Finite Family Developments

Regular Paper

Vincent van Oostrom
Vrije Universiteit, Department of Mathematics and Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam, Netherlands
`http://www.cs.vu.nl/~oostrom`
`tel: +31 20 4447735`
`fax: +31 20 4447653`

Abstract

Consider Adam and Eve. Count generations starting from them. Supposing that there will always be people, then it's true that for any generation X, eventually there will be people belonging to the next generation $X+1$.

In this paper the same result is established for the class of higher order pattern rewriting systems.

1 Introduction

Consider a set of structures and a set of transformations on them specifying how a structure may be transformed into another one. Suppose the transformations are of the following form: first a structure is decomposed into substructures, next some substructure is replaced by another one, and finally the substructures are composed into a structure again.

(destroy) The parts of the initial structure eliminated in the course of the transformation (i.e. the parts of the replaced substructure as well as the parts eliminated in the initial decomposition) can be thought of as being *destroyed*.

(create) The parts of the final structure introduced in the course of the transformation (i.e. the parts of the replacing substructure as well as the parts introduced in the final composition) can be thought of as being *created*.

(context) The other parts (i.e. belonging to the other substructures) are not affected by the transformation (although they might be erased or duplicated!).

Next, suppose a sequence of transformations starting from some initial structure is given and one is interested in to which generation (a part of) the final structure belongs. To that end each part of a structure is equiped with a natural number indicating to which generation it belongs.

(init) In the initial structure of the transformation sequence all parts belong to generation 0 (so Adam an Eve belong to generation 0).

(step) The parts created by a transformation step belong to the generation succeeding the maximum generation of the destroyed parts. The other parts are not affected.

Now *Finite Family Developments* (FFD) states:

> *If a transformation sequence is unbounded in length, then the generation the parts occurring in the sequence belong to, is unbounded.*

In this paper it is shown that FFD holds for *higher order pattern rewriting systems* (PRSs, [Nip, MN94]), a class of term rewriting systems comprising both first order *term rewriting systems* (TRSs, cf. [Klo]) and *the lambda calculus* (cf. [Bar84]). The outline of the paper is as follows.

In Section 2 higher order pattern rewriting systems are recapitulated.

In Section 3 a simpler version of FFD is established first: *Finite Developments* (FD), roughly expressing that if the generation of the parts occurring in a transformation sequence is not larger than 1, the sequence is finite. We give two proofs. In Subsection 3.1 a proofmethod originally employed by Tait to prove termination of the simply typed lambda calculus [Tai67] is adapted yielding FD *sec*. Its merit is that specialising it to TRSs and the lambda calculus yields the simplest proofs of FD we know of. In Subsection 3.2 a generalisation of a proof due to De Vrijer [Vri85] for the λ-calculus is presented, not only showing finiteness but also providing an upperbound on the length of any development starting from a given term.

In Section 4 the finite family developments theorem is proven for higher order pattern rewriting systems by generalising a proof of FFD for the lambda calculus due to Van Daalen (cf. [Bar84, Thm. 14.1.11], [Lév78, Prop. II.1.5.6], [NGV94, Part C.5]).

Some applications of FFD are discussed in Section 5 and we conclude with mentioning some possible directions for future research in Section 6.

Let us mention that FFD has been obtained for various classes of rewriting systems already (apart from the lambda calculus also for orthogonal combinatory reduction systems [Klo80, Cor. II.6.2.7.15], first order term rewriting systems [Mar92, Prop. 3.3.11], combinatory reduction systems [Melon, Conj. 1]), which we will comment on later. However we think it is worthwhile to have simple and complete proofs of theorems for the concrete systems one is interested in, i.e. for PRSs in our case.

2 Higher-Order Rewriting

Higher-order term rewriting systems are a generalisation of both first order term rewriting systems (cf. [Klo]) and the lambda calculus (cf. [Bar84]). A basic understanding of both of these classes and of abstract rewriting systems (cf. [Klo, Sec. 2]) is assumed.

In this paper we will employ higher order pattern rewriting systems (PRSs) as introduced in [Nip] (there called HRSs). Rewriting in PRSs is rewriting modulo the simply typed lambda calculus so could also be called *simply typed term rewriting*.

Definition 2.1 *An object of an abstract rewriting system is* terminating *if no infinite rewrite sequence is possible starting from that object. An abstract rewriting system is terminating if all its objects are.*

A *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. We first define the objects of a PRS.

Definition 2.2 (Alphabet) Simple types *(θ, ι, κ) are defined by the grammar: $\theta ::= b \mid \theta \to \theta$, for some set ($b \in$) B of* base *types, e.g. booleans or natural numbers. A* rewrite alphabet *is a set denoted by ($\mathtt{f}$, $\mathtt{g}$, $\mathtt{h} \in$) $\mathcal{A}$, the elements of which are called* function *symbols such that to each function symbol $\mathtt{f}$ a unique simple type $\theta_{\mathtt{f}}$ is associated.* Preterms *are objects M such that $M{:}\theta$ for some simple type θ, can be inferred from*

(sym) $\mathtt{f}{:}\theta_{\mathtt{f}}$, *for* $\mathtt{f} \in \mathcal{A}$,

(var) $x^\theta{:}\theta$ *for termvariables* x,

(app) $M{:}\theta \to \iota,\ N{:}\theta \Longrightarrow M(N){:}\iota$.

(abs) $M{:}\iota \Longrightarrow x^\theta.M$[1]$:\theta \to \iota$.

[1] We omit the usual λ in abstractions.

We use x, X, y, Y, z, Z, ... to range over termvariables. Functional notation $\mathtt{f}(M_1,\ldots,M_m)$ is employed instead of applicative notation $\mathtt{f}(M_1)\ldots(M_m)$ in case $m > 1$ (cf. [Nip]). Higher-order terms *are obtained from the preterms by quotienting by the theory consisting of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ (that is, the theory $\boldsymbol{\lambda\eta}$ 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:*

$$\begin{array}{rcl} (x.M)(N) & \to_\beta & M^{[x\mapsto N]} \\ M & \to_{\overline{\eta}} & x.M(x) \end{array}$$

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

When we write preterms we always mean their representative, i.e. their $\beta\overline{\eta}$-normal form.

We employ common shorthand notations for types (cf. [Bar92, Sec. 3]) and (pre)terms (cf. [Bar84, Sec. 2.1]). When a type is of the form $\vec{\theta} \to b$, $\vec{\theta}$ is called the *factor* and the length of that vector the *arity* of the type. The notions of free (ranged over by X, Y, Z) and bound (ranged over by x, y, z) variables are the usual ones from the lambda calculus. The terms, i.e. the representatives of the preterms, can be characterized in the following inductive way: A term is of the form

$$\vec{X}.\mathsf{a}(\vec{M})$$

where the *matrix* $\mathsf{a}(\vec{M})$ is of base type, and the *arguments* $\vec{M}$ are terms again. Following [Wol93, Not. 2.29], $\vec{X}$ and a (which is either a free or bound variable or a function symbol) will be called the *binder* and *head*, respectively.

The rewrite rules of a PRS is a set of pairs of terms satisfying the following constraints.

Definition 2.3 (Rule) *A* rewrite rule *is a pair $\vec{X}.l \to \vec{X}.r$ of closed terms of the same type $\vec{\theta} \to b$ such that*

1. *$\vec{X}{:}\vec{\theta}$ is a sequence of termvariables,*
2. *the first or* head-*symbol of l is a function symbol, i.e. l is of the form $\mathtt{h}(\vec{M})$, for some function symbol $\mathtt{h}$,*
3. *l is a* pattern *[Mil91], i.e. every occurrence of an X_k among $\vec{X}$ in l has only (representatives of) pairwise distinct variables not among $\vec{X}$ as arguments. The pattern is* linear *when each variable among $\vec{X}$ occurs exactly once in l.*

In general, we will call a term l satisfying those conditions a pattern and we use $l \to r$ as a shorthand notation for the rule.

Notions for types extend to terms and rewrite rules via their type.

Definition 2.4 (Rewrite relation) *The* rewrite relation $\to_{\mathcal{P}}$ *on the set of terms of a PRS $\mathcal{P}$ is generated from the rules via the following inference system.*

$$\frac{l \to r}{\vec{X}.l(\vec{M}) \to_{\mathcal{P}} \vec{X}.r(\vec{M})}\,(\mathsf{sub}) \qquad \frac{l \to_{\mathcal{P}} r}{\vec{X}.\mathsf{a}(M_1,\ldots,l,\ldots,M_m) \to_{\mathcal{P}} \vec{X}.\mathsf{a}(M_1,\ldots,r,\ldots,M_m)}\,(\mathsf{ctx})$$

Notions for abstract rewriting systems extend to $\mathcal{P}$, via its rewrite relation $\to_{\mathcal{P}}$.

So an inference starts with the substitution rule (sub) followed by an arbitrary number of context rules (ctx). Observe that an explicit notion of 'substitution' can be dispensed with in principle in higher order rewriting as shown by the (sub)-inference rule. Nonetheless we introduce it for convenience.

Definition 2.5 *A* substitution σ *is the extension to terms of a mapping from variables to terms, defined in the following way:*

$$M^\sigma =^{\text{def}} (\vec{X}.M)(\sigma(\vec{X}))$$

where $\vec{X}$ *is the vector of variables in the domain of* σ.

Note that the real substitution takes place implicitly in taking the representative of the preterm. A direct inductive definition of substitution is of course possible as well by performing β-reduction on the fly (see [Kah93, Def. 13]). Terminology for terms extends to substitutions via their range, e.g. a substitution is terminating if all terms in its range are so. Using substitution the (sub) rule can be written in the following more familiar way

$$\frac{\vec{X}.l \to \vec{X}.r}{\vec{Y}.l^\sigma \to_{\mathcal{P}} \vec{Y}.r^\sigma}\ (\mathsf{sub})$$

A fundamental fact for higher order rewriting is that it is compatible with preterm formation. Specifically, if $M \twoheadrightarrow_{\mathcal{P}} N$ and $M' \twoheadrightarrow_{\mathcal{P}} N'$, then $M(M') \twoheadrightarrow_{\mathcal{P}} N(N')$. In view of the above, this is equivalent to the so-called substitution lemma. We write $\sigma \twoheadrightarrow_{\mathcal{P}} \tau$ ($\sigma \to_{\mathcal{P}} \tau$) when $\sigma(X) \twoheadrightarrow_{\mathcal{P}} \tau(X)$ ($\sigma(X) \to_{\mathcal{P}} \tau(X)$) for all (exactly one) variables in the common domain (and the identity on the other variables).

Lemma 2.6 (Substitution) *Let* $\mathcal{P}$ *be a PRS.*

1. *If* $M \twoheadrightarrow_{\mathcal{P}} N$ *and* $\sigma \twoheadrightarrow_{\mathcal{P}} \tau$*, then* $M^\sigma \twoheadrightarrow_{\mathcal{P}} N^\tau$.
2. *If* P *is a linear pattern and* $\sigma \to_{\mathcal{P}} \tau$*, then* $P^\sigma \to_{\mathcal{P}} P^\tau$.

Proof 1. This is (a special case of) [MN94, Thm. 3.9].

2. By induction on the pattern P. □

Note that the second item fails in general for non-linear patterns or for terms which are not patterns. *In the sequel we assume all patterns to be linear, hence all rewrite systems to be left-linear.* This entails no loss in generality, since a rewrite sequence in a non-left-linear system can be simulated in its 'linearised' version, and (F)FD of the latter implies (F)FD for the former.

Definition 2.7 *A term is a* direct subterm *of another, in either of two cases.*

1. *If the former is* M *and the latter is* $X.M$*, or*
2. *if the former is* $\vec{X}.M_i$*,* M_i *is among* $\vec{M}$ *and the latter is* $\vec{X}.\mathsf{a}(\vec{M})$.

The proper subterm *relation* $\lhd$ *is the transitive closure of the direct subterm relation.*

Note that the proper subterm relation is well-founded.

Lemma 2.8 *Let* $\mathcal{P}$ *be a PRS, then* $\to_{\mathcal{P}}$ *is terminating iff* $\to_{\mathcal{P}} \cup \rhd$ *is terminating.*

Proof The if direction is trivial. For the only if direction one first observes that since $\rhd$ is terminating, an infinite sequence contains infinitely many $\to_{\mathcal{P}}$-steps. Now it suffices to show that $\rhd$ can always be postponed, i.e. if $M \rhd M' \to_{\mathcal{P}} N'$, then there exists a term N' such that $M \to_{\mathcal{P}} N \rhd N'$, since this would allow for the construction of an infinite $\to_{\mathcal{P}}$-sequence contradicting termination. This is shown by induction on the number of 'direct subterm' steps in $M \rhd M'$, using $M \to_{\mathcal{P}} N$ iff $X.M \to_{\mathcal{P}} X.N$. □

The well-founded order $\ll =^{\text{def}} (\leftarrow_{\mathcal{P}} \cup \lhd)^+$ will be used as an induction measure in the proof of FFD.

Lemma 2.9 *If* $M = P^\sigma$ *for a pattern* P*, then* $\sigma(X) \lhd M$ *for all* X *in* P.

Proof By induction on P. □

Characterising termination via right-hand sides In this paragraph termination of a PRS is characterised by termination of terminating instances of its right-hand sides.

Lemma 2.10 (RHS) *A PRS is terminating iff r^σ is terminating, for every right-hand side $\vec{X}.r$ and terminating substitution σ.*

Proof ($\Rightarrow$) Direct from the definition, using the fact that terms are closed under substitution.

($\Leftarrow$) Consider a term $\vec{X}.\mathtt{f}(\vec{M})$ of minimal size, allowing an infinite rewrite sequence. By the minimality assumption we may assume that $\vec{X}$ is empty and that the rewrite sequence is of the form $\mathtt{f}(\vec{M}) \twoheadrightarrow \mathtt{f}(\vec{M}') = l^\sigma \rightarrow r^\sigma \rightarrow \ldots$. By definition l has a function symbol as head, hence by Lemma 2.9 σ assigns subterms of terms reachable from terms among $\vec{M}$ to variables, so σ itself is terminating. Hence, r^σ is terminating by assumption. □

Without the termination condition on the substitution (which we make essential use of in our applications), this lemma is well-known (cf. [BKM89, Prop. 4.1.2]). Note that essential use is made of Lemma 2.9, i.e. of the pattern structure of left-hand sides in PRSs. Lemma 2.10 is false otherwise, as witnessed by e.g. the rule $\mathtt{f}(X(Y)) \rightarrow Y$. If σ is terminating, $\sigma(Y)$ is terminating by definition. Nevertheless, the PRS is not terminating as witnessed by the rewrite step $\mathtt{f}(\mathtt{a}) \rightarrow \mathtt{f}(\mathtt{a})$ via the substitution $\sigma(X) = x.\mathtt{a}$, $\sigma(Y) = \mathtt{f}(\mathtt{a})$. The problem is that to Y not a subterm of $\mathtt{a}$ is assigned.

Term Rewriting Systems as PRSs To every first order term rewriting system (see e.g. [Klo]) one can associate a PRS via so-called currying, i.e. by associating to every m-ary function symbol $\mathtt{f}$ of the TRS, a function symbol $\mathtt{f}\colon \underbrace{o \rightarrow \cdots \rightarrow o}_{m} \rightarrow o$ and translating terms and rules in the obvious homomorphic way (see e.g. [Oos94, Sec. 3.3.1]). In fact, for rules we take the *closure* of the translated rule, i.e. if $\vec{X}$ is the (non-repetative) list of free variables from left to right in the result $l \rightarrow r$ of the translation, then the resulting PRS rule will be $\vec{X}.l \rightarrow \vec{X}.r$. Hence the arity of the PRS rule is just the number of free variables in the original TRS rule In the resulting PRS we are of course only interested in terms which can be obtained from translating TRS terms.

Example 2.11 *The prime example of a TRS is Curry's Combinatory Logic ([Cur30, p. 513],cf. [Klo, p. 16]). Its rendering as a PRS is as follows.*

1. *Its alphabet consists of* $\mathtt{S}, \mathtt{K}, \mathtt{I}\colon o$ *and* $@\colon o \rightarrow o \rightarrow o$.
2. *It has three rewrite rules:*

$$\begin{array}{rcl} @(\mathtt{I}, X) & \rightarrow & X \\ @(@(\mathtt{K}, X), Y) & \rightarrow & X \\ @(@(@(\mathtt{S}, X), Y), Z) & \rightarrow & @(@(X, Z), @(Y, Z)) \end{array}$$

The Lambda Calculus as a PRS The lambda calculus (cf. [Bar84]) can be rendered as a PRS as follows (see e.g. [Nip, Sec. 5.1])

1. Its alphabet consists of $\lambda\colon (o \rightarrow o) \rightarrow o$ and $@\colon o \rightarrow o \rightarrow o$.
2. The translation of the β-rule is: $@(\lambda(X), Y) \rightarrow X(Y)$.

Again, in the PRS attention is restricted to proper lambda calculus terms. E.g. $x.@(x, x)$ is not a proper term, but $\lambda(x.@(x, x))$ is.

Combinatory Reduction Systems as a PRS The class of combinatory reduction systems (CRSs, [Klo80, KOR93]) was the first class of 'general purpose' higher order rewriting systems, encompassing both first order term rewriting systems and the lambda calculus. CRSs roughly correspond to second order PRSs (cf. [Kah93]). A translation from combinatory reduction systems into PRSs was presented in [OR]. Because of lack of space this (natural) embedding is not presented here, but we only remark that via it the results presented here for PRSs immediately specialise to CRSs.

3 Finiteness of Developments

Consider a term and decompose it into left-hand sides of rules and other parts. The decomposition can be indicated by marking the left-hand sides with the rule to be applied. Then *Finiteness of Developments* (FD) expresses that every rewrite sequence contracting marked redexes, will always terminate. Such a sequence is *complete* if it ends in a term without markings. A *(complete) development* is the projection of a marked rewrite sequence by forgetting the marking (see Definition 3.5).

Example 3.1 *Consider the TRS having (infinitely many) rules*

$$\begin{array}{rcl} \mathtt{a} & \to_1 & \mathtt{b} \\ \mathtt{f}(X) & \to_2 & \mathtt{g}(X,X) \\ \mathtt{f}(X) & \to_3 & \mathtt{g}(X,\mathtt{g}(X,X)) \\ \mathtt{f}(X) & \to_4 & \mathtt{g}(X,\mathtt{g}(X,\mathtt{g}(X,X))) \\ \text{etc.} & & \end{array}$$

and the term $\mathtt{f}(\mathtt{a})$. *It can be marked as e.g.* $\mathtt{f}(\mathtt{a})$, $\mathtt{f}(\mathtt{a}^1)$, $\mathtt{f}^2(\mathtt{a})$, $\mathtt{f}^2(\mathtt{a}^1)$, $\mathtt{f}^n(\mathtt{a}^1)$, $\ldots$. *Since there are no markings in* $\mathtt{f}(\mathtt{a})$, *only the empty sequence is a marked rewrite sequence. From both* $\mathtt{f}(\mathtt{a}^1)$ *and* $\mathtt{f}^2(\mathtt{a})$ *there's only one complete marked sequence* $\mathtt{f}(\mathtt{a}^1) \to_1 \mathtt{f}(\mathtt{b})$ *and* $\mathtt{f}^2(\mathtt{a}) \to_2 \mathtt{g}(\mathtt{a},\mathtt{a})$. *From* $\mathtt{f}^2(\mathtt{a}^1)$ *there are three complete marked sequences* $\mathtt{f}^2(\mathtt{a}) \to_2 \mathtt{g}(\mathtt{a}^1,\mathtt{a}^1) \to_1 \mathtt{g}(\mathtt{b},\mathtt{a}^1) \to_1 \mathtt{g}(\mathtt{b},\mathtt{b})$, $\mathtt{f}^2(\mathtt{a}^1) \to_2 \mathtt{g}(\mathtt{a}^1,\mathtt{a}^1) \to_1 \mathtt{g}(\mathtt{a}^1,\mathtt{b}) \to_1 \mathtt{g}(\mathtt{b},\mathtt{b})$, *and* $\mathtt{f}^2(\mathtt{a}^1) \to_1 \mathtt{f}^2(\mathtt{b}) \to_2 \mathtt{g}(\mathtt{b},\mathtt{b})$. *Since each marked term is finitely branching, i.e. only finitely many (marked) rewrite steps are possible from it, it follows from FD and Königs Lemma that there exists an upperbound on the length of each marked rewrite sequence. In general, this does not imply the existence of an upperbound on the length of the developments starting from some term, as witnessed for* $\mathtt{f}(\mathtt{a})$ *by the existence of a marked sequence of length* $n+1$ *starting from* $\mathtt{f}^n(\mathtt{a}^1)$, *for every natural number* n. *An upperbound does exist for* non-ambiguous *rewriting systems ([Klo, Def. 3.1.1]) since these are finitely branching (see also Subsection 3.2).*

Note that the marking of the initial term fixes not only the pattern (left-hand side) of the rewrite rule to be applied, but also the rule itself. For example, if we would add a rule $\mathtt{a} \to_{1'} \mathtt{b}$ to the TRS of Example 3.1 then $\mathtt{f}(\mathtt{a}) \to \mathtt{g}(\mathtt{a},\mathtt{a}) \to \mathtt{g}(\mathtt{b},\mathtt{a}) \to \mathtt{g}(\mathtt{b},\mathtt{c})$ is not a development. In contrast, in Finite Family Developments in Section 4 distinct 'residuals' can be contracted by distinct rules.

Finiteness of developments (and strengthened versions of it) have been studied extensively in the literature for various classses of rewriting systems (see e.g. [CR36, Sch65, Hin78, Klo80, Kha92, Raa, Oos94, Melon]) In Subsection 3.1 a simple proof of FD is presented for the class of PRSs. In Subsection 3.2 upperbound information is added to the termination proof of Subsection 3.1, yielding (exact) upperbounds on the lengths of marked rewrite sequences. As a consequence this yields upperbounds on the length of developments in non-ambiguous PRSs.

3.1 FD without bounds

In this section we give a proof of finiteness of developments for the class of PRSs by elementary means. To illustrate its simplicity we present the specialisation of the proof to two important classes of PRSs: first order term rewriting systems and the lambda calculus.

3.1.1 FD for PRSs

Definition 3.2 *The marked version $\underline{\mathcal{P}}$ of a PRS $\mathcal{P}$ is defined as follows.*

1. *The alphabet of $\underline{\mathcal{P}}$ is the alphabet of $\mathcal{P}$ extended with a function symbol $\underline{l \to r}$ of type θ for every rule $l \to r$ of type θ in $\mathcal{P}$.*
2. *For every rule $l \to r$ in $\mathcal{P}$, there's a rule $\underline{l \to r} \to r$ in $\underline{\mathcal{P}}$.*

In order to describe the relationship between a PRS and its marked version a mapping from marked to ordinary terms is defined.

Definition 3.3 *The mapping* lft *from marked preterms to unmarked preterms is the homomorphic extension of the mapping $\underline{l \to r} \mapsto l$.*

Note that lft is the identity on terms not containing marked symbols. The following lemma enables the definition of developments.

Lemma 3.4 *If $M \to_{\underline{\mathcal{P}}} N$, then $\mathsf{lft}(M) \to_{\mathcal{P}} \mathsf{lft}(N)$.*

Proof By induction on the generation of $\to_{\underline{\mathcal{P}}}$ using Lemma 2.6. □

Definition 3.5 *A* development *is the projection $\mathsf{lft}(M_1) \to_{\mathcal{P}} \mathsf{lft}(M_2) \to_{\mathcal{P}} \mathsf{lft}(M_3) \to_{\mathcal{P}} \dots$ of some marked rewrite sequence $M_1 \to_{\underline{\mathcal{P}}} M_2 \to_{\underline{\mathcal{P}}} M_3 \to_{\underline{\mathcal{P}}} \dots$.*

In order to show that all developments are finite, the following strengthening of termination will be employed.

Definition 3.6 *A term $\vec{X}.\mathsf{a}(\vec{M})$ is* terminable, *if $\mathsf{a}(\vec{M})^{\sigma}$ is terminating for terminable σ (on a subset of $\vec{X}$).*

This is well-defined since the types of the terms in the substitution are smaller than the type of the term. Note that terminability implies termination (by taking a substitution with empty domain) and is closed under rewriting.

Theorem 3.7 (FD) *$\underline{\mathcal{P}}$ is terminating for a higher order pattern rewriting system $\mathcal{P}$.*

Proof We prove *every marked term is terminable* by induction, first on the number of marked symbols and then on the number of head symbols in a term. So we need to show termination of $\mathsf{a}(\vec{M})^{\sigma}$ for terminable σ. Without loss of generality we may assume that $\vec{X}$ is the domain of σ since variables are terminable. By induction hypothesis we know that $\vec{X}.M_i$ is terminable for every M_i among $\vec{M}$. We make a case distinction on a.

- (var) If a is a variable X_i among $\vec{X}$ and $\sigma(X_i) =^{\text{def}} \vec{Y}.\mathsf{b}(\vec{N})$, then $\mathsf{a}(\vec{M})^{\sigma} = \sigma(X_i)(\vec{M}^{\sigma}) = \mathsf{b}(\vec{N})^{\sigma[\vec{Y} \mapsto \vec{M}^{\sigma}]}$, which is terminating by induction hypothesis.
- (fun) If a is a function symbol $\mathtt{f}$, then termination follows from the induction hypothesis on the arguments.
- (rule) In a is a marked symbol $\underline{l \to r}$ for some rule $l \to r$ of $\mathcal{P}$, then since the arguments are terminating by the induction hypothesis, an infinite rewrite sequence must look like

$$\underline{l \to r}(\vec{M}^{\sigma}) \twoheadrightarrow \underline{l \to r}(\vec{M}') \to r(\vec{M}') \to \dots$$

  But since r contains no marked symbols, and $\vec{M}'$ is terminable by the induction hypothesis and closure of terminability under rewriting, if $r =^{\text{def}} \vec{Y}.\mathsf{b}(\vec{N})$, then $r(\vec{M}') = \mathsf{b}(\vec{N})^{[\vec{Y} \mapsto \vec{M}']}$ which is terminable by the induction hypothesis, hence terminating. □

Not so many proofs of FD are known for classes of higher order rewriting systems. In the seminal thesis [Klo80], a first such proof[2] was presented for the class of combinatory reduction systems. Later proofs can be found in [Kha92, Oos94, Melon]. The proof in [Kha92] is for the class of expression reduction systems, roughly equivalent to the class of combinatory reduction systems (cf. [Raa96, Sec. 4.6]). The proof in [Oos94] is more abstract in the sense that it's based on a number of conditions which were verfied there to hold for PRSs. The axiomatic proof in [Melon] is probably the most abstract. However, the axioms were verfied to hold there only for the class of combinatory reduction systems.

Remark 3.8 *It might quite well be the case that Theorem 3.7 is a consequence of termination of rewriting systems satisfying the so-called* general schema *[JO], however the situtation is not clear to us yet.*

Remark 3.9 *In many of the publications above stronger versions of FD are proven. For example, that all complete marked rewrite sequences end in the same term. Formally, define the mapping* rgt *from marked preterms to unmarked preterms as the homomorphic extension of the mapping* $\underline{l \to r} \mapsto r$.

Lemma 3.10 *If M is a marked term, then*

1. $M \twoheadrightarrow_{\underline{\mathcal{P}}} \mathsf{rgt}(M)$*, and*
2. *if* $M \to_{\underline{\mathcal{P}}} N$*, then* $N \twoheadrightarrow_{\underline{\mathcal{P}}} \mathsf{rgt}(M)$.

Proof 1. *By induction on M.*

2. *By induction on the derivation of* $M \to_{\underline{\mathcal{P}}} N$*, using the first item and Lemma 2.6.* □

This corresponds to [Tak, Lem. 3.2] and [Raa96, Lem. 5.3.3]. We claim that other strengthenings in literature can be obtained without too much effort as well.

3.1.2 FD for TRSs

In this subsubsection we give a short proof of FD for first order term rewriting systems ([Klo]). See the previous section on how this translates to PRS notation.

Theorem 3.11 $\underline{\mathcal{R}}$ *is terminating for a first order term rewriting system* $\mathcal{R}$.

Proof By Lemma 2.10 it suffices to prove termination of terminating instances of right-hand sides of $\underline{\mathcal{R}}$. Remark that these right-hand sides contain no marked symbols by definition, hence the theorem follows from showing (by a trivial induction on M) that *for any term M not contaning marked symbols, and any terminating substitution σ, M^σ is terminating (in $\underline{\mathcal{R}}$!).* □

Of course, this theorem can be proven easily (for TRSs) by an appeal to recursive path orders (cf. [Klo, Sec. 2.3]). Ordering markers above other function symbols yields that left-hand sides are larger than right-hand sides in the RPO-ordering from which termination of the marked TRS $\underline{\mathcal{R}}$ follows.

3.1.3 FD for the lambda calculus

Although FD for the lambda calculus has been studied quite extensively, specialisation of the proof of FD for PRSs to it leads to the shortest proof of FD we are aware of. The proof and its relationship to existing proofs can be found in Appendix A.

[2] Although the proofmethod in [Klo80, pp. 141–163] is perfectly sound, the proof contains some inaccuracies.

3.2 FD with (exact) bounds

It has been observed at several places in the literature that termination proofs without bounds can be enriched with upperbound information (cf. [dV87, Pol96, Loa95]) (or the other way around; termination proofs *sec* are obtained by just omitting the upperbound information). The same holds true for finiteness of developments. Since the enrichment usually is rather tedious and takes up much space, we only perform this task for the class of combinatory reduction systems ([KOR93]).

We explain our enrichment of an FD proof for CRSs, which is essentially a representation of the ideas in [Vri85], by means of the lambda calculus. We employ CRS notation as in [KOR93]. Suppose we want to define an upperbound on the length of any marked rewrite sequence starting from some term, by induction on the structure of the term. Consider the $\underline{\beta}$-rewrite rule $\underline{\beta}(x.M,N) \to M^{[x \mapsto N]}$. The following two observations can be made about it. First, note that although N can be copied many times when substituted for x in M by the $\underline{\beta}$-rule, this number is bounded by a (*multiplicity*) function of M which is invariant with respect to substitution for variables other than x in M. Second, suppose we want to define an upperbound on the length of any marked rewrite sequence starting from some term, by induction on the structure of the term. An obvious idea is then to take one plus the upperbound on $M^{[x \mapsto N]}$ as upperbound on $\underline{\beta}(x.M,N)$. However, in order to make the computed value monotonous in both M and N, care has to be taken of the situation that N is *erased*, i.e. in case x does not occur in M. Erasure could be avoided by transforming each term $\underline{\beta}(x.M,N)$ into e.g. $\underline{\beta}(x.a(x,M),N)$, for some constant a (cf. [CF58, p. 130, footnote 10], and [Klo80, Intuition II.4.3]). Instead of such a syntactic transformation, monotonicity is enforced here by semantical brute force: the value of the left-hand side after removing the marking is added in (that is, for the lambda calculus a summand of the form $[\![(\lambda x.M)N]\!]$). The price to be paid for the brute force method is that only an upper bound, not an exact upperbound on the length of any development is obtained. The gain lies in the smoother induction proofs.

The *value* $[\![M]\!]$ of a marked term M is a *height*, *multiplicity* pair $([\![M]\!]_h, [\![M]\!]_m)$, where the former is a natural number, and the latter is a mapping from first order variables to natural numbers.

Definition 3.12 *The* value $[\![M]\!]$ *for* M *a marked term, and* $[\![\sigma,N]\!]$ *for* σ *a marked term substitution and* N *an unmarked term are defined by simultaneous induction.*

(var0) $[\![x]\!] = (0,m)$*, where* $m(x) = 1$ *and* $m(y) = 0$ *for all* $y \neq x$*,*

(fun) $[\![\mathtt{f}(M_1,\dots,M_n)]\!] = \sum_{1\leq i\leq n} [\![M_i]\!]$*,*

(abs) $[\![[x]M]\!] = [\![M]\!]_{-x}$ *where* $[\![M]\!]_{-x}$ *denotes the same pair as* $[\![M]\!]$*, except that the multiplicity of* x *is set to* 0*.*

(rule) $[\![\underline{l \to r}(\vec{Z})^{\sigma}]\!] = (1,0) + [\![\sigma,l]\!] + [\![\sigma,r]\!]$*, where* 0 *denotes the constant* 0 *function.*

and

(var0) $[\![\sigma,x]\!] = (0,m)$*, where* $m(x) = 1$ *and* $m(y) = 0$ *for all* $y \neq x$*,*

(var1) $[\![\sigma,Z(N_1,\dots,N_n)]\!] = [\![Z^{\sigma}]\!]_{-x_1\dots x_n} + \sum_{1\leq i\leq n} [\![Z^{\sigma}]\!]_m(x_i) \cdot [\![\sigma,N_i]\!]$*,*

(fun) $[\![\sigma,\mathtt{f}(N_1,\dots,N_n)]\!] = \sum_{1\leq i\leq n} [\![\sigma,N_i]\!]$*, and*

(abs) $[\![\sigma,[x]N]\!] = ([\![\sigma,N]\!])_{-x}$*.*

Values are ordered by height.

These definitions of value mutually depend on each other via the clauses (rule) and (var1). Well-definedness is checked by measuring a marked term M by (m,n), where m is the number of marked symbols and n the number of symbols of M, and by measuring a marked term substitution, metaterm pair σ, N by taking the sum of the measures of N and the marked terms Z^{σ}, for $Z \in N$.

Lemma 3.13 *Let M, M_1, …, M_n be marked terms, let σ be a marked term substitution, and let N be an unmarked term.*

1. *If $y \neq x_i$ and $y \notin M_i$, for $1 \leq i \leq n$, then*

   (marked) $[\![M]\!]_m(y) = [\![M^{[\vec{x} \mapsto \vec{M}]}]\!]_m(y)$,

   (unmarked) $[\![\sigma, N]\!]_m(y) = [\![\sigma^{[\vec{x} \mapsto \vec{M}]}, N]\!]_m(y)$.

2. (marked) $[\![M^{[\vec{x} \mapsto \vec{M}]}]\!] = [\![M]\!]_{-x_1 \ldots x_n} + \sum_{1 \leq i \leq n} [\![M]\!]_m(x_i) \cdot [\![M_i]\!]$,

   (unmarked) $[\![\sigma^{[\vec{x} \mapsto \vec{M}]}, N]\!] = [\![\sigma, N]\!]_{-x_1 \ldots x_n} + \sum_{1 \leq i \leq n} [\![\sigma, N]\!]_m(x_i) \cdot [\![M_i]\!]$,

3. $[\![N^\sigma]\!] = [\![\sigma, N]\!]$.

Proof 1. By simultaneous induction.

2. By simultaneous induction using the previous item.

3. By induction on N, using the previous item. □

Theorem 3.14 *For all marked terms M, N, if $M \rightarrow_{\underline{\mathcal{P}}} N$, then $[\![M]\!] > [\![N]\!]$.*

Proof By induction on the generation of $M \rightarrow N$.

(sub) $[\![\underline{l \rightarrow r}(\vec{Z})^\sigma]\!] = (1,0) + [\![\sigma, l]\!] + [\![\sigma, r]\!] > [\![\sigma, r]\!] = [\![r^\sigma]\!]$, by definition of $[\![\_]\!]$, of $>$ and Lemma 3.13,

(ctx) We consider cases depending on the head symbol.

- (fun) by induction hypothesis and monotonicity of $\sum$ in each summand,
- (abs) by induction hypothesis, and
- (rule) by induction hypothesis and monotonicity of $[\![\_]\!]$ in each metavariable, which holds true due to the fact that we roughly have that $[\![\sigma, l]\!] = \sum_{Z \in l} [\![\sigma, Z]\!]$ and $\sum$ is monotonous in each summand. □

Since the height decreases in every rewrite step by Theorem 3.14, $[\![M]\!]$ is in fact an upperbound on the length of any rewrite sequence starting from a marked term M. In Appendix B, we sketch how the definition of value can be adapted so as to yield exact upperbounds.

4 Finiteness of Family Developments

In the introduction we have given an informal account of finiteness of family developments. In this paper, we have chosen to put the labels recording the generation a part belongs to 'on the edges' (in the graph representation of terms).

Definition 4.1 *Let (m, $n \in$) $\mathbb{N}$ be a set of* labels *ordered by $<$ in the usual way. The* ($\mathbb{N}$)-labelled *version $\mathcal{P}\mathbb{N}$ of a PRS $\mathcal{P}$ is defined as follows.*

1. *The alphabet of $\mathcal{P}\mathbb{N}$ is the alphabet of $\mathcal{P}$ extended with for every label $m \in \mathbb{N}$ and every base type b, a function symbol $m_b\colon b \rightarrow b$. Since the types of the labels do not play a rôle in the sequel, we will not write them henceforth.*

2. *The rewrite rules of $\mathcal{P}\mathbb{N}$ are labelled versions of the rules of $\mathcal{P}$ together with rewrite rules managing merger of labels. Their definition employs some auxiliary definitions. Let $m \in \mathbb{N}$ be a label.*

(a) *The m-*pattern P^-_m *of a pattern* $P =^{\text{def}} \vec{X}.\mathtt{a}(\vec{P})$ *in* $\mathcal{P}$ *is* $\vec{X}.\mathtt{a}(\vec{P}^-_{m,\vec{X}})$*, where*

$$\begin{aligned} Q^-_{m,\vec{X}} &=^{\text{def}} \vec{Y}.m(\mathtt{b}(\vec{Q}^-_{m,\vec{X}})), \text{ if } Q =^{\text{def}} \vec{Y}.\mathtt{b}(\vec{Q}) \text{ and } \mathtt{b} \text{ is not among } \vec{X} \\ &=^{\text{def}} Q, \text{ otherwise} \end{aligned}$$

(b) *The m-*term M^+_m *of a term* $M =^{\text{def}} \vec{X}.\mathtt{a}(\vec{M})$ *in* $\mathcal{P}$ *is inductively defined as* $\vec{X}.m(\mathtt{a}(\vec{M}^+_m))$*.*

Finally, we can specify the rewrite rules of $\mathcal{P}\mathbb{N}$.

(a) *For every $\mathcal{P}$-rule $l \to r$, its* labelled versions *are of the form*

$$l^-_{m+1} \to r^+_m$$

for every $m \in \mathbb{N}$.

(b) *For all labels $n < m \in \mathbb{N}$, there are* merger *rules (*contract *and* decrease*)*

$$\begin{aligned} X.n(n(X)) &\to_c X.n(X) \\ X.m(X) &\to_d X.n(X) \end{aligned}$$

Remark 4.2 *Although the idea of labelling is quite simple, its formalisation is not so nice. We could have made the definition more general by basing it on an arbitrary well-founded order instead of on the natural numbers. This would only have cluttered up things more than they are already.*

Note that since only the 'internal' edges of a left-hand side are labelled, all labelled versions of a rule of the form $X.h(\vec{X}) \to \ldots$ would be of that same form. That is, the left-hand side does not contain 'generation' information. However, without loss of generality we may assume that left-hand sides contain at least two function symbols. Without loss of generality, since if h is the head-symbol of a violating rule, one can add a fresh unary symbol v to the alphabet and prefix this to any h occurring in the rules, e.g. for the rule above $X.v(h(\vec{X})) \to \ldots$.

Remark 4.3 *This technical problem is caused by the fact that only edges are labelled. Alternatively, one could decide to label vertices. However, this yields a technical problem as well: in so-called* collapsing *rules of the form $\ldots \to X$ there's no function symbol in the right-hand side to record the 'generation' information. The syntactic solution then is to add a fresh unary symbol e to the alphabet and prefix the right-hand side X with it (and add infinitely many rules taking care of the presence of e's in left-hand sides $\ldots$).*

Neither of these two technical problems is serious enough to seriously consider labelling both vertices and edges.

The following lemma enables the main definition of the paper, that of a family development.

Definition 4.4 *For M a labelled term in $\mathcal{P}\mathbb{N}$, it's* unlabelled version $|M|$ *in $\mathcal{P}$ is defined as the homomorphic extension of the mapping $m \mapsto x.x$, for every $m \in \mathbb{N}$.*

Lemma 4.5 *1. $|P^-_m| = P$ and $|M^+_m| = M$, for every label $m \in \mathbb{N}$, pattern P and term M in $\mathcal{P}$.*

2. If $M \to_{\mathcal{P}\mathbb{N}} N$, then $|M| \to_{\mathcal{P}} |N|$ in case a labelled version of a rule in $\mathcal{P}$ is applied, and $|M| = |N|$ in case a merger rule is applied.

Proof 1. By induction on patterns and terms.

2. By induction on the generation of $\to_{\mathcal{P}\mathbb{N}}$, using the first item in the base case. □

Definition 4.6 *A* family development *is the projection $|M_1| \to^=_{\mathcal{P}} |M_2| \to^=_{\mathcal{P}} |M_3| \to^=_{\mathcal{P}} \ldots$ of some labelled rewrite sequence $M_1 \to_{\mathcal{P}\mathbb{N}} M_2 \to_{\mathcal{P}\mathbb{N}} M_3 \to_{\mathcal{P}\mathbb{N}} \ldots$.*

Since by well-foundedness of $<$ the merger rules on their own are terminating, all family developments being finite is equivalent to all labelled rewrite sequences being finite.

We start the investigation of the correspondence between a PRS and its labelled version with a simple observation due to Lévy (cf. [Bar84, Sec. 14.2]).

Theorem 4.7 (Lift) *Every finite rewrite sequence is a family development.*

Proof See Appendix C. $\square$

The rest of this subsection will be devoted to proving the converse of this theorem. This is analogous to the first order case. The main difference is that in the induction on the right-hand side in the main lemma, we now have to deal with higher-order variables.

Lemma 4.8 *Termination of a labelled term M is equivalent to*

1. *termination of $m(M)$, for any label $m \in \mathbb{N}$,*
2. *termination of $\vec{X}.M$, for any binder $\vec{X}$.*

Proof 1. The outermost label of a term can only be involved in merger steps, which can be postponed after other steps.

2. Direct from the definition of the rewrite relation. $\square$

Definition 4.9 *For M a labelled term, it's* (minimal) label $\bigwedge M$ *is defined as the minimal label occurring in M. Note that this is in general undefined if the term doesn't contain labels.*

Lemma 4.10 (Extract) *Let $M \to_{\mathcal{P}\mathbb{N}} Q^\tau$, for some term M, pattern Q and substitution τ. Then there exist a pattern P and substitution σ such that $M = P^\sigma$, $\bigwedge Q \leq \bigwedge P$, and either*

*(*contribute*) $P' \to_{\mathcal{P}\mathbb{N}} P$ and $\tau' = \tau$, or*

*(*external*) $P' = P$ and $\tau' \to_{\mathcal{P}\mathbb{N}} \tau$.*

Proof By induction on the generation of $\to_{\mathcal{P}\mathbb{N}}$, and by induction on Q in the base case. $\square$

The following lemma was dubbed the *square bracket lemma* by Van Daalen (cf. [NGV94, p. 505], [Lév78, 1.5.4. Lem. 3], [Bar84, Lem. 14.1.8]).

Lemma 4.11 (Square Bracket) *Let M be a term, P a pattern, and σ, τ be substitutions, such that $M^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} P^\tau$. Let moreover m be the label of the heads of σ, then whether the substitution σ contributes to P can be characterised via these labels. Formally, (at least) one of the following situations must hold.*

*(*contribute*) $\bigwedge P \leq m$, or*

*(*external*) $M \twoheadrightarrow_{\mathcal{P}\mathbb{N}} P^\upsilon$, and $\upsilon^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \tau$ for some substitution υ.*

Proof See Appendix C. $\square$

Theorem 4.12 (FFD) *$\mathcal{P}\mathbb{N}$ is terminating for higher-order pattern rewriting systems $\mathcal{P}$.*

Proof See Appendix C. $\square$

In [Oos] we reported a proof of FFD for orthogonal PRSs based on a method due to Klop [Klo]. The method here is an extension of a method due to Van Daalen (cf. [Bar84]) and works for orthogonal and non-orthogonal PRSs alike. As far as we know, it's the first proof of FFD for PRSs.

4.1 FFD for related classed of rewriting systems

Since both the lambda calculus and combinatory reduction systems can be naturally embedded into PRSs (see Section 2), we have the following corollary.

Corollary 4.13 *1. The labelled version of a CRS satisfies FFD.*

2. The labelled version of the lambda calculus satisfies FFD.

3. The labelled version of a TRS satisfies FFD.

Proof 1. By the embedding of CRSs into PRSs and Theorem 4.12.

2. By the embedding of the lambda calculus into PRSs and Theorem 4.12 (alternatively, by the embedding of the lambda calculus into CRSs and the first item).

3. By the embedding of TRSs into PRSs and Theorem 4.12 (alternatively, by the embedding of TRSs into CRSs and the first item). □

For the lambda calculus FFD is equivalent to termination of the lambda calculus with so-called *Hyland-Wadsworth*-labels ([Bar84, p. 14]) and well-known, as discussed above.

On the other hand, the case of combinatory reduction systems is not so clear. A first proof sketch of FFD for orthogonal (there called regular) CRSs is due to Klop ([Klo80, Rem. II.6.2.7.16]). However, as observed by Melliès ([Melon, Sec. 6.2.2]) that method is based on a false lemma. Although we claim that the method can be repaired (and even extended to the class of orthogonal PRSs [Oos]), the method is intimately linked with the orthogonality restriction. For example, it yields an upperbound on the length of all (m-bounded) family developments starting from some term, and this is impossible for general CRSs as discussed above. An axiomatic approach to FFD was introduced by Melliès in [Melon], and it is conjectured there ([Melon, Conj. 1,p. 203]) that the axioms are verified by all (not only orthogonal) CRSs. Hence, Corollary 4.13 can be seen as resolving that conjecture in the positive.

Remark 4.14 *Actually in [Melon], an extension of CRSs with names is discussed. These can equally easy be embedded into PRSs by introducing a new base type 'names'.*

5 Applications

The applications in [Bar84, Sec. 14.2] of FFD in the untyped lambda calculus, can immediately be generalised to (orthogonal) PRSs. In this section we only sketch the generalisations.

Finiteness of Developments via FFD Finiteness of developments should follow from finiteness of family developments, since in a development never newly created redexes are contracted hence one can put a bound of 1 on the labels.

Definition 5.1 *Define a mapping f from marked to labelled preterms as the homomorphic extension of the map $\underline{l \to r} \mapsto l_1^-$.*

Lemma 5.2 *If $M \to_{\underline{\mathcal{P}}} N$, then $f(M_0^+) \to_{\mathcal{P}\mathbb{N}}^+ f(N_0^+)$.*

Proof By induction on the generation of $\to_{\underline{\mathcal{P}}}$. □

Note that the inverse does not hold in general, as witnessed by the rewrite sequence $\mathtt{f}(\mathtt{a}) \to \mathtt{g}(\mathtt{a},\mathtt{a}) \to \mathtt{g}(\mathtt{b},\mathtt{a}) \to \mathtt{g}(\mathtt{b},\mathtt{c})$ for the TRS of Example 3.1 in which indeed only redexes in generation 0 are contracted, but which is not a development (according to our definition). It's easy to check that the inverse does hold when the PRS is orthogonal.

Corollary 5.3 *Developments in PRSs are finite.*

Proof From Definition 3.5, Theorem 4.12, and Lemma 5.2. □

Standardisation via FFD If one defines two rewrite steps to be *dependent* if the occurrences of their left-hand sides have 'overlap', then by quotienting rewrite sequences by the corresponding independence relation one obtains 'concurrent rewrite sequences', just like in concurrency theory. As a representative of such equivalence classes one can take so-called *standard* rewrite sequences. Standard sequences, originally introduced in [CF58, p. 140] for the lambda calculus and later generalised to other classes of rewriting systems (cf. [Klo80, Oos, Melon]), are rewrite sequences where the rewrite steps are performed in outside-in, left-to-right order.

Standardisation is the process of transforming a non-standard into a standard rewrite sequence, i.e. of computing the representative of the equivalence class.

Definition 5.4 *1. If $M \rightarrow_{\underline{\mathcal{P}}} N_1$, $M \rightarrow_{\underline{\mathcal{P}}} N_2$ and M contains exactly two marked symbols, then the resulting $\mathcal{P}$-steps are* simultaneous.

2. *Extend both steps to complete marked rewrite sequences. The corresponding complete developments (starting and ending in the same terms) possibly prefixed and suffixed by arbitrary rewrite sequences are* permutations *of each other.* Permutation equivalence *is the equivalence closure of the permutation relation.*

3. *The steps in the first item can be ordered according to the relative positions of the corresponding marked symbols. In particular, we say that the resulting (prefixed and suffixed) complete development of the former is to the* left *(*right, outside, inside*) of the latter, if the positions of the steps are thus related. A transformation from a sequence inside (to the right of) some other sequence, to that other sequence is called an* outside-in *(*left-to-right*) step.*

4. *The* standardisation order *is the outside-in modulo left-to-right order.*

The toplevel proofstructure of the following theorem is due to Klop ([Klo80, II.6.2.8.5], [Bar84, Prop. 14.2.8])

Corollary 5.5 *The standardisation relation is terminating on finite rewrite sequence in PRSs.*

Proof A rewrite sequence $M \twoheadrightarrow_{\mathcal{P}} N$ of length m can be lifted to a labelled rewrite sequence $M_m^+ \twoheadrightarrow_{\mathcal{P}_{\mathbb{N}}} M_0^+$ by Theorem 4.7. Observe that permutations can be lifted as well, without changing the labellings of the initial and final terms. Moreover, if the initial part of some sequence is fixed, permutations can only yield finitely many distinct initial rewrite steps from it since the initial rewrite sequence is finite. Hence, for each term reachable from M_m^+, only finitely many distinct rewrite steps can be (obtained by standardisation; this is false in general). Then from König's Lemma and Theorem 4.12, we have finiteness of the number of rewrite sequences starting from M_m^+ obtained by standardisation. Therefore, to prove termination, it suffices to show acyclicity of standardisation. This follows by the minimal counterexample method in [Bar84, Lem. 14.2.7], which is only based on finiteness of the initial term. □

6 Conclusion

We've presented an inductive proof of the finite family developments theorem for the class of higher-order pattern rewriting systems and sketched some applications of this result.

Many questions remain open. We mention some of them.

1. FFD was proven for first-order term rewriting systems in [Mar92, Prop. 3.3.11], using Kruskal's Tree Theorem. Is this also possible for the higher-order case? (None of the higher-order extensions of RPO in the literature seems to work.)

2. Do there exist useful critical pair criteria implying confluence based on family developments? (Analogous to critical pair criteria based on developments (cf. [Oos97]).

3. For what kind of rewriting systems does FFD hold? An axiomatic geometric approach to this question is presented in [Melon].

4. Does FFD imply finiteness of superdevelopments ([Raa])? (We think yes, since intuitively a superdevelopment from some term should be bounded by the marking such that each symbol is marked by its distance to the root in the term.)

5. Can the same technique be employed to prove modularity of termination for *non-collapsing* PRSs? (A rewrite systems is non-collapsing when no two 'disconnected' parts of a term can be 'connected' by performing a rewrite step.)

Acknowledgements Thanks to Femke van Raamsdonk and Roel de Vrijer for discussions and comments on the contents of this paper.

References

[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*. North-Holland, Amsterdam, revised edition, 1984.

[Bar92] H. P. Barendregt. Lambda calculi with types. in [AGM92, pp. 117–310], 1992.

[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*. Springer-Verlag, 1993.

[BKM89] J.A. Bergstra, J.W. Klop, and A. Middeldorp. *Termherschrijfsystemen*. Kluwer Bedrijfswetenschappen, Deventer, 1989. in Dutch.

[CF58] Haskell B. Curry and Robert Feys. *Combinatory Logic, volume I*. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1958.

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

[Cur30] H. B. Curry. Grundlagen der kombinatorischen Logik. Teil I, II. *American Journal of Mathematics*, LII:509–536 and 789–834, 1930.

[dV87] Roel de Vrijer. Exactly estimating functionals and strong normalization. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A*, 90(4):479–493, December 1987. Also in [Vri87, Ch. 3].

[EMW94] P. Enjalbert, E. W. Mayr, and K. W. Wagner, editors. *STACS 94, 11th Annual Symposium on Theoretical Aspects of Computer Science, Caen, France, February 1994*, volume 775 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[Gan96] Harald Ganzinger, editor. *Rewriting Techniques and Applications, 7th International Conference, RTA-96, New Brunswick, NJ, USA, July 27–30,1996*, volume 1103 of *Lecture Notes in Computer Science*. Springer, 1996.

[Hin78] R. Hindley. Reductions of residuals are finite. *Transactions of the American Mathematical Society*, 240:345–361, June 1978.

[HMMN94] Jan Heering, Karl Meinke, Bernhard Möller, and Tobias Nipkow, editors. *Higher-Order Algebra, Logic, and Term Rewriting, First International Workshop, HOA '93, Amsterdam, The Netherlands, September 23–24 1993, Selected Papers*, volume 816 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[JO] Jean-Pierre Jouannaud and Mitsuhiro Okada. A computation model for executable higher-order algebraic specification languages. in [LIC91, pp. 350–361].

[Kah93] Stefan Kahrs. Context rewriting. In M. Rusinowitch and J. L. Rémy, editors, *Conditional Term Rewriting Systems, Third International Workshop, CTRS-92, Pont-à-Mousson, France, July 8–10, 1992*, volume 656 of *Lecture Notes in Computer Science*, pages 21–35. Springer-Verlag, 1993.

[Kha] Zurab Khasidashvili. Perpetuality and strong normalization in orthogonal term rewriting systems. In [EMW94, pp. 163–174].

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

[Kha96] Zurab Khasidashvili, October 1996. Personal communication.

[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.

[KOR93] J. W. Klop, V. van Oostrom, and F. van Raamsdonk. Combinatory reduction systems, introduction and survey. *Theoretical Computer Science*, 121(1–2):279–308, December 1993.

[Lév78] Jean-Jacques Lévy. *Réductions correctes et optimales dans le λ-calcul*. Thèse de doctorat d'etat, Université Paris VII, 1978.

[LIC91] *Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science, Amsterdam, The Netherlands, July 15–18, 1991*, Los Alamitos, California, 1991. IEEE Computer Society Press.

[Loa95] Ralph Loader. Normalisation by translation. Note distributed on "types" mailing list, April 1995. Obtainable via `http://sable.ox.ac.uk/~loader/`.

[Mar92] Luc Maranget. *La stratégie paresseuse*. Thèse de doctorat, Université Paris VII, 6 juilliet 1992.

[Melon] Paul-André Melliès. *Description Abstraite des Systèmes de Réécriture*. Thèse de doctorat, Université Paris VII, 23 October 1996 Preliminary version. (in French).

[Mil91] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. In P. Schroeder-Heister, editor, *Extensions of Logic Programming, International Workshop, Tübingen, FRG, December 8–10, 1989*, volume 475 of *Lecture Notes in Artificial Intelligence*, pages 253–281. Springer-Verlag, 1991.

[MN94] Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. Technical Report I9433, Institut für Informatik, TU München, November 1994. To appear in TCS.

[NGV94] R. P. Nederpelt, J. H. Geuvers, and R. C. de Vrijer. *Selected Papers on Automath*, volume 133 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1994.

[Nip] Tobias Nipkow. Higher-order critical pairs. in [LIC91, pp. 342–349].

[Oos] Vincent van Oostrom. Higher-order families. In [Gan96, pp. 392–407].

[Oos94] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, March 1994.

[Oos96] Vincent van Oostrom. Take five. IR 406, Vrije Universiteit, Amsterdam, June 5, 1996.

[Oos97] Vincent van Oostrom. Developing developments. *Theoretical Computer Science*, 175:28 pp., 1997.

[OR] Vincent van Oostrom and Femke van Raamsdonk. Comparing combinatory reduction systems and higher-order rewrite systems. In [HMMN94, pp. 276–304].

[Pol96] Jan Cornelis van de Pol. *Termination of Higher-order Rewrite Systems*. PhD thesis, Universiteit Utrecht, Preliminary version of August 27, 1996.

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

[Raa96] Femke van Raamsdonk. *Confluence and Normalisation for Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, May 1996.

[RS95] F. van Raamsdonk and P. Severi. On normalisation. Report CS-R9545, CWI, June 1995.

[Sch65] D. E. Schroer. *The Church-Rosser Theorem*. PhD thesis, Cornell University, Ithaca, N.Y., 1965.

[Tai67] W. W. Tait. Intensional interpretations of functionals of finite type *I*. *the Journal of Symbolic Logic*, 32(2):198–212, June 1967.

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

[Vri85] Roel de Vrijer. A direct proof of the finite developments theorem. *The Journal of Symbolic Logic*, 50(2), June 1985. Also in [Vri87, Ch. 2].

[Vri87] Roelof Cornelis de Vrijer. *Surjective Pairing and Strong Normalization: Two Themes in Lambda Calculus*. PhD thesis, Universiteit van Amsterdam, January 1987.

[Wol93] D. A. Wolfram. *The Clausal Theory of Types*, volume 21 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1993.

A FD for the lambda calculus

We give a short proof of FD for the lambda calculus, using standard notation ([Bar84]). See the section on higher-order rewriting on how this translates to PRS notation.

In the marked lambda calculus, except for adding the rule

$$M, N \text{ are terms} \implies \underline{\beta}(x.M, N) \text{ is a term}$$

nothing changes in term formation (cf. [Bar84, Def. 2.1.1]). Each occurrence of x in M is bound by (the displayed) x in $\underline{\beta}(x.M, N)$. The only rewrite rule of the marked lambda calculus is

$$\underline{\beta}(x.M, N) \to M^{[x \mapsto N]}$$

. Termination of the marked lambda calculus corresponds to all developments (see Definition 3.5) being terminating (cf. [Bar84, Thm. 11.2.21]).

Theorem A.1 *The marked lambda calculus is terminating.*[3]

Proof We prove *for all marked lambda terms M and terminating substitutions σ, M^σ is terminating* by induction on M, from which the theorem follows by taking the identity for σ.

- (var) In case of a variable, the statement follows by the termination assumption on the substitution.
- (@) In case of an application of the form MN, the statement follows by the induction hypothesis for M and N.
- (λ) In case of an abstraction of the form $\lambda x.M$, the statement follows by the induction hypothesis for M.
- ($\underline{\beta}$) Consider a $\underline{\beta}$-redex of the form $\underline{\beta}(x.M, N)$. Since M^σ and N^σ are terminating by induction hypothesis, an infinite rewrite sequence must look like $\underline{\beta}(x.M^\sigma, N^\sigma) \twoheadrightarrow \underline{\beta}(x.M', N') \to M'^{[x \mapsto N']} \to \ldots$. However $M'^{[x \mapsto N']} \twoheadleftarrow M^{\sigma[x \mapsto N^\sigma]}$ which is terminating by the induction hypothesis. □

Many proofs of the finite developments theorem for the lambda calculus are known. See [Raa96, Oos96] for (pointers to) some of these. Apart from by specialisation of the proof of FD for PRSs, there are several other ways to arrive at the present proof (1) The proof presented here can be viewed upon as a specialisation of the computability proof of termination for the simply typed lambda calculus ([Tai67]), exploiting the observation that induction-loading on the termination predicate is not necessary here since no redexes can be created. (2) Removing the indirection via the set $\mathcal{SN}$ in the proof in [RS95] gives rise to it. (3) It can be found by omitting the (exact) upperbound information in the FD proof in [Vri85] (see Subsection 3.2).

[3] This proof was originally presented in [Oos96, FD à la Tait].

B Exact upperbounds

Exact upperbounds for developments in combinatory reduction systems can in fact be obtained quite easily. The idea is to use a more precise notion of value, by forcing monotonicity only for metavariables which are *erased* on the right-hand side, not for all metavariables appearing in the left-hand side. So, instead of adding in the value of the left-hand side after removing the marking ($[\![\sigma, l]\!]$ in the (rule)-case above), only the values of those metavariables are added in, that either do not appear in the right-hand side or that are always erased since all its occurrences are inside erased arguments of other metavariables.

Only the changes with respect to the non-exact upperbound case are given.

Definition B.1 *(rule)* $[\![\underline{l \to r}(\vec{Z})^{\sigma}]\!] = (1,0) + \sum_{Z \in l | Z \notin r^{\sigma_{-Z}}} [\![\sigma, Z]\!] + [\![\sigma, r]\!]$, *and* σ_{-Z} *is* σ *except that* $\sigma(Z) = Z$.

When specialised to the lambda calculus the ensuing definition of value when projected onto its height (multiplicity) yields exactly the definition of height (multiplicity) in [Vri85] (after some unfolding). The proof of Lemma 3.13 goes through easily, using the following auxiliary lemma (under the same assumptions).

Lemma B.2 *1. If* $y \neq x_i$ *and* $y \notin M_i$, *for* $1 \leq i \leq n$, *then* $y \in M$ *iff* $y \in M^{[\vec{x} \mapsto \vec{M}]}$,

2. $Z \in N^{\sigma_{-Z}}$ *iff* $Z \in N^{\sigma_{-Z}[\vec{x} \mapsto \vec{M}]}$.

Proof 1. By induction on marked terms.

2. By induction on unmarked terms, using the first item in the (var1)-case. □

The proof of Theorem 3.14 goes through easily, using the following auxiliary theorem for the (rule)-case.

Theorem B.3 *1. If* $x \in M$, *then* $[\![M]\!]_m$ *is monotonous in* x.

2. If $Z \in r^{\sigma_{-Z}}$, *then* $[\![\sigma, r]\!]$ *is monotonous in* Z.

In order to show that the upperbound computed (the height of the value) is indeed an exact upperbound, it suffices to specify a so-called *maximal* strategy achieving that bound. Maximal strategies specify how to obtain the worst-case (w.r.t. the number of steps performed) behaviour of a term. The following inductively defined maximal strategy taken from [Kha], which specialises to the maximal strategy in [Vri85, Def. 2.10] for the lambda calculus, does the job.

Definition B.4 *For* M *a marked term, such that* $[\![M]\!]_h > 0$, *we define* M^* *by induction on the structure of* M.

(fun) $\mathtt{f}(M_1, \ldots, M_i, \ldots, M_n)^* =^{\mathrm{def}} \mathtt{f}(M_1, \ldots, M_i^*, \ldots, M_n)$, *if* $[\![M_i]\!]_h > 0$ *and* $[\![M_j]\!]_h = 0$, *for* $1 \leq j < i$,

(abs) $(x.M)^* = x.M^*$,

(rule) $(\underline{l \to r}(\vec{Z})^{\sigma})^* = \underline{l \to r}(\vec{Z})^{\sigma^*}$, *if* $Z \in l$, $Z \notin r^{\sigma_{-Z}}$, *and* $[\![\sigma, Z]\!]_h > 0$,
$(\underline{l \to r}(\vec{Z})^{\sigma})^* = r^{\sigma}$, *otherwise.*

Theorem B.5 (FD♯) $[\![M]\!]_h$ *is exactly the number of rewrite steps in a complete marked sequence from* M *of maximal length.*[4]

Proof It suffices to prove that if $[\![M]\!]_h > 0$, then $M \to_{\mathcal{P}} M^*$ and $[\![M]\!]_h = [\![M^*]\!]_h + 1$. This is proven by induction on M. □

Note that the strategy 'respects' non-left-linearity, hence also provides an exact upperbound on the length of developments in non-left-linear combinatory reduction systems.

[4] A similar result has been independently obtained by Khasidashvili [Kha96].

C Proofs of FFD

Proof (of Lift Theorem 4.7) We prove that *any rewrite sequence $M \twoheadrightarrow_{\mathcal{P}} N$ of length m is the projection of a labelled rewrite sequence $M^+_m \twoheadrightarrow_{\mathcal{P}\mathbb{N}} N^+_0$* by induction on m.

(0) By Lemma 4.5.

$(m+1)$ Suppose $M \to_{\mathcal{P}} M' \twoheadrightarrow_{\mathcal{P}} N$. By induction hypothesis we know $M'^+_m \twoheadrightarrow_{\mathcal{P}\mathbb{N}} N^+_0$, hence it suffices to show $M^+_{m+1} \twoheadrightarrow_{\mathcal{P}\mathbb{N}} M'^+_m$, by induction on the derivation of $M \to_{\mathcal{P}} M'$. In the base case (sub) the following two properties are used.

1. $\vec{X}.l(\vec{M})^+_m = \vec{X}.m(l^-_m(\vec{M}^+_m))$ (for m+1), and
2. $\vec{X}.m{+}1(r^+_m(\vec{M}^+_{m+1})) \twoheadrightarrow_d \vec{X}.m(r^+_m(\vec{M}^+_m)) \twoheadrightarrow_c \vec{X}.r(\vec{M})^+_m$,

which are shown by induction on patterns and terms respectively. □

Proof (of Square Bracket Lemma 4.11) By induction on the length of the rewrite sequence $M^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} P^\tau$

(0) Either, a substitution exists υ such that $M = P^\upsilon$, or it does not exist. If it does exist, the conditions of ($\mathsf{external}$) are trivially satisfied. If it does not exist then one shows by induction on P that $\bigwedge P \leq m$.

$(m+1)$ Suppose $M^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} M' \to_{\mathcal{P}\mathbb{N}} P^\tau$. By Lemma 4.10, we know that there exists a pattern P' and substitution τ' such that $M' = P'^{\tau'}$, $\bigwedge P \leq \bigwedge P'$, and either

1. $P' \to_{\mathcal{P}\mathbb{N}} P$ and $\tau' = \tau$, or
2. $P' = P$ and $\tau' \to_{\mathcal{P}\mathbb{N}} \tau$.

By the induction hypothesis, we know that for $M^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} P'^{\tau'}$ we are in either of the following situations

($\mathsf{contribute}$) $\bigwedge P' \leq m$ but then we can conclude $\bigwedge P \leq m$ by transitivity of $<$, or

($\mathsf{external}$) $M \twoheadrightarrow_{\mathcal{P}\mathbb{N}} P'^\upsilon$, and $\upsilon^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \tau'$ for some substitution υ.

- ($\mathsf{contribute}$) If $P' \to_{\mathcal{P}\mathbb{N}} P$ and $\tau' = \tau$, then $M \twoheadrightarrow_{\mathcal{P}\mathbb{N}} P'^\upsilon \to_{\mathcal{P}\mathbb{N}} P^\upsilon$ by Lemma 2.6, and $\upsilon^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \tau$,
- ($\mathsf{external}$) If $P' = P$ and $\tau' \to_{\mathcal{P}\mathbb{N}} \tau$, then $M \twoheadrightarrow_{\mathcal{P}\mathbb{N}} P^\upsilon$, and $\upsilon^\sigma \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \tau' \to_{\mathcal{P}\mathbb{N}} \tau$ and we are done for. □

Proof (of the Main Theorem 4.12) By Lemma 2.10 it suffices to prove $M^{+\sigma}_m$ *is terminating, for every term M in $\mathcal{P}$, label $m \in \mathbb{N}$, and terminating substitution σ*, by induction first on m ordered by $<$, and then on M ordered by $\lhd$ (see Lemma 2.8). Consider a minimal term $M =^{\text{def}} \vec{X}.\mathsf{a}(\vec{M})$ for which $M^{+\sigma}_m =^{\text{def}} \vec{X}.m(\sigma(\mathsf{a})(\vec{M}^{+\sigma}_m))$. is not terminating. By Lemma 4.8, this is the same as non-termination of $\sigma(\mathsf{a})(\vec{M}^{+\sigma}_m)$. By induction hypothesis $\vec{M}^{+\sigma}_m$ is terminating. We make a case distinction on a.

(fun) If $\mathtt{a}$ is a function symbol $\mathtt{f}$, then $\sigma(\mathsf{a}) =^{\text{def}} \mathtt{f}$, and since by induction hypothesis the arguments of $\mathtt{f}$ are terminating, an infinite rewrite sequence must look like

$$\sigma(\mathsf{a})(\vec{M}^{+\sigma}_m) \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \mathtt{f}(\vec{M}') = {l^-_{n+1}}^\tau \to_{\mathcal{P}\mathbb{N}} {r^+_n}^\tau \to_{\mathcal{P}\mathbb{N}} \ldots$$

for some rule $\vec{Z}.l \to \vec{Z}.r$ in $\mathcal{P}$ where $l =^{\text{def}} \mathtt{f}(\vec{P})$, and substitution τ having domain $\vec{Z}$. Two observations can be made.

1. $\vec{M}'$ is a vector of terminating terms by the induction hypothesis, hence the substitution τ is terminating by Lemma 2.9.

2. $n < m$, by the assumption that every left-hand side contains at least two function symbols and the label of the head symbols of the arguments of $\mathtt{f}$ are initially m and cannot increase by rewriting.

Hence, by the induction hypothesis ${r^{+}_{m'}}^{\tau}$ is terminating.

(var) If $\mathtt{a}$ is a variable not in the domain of σ, then termination follows as in the previous item (even simpler). If $\mathtt{a}$ is in the domain of σ, then we prove termination by induction on the order $\ll$ starting from $\sigma(\mathtt{a})$, which is well-founded by the termination assumption on σ and Lemma 2.8.[5] If $\sigma(\mathtt{a}) =^{\mathrm{def}} \vec{Y}.\mathtt{b}(\vec{N})$, then $\sigma(\mathtt{a})(\vec{M}^{+\sigma}_{m}) =^{\mathrm{def}} \mathtt{b}(\vec{N})^{[\vec{Y}\mapsto \vec{M}^{+\sigma}_{m}]}$. By the (inner) induction hypothesis, we have that $\vec{N}{*} =^{\mathrm{def}} \vec{N}^{[\vec{Y}\mapsto \vec{M}^{+\sigma}_{m}]}$ is terminating. We perform case analysis on $\mathtt{b}$.

(lab) If $\mathtt{b}$ is a label, then termination follows from the inner induction hypothesis and Lemma 4.8.

(fun) If $\mathtt{b}$ is a function symbol $\mathtt{f}$ in $\mathcal{P}$, then since the arguments are terminating by the inner induction hypothesis, an infinite sequence must look like

$$\mathtt{f}(\vec{N}{*}) \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \mathtt{f}(\vec{N}') = {l^{-}_{n+1}}^{\tau} \rightarrow_{\mathcal{P}\mathbb{N}} {r^{+}_{n}}^{\tau} \rightarrow_{\mathcal{P}\mathbb{N}} \dots$$

for some rule $\vec{Z}.l \rightarrow \vec{Z}.r$ in $\mathcal{P}$ where $l =^{\mathrm{def}} \mathtt{f}(\vec{P})$, and substitution τ having domain $\vec{Z}$. By Lemma 4.11, either of two situations must be the case; either the substitution $\sigma' =^{\mathrm{def}} [\vec{Y} \mapsto \vec{M}^{+\sigma}_{m}]$ having m as head label did contribute to one among $\vec{P}' =^{\mathrm{def}} \vec{P}^{-}_{n+1}$ or it didn't.

(contribute) If $\bigwedge \vec{P}' = n{+}1 \leq m$, then $n < m$, and by the inner induction hypothesis and Lemma 2.9, the outer induction hypothesis applies to r^{+}_{n} and τ.

(external) If $\vec{N} \twoheadrightarrow_{\mathcal{P}\mathbb{N}} {\vec{P}'}^{\upsilon}$, and $\upsilon^{\sigma'} \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \tau$ for some substitution υ, then

$$\mathtt{f}(\vec{N}) \twoheadrightarrow_{\mathcal{P}\mathbb{N}} \mathtt{f}({\vec{P}'}^{\upsilon}) = {l^{-}_{n+1}}^{\upsilon} \rightarrow_{\mathcal{P}\mathbb{N}} {r^{+}_{n}}^{\upsilon}$$

and ${r^{+}_{n}}^{\upsilon\,\sigma'} \twoheadrightarrow_{\mathcal{P}\mathbb{N}} {r^{+}_{n}}^{\tau}$ by Lemma 2.6. Since the former sequence contains at least one step (the final one), the inner induction hypothesis applies to ${r^{+}_{n}}^{\upsilon}$ and σ', yielding termination of ${r^{+}_{n}}^{\tau}$ by the latter sequence.

(var) If $\mathtt{b}$ is a variable not among $\vec{Y}$, then termination follows as in the previous item (even simpler). If $\mathtt{b} = Y_i$, and $M_i =^{\mathrm{def}} \vec{Z}.\mathtt{c}(\vec{O})$, then $\mathtt{b}(\vec{N})^{[\vec{Y}\mapsto \vec{M}^{+\sigma}_{m}]} =^{\mathrm{def}} {\mathtt{c}(\vec{O})^{+}_{m}}^{\sigma[\vec{Z}\mapsto \vec{N}{*}]}$ which is terminating by the outer induction hypothesis, since $\mathtt{c}(\vec{O})$ is a proper subterm of M, σ is terminating by assumption, and $\vec{N}{*}$ is terminating by the inner induction hypothesis. $\square$

[5]In the first order term rewriting case we are done with the proof now. In the lambda calculus case we have arrived at the *Main Lemma* (cf. [Bar84, Cor. 14.1.11], [Lév78, 1.5.5. Lem. 4], [NGV94, p. 511]).