsterdam, The Netherlands, and an HCM grant at the Technische Universität München, Germany. Current address: `http://www.cs.vu.nl/~oostrom`.

2. The rules `F` $\rightarrow$ `G` and `F` $\rightarrow$ `H` are not orthogonal, since they both depend on the symbol `F`, and for this reason they're said to be *ambiguous*. The system is not confluent since `F` can be rewritten to the normal forms `G` and `H`.
3. The rules ([Hue80])

$$\begin{aligned} EQ(x,x) &\rightarrow T \\ EQ(x,S(x)) &\rightarrow F \\ \infty &\rightarrow S(\infty) \end{aligned}$$

   are not orthogonal, since application of the third rule to $EQ(\infty,\infty)$ destroys the possibility of applying the first rule. The system is not confluent since the term $EQ(\infty,\infty)$ can be rewritten to both T and F. The first and second rule are said to be *non-left-linear*, because of the presence of a repeated variable in its left-hand side.

A fundamental result in rewriting is that forbidding the kind of dependence in the latter two items suffices for orthogonality of rewriting systems:

Left-linear, non-ambiguous term rewriting systems are confluent.

Our aim is to show that for the class of higher-order Pattern Rewriting Systems (PRSs [MN94]), the non-ambiguity condition can be relaxed somewhat without jeopardising confluence. The condition we name *development closed* and it expresses that if two steps are dependent, then from the result of performing the inner step, the result of performing the outer step can be reached by one *development step*.

Remark. The notion of inner and outer are the usual ones obtained from viewing terms as trees. Outer means closer to the root. Observe that steps in disjoint subtrees cannot depend on each other.

1. Adding the rules `G` $\rightarrow$ `H` and `H` $\rightarrow$ `G` to the rewriting system in the second item above makes it development closed. The results `G` and `H` of two dependent steps can be rewritten to each other in one step, which is a special case of a development step. (Note that in this case the dependent steps are both inner and outer.)
2. The rewrite rules for 'parallel or':

$$\begin{aligned} por(x,T) &\rightarrow T \\ por(T,x) &\rightarrow T \end{aligned}$$

   are development closed since steps only depend on each other in the case of $por(T,T)$. The result then is T for either rewrite rule (so in fact the system is *weakly orthogonal* [ORb]) and since the empty step is a special case of a development step, the system is confluent.

Remark. Throughout the text there are many references to [Oos94]. This is only meant for easy reference. Many of the results can be found at many other places.

2 Rewriting

We fix the no(ta)tions for the rewriting systems we're interested in.

Definition 1. An *abstract rewriting system (ARS)* $\rightarrow$ is a binary relation on some set $(a, b \in)A$. The denotation of the 'repeated' notation $\twoheadrightarrow$ is the transitive-reflexive closure of the denotation of $\rightarrow$. Similarly, the denotation of the 'inverse' notation $\leftarrow$ is the inverse of the denotation of $\rightarrow$ and the denotation of the 'union with the inverse' notation $\leftrightarrow$ is the symmetric closure of the denotation of $\rightarrow$. We use infix notation for ARSs. If $a \rightarrow b$, then we say that *a ($\rightarrow$-)rewrites to b*, and the structure $\langle a, \rightarrow, b\rangle$ is called a *$\rightarrow$-step from a to b*. A *rewrite sequence* is a (finite or infinite) sequence of rewrite steps, such that for successive steps the object to which the former rewrites is the same as the object from which the latter rewrites. An ARS $\rightarrow$ has the *diamond property*, if $\leftarrow ; \rightarrow \subseteq \rightarrow ; \leftarrow$. An ARS $\rightarrow$ is *strongly confluent* ([Hue80]), if $\leftarrow ; \rightarrow \subseteq \rightarrow ; \twoheadleftarrow$. An ARS $\rightarrow$ is *confluent*, if $\twoheadrightarrow$ has the diamond property. An object is a *($\rightarrow$)-normal form* if no rewrite steps are possible from it. An ARS is *terminating* if no infinite rewrite sequences are possible.

It is well-known that ARSs having the diamond property (or which are strongly confluent) are confluent ([Hue80]).

Definition 2. Let $\rightarrow$ be an ARS. An ($\rightarrow$-)*span* is a pair $(\langle a, \rightarrow, b\rangle, \langle a, \rightarrow, c\rangle)$, which we will usually write as $b \leftarrow a \rightarrow c$.

For cultural reasons we employ the concrete class of higher-order pattern rewriting systems (PRSs [MN94])[2] as a vehicle to present our ideas. The proof-method will be seen to rely on two essential ingredients: trees and the notion of substructure. In PRSs the substructures employed are called *patterns* and the trees arise by viewing the objects of PRSs (λ-terms) as trees in the usual way. Our proofs could be stated in the general framework of HORSs (see [Raa96]) without difficulty, but this would not make them clearer. Stating them for PRSs is a hassle already.

Definition 3. 1. We first define the objects of a higher-order term rewriting system. *Simple types* are defined by the grammar:

$$\sigma ::= o \mid \sigma \rightarrow \sigma$$

Preterms are objects s such that $s\colon \sigma$ for simple type σ can be inferred from
(var) $x^\sigma\colon \sigma$ for termvariables x,
(app) $s\colon \sigma \rightarrow \tau,\ t\colon \sigma \Longrightarrow s(t)\colon \tau$.
(abs) $s\colon \tau \Longrightarrow x^\sigma.s^3\colon \sigma \rightarrow \tau$.

[2] Actually we employ a slight variation on PRSs as presented in [ORb] in which all rules are closed, which we find technically and conceptually more convenient.
[3] We omit the usual λ in abstractions.

Higher-order *terms* are obtained from the preterms by quotienting by the theory consisting of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ (that is, $\boldsymbol{\lambda\eta}$ in [Bar84]). To make terms concrete we use their $\beta\overline{\eta}$-normal forms as representatives (unique up to α-conversion) of the equivalence classes. Here, the rewrite relations β and $\overline{\eta}$ are generated by the rules:

$$(x.s)(t) \rightarrow_\beta s[x := t]$$
$$s \rightarrow_{\overline{\eta}} x.s(x)$$

where $\overline{\eta}$ is not allowed to create β-redexes and x does not occur in s.

2. An alphabet of (simply typed) variables $\mathcal{A}$ is distinguished from the set of all variables, and elements of $\mathcal{A}$ will be called *function symbols.* They will be used as constants, i.e. bound externally, in higher-order rewriting. To stress the functional nature of function symbols we use functional notation $\mathtt{F}(s_1, \ldots, s_m)$ instead of applicative notation $\mathtt{F}(s_1)\ldots(s_m)$ in case $m > 1$ (cf. [MN94]).
3. A *pattern rewrite rule* R is a pair $l \rightarrow r$ of closed (note that function symbols are considered bound already, as per the previous item) terms of the same simple type σ, where the *left-hand side* l is a linear pattern. Here a *linear pattern* is a term of the form $\mathbf{x}.s$,[4] such that
   (a) $s{:}\,o$. To understand this it is convenient to think of o as the set of terms.
   (b) s is of the form $\mathtt{F}(s_1, \ldots, s_m)$, $\mathtt{F}$ is called the *head-symbol* of the term,
   (c) each x_k among $\mathbf{x}$ occurs exactly once in s and has only ($\overline{\eta}$-normal forms of) pairwise distinct variables not among $\mathbf{x}$ as arguments.

   We'll just say pattern instead of linear pattern and rewrite rule instead of pattern rewrite rule. A higher-order *pattern rewriting systems* (PRS) is a pair ($\mathcal{A}$,$\mathcal{R}$) consisting of an alphabet $\mathcal{A}$ and a set $\mathcal{R}$ of rewrite rules.
4. Let $s =^{\mathrm{def}} C\boxed{l_1,\ldots,l_m}$, $t =^{\mathrm{def}} C\boxed{r_1,\ldots,r_m}$ be preterms, where $R_1 =^{\mathrm{def}} l_1 \rightarrow r_1$, $\ldots$, $R_m =^{\mathrm{def}} l_m \rightarrow r_m$ are rewrite rules, and C is an m-ary *precontext*, i.e. a preterm containing variables $\Box_1,\ldots,\Box_m$. Then we say that t can be obtained from s by *contracting* the *(complete) development redex* $C\boxed{R_1,\ldots,R_m}$.[5] This will be denoted by $s \multimap\!\!\rightarrow_{C\boxed{R_1,\ldots,R_m}} t$. We use u, v, w to denote redexes, and idenify them with their induced steps whenever convenient. This is extended to (the unique representatives of) terms by defining a *development step* $s \multimap\!\!\rightarrow t$ if there exist preterms s' and t' in the same $\beta\overline{\eta}$-equivalence classes as s and t, respectively, such that $s' \multimap\!\!\rightarrow t'$. The ARS associated to the PRS $\mathcal{H}$ is the relation on terms obtained by requiring the precontext employed in a development step to be a unary *context*, i.e. a term containing exactly one occurrence of $\Box$. The so-obtained relation $\rightarrow_{\mathcal{H}}$ (or simply $\rightarrow$) is called the *rewrite step* relation.

Note that we've only defined the linear PRSs, but that suffices for our purposes. For thorough investigations carried out using (also non-linear) PRSs we refer the reader to [MN94, Pre95].

[4] We employ boldface to denote sequences, i.e. $\mathbf{x}$ is a sequence of variables.

[5] This essentially amounts to replacing some derivation subtrees of a simply typed lambda term (in the usual sense) by other ones.

The development rewrite relation defined above is somewhat overly general. Without loss of generality the precontexts can be restricted to contexts, since the precontext can be $\beta\overline{\eta}$-normalised, preserving that there's a development step, i.e. if $C\boxed{\mathbf{l}} \mathrel{-\!\!\circ\!\!\!\rightarrow} C\boxed{\mathbf{r}}$, then $D\boxed{\mathbf{l}} \mathrel{-\!\!\circ\!\!\!\rightarrow} D\boxed{\mathbf{r}}$ where D is the $\beta\overline{\eta}$-normal form of C (cf. [Oos94, p. 68]). Furthermore, every step between terms $s \leftrightarrow^{*}_{\beta\overline{\eta}} s' \mathrel{-\!\!\circ\!\!\!\rightarrow} t' \leftrightarrow^{*}_{\beta\overline{\eta}} t$ may be assumed of the form $s \twoheadleftarrow_{\beta} s' \mathrel{-\!\!\circ\!\!\!\rightarrow} t' \twoheadrightarrow_{\beta} t$, by confluence of β and the fact that β-reduction preserves $\overline{\eta}$-normal forms (cf. [Oos94, Prop. 3.2.10]). This justifies restricting attention to β-reduction in the following.

Definition 4. 1. If $C\boxed{l} \twoheadrightarrow_{\beta} s$ for some context C, pattern l and term s, then we say that l *is a pattern (at* C*) in* s. A set $l_1,\ldots,l_m$ of patterns (at $C_1,\ldots,C_m$) in s is said to be *independent (at* C*)*, if C is an m-ary context such that $C\boxed{l_1,\ldots,l_m} \twoheadrightarrow_{\beta} s$, and such that the head-symbol of l_k descends to the same symbol in s in both $C\boxed{l_1,\ldots,l_m} \twoheadrightarrow_{\beta} s$ and $C_k\boxed{l_k} \twoheadrightarrow_{\beta} s$. Here, the descendant relation on function symbols is the usual one for β-reduction ([Bar84, Hue93] or [Oos94, Def. 3.2.12]). It's important to note that due to the linearity of patterns, each function symbol always has exactly one descendant along these β-reductions to s. Independence of sets of redexes is defined via their left-hand sides. We use $\mathcal{U}$, $\mathcal{V}$, $\mathcal{W}$ to range over sets of independent redexes and identify them with their induced development steps whenever convenient.

2. The descendant relation of a development step $s \twoheadleftarrow_{\beta} C\boxed{l_1,\ldots,l_m} \mathrel{-\!\!\circ\!\!\!\rightarrow}_{C\boxed{R_1,\ldots,R_m}} C\boxed{r_1,\ldots,r_m} \twoheadrightarrow_{\beta} t$ of a set of independent redexes is the relation composition of the descendant relations of its three components, the β-expansion, the replacement, and the β-reduction. ([Oos94, Def. 3.1.25]). In the replacement step $C\boxed{l_1,\ldots,l_m} \mathrel{-\!\!\circ\!\!\!\rightarrow}_{C\boxed{R_1,\ldots,R_m}} C\boxed{r_1,\ldots,r_m}$, function symbols in the context C descend to themselves. Function symbols in the left-(right-)hand sides are said to be *destroyed (created)*.
3. Having defined descendants of function symbols along steps, descendants of patterns can be defined directly via their head-symbol and subsequently descendants of redexes (so called *residuals*) via their left-hand sides ([Oos94, sec. 3.1.1]). A pattern must be independent of the contracted redex to have a descendant. The set of residuals of a set of (implicitly independent) redexes $\mathcal{V}$ *after* a development step contracting $\mathcal{U}$ is written as $\mathcal{V}/\mathcal{U}$.

In the following sections we recapitulate some standard general abstract nonsense about rewriting (in)dependent sets of redexes, paving the technical way for the proof of our main corollaries.

3 Independence

We list the main ingredients of the confluence by developments method and do some shopping in literature to get the results needed for the class of PRSs.

Theorem 5 Prism. *1. Let* $t \mathrel{\leftarrow\!\!\!\circ\!\!-}_{\mathcal{U}} s \mathrel{-\!\!\circ\!\!\!\rightarrow}_{\mathcal{V}} r$ *be a development span contracting the sets of independent redexes* $\mathcal{U} \subseteq \mathcal{V}$*. Then* $t \mathrel{-\!\!\circ\!\!\!\rightarrow}_{\mathcal{V}/\mathcal{U}} r$ *and the descendant relations induced by* $\mathrel{-\!\!\circ\!\!\!\rightarrow}_{\mathcal{U}} ; \mathrel{-\!\!\circ\!\!\!\rightarrow}_{\mathcal{V}/\mathcal{U}}$ *and* $\mathrel{-\!\!\circ\!\!\!\rightarrow}_{\mathcal{V}}$ *are the same.*

2. *If $\mathcal{U} \cup \mathcal{V}$ is independent, then $\mathcal{V}/\mathcal{U}$ is independent again.*
3. *Every development step can be* serialised*, i.e. if $s \mathrel{-\!\!\circ\!\!\!\rightarrow} t$, then there exists a rewrite sequence $s \twoheadrightarrow t$ consisting of rewrite steps, inducing the same descendant relation.*

Proof. 1. This is the Prism Theorem as stated in [Hue93] for the untyped lambda calculus with beta reduction. We will illustrate our proof for PRSs in the case of a set of two rules $l \to r$ and $g \to d$.

$$\begin{tikzcd}
C\boxed{l,g} \arrow[rrr, "\circ" marking] \arrow[dr, "\beta"'] \arrow[dd, two heads, "\beta"'] & & & C\boxed{r,g} \arrow[rrr, "\circ" marking] \arrow[dl, "\beta"] \arrow[dr, "\beta"'] \arrow[dd, two heads, "\beta"'] & & & C\boxed{r,d} \arrow[dl, "\beta"] \arrow[dd, two heads, "\beta"] \\
 & D\boxed{l} \arrow[r, "\circ" marking] \arrow[dl, "\beta"] & D\boxed{r} \arrow[dr, "\beta"'] & & E\boxed{\mathbf{g}} \arrow[r, "\circ" marking] \arrow[dl, "\beta"] & E\boxed{\mathbf{d}} \arrow[dr, "\beta"'] & \\
s \arrow[rrr] & & & s' \arrow[rrr, "\circ" marking] & & & t
\end{tikzcd}$$

The diagram represents contraction of two independent redexes (l and g) in the term s. The top-line going from $C\boxed{l,g}$ to $C\boxed{r,d}$ corresponds to contracting both in one step, while the middle lines from $D\boxed{l}$ to $D\boxed{r}$ and from $E\boxed{\mathbf{g}}$ to $E\boxed{\mathbf{d}}$ correspond to first contracting l and then the residuals of g. The left-part of the diagram shows how the development step contracting $D\boxed{l}$ can be lifted into a context (C) where also the g-redex is made explicit (we called this the Envelop Lemma [Oos94, Lem. 3.1.42]). Now from this lifted step, it is obvious what should be done next: just contract the residual g in $C\boxed{r,g}$. The problem then is that $C\boxed{r,\ }$ is not a context, only a precontext, but to get the development step we want, we only need to reduce the precontext to its β-normal form E, possibly erasing or duplicating g on the way, and we are done (this we called the Develop Lemma [Oos94, Lem. 3.1.43]). Of course, one needs to check that the descendant relations behave well, but that's a matter of routine and can be found in [Oos94].

2. This is intuitively obvious, since the union being independent, the contraction of a subset may only lead to erasure or duplication of the others. Technically, we've already used this in the proof of the previous item (consider what happened to g in the right-hand side of the diagram above) and is a consequence of the Develop Lemma in [Oos94, Lem. 3.1.43]. Cf. also Section 6.2 in [Hue93], where his compatibility corresponds to our independence.
3. This follows from the Prism Theorem if we can find a particular strategy which decreases the size of the set of independent redexes in every step. This is easy: always contract an innermost redex. This cannot lead to duplication of other redexes. $\odot$

From the first two items of the theorem it follows immediately that developments of mutually independent sets of redexes satisfy the diamond property. From serialisation it moreover follows that confluence of $\mathrel{-\!\!\circ\!\!\!\rightarrow}$ is the same as confluence of $\to$ (cf. [Hue80]).

Note that like [Hue93], we didn't employ the *finiteness of developments theorem* (FD, i.e. that any serialisation of a development step is finite). How the techniques employed above can be slightly strengthened to yield FD as a bonus can be found in [Oos94], but since we don't need it here we omit it. Moreover, taking development steps for the multi-step derivations in [HL] there's no problem in setting up the theory of *permutation equivalence*. Again, since we don't need that part of the general abstract theory of independence here, we don't develop it.

In general steps need of course not be independent, but if steps operate on disjoint subterms, they are.

Definition 6. Let l, g be patterns at C, D in s. The patterns are related to each other via the positions (in the usual tree representation, see [Oos94, Def. 3.2.6] for a formal definition of this) of the descendants of their head-symbols in s. If l is on the path to the root of g, then l is *outside* g, or also g is *inside* l. If neither is outside the other, then they're said to be *disjoint*. Redexes are related to each other via their left-hand sides. A redex is said to be *innermost* among a set of (not necessarily independent) redexes if all redexes in the set are either outside or disjoint from it.

In [Hue80] the notation $\twoheadrightarrow\!\!\!\!\!\!\!\,\parallel$ was introduced to denote the development of a set of disjoint ('parallel') redexes.

Lemma 7. *Sets of disjoint redexes are independent.*

Proof. Trivial from the restriction to trees. $\odot$

4 Dependence

After having sketched the behaviour of independent steps, we briefly recapitulate some basic theory of dependent steps as known from e.g. [Mil91, Pfe, MN94, Pre95]. The basic results ([Mil91, Pre95]) we need are:

1. *Matching* for patterns is decidable. This yields that the rewrite relation $\rightarrow$ of a PRS is decidable.
2. *Unification* of patterns is decidable as well, and *most general unifiers* do exist.

This yields that one can compute so-called critical pairs and that this preserves decidability of matching and unification. Basically, patterns allow you to carry over the techniques from first- to higher-order.

Definition 8. Let $t \leftarrow_{C\boxed{l \rightarrow r}} s \rightarrow_{D\boxed{g \rightarrow d}} r$ be a span of PRS-steps.

1. An *intersector* of the span is a pattern p at C' in l and at D' in g, such that C' at C and D' at D are patterns in the same term (context). This expresses that p is a substructure on which both redexes operate.

2. The intersector is called *critical*, if there does not exist an intersector p' at C'' and D'', such that p is at E in p', and E is a pattern (context) at C'' in C' and at D'' in D'. The term p is said to be the *critical intersection* of the span, denoted sloppywise by $l \sqcap g$. This expresses that p is the maximal substructure on which both redexes operate.
3. The span is *dependent* if an intersector exists for it, that is there is some substructure on which both operate.
4. The dependent span is *critical*, if there does not exist a dependent span $t' \leftarrow_{C'\boxed{l\to r}} s' \rightarrow_{D'\boxed{g\to d}} r'$ such that s' is at E in s, and C' in C and D' in D are both at E. The term s' is said to be the *critical unification* of the span, (imprecisely) denoted by $l \sqcup g$. The critical unification is the largest substructure on which either of the steps operate.

That patterns are closed under all these operations and critical unifications and intersections exist in case of dependent steps follows from results in [Mil91, Pfe, MN94, Pre95].

Lemma 9. *1. A set of redexes is independent if and only if its elements are pairwise independent.*
2. Two redexes are independent if and only if they don't form a dependent span.
3. Every dependent span is obtained by putting a critical span in a context.

Proof. This is tedious. The first item was shown in [Oos94, Prop. 3.1.49]. The second and third item can be dug out from Section 4 in [MN94]. $\odot$

Lemma 10. *If two steps u, u' (with left-hand sides l, l') are both not inside v (with left-hand side g) and do both depend on v, then they depend on each other.*

Proof. Since both steps do not depend on v, by Lemma 7 they cannot be disjoint from v and since they're also not inside v, they must be both outside v. Now, because of the tree restriction if some pattern $\tilde{l}$ is outside another one $\tilde{g}$ on which it depends, the head-symbol of $\tilde{g}$ must be in $\tilde{l} \sqcap \tilde{g}$. In particular both $l \sqcap g$ and $l' \sqcap g$ must contain the head-symbol of g, so $l \sqcap l'$ is non-empty. $\odot$

For PRSs at least one of the contexts in a critical span must be of the form $\mathbf{x}.\Box(\mathbf{s})$, since if not the contexts would have a common head contradicting minimality ([MN94],[Oos94, p. 102]). A context of the form $\mathbf{x}.\Box(\mathbf{s})$ is called a *substitution context*. Then, the symmetry in the notion of critical span can be avoided by requiring that D is a substitution context, in the definition above. For such critical spans the pair (t,r) is called a *critical pair*.[6] A critical pair is called *trivial*, if $t \equiv r$. The only symmetric case remaining is when both C and D are substitution contexts. In that case the critical pair is called a *root critical pair*, giving rise to a *critical overlay* when both elements of the pair are put into the same context.

Definition 11. A critical pair (t,r) is *development closed*, if $t \mathrel{-\!\!\circ\!\!\rightarrow} r$.

[6] This notion of critical pair is (apart from the initial binder) the usual one! Cf. [Hue80].

5 Development Closed

Huet studied confluence of (first-order) term rewriting systems (TRSs) in [Hue80]. He showed that *parallel closed* left-linear TRSs, i.e. TRSs such that for every critical pair (s,t), we have that $s \twoheadrightarrow\!\!\!\!\!\!\!+\!\!\!+ \; t$, are confluent ([Hue80, Lem. 3.3], see the left-hand side of Figure 1).

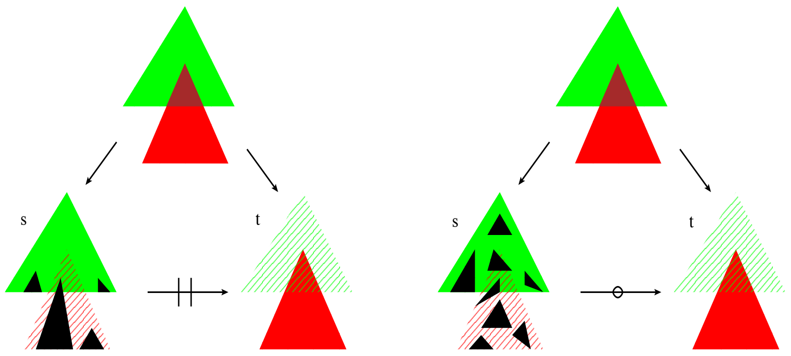

Fig. 1. Parallel closed and Development closed

We show that if a PRS is *development closed*, i.e. for all critical pairs (s,t), $s \multimap t$ (see the right-hand side of Figure 1), confluence can be concluded. Since sets of disjoint redexes are independent by Lemma 7, this extends Huet's result. Since left-linear TRSs are PRSs, this moreover generalises it to the higher-order case. Note that our result holds for Klop's combinatory reduction systems ([Klo80]) and Khasidashvili's expression reduction systems ([Kha90]) as well, since these can be embedded into PRSs in a natural way (see [ORa, Raa96]).

Example 1. Consider a PRS having function symbols $\mathtt{app}\colon o \to (o \to o)$, $\mathtt{abs}\colon (o \to o) \to o$, $\mathtt{or}\colon o \to o \to o$, $\mathtt{tt}\colon o$ and rewrite rules[7]

$$\begin{aligned} y.z.\mathtt{app}(\mathtt{abs}(x.y(x)),z) &\to_{beta} y.z.y(z) \\ y.\mathtt{abs}(x.\mathtt{app}(y,x)) &\to_{eta} y.y \\ x.\mathtt{or}(\mathtt{tt},x) &\to_{lor} x.\mathtt{tt} \\ x.\mathtt{or}(x,\mathtt{tt}) &\to_{ror} x.\mathtt{tt} \end{aligned}$$

Rewrite steps in this system need not be independent. Critical pairs arise from different ways to unify left-hand sides of rules (cf. [MN94]). The critical pair $(\mathtt{app}(s,t),\mathtt{app}(s,t))$ arises from the critical unification $\mathtt{app}(\mathtt{abs}(x.\mathtt{app}(s,x)),t)$ of *beta* and *eta*, $(\mathtt{abs}(y.s(y)),\mathtt{abs}(x.s(x)))$ arises from the critical unification $\mathtt{abs}(x.\mathtt{app}(\mathtt{abs}(y.s(y)),x))$ of *eta* and *beta*, and $(\mathtt{tt},\mathtt{tt})$ arises from the critical unification $\mathtt{or}(\mathtt{tt},\mathtt{tt})$ of *lor* and *ror* (and vice versa). All the critical pairs are trivial so surely development closed, hence the system is confluent.

[7] The beta and eta rules are just the usual beta ($(\lambda x.M)N \to M[x := N]$) and eta ($\lambda x.Mx \to M$, if x not free in M) rules of lambda calculus using higher-order notation (cf. [MN94]).

5.1 Huet

We proceed with our adaptation of the proofmethod in [Hue80].

Lemma 12. *Let $\mathcal{H}$ be a development closed PRS. Then $\mathrel{-\!\!\circ\!\!\to}$ satisfies the diamond property.*

Proof. The structure of the proof is the same as Huet's and is by induction on the 'critical intersection' between the sets in a development span. Let $t \mathrel{\leftarrow\!\!\circ\!\!-}_{\mathcal{V}} s \mathrel{-\!\!\circ\!\!\to}_{\mathcal{U}} r$ be a development span.

1. If $\mathcal{V} \cup \mathcal{U}$ is independent, then the result follows from Theorem 5.
2. If $\mathcal{V} \cup \mathcal{U}$ is not independent, then let $|s, \mathcal{V}, \mathcal{U}|$ denote the number of function symbols in all critical intersections between steps in $\mathcal{V}$ and $\mathcal{U}$. By Lemma 9 this number is greater than zero. We will transform the development span into another one $t \mathrel{\leftarrow\!\!\circ\!\!-}_{\mathcal{V}'} s' \mathrel{-\!\!\circ\!\!\to}_{\mathcal{U}'} r$, such that $|s', \mathcal{V}', \mathcal{U}'|$ is smaller than $|s, \mathcal{V}, \mathcal{U}|$. The idea is to do this by removing an innermost dependency using the assumption of development closedness. For that purpose consider among all the dependent steps between $\mathcal{V}$ and $\mathcal{U}$ an innermost one (not necessarily unique), say (wlog) $v =^{\text{def}} D\boxed{l \to r} \in \mathcal{V}$ and $s \to_v t'$. By the innermost assumption, Lemma 10, and the assumption that $\mathcal{U}$ is independent, there exists a unique step $s \to_u r'$, depending on v, with $u =^{\text{def}} E\boxed{g \to d} \in \mathcal{U}$. The resulting dependence span: $t' \leftarrow_v s \to_u r'$ can be written as an F_0-instance of a critical span $t_0 \leftarrow_{D_0\boxed{l \to r}} s_0 \to_{E_0\boxed{g \to d}} r_0$ for some context F_0, by Lemma 9. By assumption the critical pair (t_0,r_0) is development closed, so $t_0 \mathrel{-\!\!\circ\!\!\to}_{C_0} r_0$ for some set C_0 of independent redexes. By the Prism Theorem 5, this development can be put inside the context F_0 giving rise to a development $t' \mathrel{-\!\!\circ\!\!\to}_C r'$. Define $\mathcal{U}^- =^{\text{def}} \mathcal{U} - \{u\}$ and $\mathcal{V}^- =^{\text{def}} \mathcal{V} - \{v\}$, then we can make the following claim.

   <span style="font-variant:small-caps">Claim</span> Let $s' =^{\text{def}} t'$, $\mathcal{V}' =^{\text{def}} \mathcal{V}^-/v$, and $\mathcal{U}' =^{\text{def}} C \cup (\mathcal{U}^-/v)$. Then $t \mathrel{\leftarrow\!\!\circ\!\!-}_{\mathcal{V}'} s' \mathrel{-\!\!\circ\!\!\to}_{\mathcal{U}'} r$ is a development span for which $m' =^{\text{def}} |s', \mathcal{V}', \mathcal{U}'|$ is smaller than $m =^{\text{def}} |s, \mathcal{V}, \mathcal{U}|$ (see Figure 2).

   Proof of claim By independence of $\mathcal{V}^-$ and the Prism Theorem, $s' \mathrel{-\!\!\circ\!\!\to}_{\mathcal{V}'} t$. By independence of $\mathcal{U}^-$ and the Prism Theorem, $r' \mathrel{-\!\!\circ\!\!\to}_{\mathcal{U}^-/u} r$. The critical unification $l \sqcup g$ of u and v at context F in s is independent from $\mathcal{U}^-$ by construction (and since independence is preserved by unification). This means by the Prism Theorem that the descendant relation is obtained from combining the ones on F and $l \sqcup g$. Since performing 'u' from $l \sqcup g$ induces the same (empty) descendant relation as performing '$v\,; C$' this entails that $\mathcal{U}^-/(v\,; C) = \mathcal{U}^-/u$, hence also $(C \cup (\mathcal{U}^-/v))/C = \mathcal{U}^-/u$. Note that the union here is independent since it is obtained by combining independent sets of redexes in independent terms. Using this the result follows from the Prism Theorem since the sequence $C\,; (\mathcal{U}^-/u)$ is a complete development of the set $C \cup (\mathcal{U}^-/v)$.
   It remains to show that $m' < m$. By the innermost assumption redexes inside v are all independent, and by the Prism Theorem their residuals after v are

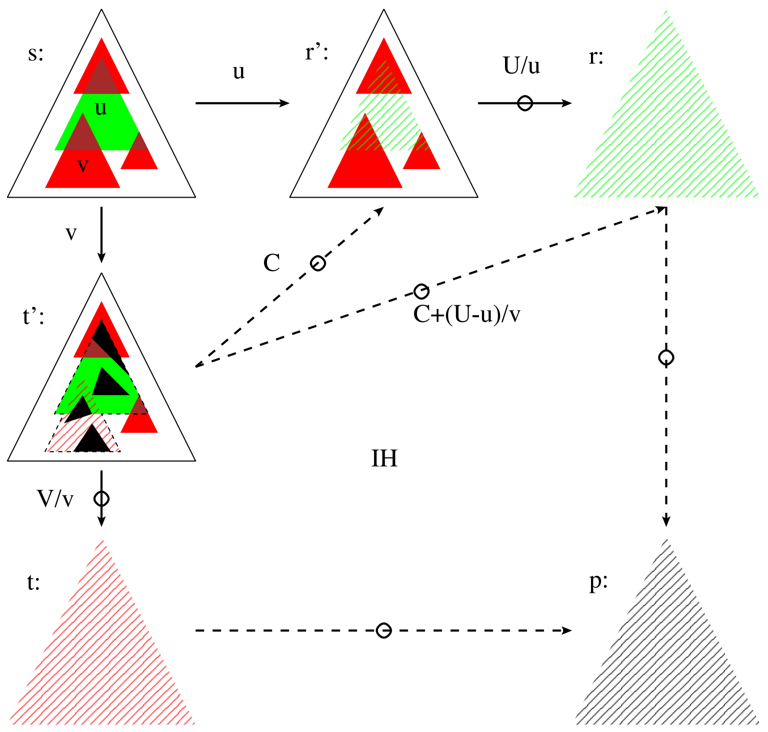

Fig. 2. proof of the claim (only patterns of steps having overlap with u are shown)

independent again. Critical intersections independent from v must therefore be outside or disjoint it, so have unique descendants after v. The only thing which could happen is that they're not longer critical intersections. This can happen when they were critical intersections between redexes in $\mathcal{W} \subseteq \mathcal{V}$ and u. But since every intersection between a redex in C and $\mathcal{W}$ is also an intersection between u and $\mathcal{W}$, per construction, the measure cannot have increased for these intersections either. We conclude by noting that the critical intersection between u and v was non-empty by assumption, contributing to m, but not to m'. $\odot$

As remarked above, the diamond property for $\multimap$ implies its confluence, which in turn implies confluence of $\rightarrow$, so we have our main result:

Corollary 13. *Development closed PRSs are confluent.*

5.2 Toyama

The main result of Toyama's [Toy88] is an improvement of Huet's result by weakening the condition on a critical pair (s,t) in case of overlay (i.e. root overlap) to the existence of a term r such that $s \,\#\!\!\!\rightarrow r \twoheadleftarrow t$. Analogous to the above extension of Huet's result, Toyama's result can be extended to only requiring $s \xrightarrow{\circ} r \twoheadleftarrow t$ in case of overlay.

Lemma 14. *Let $\mathcal{H}$ be an* almost development closed *PRS, that is,*

1. *$s \xrightarrow{\circ} t$, or*
2. *(s,t) is a root critical pair and $s \xrightarrow{\circ} ; \twoheadleftarrow t$,*

for every critical pair (s,t). Then $\xrightarrow{\circ}$ is strongly confluent.

Proof. The second part of the proof of Lemma 12 can be essentially followed, proving strong confluence instead of the diamond property (note that this part of the proof does not depend on the conclusion only the hypothesis of strong confluence) for development steps and changing the measure defined above by not counting the function symbols in critical intersections for overlays.

The base case then has to show strong confluence for sets $\mathcal{U}$ and $\mathcal{V}$ which are independent except for some possible overlays. We prove strong confluence for such cases, by induction of the size of $\mathcal{U}$ in a development span $t \xleftarrow{\circ}_{\mathcal{V}} s \xrightarrow{\circ}_{\mathcal{U}} r$. Consider an innermost step $u \in \mathcal{U}$.

1. if u is independent from any step in $\mathcal{V}$, strong confluence follows as in the diagram

$$\begin{tikzcd}
s \arrow[r, "u"'] \arrow[d, "\circ" marking, "\mathcal{V}"] & s' \arrow[r, "\circ" marking, "\mathcal{U}/u"'] \arrow[d, dashed, "\circ" marking, "\mathcal{V}/u"'] \arrow[rd, phantom, "\text{IH}"] & r \arrow[d, dashed, "\circ" marking, "="] \\
t \arrow[r, dashed, "\circ" marking, "u/\mathcal{V}"'] & t' \arrow[r, dashed, two heads] & {}
\end{tikzcd}$$

   By the innermost assumption u cannot duplicate other redexes, so $\mathcal{U}/u$ has one element less than $\mathcal{U}$.
2. if u has overlay overlap with some step $v \in \mathcal{V}$, the following diagram can be constructed

$$\begin{tikzcd}
s \arrow[r, "u"'] \arrow[d, "v"'] \arrow[rd, phantom, "\text{Ass}"] & s' \arrow[r, "\circ" marking, "\mathcal{U}/u"'] \arrow[d, dashed, "\circ" marking, "\mathcal{W}"] & r \arrow[dd, dashed, "\circ" marking] \\
{} \arrow[r, dashed, two heads, "d"] \arrow[d, "\circ" marking, "\mathcal{V}^-/v"] & {} \arrow[d, dashed, "\circ" marking, "\mathcal{V}^-/(v\,;d)"] \arrow[ru, phantom, "\text{IH}"] & \\
t \arrow[r, dashed, two heads] & t' \arrow[r, dashed, two heads] & {}
\end{tikzcd}$$

   where $\mathcal{V}^- =^{\text{def}} \mathcal{V} - \{v\}$, and *Ass* refers to the overlay case of the definition of almost development closedness. One can construct the critical unification $l \sqcup g$ of u and v at context F in s. Since u and v are overlays, $\mathcal{V}^-$ is independent from both of them, hence also from their critical unification $l \sqcup g$. From this and since the descendant relations induced by the rewrites 'u ; $\mathcal{W}$' and 'v ; d' from the almost development closedness condition are both empty, the rewrites from s obtained by putting them in context F induce the same descendant relation, like in the proof of Lemma 12. Hence $\mathcal{V}^-/(u\,;\mathcal{W}) = \mathcal{V}^-/(v\,;d)$ and $(\mathcal{W}\cup(\mathcal{V}^-/u))/\mathcal{W} = \mathcal{V}^-/(v\,;d)$, and again we see that the sequential composition of the two development steps $\mathcal{W}$ and $\mathcal{V}^-/(v\,;d)$ can be combined into one development step contracting the set $\mathcal{W}\cup(\mathcal{V}^-/u)$ of independent redexes, and the induction hypothesis can be indeed applied for the right part of the diagram, for the same reason as in the previous case. ⊙

As remarked above, strong confluence of $\multimap$ implied its confluence, which in turn implies confluence of $\mathcal{H}$, thereby extending and generalising Toyama's [Toy88, Cor. 3.2].

Corollary 15. *Almost development closed PRSs are confluent.*

Although our results don't seem to have a great number of killer-applications, they do support our feeling that many first-order techniques based on orthogonality carry over to the higher-order case.

Example 2. 1. Consider the non-orthogonal, non-terminating, non-right-linear, highly-artificial TRS having rules:

$$\begin{aligned}\mathtt{A} &\to \mathtt{G}(\mathtt{B})\\ \mathtt{A} &\to \mathtt{H}(\mathtt{K})\\ x.\mathtt{G}(x) &\to x.\mathtt{F}(x,x)\\ x.\mathtt{H}(x) &\to x.\mathtt{F}(x,x)\\ \mathtt{B} &\to \mathtt{A}\\ \mathtt{K} &\to \mathtt{A}\end{aligned}$$

This TRS is confluent by Lemma 14.

2. A more useful higher-order example to which the lemma also applies, but not immediately since the system is not a PRS (because the pattern condition is violated) is the following definitional expansion rule:

$$y.z.\mathsf{app}(\mathsf{abs}(x.y(x,x)),z) \to_{betadelta} y.z.\mathsf{app}(\mathsf{abs}(x.y(x,z)),z)$$

This rule allows to 'expand' some 'abbreviations' x to their 'definition' z in the 'text' y.

6 Conclusion

In this paper we have proven all development closed PRSs to be confluent, thereby obtaining a critical pair based confluence result, for non-terminating higher-order term rewriting systems.

The idea of our confluence proof for development closed rewriting system is the same as in [Hue80, Toy88], but for the refined measure function. The reason for this adaptation is that the confluence results obtained in those papers were based on the diamond property for 'parallel rewriting'. Since parallel rewriting doesn't satisfy the diamond property for higher-order rewriting systems, not even for such simple ones as the lambda calculus, it has to be replaced by the notion of 'development rewriting'. Trying this back in 1992, I only observed that the measure function used by Huet didn't work. It took two years before looking at it again and realising that a straightforward adaptation of the measure function did the trick.

The proof is modular in the following sense. The basis of the result is formed by a more or less abstract theory of independence of redexes as found at many places ([CR36, Klo80, HL, Kha92, Hue93, Oos94, Mel95]) and briefly recapitulated here. For orthogonal systems this immediately yields confluence. The development closed condition requires on top of that also a term structure of the objects of the rewriting system.

The known relaxation of orthogonality to weak orthogonality (having only trivial critical pairs, [ORb]) puts weaker demands on the structure of objects than the development closed condition does and in particular it doesn't require them to be trees. So, for term rewriting, development closedness is better, but for graph rewriting our proof fails and weak orthogonality is the strongest result available at this moment. Of course, this should be formalised in some formal system, and we believe that the framework as introduced in [Oos94, ORb, Raa96] is appropriate for that purpose.

One can also choose not to work with a notion of descendants at all, and to prove the diamond property directly for an inductively defined notion of complete development, by induction over that definition. This method (and variations) one can find described in [Acz78, Raa93, Tak, Nip, MN94, Raa96] and in many ad hoc confluence proofs in literature as well.

The advantages of keeping the descendants around is that it yields such nice byproducts as the theory of permutation equivalence. The advantage of not using descendants is that it yields short confluence proofs. A paper displaying both for weakly orthogonal rewriting system in a general setting is [ORb].

Some random questions remaining are:

1. Can the Tait & Martin-Löf method be employed to yield our main results? This is a methodological question (there's no sense an sich in reproving known results by known methods).
2. Do the results carry over to the term graph rewriting world? From the abstract nature of the proof I expect the answer to be affirmative. The problem is more that existing notions of term graph rewriting and concepts such as 'critical pairs' for these are rather ad hoc.
3. This brings us immediately to an important issue: unlike the theory of independence there's not much been done for dependence. We expect that our setting is convenient for formulating such a theory.
4. Can the requirement for development closedness that for each critical pair (s,t), t can be reached from s by a *complete* development of some set of re-

dexes can be relaxed by omitting the completeness requirement?[8] I would say yes, but could imagine an out-of-sync phenomenon as in the counterexample against confluence of non-ambiguous TRSs as well.

5. There's no reason why one should restrict attention to critical *pairs*. It seems worthwhile to study such intuitive notions as critical *clusters*.

Acknowledgements I thank Femke van Raamsdonk for collaboration on the general setting, Mizuhito Ogawa, Aart Middeldorp, Jan Willem Klop and Fer-Jan de Vries for feedback, Paul Taylor for his diagrams and Hideki Isozaki for colour-support. Finally, thanks to the referees for useful comments.

References

[Acz78] Peter Aczel. A general Church-Rosser theorem. Technical report, University of Manchester, July 1978.

[AGM92] S. Abramsky, Dov M. Gabbay, and T. S. E. Maibaum, editors. *Handbook of Logic in Computer Science*, volume 2, Background: Computational Structures. Oxford University Press, New York, 1992.

[Bar84] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Publishers B.V., Amsterdam, revised edition, 1984.

[BG93] M. Bezem and J. F. Groote, editors. *Proceedings of the International Conference on Typed Lambda Calculi and Applications, TLCA'93, March 1993, Utrecht, The Netherlands*, volume 664 of *Lecture Notes in Computer Science*, Berlin Heidelberg, 1993. Springer-Verlag.

[CR36] Alonzo Church and J. B. Rosser. Some properties of conversion. *Transactions of the American Mathematical Society*, 39:472–482, January to June 1936.

[DJ] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In [Lee90, Ch. 6 pp. 243–320].

[GLM] Georges Gonthier, Jean-Jacques Lévy, and Paul-André Melliès. An abstract standardisation theorem. pp. 72–81 in LICS'92.

[HL] Gérard Huet and Jean-Jacques Lévy. Computations in orthogonal rewriting systems. Chch. 11,12 in [LP91].

[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.

[Hue93] Gérard Huet. Residual theory in λ-calculus : A formal development. Rapport de Recherche 2009, INRIA, Août 1993.

[Kha90] Z. O. Khasidashvili. Expression reduction systems. In *Proceedings of I. Vekua Institute of Applied Mathematics*, volume 36, pages 200–220, Tbilisi, 1990.

[Kha92] Zurab Khasidashvili. The Church-Rosser theorem in orthogonal combinatory reduction systems. Rapports de Recherche 1825, INRIA-Rocquencourt, December 1992.

[8] Question asked by Aart Middeldorp.

[Kir93] Claude Kirchner, editor. *Rewriting Techniques and Applications, 5th International Conference, RTA-93, Montreal, Canada, June 16–18, 1993*, volume 690 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Heidelberg, 1993.

[Klo] J. W. Klop. Term rewriting systems. In [AGM92, pp. 1–116].

[Klo80] J. W. Klop. *Combinatory Reduction Systems*. PhD thesis, Rijksuniversiteit Utrecht, June 1980. Mathematical Centre Tracts 127.

[Lee90] Jan van Leeuwen, editor. *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics. Elsevier Science Publishers B.V., Amsterdam, 1990.

[LP91] Jean-Louis Lassez and Gordon Plotkin, editors. *Computational Logic: Essays in Honor of Alan Robinson*. The MIT Press, Cambridge, Massachusetts, 1991.

[Mel95] Paul-André Melliès. *Description Abstraite des Systèmes de Réécriture*. Thèse de doctorat, Université Paris VII, Preliminary version of September 28, 1995.

[Mil91] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. in [SH91, pp. 253–281], 1991.

[MN94] Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. On `ftp.informatik.tu-muenchen.de` as `local/lehrstuhl/nipkow/hrs.dvi.gz`, November 1994. To appear in TCS.

[Nip] Tobias Nipkow. Orthogonal higher-order rewrite systems are confluent. In [BG93, pp. 306–317].

[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`.

[ORa] Vincent van Oostrom and Femke van Raamsdonk. Comparing combinatory reduction systems and higher-order rewrite systems. pp. 276–304 in HOA'93, LNCS 816.

[ORb] Vincent van Oostrom and Femke van Raamsdonk. Weak orthogonality implies confluence: the higher-order case. pp. 379–392 in LFCS'94, LNCS 813.

[Pfe] Frank Pfenning. Unification and anti-unification in the calculus of constructions. pp. 74–85 in LICS'91.

[Pre95] Christian Prehofer. *Solving Higher-Order Equations: From Logic to Programming*. PhD thesis, Technische Universität München, February 1995. Report TUM-19508.

[Raa93] Femke van Raamsdonk. Confluence and superdevelopments. in [Kir93, pp. 168–182], 1993.

[Raa96] Femke van Raamsdonk. *Confluence and Normalisation for Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, 1996. Preliminary version of 19 January 1996.

[SH91] P. Schroeder-Heister, editor. *Extensions of Logic Programming, International Workshop, Tübingen, FRG, December 8–10, 1989*, volume 475 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1991.

[Tak] Masako Takahashi. λ-calculi with conditional rules. In [BG93, pp. 406–417].

[Toy88] Yoshihito Toyama. Commutativity of term rewriting systems. In F. Fuchi and L. Kott, editors, *Programming of Future Generation Computer*, volume II, pages 393–407. North-Holland, 1988.

This article was processed using the LaTeX macro package with LLNCS style