ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 333 (2005) 91–125

Theoretical Computer Science

www.elsevier.com/locate/tcs

Normalisation for higher-order calculi with explicit substitutions☆

Eduardo Bonelli[a, b, *]

[a] *LIFIA, Facultad de Informática, Universidad-Nacional de La Plata, 50 y 115, La Plata (1900), Buenos Aires, Argentina*

[b] *Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA*

Abstract

Explicit substitutions (ES) were introduced as a bridge between the theory of rewrite systems with binders and substitution, such as the λ-calculus, and their implementation. In a seminal paper Melliès observed that the dynamical properties of a rewrite system and its ES-based implementation may not coincide: he showed that a strongly normalising term (i.e. one which does not admit infinite derivations) in the λ-calculus may lose this status in its ES-based implementation. This paper studies normalisation for the latter systems in the general setting of higher-order rewriting: Based on recent work extending the theory of needed strategies to non-orthogonal rewrite systems we show that needed strategies normalise in the ES-based implementation of any orthogonal pattern higher-order rewrite system.

Keywords: Higher-order rewriting; Lamda calculus; Explicit substitutions; Normalisation; Needed-strategies

1. Introduction

This paper studies normalisation for calculi of explicit substitutions (ES) implementing higher-order term rewrite systems (HORS). The latter are rewrite systems in which binders

☆ Extended version of the work entitled "A normalisation result for higher-order calculi with explicit substitutions" which appeared in the proceedings of the conference FOSSACS 2003.

* Corresponding author. LIFIA, Facultad de Informática, Universidad-Nacional de La Plata, 50 y 115, La Plata (1900), Buenos Aires, Argentina.

E-mail address: ebonelli@cs.stevens-tech.edu (E. Bonelli).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.10.019

$$\begin{array}{ll} \text{Terms} & X := 1 \mid XX \mid \lambda X \mid X[s] \\ \text{Substitutions} & s := id \mid \uparrow \mid X \cdot s \mid s \circ s \end{array}$$

$$\begin{array}{llll} (\lambda X)\,Y & \to_{Beta} & X[Y \cdot id] & \\ (X\,Y)[s] & \to_{App} & X[s]\,Y[s] & (X \cdot s) \circ t \quad \to_{Map} \quad X[t] \cdot (s \circ t) \\ (\lambda X)[s] & \to_{Lam} & \lambda X[1 \cdot (s \circ \uparrow)] & id \circ s \quad \to_{IdL} \quad s \\ X[s][t] & \to_{Clos} & X[s \circ t] & (s_1 \circ s_2) \circ s_3 \quad \to_{Ass} \quad s_1 \circ (s_2 \circ s_3) \\ 1[X \cdot s] & \to_{VarCons} & X & \uparrow \circ (X \cdot s) \quad \to_{ShiftCons} \quad s \\ 1[id] & \to_{VarId} & 1 & \uparrow \circ id \quad \to_{ShiftId} \quad \uparrow \end{array}$$

Fig. 1. The $\lambda\sigma$ calculus.

and substitution are present, the λ-calculus [3] being a typical example. A recent approach to the implementation of HORS is the use of ES [1,4,12,18]. ES were introduced as a bridge between HORS and their concrete implementations. Their close relation with abstract reduction machines [1,4,16] allows us to speak of ES-based *implementations* of HORS. The idea behind these implementations is that the complex notion of substitution is promoted to the object-level by introducing new operators into the language in order to compute substitutions explicitly. This allows HORS to be expressed as more fine-grained (first-order) rewrite systems in which no complex substitution nor binders are present. Such a process may be applied to any HORS [9]. As an example Fig. 1 shows the rules of $\lambda\sigma$ [1], a calculus of ES implementing the λ-calculus (based on de Bruijn indices notation [13] in order to discard annoying issues related to the renaming of variables [1]).

An obstacle which arises when using ES for implementing HORS is that results on normalisation which hold in the higher-order rewriting setting may not be preserved in its implementation. A well-known example of this mismatch is due to Melliès [25]: he exhibited a strongly normalising (typed) term in the λ-calculus for which the $\lambda\sigma$-calculus introduces an infinite derivation. The problem is not confined to the setting of λ-calculus but rather affects any HORS. For example the following well-typed Haskell program:

```
map (\x → (map id [ map id [true] ])) [ map id [true] ] ,
```

where `id` abbreviates `\x → x`, is easily seen to be strongly normalising (and reduces to `[[[true]]]`), however its ES-based implementation (cf. Example 7) may introduce infinite derivations for it in a similar way to [25] (cf. Appendix A).

This mismatch calls for careful consideration of normalising strategies in the context of ES-based implementations of HORS. This paper studies normalisation in the latter systems based on *needed* strategies, a notion introduced in [17]. Needed strategies are those which rewrite redexes which are "needed" (cf. Section 2.3) in order to attain a normal form, assuming it exists. For example, the underlined redex in $1[2 \cdot \underline{(\lambda 1)\ 1} \cdot id]$ is not "needed" in order to achieve a normal form since there is a derivation, namely $1[2 \cdot (\lambda 1)\ 1 \cdot id] \to_{VarCons} 2$, that never reduces it. In fact the infinite derivation of the aforementioned Haskell program in the ES-based implementation takes place inside a substitution s in a term of the form $1[X \cdot s]$.

[1] Variables in terms are represented as positive integers. For example, $\lambda x.x$ is represented as $\lambda 1$, $\lambda x.\lambda y.x$ as $\lambda\lambda 2$, and $\lambda x.y$ as $\lambda 2$

The literature on needed strategies for higher-order rewrite systems has required them to be *orthogonal* [15,17,24] (an exception is [35], however, only weakly orthogonal systems are studied). A system is orthogonal if no conflicts (overlap) between redexes may arise. The theory of neededness for orthogonal systems does not suffice in our setting since HORS (even orthogonal ones) are implemented as *non-orthogonal* systems in the ES-based approach. For example, although the λ-calculus is orthogonal, $\lambda\sigma$ is not, as witnessed by the critical pair: $(\lambda X)[s]\,Y[s] \leftarrow_{App} ((\lambda X)\,Y)[s] \rightarrow_{Beta} X[Y \cdot id][s]$. However, recently an extension has been introduced for non-orthogonal systems [26,27]. Motivated by this work on needed derivations for non-orthogonal systems we prove the following new result: all needed strategies in ES-based implementations of arbitrary orthogonal pattern HORS normalise. This extends the known result [27] that needed strategies in $\lambda\sigma$ normalise.

As an example of (one of) the issues which must be revisited in the extended setting of HORS is a result [27, Lemma 6.4] which states that if the first redex in a *standard* $\lambda\sigma$ derivation occurs under a "λ" symbol, then no other redex in the derivation occurs at that symbol or above it. A standard derivation is one in which computation takes place in an outside-in fashion (cf. Definition 12). What makes "λ" so special in this regard in the $\lambda\sigma$-calculus is that creation of redexes above "λ", from below it, is not possible. We introduce the notion of *insulating symbol*[2] in order to identify these special symbols in the ES-based implementation of arbitrary higher-order rewrite systems (under our definition "λ" is insulating), and prove an analogous result.

Another contribution is the notion of *correspondence* for tracing redexes along derivations of the substitution calculus (in the $\lambda\sigma$-calculus, the substitution calculus is σ and consists of all the rewrite rules of Fig. 1 except for *Beta*). The need for such a notion arises from the fact that the residual theory for overlapping rewrite systems may cause redexes to be lost after some number of σ rewrite steps (cf. Section 4). For example, the *Beta* redex in M is lost in the following σ derivation:

$$M = ((\lambda X)\,Y)[id] \rightarrow_{App} (\lambda X)[id]\,Y[id] \rightarrow_{Lam} (\lambda(X[1 \cdot id \circ \uparrow]))\,Y[id]. \tag{1}$$

The correspondence relation shall allow such redexes to be retrieved once they "re-appear". The ideas developed suggest further interesting approaches towards an axiomatic treatment of the issue in the style of [26] or more recently [29].

Structure of the paper: Section 2 reviews the ERS_{db} higher-order rewriting formalism, the notion of standard derivation using the redex-permutation approach and its applications to the theory of neededness for orthogonal systems. Also, Melliès' extension to non-orthogonal systems are briefly discussed. Section 3 identifies the Standard-Projection Proposition as the only requirement for an orthogonal pattern HORS to verify normalisation of needed strategies. Section 4 is devoted to verifying that the latter holds for HORS. We then conclude and suggest possible future research directions. An accompanying appendix provides further details and proofs.

[2] In [7] we used the terminology "uncontributable" however the author deems that the word "insulating", as suggested by a referee, is more appropriate.

2. Setting the scene

This section provides a short introduction to the ERS_{db} higher-order rewriting formalism and its ES-based implementations. Further details may be consulted in [6] or [8].

2.1. The ERS_{db} formalism and its ES-based implementations

The ERS_{db} formalism: ERS_{db} is a higher-order rewriting formalism based on de Bruijn indices notation. Rewrite rules are constructed from metaterms; metaterms are built from: de Bruijn *indices* $1, 2, \ldots$, *metavariables* $X_l, Y_l, Z_l, \ldots$, where l is a label (i.e. a finite sequence of symbols) over an alphabet of *binder indicators* $\alpha, \beta, \ldots$, *function symbols* $f, g, h, \ldots$ equipped with an arity n with $n \geqslant 0$, *binder symbols* $\lambda, \mu, \nu, \xi, \ldots$ equipped with an arity n with $n > 0$, and a *metasubstitution* operator $M[\![N]\!]$.

Definition 1 (*Metaterms*). The grammar for metaterms is

$$M ::= n \mid X_l \mid f(M, \ldots, M) \mid \xi(M, \ldots, M) \mid M[\![M]\!].$$

Labels in metavariables reference binders occurring "above" them (when depicted as trees) and henceforth we shall assume that their length coincides with the number of these binders and that they contain no repeated elements. More precisely, we restrict our attention to all metaterms M such that the predicate $\mathsf{WF}_{\varepsilon}(A)$ holds, where ε is the empty label and $\mathsf{WF}_l(A)$ is defined as follows:

- $\mathsf{WF}_l(n)$ holds.
- $\mathsf{WF}_l(X_k)$ iff $l = k$ and l contains no repeated elements.
- $\mathsf{WF}_l(f(M_1, \ldots, M_n))$ iff for all $1 \leqslant i \leqslant n$ we have $\mathsf{WF}_l(M_i)$.
- $\mathsf{WF}_l(\xi(M_1, \ldots, M_n))$ iff there exists $\alpha \notin l$ s.t. for all $1 \leqslant i \leqslant n$, $\mathsf{WF}_{\alpha l}(M_i)$.
- $\mathsf{WF}_l(M_1[\![M_2]\!])$ iff $\mathsf{WF}_l(M_2)$ and there exists $\alpha \notin l$ s.t. $\mathsf{WF}_{\alpha l}(M_1)$.

For example, $app(\lambda X_{\alpha}, Y_{\varepsilon})$ is a metaterm, however $f(\xi\lambda X_{\alpha\alpha})$ and $f(X_{\alpha})$ are not. As is standard, *positions* in metaterms are defined as the paths of their tree representation [2,14]. For example, the position of X_{α} and Y_{ε} in $app(\lambda X_{\alpha}, Y_{\varepsilon})$ is 1.1 and 2, respectively; also the position of λX_{α} is 1. We use ε for the root position. We write $Pos(M)$ for the set of all positions of M and $M|_p$ for the subterm of M at position p assuming $p \in Pos(M)$. For example, $Pos(app(\lambda X_{\alpha}, Y_{\varepsilon})) = \{\varepsilon, 1, 2, 1.1\}$ and $app(\lambda X_{\alpha}, Y_{\varepsilon})|_{1.1} = \lambda X_{\alpha}$. We use $<$ for the strict prefix ordering on positions, $\leqslant$ for the prefix ordering on positions and $\|$ to indicate that two positions are disjoint (i.e. $p \| q$ if neither $p < q$ nor $q < p$). If $p < q$ we say that p is *strictly above* q and if $p \leqslant q$ we say that p is *above* q. The leftmost symbol of a metaterm is its *head symbol*.

Definition 2 (*Rewrite rule*). A *rewrite rule* is a pair of metaterms $L \to R$ s.t.

(1) the head symbol of L is either a function or a binder symbol (hence L may not be a metavariable),
(2) all metavariables occurring in R also occur in L (disregarding labels), and
(3) there are no metasubstitutions in L.

We refer to L as the LHS and to R as the *RHS*, of the rewrite rule. An ERS_{db} $\mathcal{R}$ is a set of rewrite rules. The $\lambda\sigma$-calculus of Fig. 1 is an ERS_{db} as are all first-order rewrite systems. Two other examples are:

Example 3 (ERS_{db} *rewrite rules*). The β rewrite rule of the Lambda Calculus in which terms are represented using de Bruijn indices notation:

$$app(\lambda X_\alpha, Y_\varepsilon) \to_{\beta_{\mathrm{db}}} X_\alpha[\![Y_\varepsilon]\!].$$

The map ERS_{db}, encoding the higher-order function which maps an argument function over a list:

$$map(\xi X_\alpha, nil) \to_{map.1} nil$$
$$map(\xi X_\alpha, cons(Y_\varepsilon, Z_\varepsilon)) \to_{map.2} cons(X_\alpha[\![Y_\varepsilon]\!], map(\xi X_\alpha, Z_\varepsilon)).$$

An ERS_{db} $\mathcal{R}$ is *left-linear* if for every rewrite rule $L \to R$ in $\mathcal{R}$ and for every X, X occurs at most once in L. For example, β_{db} and *map* are left-linear but the following rewrite rule for quantifier permutation $imply(\exists\forall X_{\alpha\beta}, \forall\exists X_{\beta\alpha}) \to_{Comm} true$ is not since X occurs twice in $imply(\exists\forall X_{\alpha\beta}, \forall\exists X_{\beta\alpha})$, once with label $\alpha\beta$ and once with label $\beta\alpha$. In the sequel of this article we shall consider only left-linear ERS_{db}.

Terms are metaterms without occurrences of metavariables nor metasubstitution. A *rewrite step* is obtained by instantiating rewrite rules with *valuations*; the latter result from extending assignments (partial functions from metavariables to terms) to the full set of metaterms.

Definition 4 (*Valuation*). Let κ be a (partial) function from metavariables to terms. The *valuation* $\overline{\kappa}$ over metaterms (uniquely) determined by κ, is the (partial) function that satisfies the following conditions:

$$\begin{aligned}
\overline{\kappa}n &= n,\\
\overline{\kappa}X_l &= \kappa X_l,\\
\overline{\kappa}f(M_1,\ldots,M_n) &= f(\overline{\kappa}M_1,\ldots,\overline{\kappa}M_n),\\
\overline{\kappa}\xi(M_1,\ldots,M_n) &= \xi(\overline{\kappa}M_1,\ldots,\overline{\kappa}M_n),\\
\overline{\kappa}(M_1[\![M_2]\!]) &= \overline{\kappa}(M_1)\{\{1 \leftarrow \overline{\kappa}M_2\}\},
\end{aligned}$$

where $M\{\{n \leftarrow N\}\}$ is the result of substituting a term N for the index $n \geqslant 1$ in a term M and is defined as follows:

$$\begin{aligned}
f(M_1,\ldots,M_n)\{\{n \leftarrow N\}\} &= f(M_1\{\{n \leftarrow N\}\},\ldots,M_n\{\{n \leftarrow N\}\}),\\
\xi(M_1,\ldots,M_n)\{\{n \leftarrow N\}\} &= \xi(M_1\{\{n+1 \leftarrow N\}\},\ldots,M_n\{\{n+1 \leftarrow N\}\}),\\
m\{\{n \leftarrow N\}\} &= \begin{cases} m-1 & \text{if } m > n,\\ \mathsf{U}_0^n(N) & \text{if } m = n,\\ m & \text{if } m < n, \end{cases}
\end{aligned}$$

where for $i \geqslant 0$ and $n \geqslant 1$ we define the *updating functions* $\mathsf{U}_i^n(.)$ as follows:

$$\begin{aligned}
\mathsf{U}_i^n(f(M_1,\ldots,M_n)) &= f(\mathsf{U}_i^n(M_1),\ldots,\mathsf{U}_i^n(M_n)),\\
\mathsf{U}_i^n(\xi(M_1,\ldots,M_n)) &= \xi(\mathsf{U}_{i+1}^n(M_1),\ldots,\mathsf{U}_{i+1}^n(M_n)),\\
\mathsf{U}_i^n(m) &= \begin{cases} m+n-1 & \text{if } m > i,\\ m & \text{if } m \leqslant i. \end{cases}
\end{aligned}$$

In the rewrite rule $\exists\forall X_{\alpha\beta} \to_{Comm'} \forall\exists X_{\beta\alpha}$, a valuation that assigns the de Bruijn index 1 to the metavariable $X_{\alpha\beta}$ and also to $X_{\beta\alpha}$ does not reflect the binder commutation that the labels of metavariables are expressing. Thus, in the sequel we restrict our attention to the subset of all valuations that are *coherent* with the contextual information described by the labels of binder indicators in metavariables. Such valuations are dubbed *coherent valuations*.

Definition 5 (*Coherent valuations*).

- A valuation κ is *coherent* if for every pair of metavariables X_l and $X_{l'}$ in the domain of κ, $\mathsf{Coh}(l, \kappa X_l) = \mathsf{Coh}(l', \kappa X_{l'})$.
- Let S be a set of variables $\{x_0, x_1, \ldots\}$, M a metaterm and l be a label of binder indicators. Also, let $|l|$ denote the number of elements in l and $l@i$ the ith symbol in l ($i \leqslant |l|$). Then $\mathsf{Coh}(l, M)$ is defined as $\mathsf{Coh}^0(l, M)$, where

$$\begin{aligned}
\mathsf{Coh}^i(l,n) &= \begin{cases} n & \text{if } n \leqslant i,\\ l@(n-i) & \text{if } 0 < n-i \leqslant |l|,\\ x_{n-i-|l|} & \text{if } n-i > |l|, \end{cases}\\
\mathsf{Coh}^i(l, f(M_1,\ldots,M_n)) &= f(\mathsf{Coh}^i(l,M_1),\ldots,\mathsf{Coh}^i(l,M_n)),\\
\mathsf{Coh}^i(l, \xi(M_1,\ldots,M_n)) &= \xi(\mathsf{Coh}^{i+1}(l,M_1),\ldots,\mathsf{Coh}^{i+1}(l,M_n)).
\end{aligned}$$

For example, the valuation that is obtained by instantiating all occurrences of X_α, Y_ε and Z_ε in the rule *map*.2 by the terms $cons(1, nil)$, 2 and nil, respectively, is coherent. The resulting rewrite step is

$$\begin{aligned}
&map(\xi(cons(1,nil)), cons(2,nil))\\
&\quad \to_{map.2} cons(cons(2,nil), map(\xi(cons(1,nil)), nil)).
\end{aligned} \tag{2}$$

Determining whether a term matches the LHS of a rewrite rule in an arbitrary ERS_{db} is not as simple as syntactic or first-order matching. For example, in order to determine if a term is an instance of the LHS of the η_{db} rule, where η_{db} is $\lambda(app(X_\alpha, 1)) \to_{\eta_{\mathrm{db}}} X_\varepsilon$, it must be determined whether the one term to be substituted for X_α and X_ε does not have occurrences of free 1-level indices. Likewise, in the case of the aforementioned *Comm* and *Comm*′ rewrite rules, it must be determined whether the term assigned to $X_{\alpha\beta}$ results from the one assigned to $X_{\beta\alpha}$ by interchanging 1- with 2-level indices. As a consequence, we shall focus our attention on a subset of ERS_{db} for which the matching process is simpler. Such ERS_{db} are called *pattern* ERS_{db}. Definition 6 corresponds to the usual requirement that LHS of rewrite rules be higher-order patterns in other rewrite formalisms such as [22,30,31].

Definition 6 (*Pattern ERS_{db}*).

- The binding allowance of X in $L \to R$ is the set $\bigcap_{i=1}^{n} l_i$,[3] where $\{X_{l_1}, \ldots, X_{l_n}\}$ is the set of all X-based metavariables (i.e. of the form X_l for some label l) in $L \to R$.
- An ERS_{db} $\mathcal{R}$ is a *pattern* ERS_{db} ($PERS_{\mathrm{db}}$) if the following condition holds for every $L \to R$ in $\mathcal{R}$: For every X, if we let $X_{l_1}, \ldots, X_{l_n}$ be all the X-based metavariables in L, then
  (1) $l_1 = l_2 = \cdots = l_n$ and l_1 is the binding allowance of X in $L \to R$ and
  (2) for all $X_k \in R$, $|k| \geqslant |l_1|$.

For example, η_{db} and *Comm* are not pattern ERS_{db}. However, those described in Example 3 are. The fact that matching is simpler in pattern ERS_{db} is also reflected in the ES-based implementations of pattern ERS_{db} (described below). The latter do not contain explicit substitutions on the LHS (except for those which encode de Bruijn indices). In contrast, ES-based implementation of η_{db} and *Comm* do.[4] This is witness to the fact that when translating to a first-order ES-based setting, higher-order matching may not always be coded as syntactic matching. The "occurs check" imposed by η_{db} or the "commutation of indices check" imposed by *Comm* are complex features of higher-order matching that require further machinery (matching *modulo* a calculus of ES) in order to be mimicked in a first-order setting.

A *redex* r is a quadruple consisting of a term M, a rewrite rule $L \to R$, a (coherent) valuation, and a position p in M such that M at position p is an instance of the LHS of the rewrite rule $L \to R$ via the aforementioned valuation; the induced *rewrite step* is written $M \to_r N$ and we say that we *contract* r in M and obtain N. M is the *source* and N the *target* of r. Letters $r, s, t, \ldots$ stand for redexes. Let s, r be redexes with source M, then s *nests* r, written $s < r$, if the position of s is a prefix of the position of r in M; if neither s nests r nor r nests s, then they are said to be *disjoint* in M and we write $s \| r$. Two redexes r, s which are instances of the same rewrite rule are called *similar*, we also say that r is similar to s. We use $\to_{\mathcal{R}}$ for the rewrite step relation induced by an ERS_{db} $\mathcal{R}$ and $\twoheadrightarrow_{\mathcal{R}}$ for its reflexive–transitive closure. A *derivation* is a sequence of rewrite steps; we use letters $\phi, \varphi, \ldots$ for derivations; $|\phi|$ stands for the length (number of rewrite steps) of a derivation ϕ. Two rewrite steps s, r are *composable* if the target of s coincides with the source of r; in this case we write $s; r$ for their composition (i.e. the derivation in which r is computed after s); this notion is extended to a sequence of rewrite steps as expected. If $r_1, \ldots, r_n$ are composable rewrite steps, then $r_1; \ldots; r_n$ is the derivation resulting from composing them. We use e_M for the empty rewrite step whose source and target is M. Note that $e_M; \chi = \chi = \chi; e_N$ if χ is a derivation from M to N. Derivations that start (resp. end) at the same term are called *coinitial* (resp. *cofinal*).

ES-based implementations of HORS: Any ERS_{db} may be implemented as a first-order rewrite system with the use of explicit substitutions [9,6]. The implementation process roughly goes about dropping labels in metavariables, encoding indices n as $1[\uparrow^{n-1}]$ and replacing metasubstitution operators $\bullet[\![\bullet]\!]$ in rewrite rules with explicit substitutions $\bullet[\bullet \cdot id]$. Note that the operation of substitution is promoted to the object-level language. Therefore,

[3] Note that although labels are sequences of binder indicators, here we are referring to the underlying set of the labels l_i. For example, the underlying set of the label $\alpha\beta$ is $\{\alpha, \beta\}$.

[4] Namely, $\lambda(app(X[\uparrow], 1)) \to_{ES(\eta_{\mathrm{db}})} X$ and $imply(\exists\forall X, \forall\exists X[2 \cdot 1 \cdot (\uparrow \circ \uparrow)]) \to true$.

new rules—the substitution calculus—are added in order to define the behaviour of the new explicit substitutions; this calculus is in charge of propagating substitutions until they reach indices and then discarding the substitutions or replacing the indices. The details of the implementation process may be consulted in [9,6].

In this paper we fix the σ-calculus [1] as substitution calculus. Its rules are those of Fig. 1 (disregarding *Beta*) in which the rules *App* and *Lam* are generalised to arbitrary function symbols f and binder symbols ξ as follows:

$$f(X_1,\ldots,X_n)[s] \to_{Func_f} f(X_1[s],\ldots,X_n[s]),$$
$$\xi(X_1,\ldots,X_n)[s] \to_{Bind_\xi} \xi(X_1[1\cdot(s\circ\uparrow)],\ldots,X_n[1\cdot(s\circ\uparrow)]).$$

Since the σ-calculus is confluent [1] and strongly normalising [11] we shall write $\sigma(N)$ for the (unique) σ-normal form of N. If $\mathcal{R}$ is an ERS_{db}, then we write $\mathcal{R}_\sigma^{\mathsf{ES}}$ for its ES-based implementation and refer to it as an *implementation* (of $\mathcal{R}$).

Example 7. The implementation of the HORS β_{db} is $\lambda\sigma$, in other words $(\beta_{\mathrm{db}})_\sigma^{\mathsf{ES}} = \lambda\sigma$. That of *map* is the first-order rewrite system consisting of σ and the following rewrite rules:

$$map(\xi X, nil) \to_{ES(map.1)} nil,$$
$$map(\xi X, cons(Y,Z)) \to_{ES(map.2)} cons(X[Y\cdot id], map(\xi X, Z)).$$

Two basic properties of ES-based implementations of HORS are *Simulation* (if $M\to_{\mathcal{R}} N$, then for some M', $M\to_{\mathcal{R}^{\mathsf{ES}}} M' \twoheadrightarrow_\sigma \sigma(N)$) and *Projection* ($M\to_{\mathcal{R}_\sigma^{\mathsf{ES}}} N$ then $\sigma(M)\twoheadrightarrow_{\mathcal{R}} \sigma(N)$). The rewrite step (2) may be simulated in its ES-based implementation as

$$\begin{aligned} &map(\xi(cons(1,nil)), cons(2,nil))\\ &\quad\to_{ES(map.2)} cons(cons(1,nil)[2\cdot id], map(\xi(cons(1,nil)),nil))\\ &\quad\to_{Func_{cons}} cons(cons(1[2\cdot id], nil[2\cdot id]), map(\xi(cons(1,nil)),nil))\\ &\quad\to_{VarCons} cons(cons(2,nil[2\cdot id]), map(\xi(cons(1,nil)),nil))\\ &\quad\to_{Func_{nil}} cons(cons(2,nil), map(\xi(cons(1,nil)),nil)). \end{aligned}$$

Remark 8. Although we have restricted our attention to left-linear pattern ERS_{db}, the implementation procedure described in [9,6] applies to any ERS_{db}.

2.2. Standardisation

Informally, a standard derivation is one in which computation takes place in an outside-in manner. This notion may be formalised using the *redex-permutation* approach [20,26,28]. We briefly recall this approach.

Given two non-overlapping redexes r, s in some term M we define the notion of a redex *residual* of r after contracting s (written r/s) with an example. In $M = \underline{(\lambda\lambda(2\,2))\,((\lambda 1)\,2)}\,3 \to_{\beta_{\mathrm{db}}} (\lambda((\lambda 1)\,3)\,((\lambda 1)\,3))\,3 = N$, the residuals of $r = ((\lambda 1)\,2)$ in M after contracting the underlined redex s are the two copies of $(\lambda 1)\,3$ in N. Note that s has no residuals in N (i.e. for any s, $s/s = \emptyset$), and also that r/s is a finite set of redexes. The outermost redex in N is said to be *created* since it is not the residual of any redex in M. If U_M is a finite set of non-overlapping redexes (i.e. $r, s \in U_M$ implies r does not overlap s) in M and s is a redex

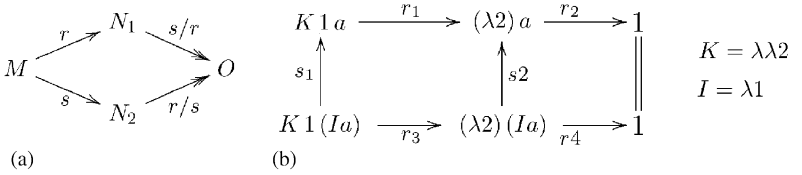


Fig. 2. Basic tiles.

Fig. 3. Oriented tiles.

in M, then $U_M/s = \{v | \exists u \in U_M, v \in u/s\}$. The residuals of r after a derivation $r_1; \ldots; r_n$ coinitial with it, is defined as $((r/r_1)/r_2)\ldots/r_n$. A *development* of U_M is a derivation $\phi = r_1; \ldots; r_n$ s.t. $r_i \in U_M/(r_1; \ldots; r_{i-1})$ for $i \in 1, \ldots, n$ and $U_M/\phi = \emptyset$ (if this last condition is not satisfied we say ϕ is a *partial development* of U_M). A well-known result called Finite Developments states that all partial developments are finite [3,33]. This allows one to prove the following property for any left-linear pattern ERS_{db}.

Proposition 9 (*Parallel moves or basic tile lemma Barendregt [3], Huet and Levy [17]*). *Given two non-overlapping redexes r, s in some term M, the divergence resulting from contracting r and s may be settled by developing their corresponding residuals* (*Fig. 2*(a)). *Moreover, for any u in M, $u/(r; s/r) = u/(s; r/s)$.*

Fig. 2(b) shows two basic tiles in the λ-calculus. As depicted, these basic tiles may be "glued" in order to construct tilings between some coinitial and cofinal derivations ϕ and φ, in which case we write $\phi \equiv \varphi$ and say that ϕ and φ are *Lévy-permutation equivalent* [20,23,26,28]. For example, $s_1; r_1; r_2 \equiv r_3; s_2; r_2$, however $I\,(\underline{I\ 1}) \to_{\beta_{\text{db}}} I\ 1 \not\equiv \underline{I\,(I\ 1)} \to_{\beta_{\text{db}}} I\ 1$. Roughly speaking, $\phi \equiv \varphi$ means that ϕ and φ do the same work.

By comparing the relative positions of r and s in a basic tile [26,28] we can further classify tiles into disjoint and nested. The left tile in Fig. 2(b) is an example of the former and the right tile in the same figure an example of the latter. Furthermore, we can assign an orientation to these tiles as indicated in Fig. 3 and define standard derivations as an appropriate "minimal" derivation.

Definition 10 (*Reversible/irreversible tiles and permutations*).

- A *reversible tile* is a diagram of the form depicted in Fig. 2(a) such that r and s are disjoint. We denote such a tile as $r; s/r \diamond s; r/s$. An irreversible tile is a diagram of the form depicted in Fig. 2(a) such that s nests r. We denote such a tile as $r; s/r \triangleright s; r/s$.

- A *reversible permutation* ($\overset{\mathtt{r}}{\Rightarrow}$) is defined as $\phi_1; r; s/r; \phi_2 \overset{\mathtt{r}}{\Rightarrow} \phi_1; s; r/s; \phi_2$ where $r; s/r \diamond s; r/s$; an *irreversible* one ($\overset{\mathtt{i}}{\Rightarrow}$) as $\phi_1; r; s/r; \phi_2 \overset{\mathtt{i}}{\Rightarrow} \phi_1; s; r/s; \phi_2$ where $r; s/r \triangleright s; r/s$.

We write $\Rightarrow$ for the reflexive–transitive closure of $\overset{\mathtt{r}}{\Rightarrow} \cup \overset{\mathtt{i}}{\Rightarrow}$ and $\simeq$ for the least equivalence relation containing $\overset{\mathtt{r}}{\Rightarrow}$. The aforementioned notion of Lévy-permutation equivalence may now be formalised as follows.

Definition 11 (*Lévy-permutation equivalence*). Two coinitial and cofinal derivations ϕ, φ are *Lévy-permutation equivalent* if $\phi \equiv \varphi$, where $\equiv$ is the least equivalence relation containing $\Rightarrow$.

Finally, we may formulate a precise definition of standard derivations.

Definition 12 (*Standard derivation Melliès [26,28]*). A derivation ϕ is standard if it is minimal in the following sense: there is no sequence of the form $\phi = \phi_0 \overset{\mathtt{r}}{\Rightarrow} \cdots \overset{\mathtt{r}}{\Rightarrow} \phi_{k-1} \overset{\mathtt{i}}{\Rightarrow} \phi_k$, where $k \geqslant 1$.

For example $r_3; r_4$ is standard, but the following derivation $r_3; s_2; r_2$

$$\underline{K\ 1\ (I\,a)} \to_{r_3} (\lambda 2)\ \underline{(I\,a)} \to_{s_2} \underline{(\lambda 2)\ a} \to_{r_2} 1$$

is not since s_2 is nested by a redex (in this case, $\underline{(\lambda 2)\ (I\,a)}$) whose residual is immediately contracted in the next step, namely r_2.

Reformulated in this way by Melliès, the standardisation theorem proved for the λ-calculus [3,23] and term-rewriting systems [10,17] was extended in [26,28] to any left-linear pattern HORS.

Theorem 13. *(Standardisation Barendregt [3], Huet and Lévy [17], Lévy [23], Melliès [26,28])*.
(1) (*Existence*) *For any ϕ there exists a standard derivation φ s.t. $\phi \Rightarrow \varphi$.*
(2) (*Unicity*) *If $\varphi_1 \equiv \varphi_2$ and φ_1, φ_2 are standard derivations, then $\varphi_1 \simeq \varphi_2$.*

The first item states that every derivation may be transformed into a standard derivation (we say the derivation is "standardised") by oriented tiling. The second item states that standard derivations are unique up to disjoint tilings in Lévy-permutation equivalence classes. If we let $\mathtt{std}(\phi)$ stand for the (unique modulo $\simeq$) standard derivation in the $\equiv$-equivalence class of ϕ, then, for example, $\mathtt{std}(s_1; r_1; r_2) = \mathtt{std}(r_3; s_2; r_2) = r_3; r_4$ as illustrated in Fig. 2(b).

2.3. Neededness

Standard derivations are used to show that needed strategies are normalising in *orthogonal systems*. A rewrite system is orthogonal if any pair of coinitial redexes r, s do not overlap.

Definition 14 (*Needed redexes in orthogonal systems*). A redex r in M is needed if it has at least one residual in any coinitial derivation ϕ, unless ϕ contracts a residual of r [24].

For example, for the rewriting system $\{f(X_\varepsilon, b) \to_f c,\ a \to_a b\}$ the right occurrence of a in $f(a, a)$ is needed but not the left one since the derivation $f(a, a) \to_a f(a, b) \to_f c$ never reduces the left occurrence of a nor any of its residuals and no residuals of a are left in c. A needed rewrite strategy is one that only selects needed redexes. By defining a measure $|M|$ as "the length of the unique standard derivation (modulo $\simeq$) to M's normal form" it may be shown [17,26,28] that if $M \to_r N$ for some needed redex r, then $|M| > |N|$; hence needed strategies normalise in orthogonal rewrite systems. This measure is well-defined since in orthogonal systems any two coinitial derivations to (the unique) normal form may be tiled or, in other words, are Lévy-permutation equivalent [17,26,28].

In the case of non-orthogonal systems the notion of needed redex requires revision. Indeed, in $\mathcal{R} = \{a \to_{a_1} a,\ a \to_{a_2} b\}$ the derivation to normal form $\phi : a \to_{a_2} b$ leaves no residual of a_1. However, one cannot conclude that the a_1 redex is not needed since although it is not reduced in ϕ a redex which overlaps with this a_1 redex has. Thus the notion of needed redex is extended to needed derivations as follows.

Definition 15 (*Needed derivations in non-orthogonal systems Melliés [27]*). $\phi : M \twoheadrightarrow N$ is needed in a non-orthogonal rewrite system if for any term P and any derivation $\psi : N \twoheadrightarrow P$, $|\mathtt{std}(\phi; \psi)| > |\mathtt{std}(\psi)|$.

Note that now $a \to_{a_1} a$ is needed in the aforementioned example. The concept of needed redexes is extended to that of derivations since, in contrast to orthogonal systems, terms in non-orthogonal ones may not have needed redexes. For example, in $\{xor(true, X_\varepsilon) \to_L true$, $xor(X_\varepsilon, true) \to_R true$, $\Omega \to_\Omega true\}$ the term $xor(\Omega, \Omega)$ has no needed redexes [21].

Needed derivations get us "closer" to a normal form, however the aforementioned measure for orthogonal systems is no longer well-defined in the case of non-orthogonal systems: There may be two or more $\equiv$-distinct normalising derivations. Consider the following derivations:

$$\begin{aligned} \phi_1 &: a \to_{a_1} a \to_{a_2} b \\ \phi_2 &: a \to_{a_1} a \to_{a_1} a \to_{a_2} b \\ \phi_3 &: a \to_{a_1} a \to_{a_1} a \to_{a_1} a \to_{a_2} b. \\ &\cdots \end{aligned}$$

They are $\equiv$-distinct normalising derivations since each a_1-step creates a new copy of a. In [27] such badly-behaved systems are discarded by requiring the property of *finite normalisation cones* (*FNC*) to be fulfilled. A *normalisation cone* for a term M is a family $\{\psi_i : M \twoheadrightarrow N \mid i \in I_M\}$ of normalising derivations such that every normalising derivation $\phi : M \twoheadrightarrow N$ is Lévy-permutation equivalent to a unique derivation ψ_i (i.e. $\exists! i \in I_M$ s.t. $\phi \equiv \psi_i$). In fact, this definition specialises the definition of [27, Definition 4.8] since we make use of the fact that ES-based implementations of orthogonal HORS are confluent [9]. A rewrite system enjoys finite normalisation cones when there exists a *finite* normalisation cone for any term M.

Redefining the measure of a term $|M|$ to be "the length of the longest standard derivation in M's cone to M's normal form" allows one to show [27] that if $\phi : M \twoheadrightarrow N$ is a needed derivation, then $|M| > |N|$; hence needed strategies normalise in non-orthogonal rewrite systems satisfying *FNC*.

3. FNC for ES-based implementations of HORS

It is therefore of interest to identify conditions guaranteeing that ES-based implementations of HORS enjoy finite normalisation cones. In [27] the *FNC* property is shown for $\lambda\sigma$; this result relies on two observations:

(1) if ϕ is a standard derivation in $\lambda\sigma$ ending in a σ-normal form, then $\sigma(\phi)$ is standard in the λ-calculus, and
(2) needed strategies are normalising for the λ-calculus.

$\sigma(\phi)$ is obtained by mapping each $\lambda\sigma$ rewrite step in ϕ to its "corresponding" or "projected", if any, rewrite step in λ. We shall develop the required machinery in order to provide a formal definition of "projection" in Section 4.2 (cf. Definition 29). For now we illustrate it with an example: if $\phi : (1\,1)[(\lambda 1)\,2 \cdot id] \rightarrow_{Beta} (1\,1)[1[2 \cdot id] \cdot id]$, then $\sigma(\phi)$ takes the form

$$((\lambda 1)\,2)\,((\lambda 1)\,2) \rightarrow_{\beta_{\mathrm{db}}} 2\,((\lambda 1)\,2) \rightarrow_{\beta_{\mathrm{db}}} 2\,2.$$

In this paper we show that the *FNC* property holds for the ES-based implementation of *arbitrary* orthogonal $PERS_{\mathrm{db}}$ (recall from Section 2.1 that $PERS_{\mathrm{db}}$ are pattern ERS_{db}). This generalises the result which was originally verified for the λ-calculus. The proof follows the same lines as in [27] and is developed in Section 4; it relies on our meeting requirement (1), namely

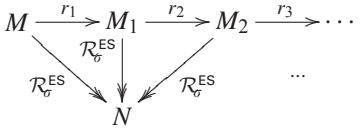
Proposition 16 (*Std-Projection Proposition*). *Let $\mathcal{R}$ be a left-linear $PERS_{\mathrm{db}}$. Every standard derivation $\phi : M \twoheadrightarrow N$ in $\mathcal{R}_\sigma^{\mathsf{ES}}$ with N in σ-normal form is projected onto a standard derivation $\sigma(\phi) : \sigma(M) \twoheadrightarrow N$ in $\mathcal{R}$. Besides, every $\mathcal{R}^{\mathsf{ES}}$ redex in ϕ is projected to a unique $\mathcal{R}$ redex in $\sigma(\phi)$.*

That *FNC* follows from the Std-Projection Proposition may be proved as follows:

Proposition 17. *The ES-based implementation of any orthogonal $PERS_{\mathrm{db}}$ $\mathcal{R}$ verifying the Std-Projection Proposition enjoys FNC: every closed $\mathcal{R}_\sigma^{\mathsf{ES}}$-term has FNC.*

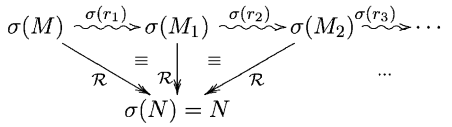
Proof. Suppose, on the contrary, that there exists a closed $\mathcal{R}_\sigma^{\mathsf{ES}}$ term with an infinite number of normalising $\mathcal{R}_\sigma^{\mathsf{ES}}$ derivations, modulo Lévy-permutation equivalence. We may construct an infinite tree whose nodes are the derivations $M \twoheadrightarrow N$ which may be extended to standard normalising derivations $M \twoheadrightarrow N \twoheadrightarrow P$, where nodes are ordered by the prefix ordering, and by König's Lemma (since every $\mathcal{R}_\sigma^{\mathsf{ES}}$ term contains only a finite number of redexes) deduce the existence of an infinite derivation ϕ_∞. Moreover, since σ is strongly normalising we know that ϕ_∞ has an infinite number of $\mathcal{R}^{\mathsf{ES}}$-steps.

Let ϕ_∞ be of the form $M \xrightarrow{r_1} M_1 \xrightarrow{r_2} M_2 \xrightarrow{r_3} \cdots$. Every finite prefix $\phi_i : M \twoheadrightarrow M_i$ of ϕ_∞ may be extended to a standard normalising path $\chi_i : M \twoheadrightarrow M_i \twoheadrightarrow N$.



And, by Proposition 16, each $\sigma(\chi_i) : \sigma(M) \twoheadrightarrow \sigma(M_i) \twoheadrightarrow \sigma(N) = N$ is a standard and normalising $\mathcal{R}$ derivation. Since $\mathcal{R}$ is orthogonal all the normalising derivations $\sigma(\chi_i) : \sigma(M) \twoheadrightarrow \sigma(N) = N$ must be Lévy-permutation equivalent. Thus we have: $\sigma(\chi_1) \equiv \sigma(\chi_2) \equiv \sigma(\chi_3) \equiv \sigma(\chi_4) \equiv \cdots$. And from Theorem 13(2) and the fact that $\phi \simeq \varphi$ implies $|\phi| = |\varphi|$ we deduce that: $|\sigma(\chi_1)| = |\sigma(\chi_2)| = |\sigma(\chi_3)| = |\sigma(\chi_4)| = \cdots$.

We reach a contradiction from the fact that there are an infinite number of $\mathcal{R}^{\mathsf{ES}}$ redexes in ϕ_∞ and that Proposition 16 projects $\mathcal{R}^{\mathsf{ES}}$ redexes to unique $\mathcal{R}$ redexes: For every $i > 0$ there is a $j > i$ such that $|\sigma(\chi_j)| > |\sigma(\chi_i)|$. See below, where the squiggly arrow labeled $\sigma(r_i)$ is the identity if r_i is a σ redex and is an $\mathcal{R}$ redex if r_i is an $\mathcal{R}^{\mathsf{ES}}$ redex (cf. Definition 29).



□

Remark 18. In [27] also the notion of leftmost-outermost computation of the λ-calculus is generalised to arbitrary left-linear pattern HORS: a derivation $\phi : M \to N$ is declared *external* when for every standard derivation $\chi : N \to P$, $\phi; \chi$ is a standard derivation too. It is shown that if $\phi : M \to N$ is an external rewriting derivation in $\lambda\sigma$, then $\sigma(\phi) : \sigma(M) \to \sigma(N)$ is an external rewriting derivation in the λ-calculus [27, Lemma 6.5].

4. The Std-Projection Proposition

We now concentrate on the proof of the Std-Projection Proposition which proceeds by contradiction and is developed in three stages. Before continuing however, we remark that it is non-trivial. In fact, in the general case in which N is not required to be a σ-normal form it fails. The following $\lambda\sigma$ derivation χ is standard:

$$((\lambda(11))1)[\underline{(\lambda 1)c} \cdot id] \to_{Beta} \underline{((\lambda(11))1)}[1[c \cdot id] \cdot id] \to_{Beta} (11)[1 \cdot id][1[c \cdot id] \cdot id].$$

However $\sigma(\chi)$ is not standard in the λ-calculus:

$$\sigma(\chi) : (\lambda(11))\underline{((\lambda 1)c)} \to_\beta \underline{(\lambda(11))c} \to_\beta cc.$$

The problem is that disjoint β redexes become nested after they are projected. We shall see that such a situation may not arise if χ is a standard derivation which ends at a σ-normal form.

Let χ be any standard $\mathcal{R}_\sigma^{\mathsf{ES}}$ derivation. The idea of the proof, inspired from [27], is to show that every reversible (resp. irreversible) permutation in the projection (cf. Definition 29) $\sigma(\chi)$ of χ may be mimicked by one or more reversible (resp. reversible permutations followed by one or more irreversible) permutations in χ, the ES-based derivation. Hence we may conclude by reasoning by contradiction. We prove in Stage 1 that the projection of an $\mathcal{R}^{\mathsf{ES}}$ step results in a unique $\mathcal{R}$ step, if χ ends in a σ-normal form. We prove then that every permutation in $\sigma(\chi)$ is mimicked by a series of permutations in χ as explained above. Reversible permutations are treated in Stage 2, and irreversible permutations in Stage 3.

4.1. Stage 1 (substitution zones)

First of all, note that χ consists of two kinds of rewrite steps: $\mathcal{R}^{\mathsf{ES}}$ steps and σ steps. We argue that it is not possible for a $\mathcal{R}^{\mathsf{ES}}$ step to take place inside a substitution if χ ends in a σ-normal form. The reason is that in that case the $\mathcal{R}^{\mathsf{ES}}$ redex would occur inside some term P in $P \cdot s$ and hence under the "$\cdot$" symbol. Since χ is standard, redexes reduced below a "$\cdot$" symbol cannot create redexes above it, and since N is a pure term we arrive at a contradiction. We formalise this argument below (Lemma 23).

Definition 19. Given an implementation $\mathcal{R}_\sigma^{\mathsf{ES}}$ of an ERS_{db} $\mathcal{R}$ with Γ the set of function and binder symbols, we define $g \in \Gamma$ of arity n as insulating in $\mathcal{R}_\sigma^{\mathsf{ES}}$ if

(1) either, g does not occur on the LHS of any rule in $\mathcal{R}_\sigma^{\mathsf{ES}}$,
(2) or, g occurs on the LHS of a rule in $\mathcal{R}_\sigma^{\mathsf{ES}}$ only under the form $g(X_1, \ldots, X_n)$ (i.e. it occurs applied to metavariables).

Example 20. The "λ" symbol is an example of an insulating symbol in the $\lambda\sigma$-calculus, i.e. in $(\beta_{\mathrm{db}})_\sigma^{\mathsf{ES}}$. Whereas, the application symbol is not insulating in $\lambda\sigma$ due to the *Beta*-rule. Also, for any $PERS_{\mathrm{db}}$ $\mathcal{R}$ the cons symbol "$\cdot$" is insulating in $\mathcal{R}_\sigma^{\mathsf{ES}}$ since it is only the rules of σ that govern the behaviour of "$\cdot$".

The notion of insulating symbol attempts to capture those symbols from which reduction below it cannot create/erase redexes above it. For example, in the rewrite step

$$M = g(h(c)[id]) \to_{Func_h} g(h(c[id])) = N, \tag{3}$$

where $\mathcal{R} = \{g(h(X)) \to c\}$, the redex v at the root position of N is created by rewriting at the position 1 in M, thus a rewrite step whose redex occurs below "g" in M has created a redex above it in N. Note that according to Definition 19 "g" is not insulating. Were we not to restrict our attention to left-linear rewrite systems other forms of redex creation different from the one discussed above (a redex contributing symbols to the pattern of the created redex) would also be present. Indeed, in left-linear systems redexes could be created simply by equating subterms as in the step $f(a, b) \to_a f(b, b)$, where $\mathcal{R} = \{f(X, X) \to_f a, a \to_a b\}$.

Remark 21. Although similar, the concept of insulating symbol does not coincide with that of a constructor symbol in a constructor Term Rewrite System [21]: given a constructor TRS there may be constructor symbols which are not insulating (for example, "s" in $f(s(s(x)))\to x$) and likewise there may be insulating symbols that are not constructor symbols (for example, "f" in $f(x)\to a$).

We say that a derivation $r_1;\ldots;r_n$ *preserves* a position p when the position of none of the redexes r_i is a prefix of p. Rephrasing in the terminology introduced in Section 2, a derivation $r_1;\ldots;r_n$ preserves a position p when none of the redexes r_i is above p.

Lemma 22. *Let $\mathcal{R}_\sigma^{\mathsf{ES}}$ implement $\mathcal{R}$. Suppose that a position p is strictly above a redex $P\to_r Q$. Every standard derivation $\phi = r;\psi$ preserves p when*
(1) *either, p is a g-node for g an insulating symbol in $\mathcal{R}_\sigma^{\mathsf{ES}}$,*
(2) *or, p is a g-node for g a function or binder symbol in $\mathcal{R}_\sigma^{\mathsf{ES}}$ and ψ is a σ derivation.*

Proof. Given the standard derivation $\phi = r;\psi$ and the position p strictly above r two cases may arise: either ϕ preserves p (in which case we are done) or otherwise ϕ may be reorganised modulo $\simeq$ into a derivation $\phi_1;u;v;\phi_2$ such that ϕ_1 preserves the position p, the position p is strictly above a redex u, and a redex v is above p. We shall see that the latter case results in a contradiction. Note that the derivation $u;v$ cannot be standard, unless u creates v. Now, in at least the following two cases creation is not possible:
(1) When p is the position of an insulating symbol in $\mathcal{R}_\sigma^{\mathsf{ES}}$. This follows from the fact that contraction of a redex below an insulating symbol may not create a redex above it.
(2) When the position p is a function or binder symbol node and u is a σ redex then an $\mathcal{R}^{\mathsf{ES}}$ redex must have been created, in other words, the only possible pattern of creation is when u is a $Func_f$ redex for some function symbol f or a $Bind_\xi$ redex for some binder symbol ξ and v is an $\mathcal{R}^{\mathsf{ES}}$ redex. See (3) for an example. Note that it is not possible for u to be an $\mathcal{R}_\sigma^{\mathsf{ES}}$ redex and v a σ redex since σ redexes above function or binder symbols cannot be created from below them.

Regarding the first item of Lemma 22 only the particular instance of it in which the g-node is a "cons" node "$\cdot$" shall be used in the sequel. Regarding the second, it shall be used as stated.

We say a redex $r : M\to N$ occurs in the left argument of a cons, if the position p of r in M is of the form $p = p_1.1.p_2$ with $M|_{p_1} = P\cdot s$ for some term P and substitution s. The following key lemma states that the left argument of a cons determines an "enclave" in a standard $\mathcal{R}_\sigma^{\mathsf{ES}}$ derivation to σ-normal form. □

Lemma 23. *Let $\mathcal{R}_\sigma^{\mathsf{ES}}$ implement $\mathcal{R}$ and let $\phi : M\twoheadrightarrow N$ be a standard $\mathcal{R}_\sigma^{\mathsf{ES}}$ derivation with N in σ-normal form. Then no $\mathcal{R}_\sigma^{\mathsf{ES}}$ redex ever occurs in the left argument of a cons.*

Proof. By contradiction. Suppose there exists an r_i with source O contracted in $\phi = r_1;\ldots;r_n$ inside the left argument P of a cons $P\cdot s$. Suppose, furthermore, that the position of the subterm $P\cdot s$ in O is p. Since the "$\cdot$" symbol is insulating, then by Lemma 22(1) the

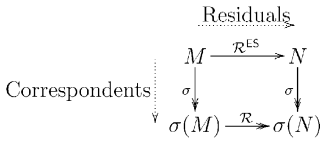


Fig. 4. Correspondence vs. residual.

derivation $r_i; \ldots; r_n$ preserves p. Since N is a pure term (i.e. has no explicit substitutions) we arrive at a contradiction. □

4.2. Correspondence: definition and properties

The next stage of the proof requires relating $\mathcal{R}^{\mathsf{ES}}$ steps in a $\mathcal{R}_\sigma^{\mathsf{ES}}$ derivation χ with their "corresponding" steps in the *projection* of χ, written $\sigma(\chi)$, via the substitution calculus σ (see Fig. 4). As mentioned in Section 1, and explained below in further detail, this notion differs from that of residuals as found in the rewriting literature and introduced in Section 2.2. Indeed, an attempt to define correspondents through the usual notion of residuals is doomed to fail. The problem is that $\mathcal{R}^{\mathsf{ES}}$ redexes may be lost when traversed by substitutions as illustrated in (1), in the introduction to this article. Therefore, an appropriate notion of correspondent and correspondence relation for tracing $\mathcal{R}^{\mathsf{ES}}$ redexes through σ derivations must be defined. We shall address this matter immediately before continuing with the next stage of the Std-Projection Proposition, namely Stage 2 (Section 4.3).

Remark 24. Note that we are interested in tracing only "what's left" of an $\mathcal{R}^{\mathsf{ES}}$ redex (and not arbitrary $\mathcal{R}_\sigma^{\mathsf{ES}}$ redexes) after σ-rewriting to σ-normal form.

The correspondence relation builds on the notion of *descendant*. If $M \to_s N$, then just as the residual relation $/s$ allows us to relate redexes in M with those in N after contracting s, the descendant relation $[\![s]\!]$ allows us to relate *positions* in M with those in N after contracting s. The concept of descendants is not novel (see [5] for references). In fact, in many presentations of rewrite systems the residual relation is defined by means of a descendant relation [10,17,19,33]: redexes are traced by keeping track of the position of their head symbol.

Call $p \in Pos(M)$ a *symbol position* if the head symbol of $M|_p$ is either a function or binder symbol and let us write $SPos(M)$ for the subset of $Pos(M)$ consisting of the symbol positions in M. A *descendant relation scheme for a rewrite rule* $L \to R$ is a binary relation $[\![L \to R]\!]$ that relates symbol positions in L with those in R and extends the relation between positions indicated by the metavariables: if p is the position of a metavariable X in L, then $p[\![L \to R]\!]q$ for all the positions q in R such that $R|_q = X$. This relation is extended to rewrite steps as expected. Consider the redex $r = (M, L \to R, \kappa, p)$ and the rewrite step $M = C[\kappa L] \to_r C[\kappa R] = N$. The induced descendant relation $[\![r]\!] \subseteq SPos(M) \times SPos(N)$

is defined as follows:

- If q is disjoint to p or $q < p$, then $q[\![r]\!]q$.
- If $q = p.q_1$, where q_1 is a non-metavariable position in L, then $q[\![r]\!]$ is the set of all positions $p.q_1'$ in N such that $q_1[\![L \to R]\!]q_1'$.
- If $q = p.q_1.q_2$, where q_1 is a metavariable position in L, then $q[\![r]\!]$ is the set of all positions $p.q_1'.q_2$ in N such that $q_1[\![L \to R]\!]q_1'$.

As in the case of the residual relation, the descendant relation may be extended to derivations by relation composition. Before stating the definition of correspondent we recall from Section 2 that two redexes r, s which are instances of the same rewrite rule are called *similar*.

Definition 25 (*Correspondent*). Consider an implementation $\mathcal{R}_\sigma^{\mathsf{ES}}$ of an ERS_{db} $\mathcal{R}$, where each σ-rewrite rule is equipped with a descendant relation scheme and an $\mathcal{R}^{\mathsf{ES}}$ redex r in M. Let $\phi : M \twoheadrightarrow_\sigma N$ and suppose that $p[\![\phi]\!]q$ where p is the position of r in M and $s = N|_q$ is a redex similar to r. We then say that s is a ϕ-correspondent of r in N and write $r[\![\phi]\!]s$.

In the sequel of this presentation we shall fix the descendant relation schemes for the rewrite rules of the σ-calculus which are described below. Note that only the ones for $Func_f$ for f a function symbol and $Bind_\xi$ for ξ a binder symbol need be specified. We recall from Section 2.1 the rewrite rules $Func_f$ and $Bind_\xi$

$$f(X_1, \ldots, X_n)[s] \to_{Func_f} f(X_1[s], \ldots, X_n[s]),$$
$$\xi(X_1, \ldots, X_n)[s] \to_{Bind_\xi} \xi(X_1[1 \cdot (s \circ \uparrow)], \ldots, X_n[1 \cdot (s \circ \uparrow)]).$$

The descendant relation scheme for $Func_f$ is

$$1[\![Func_f]\!]\varepsilon$$

$$1.i[\![Func_f]\!]i.1, \text{ for all } 1 \leqslant i \leqslant n.$$

The descendant relation scheme for $Bind_\xi$ is the same. As an example of the induced descendant relation on rewrite steps, in the derivation ϕ of (1) the *Beta* redex at position 1 in M has the *Beta* redex at position ε (the root position) as ϕ-correspondent.

These descendant relation schemes shall guarantee that two important properties of the tracing of $\mathcal{R}^{\mathsf{ES}}$ redexes through σ derivations to normal form are met: the structure of $\mathcal{R}^{\mathsf{ES}}$ redexes is preserved (Lemma 26) and the particular σ derivation used is unimportant (Lemma 27).

Let $\rightarrowtail_\sigma$ denote σ reduction to σ-normal form. The fact that the σ-calculus preserves the structure of $\mathcal{R}^{\mathsf{ES}}$ redexes is evidenced in the following result which is proved in the Appendix (Section B.4).

Lemma 26 (*Correspondence lemma*). *Suppose* $\phi : M \rightarrowtail_\sigma N$, *let* $p \in SPos(M)$ *be the position of a* $\mathcal{R}^{\mathsf{ES}}$ *redex* r, *and* $p[\![\phi]\!]q$. *Then* q *is the position of an* $\mathcal{R}^{\mathsf{ES}}$ *redex* s *similar to* r (*i.e.* $s[\![\phi]\!]r$). *Moreover, if* p *is not inside the body of a substitution, then* s *is the unique correspondent of* r.

A symbol position p in M is said to be "inside the body of a substitution" if there is a subterm of M of the form $P[s]$ at position q and $q.2 > p$ (cf. *substitution symbol positions* in Section B.2). A crucial building block on which the Correspondence Lemma is erected is the fact that the particular σ derivation chosen, called ϕ above, is of no relevance. More precisely, all σ derivations from a term M to its (unique) σ-normal form N induce the same descendent relation (and hence the same correspondence relation). This property is called *parametricity*.

Lemma 27 (*Parametricity*). *All* $\phi : M \rightarrowtail_\sigma N$ *induce the same descendant relation* $[\![\phi]\!]$ *over* $SPos(M) \times SPos(N)$.

The proof of this fact is developed in the style of [33, Proposition 3.2.14] and is relegated to the Appendix (Section B.1).

Remark 28. Parametricity fails due to "syntactic coincidences" [17] if all positions are traced (instead of just symbol positions) and arbitrary σ derivations considered (instead of σ derivations to σ-normal form). For example, consider the terms $M = 1[id][id]$ and $N = 1[id]$ and derivations $v : \underline{1[id][id]} \rightarrow_{Clos} 1[\underline{id \circ id}] \rightarrow_{IdL} 1[id]$ and $\phi : \underline{1[id][id]} \rightarrow_{VarId} 1[id]$. If we trace the subterm $\overline{1[id]}$ in M occurring at position 1 we obtain $1[\![v]\!]\varepsilon$ and $1[\![\phi]\!]1$, respectively.

We may now define formally the projection of a $\mathcal{R}^{\mathsf{ES}}_\sigma$ derivation.

Definition 29 (*Projection of* $\mathcal{R}^{\mathsf{ES}}_\sigma$ *derivations*). Let χ be a $\mathcal{R}^{\mathsf{ES}}_\sigma$ derivation. We define $\sigma(\chi)$ by induction on the length of χ:

$$\sigma(e_M) \stackrel{\text{def}}{=} e_{\sigma(M)} \qquad \sigma(u;\psi) \stackrel{\text{def}}{=} \begin{cases} \sigma(\psi) & \text{if } M \xrightarrow{u}_\sigma N, \\ v_1;\ldots;v_n;\sigma(\psi) & \text{if } M \xrightarrow{u}_{\mathcal{R}^{\mathsf{ES}}} N, \end{cases}$$

where $v_1;\ldots;v_n$ is a development in $\mathcal{R}$ of the set $u[\![\phi]\!]$ of ϕ-correspondents of u for any σ derivation ϕ from M to $\sigma(M)$.

In virtue of the parametricity property of σ, in this definition we may consider any σ derivation ϕ from M to $\sigma(M)$. Thus, Definition 29 does not depend on the σ derivation ϕ chosen. However, it does depend on the development of $u[\![\phi]\!]$ since different reductions are obtained for each such selection. However, all such reductions are $\simeq$-equivalent, and thus the projection may be seen to yield an $\simeq$-equivalence class of $\mathcal{R}$ derivations. The reason that all such reductions are $\simeq$-equivalent is that although σ may duplicate the correspondents of a $\mathcal{R}^{\mathsf{ES}}$ redex it may not nest them, therefore $u[\![\phi]\!]$ is a set of *disjoint* correspondents. [5] This fact is proved by introducing the following notion of σ-*disjoint* sets of positions and showing that σ descendants of σ-disjoint sets are, once again, σ-disjoint (see Lemma 36 in the Appendix).

[5] This is known as the *disjointness property* [3,20]. It has been considered in [20, Definition 4.3.1] where, following results due to Hyland, it shown to hold for β developments (it fails for full β reduction). See also [34].

Definition 30 (*σ-disjoint set*). Let $p, q \in SPos(M)$. We say p is σ-nested with q in M iff there exists some position o s.t. $p = o.1.p'$ and $q = o.2.q'$ and one of the following two conditions hold:
- either, the head symbol of $M|_o$ is $\bullet[\bullet]$,
- or, the head symbol of $M|_o$ is $\bullet \circ \bullet$.

A set S of disjoint symbol positions in M is said to be σ-disjoint if for every $p, q \in S$, p is not σ-nested with q.

Intuitively, a σ-disjoint set of positions captures "statically" a set of disjoint symbol positions which σ rewriting cannot nest. Indeed, note that two disjoint symbol positions risk getting nested through σ rewriting if one of the two conditions of Definition 30 hold. An example of the first condition is

$$M = f(a)[g(b) \cdot id] \to_{Func_f} f(a[g(b) \cdot id]) = N,$$

where the descendants of the disjoint symbol positions 1 and 2.1 in M get nested in N. An example of the second condition is

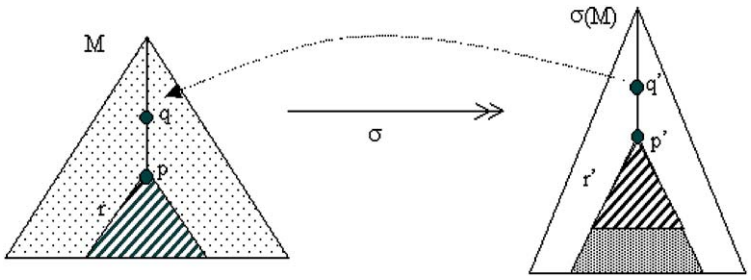
$$O = (f(a) \cdot id) \circ (g(b) \cdot id) \to_{Map} M \cdot (id \circ (g(b) \cdot id)) \to_{Func_f} N \cdot (id \circ (g(b) \cdot id)) = P,$$

where the descendants of the disjoint symbol positions 1.1 and 2.1 in O get nested in P. A formal proof of this observation if given in the Appendix (Proposition 41).

4.3. Stage 2 (reversible permutations)

From the analysis in Stage 1 we know that every $\mathcal{R}^{ES}$ redex contracted in χ does not occur inside the body of a substitution. As a consequence of Lemma 26 it has a unique correspondent $\mathcal{R}$ redex in $\sigma(\chi)$. This means that if $\sigma(\chi) = R_1; \ldots; R_o$, then there is a function $\rho : \{1, \ldots, o\} \to \{1, \ldots, n\}$ which associates to any $\mathcal{R}$ redex R_k in $\sigma(\chi)$ the unique $\mathcal{R}^{ES}$ redex $r_{\rho(k)}$ in $\chi = r_1; \ldots; r_n$ to which it corresponds.

Let R_k and R_{k+1} be two consecutive $\mathcal{R}$ redexes in $\sigma(\chi)$. Note that the $\mathcal{R}^{ES}_{\sigma}$ derivation $r_i; \ldots; r_j = r_{\rho(k)+1}; \ldots; r_{\rho(k+1)-1}$ between $r_{\rho(k)}$ and $r_{\rho(k+1)}$ contracts only σ redexes, as depicted below.



We now show that every reversible standardisation permutation $\sigma(\chi) \overset{r}{\Rightarrow} \omega$ in $\mathcal{R}$ may be mirrored as a non-empty series of reversible standardisation permutations $\chi \overset{r}{\Rightarrow} \cdots \overset{r}{\Rightarrow} \phi$ in

$\mathcal{R}_\sigma^{\mathsf{ES}}$, where $\sigma(\phi) = \omega$. It suffices to show that if R_k and R_{k+1} can be permuted using a reversible tile $(R_k; R_{k+1} \diamond R'_k; R'_{k+1})$, then a number of reversible permutations may be applied to $r_{\rho(k)}; r_i; \ldots; r_j; r_{\rho(k+1)}$ yielding ϕ s.t. $\sigma(\phi) = R'_k; R'_{k+1}$. Before proceeding a remark on the inverse of the descendent relation, also called the *ancestor* relation.

Remark 31 (*Ancestors of positions outside substitutions*). Let r be an $\mathcal{R}^{\mathsf{ES}}$ redex in M occurring at a position p not inside a substitution. As already noted, it has a unique corresponding $\mathcal{R}$ redex R in $\sigma(M)$ occurring at some position p'. Moreover, the following property on the ancestors of positions not inside substitutions holds: for every position q' in $\sigma(M)$ with $q' < p'$, we have $q < p$ where q is the (unique) ancestor of q' as illustrated below. The proof may be found in the Appendix (Lemma 43). The fact that q' is unique follows from the observation that σ may not create new function or binder symbols.

Suppose that the two $\mathcal{R}$ redexes R_k and R_{k+1} can be permuted using a reversible tile, that is, $R_k; R_{k+1} \diamond R'_k; R'_{k+1}$. We construct an $\mathcal{R}_\sigma^{\mathsf{ES}}$ derivation ϕ s.t. $\chi \simeq \phi$ and $\sigma(\phi) = R_1; \ldots; R'_k; R'_{k+1}; \ldots; R_p$.

By Lemma 22(2), the derivation $r_i; \ldots; r_j$ preserves the position of any function or binder symbol strictly above $r_{\rho(k)}$. And, in particular, the lowest symbol g appearing above $R_k : \sigma(P) \to \sigma(Q)$ and R'_k in the term $\sigma(P)$. By Remark 31, the ancestor of the symbol g is strictly above $r_{\rho(k)}$ in P. Note that, as discussed in Stage 1, g cannot be a "cons" symbol ".".

As a consequence the derivation $\psi = r_{\rho(k)}; r_i; \ldots; r_j; r_{\rho(k+1)}$ may be reorganised modulo $\simeq$ into a derivation ψ' such that $\sigma(\psi') = R'_k; R'_{k+1}$. In order to do so,

- let p be the position of this occurrence of g in P and let us assume that $P|_p = g(N_1, \ldots, N_m)$ and, moreover,
- suppose $r_{\rho(k)}$ occurs in N_{l_1} and the head symbol of $r_{\rho(k+1)}$ occurs in N_{l_2} for $l_1, l_2 \in 1..m$ and $l_1 \neq l_2$.

We may now reorganise ϕ as follows:

(1) First contract all the redexes in $r_i; \ldots; r_j$ prefixed by $p.l_2$,
(2) Second contract $r_{\rho(k+1)}$,
(3) Third contract the (unique) residual of $r_{\rho(k)}$,
(4) Finally contract the remaining redexes of $r_i; \ldots; r_j$, i.e. those prefixed by $p.1, \ldots, p.l_2 - 1, p.l_2 + 1, \ldots, p.m$.

4.4. Stage 3 (irreversible permutations)

Finally, we show that also irreversible standardisation permutations in $\mathcal{R}$ may be mimicked in the implementation: every irreversible standardisation permutation $\sigma(\chi) \overset{\mathtt{i}}{\Rightarrow} \omega$ in $\mathcal{R}$ may be mirrored as a non-empty series of standardisation permutations $\chi \overset{\mathtt{r}}{\Rightarrow} \cdots \overset{\mathtt{r}}{\Rightarrow} \phi' \overset{\mathtt{i}}{\Rightarrow} \cdots \overset{\mathtt{i}}{\Rightarrow} \phi$ with at least one irreversible permutation in $\mathcal{R}_\sigma^{\mathsf{ES}}$, where $\sigma(\phi) = \omega$.

Hence the proof of Proposition 16 follows by contradiction: indeed, every standardisation permutation acting on the projected higher-order rewrite derivation may be mimicked by projection-related standardisation permutations of the same nature (reversible/irreversible) over derivations in the implementation.

Suppose two $\mathcal{R}$ redexes $R_k : \sigma(P) \to \sigma(Q)$ and R_{k+1} can be permuted using an irreversible tile $R_k; R_{k+1} \triangleright R'_k; \psi$. Remark the following:

Remark 32. Let r be a redex in M at some position p, instance of a rule $L \to R$. Call the *pattern* of r the subterm of M at position p where the arguments of r (i.e. the terms substituted for the variables of L) are replaced by holes. Similarly to Remark 31, every symbol f in the pattern of R'_k strictly above R_k in $\sigma(P)$ has a unique ancestor in P, and this ancestor is above the occurrence of $r_{\rho(k)}$. Moreover, none of these symbols occurs embraced by a substitution operator. This follows from two facts:

(1) first, by Lemma 22, the derivation $r_i; \ldots; r_j$ preserves all these symbols (in particular the lowest one), and
(2) second, $r_{\rho(k+1)}$ is an $\mathcal{R}^{\mathsf{ES}}$ redex for $\mathcal{R}$ a $PERS_{\mathrm{db}}$ hence its LHS contains no occurrences of the substitution operator $\bullet[\bullet]$.

We now continue with Stage 3 of the proof. Let p be the occurrence of the unique σ ancestor of the head symbol g of R_{k+1} in P, and $P|_p = g(M_1, \ldots, M_m)$. Let $l \in 1..m$ such that $r_{\rho(k)}$ occurs in M_l. Notice that by Lemma 22(2) the σ derivation $r_i; \ldots; r_j$ preserves p and thus all these σ-redexes are either disjoint to p or in one of the M_is, $i \in 1..m$. We proceed to reorganise the derivation $\psi = r_{\rho(k)}; r_i; \ldots; r_j; r_{\rho(k+1)}$ as follows:

- first, we reorganise ψ modulo $\simeq$ by selecting to contract first all the σ-redexes in $r_i; \ldots; r_j$ which do not occur at the position of $r_{\rho(k)}$. In other words, we obtain

$$v = v_1; \ldots; v_m; v_{m+1}; r'_{\rho(k)}; v_{m+2}; r_{\rho(k+1)} \simeq r_{\rho(k)}; r_i; \ldots; r_j; r_{\rho(k+1)}$$

  such that
  - the derivation v_i, for $i \in 1..l-1$, is composed of all the redexes in $r_i; \ldots; r_j$ whose position is prefixed by $p.i$,
  - the derivation v_l is composed of all the redexes in $r_i; \ldots; r_j$ whose position is prefixed by $p.l$ but disjoint to the position of $r_{\rho(k)}$,
  - the derivation v_i, for $i \in l+1..m$, is composed of all the redexes in $r_i; \ldots; r_j$ whose position is prefixed by $p.i$,
  - the derivation v_{m+1} is composed of all the redexes in $r_i; \ldots; r_j$ whose position is disjoint to p,
  - $r'_{\rho(k)}$ is the unique residual of $r_{\rho(k)}$ after $v_1; \ldots; v_{m+1}$.

  By Remark 32 the redex $r_{\rho(k+1)}$ must have emerged in the target of v_{m+1}.
- Second, note that the σ-redexes remaining in $v_{m+2} = r_{k_1}; \ldots; r_{k_o}$ all occur at the position of $r'_{\rho(k)}$ or below. Thus we may apply $o+1$ irreversible permutations starting from v to obtain v', $v \underbrace{\overset{\mathtt{i}}{\Rightarrow} \cdots \overset{\mathtt{i}}{\Rightarrow}}_{o+1} v'$ where

$$v' = v_1; \ldots; v_{m+1}; r'_{\rho(k+1)}; \phi_{r'_{\rho(k)}}; \phi_{r_{k_1}}; \ldots; \phi_{r_{k_o}}.$$

Since $\sigma(v') = R'_k; \psi$, we set $\phi = v'$ and conclude.

This concludes the proof of Proposition 16. As a consequence we have:

Theorem 33. *Let $\mathcal{R}$ be any orthogonal pattern ERS_{db}. All needed derivations normalise in the ES-based implementation $\mathcal{R}_\sigma^{\mathsf{ES}}$ of $\mathcal{R}$.*

Remark 34. Although our interest is in normalisation we would like to point out that Proposition 16 may be seen as reducing standardisation for HORS to that of first-order systems. Given a derivation $\chi : M \twoheadrightarrow N$ in an orthogonal pattern ERS_{db} $\mathcal{R}$, we recast χ in the ES-based implementation of $\mathcal{R}$, then we standardise the resulting derivation in the first-order setting [10] and finally we project back to the higher-order setting. The resulting $\mathcal{R}$ derivation ϕ shall not be just any standard derivation from M to N, but also Lévy-permutation equivalent to χ, in other words, $\chi \equiv \phi$. This may be verified by proving that $\chi \Rightarrow \phi$ implies $\sigma(\chi) \equiv \sigma(\phi)$ (Lemma 45).

5. Conclusions

We have addressed normalisation by needed reduction in the ES-based approach to the implementation of HORS. Melliès [25] observed that the implementation of a higher-order rewrite system by means of calculi of explicit substitution may change its normalisation properties fundamentally; indeed, a term possessing no infinite derivations in the λ-calculus may lose this property when shifting to the $\lambda\sigma$-calculus. Based on an extension of the theory of needed redexes to overlapping systems [27] we have shown that all needed derivations normalise in the ES-based implementation of any orthogonal pattern HORS; the latter result has been established in the setting of the ERS_{db} formalism for higher-order rewriting. The key property that has been addressed in order to apply the aforementioned theory is to show that standard derivations in the ES-based implementation of a HORS project to standard derivations in the higher-order setting (Std-Projection Proposition).

In [9] the ES-based implementation of HORS does not fix a particular calculus of explicit substitutions. Instead a macro-based presentation encompassing a wide class of calculi of ES is used. The study of the abstract properties that make the proof of the Std-Projection Proposition go through would allow the results presented here to be made independent of σ, the calculus of ES which we have dealt with in this paper.

The fact that the redexes may be lost if one applies current literature on the theory of residuals as discussed in Section 4.2, seems to indicate that for the particular class of overlapping rewrite systems consisting of calculi of explicit substitution a more interesting theory can be developed. In such a theory a $\mathcal{R}^{\mathsf{ES}}$ redex which is suddenly polluted by a number of substitution operators should not disappear but rather change "status", for example it could become "passive" (since a "veil" has been drawn over it). This is certainly an improvement over it being lost! Of course, "passive" redexes could become "active" again (or "unveiled") by the works of the substitution calculus. A serious attempt at developing such a theory would require steering free of complications introduced by syntax. The author deems that an *axiomatic* approach in the sense of [26] or more recently [29] should be appropriate although this requires more detailed investigation.

Acknowledgements

To Delia Kesner, Alejandro Ríos and Paul-André Melliès for providing valuable suggestions. The comments by the anonymous referees greatly improved the presentation.

Appendix A. The `map`-example

The Haskell program

```
map (\x → (map id [ map id [true] ])) [ map id [true] ]
```

is represented as the ERS_{db} term

$$P = map(\underbrace{\lambda map(\lambda 1, \langle map(\lambda 1, \langle true\rangle)\rangle)}_{f}, \langle map(\lambda 1, \langle true\rangle)\rangle).$$

To make this section more readable we occasionally use $\langle \ldots \rangle$ for lists (for example, we abbreviate $cons(true, nil)$ by $\langle true\rangle$), and also applicative style notation (for example, we shall write $map\ id\ \langle true\rangle$ instead of $map(id, \langle true\rangle)$). We shall write I for $\lambda 1$.

Here we show how to reproduce Melliès' counter-example [25] in the modified setting of the map_σ^{ES} rewrite system of Example 7. The latter consists of the σ-calculus together with:

$$\begin{array}{ll} map(\xi X, nil) & \to_{ES(map.1)}\ nil, \\ map(\xi X, cons(Y, Z)) & \to_{ES(map.2)}\ cons(X[Y \cdot id], map(\xi X, Z)). \end{array}$$

Let us define

$$\begin{array}{rcl} s_1 & = & map\ I\ \langle true\rangle \cdot id, \\ rec(s) & = & \uparrow \circ\ (true[s] \cdot id), \\ s_{n+1} & = & rec(s_n), \\ C_x(y) & = & \uparrow \circ\ (true[y] \cdot x), \\ D_x(y) & = & cons(1[true[x] \cdot y], map\ I\ \langle\rangle) \cdot x. \end{array}$$

The reader may verify that the term P reduces to a term Q containing the subterm $1[map\ I\ \langle true\rangle \cdot id][map\ I\ \langle true\rangle \cdot id]$. We observe that

$$\begin{array}{l} 1[map\ I\ \langle true\rangle \cdot id][map\ I\ \langle true\rangle \cdot id] \\ \quad \to_{Clos} 1[(map\ I\ \langle true\rangle \cdot id) \circ \underbrace{(map\ I\ \langle true\rangle \cdot id)}_{s_1}] \\ \quad \to_{Map} 1[(map\ I\ \langle true\rangle)[s_1] \cdot (id \circ s_1)] \\ \quad \to_{IdL} 1[(map\ I\ \langle true\rangle)[s_1] \cdot s_1] \\ \quad \overset{+}{\to}_{\sigma}\ 1[(map\ (\lambda 1[1 \cdot (s_1 \circ \uparrow)])\ \langle true[s_1]\rangle) \cdot s_1] \\ \quad \to_{map.2^{\mathsf{ES}}} 1[cons(1[1 \cdot (s_1 \circ \uparrow)][true[s_1] \cdot id], map\ (\lambda 1[1 \cdot (s_1 \circ \uparrow)])\ \langle\rangle) \cdot s_1] \\ \quad \overset{+}{\to}_{\sigma}\ 1[cons(1[true[s_1] \cdot s_1 \circ (\uparrow \circ (true[s_1] \cdot id))], map\ I\ \langle\rangle) \cdot s_1] \\ \quad = 1[D_{s_1}(s_1 \circ rec(s_1))]. \end{array}$$

Note that from above $s_1 \circ s_1 \overset{+}{\to}_{map_\sigma^{\mathsf{ES}}} D_{s_1}(s_1 \circ s_2)$. Let C^n stand for C applied n times. The result is completed by verifying that the following statement, taken from [25] and instantiated to our current example where the arrows refer to map_σ^{ES} rewriting, holds

$$s_n \circ s_{n+1} \overset{+}{\to} C_{s_{n+1}}^{n-1}(D_{s_{n+1}}(s_{n+1} \circ s_{n+2})).$$

As a consequence there is an infinite computation in map_σ^{ES} of $s_1 \circ s_1$, and hence of P.

Appendix B. Proofs

B.1. Parametricity (Section 4.2)

Let $SPos(M, f)$ stand for the set of positions of a function or binder symbol f in the term M (this notion shall not be required for f a substitution symbol: "↑", "id", "∘" or "·").

Remark 35. If $p \in SPos(M, f)$ for some function or binder symbol f, $v : M \twoheadrightarrow_\sigma N$ and $p[\![v]\!]q$, then $q \in SPos(N, f)$. That is to say, σ descendants of function or binder symbols are once again function or binder symbols, and moreover, they have the same "name". This may be verified by inspecting the rewrite rules of the σ-calculus.

Recall from Section 4 that $\rightarrowtail_\sigma$ denotes σ reduction to σ-normal form. The proof of Parametricity (Lemma 27) follows.

Proof. We use a proof technique due to van Oostrom [33]. We must prove that if $v, \phi : M \rightarrowtail_\sigma N$ then $[\![v]\!] = [\![\phi]\!]$. Before proceeding two observations:

Observation 1. The σ-calculus does not create function or binder symbols, i.e. if $S \subseteq SPos(M)$ is the set of all positions in M of some (binder or function) symbol f and $M \overset{r}{\to}_\sigma N'$ then $S[\![r]\!]$ is the set of all symbol positions of f in N'.

Observation 2. If we replace some (function or binder) symbol f in M occurring at a position p with a fresh symbol g obtaining M', then the derivation v is transformed into a new σ derivation v', and

$$p[\![v]\!]q \Longleftrightarrow p[\![v']\!]q.$$

This may be verified by induction on the length of v.

Now let $p \in SPos(M, f)$ (i.e. p is a position of the symbol f in the term M) and let g be a fresh symbol. Then replacing f with g in M yields a term M' and two new σ derivations v' and ϕ' from M' to N_1 and N_2, respectively, such that

$$p[\![v]\!]q \Longleftrightarrow p[\![v']\!]q \quad \text{and} \quad p[\![\phi]\!]q \Longleftrightarrow p[\![\phi']\!]q. \tag{B.1}$$

Since g is a fresh symbol then

$$p[\![v']\!]q \Longleftrightarrow \text{the head symbol of } N_1|_q \text{ is } g. \tag{B.2}$$

The left-to-right direction of (B.2) follows from Remark 35. For the right-to-left direction of (B.2) is proved as follows: suppose the head symbol of $N_1|_q$ is g and that $p \in SPos(M, g)$ then by the first observation it follows that p must be an ancestor of q, that is, $p[\![v']\!]q$.

So, joining the equivalences (B.1) with the equivalence (B.2) we obtain:

$$p[\![v]\!]q \Longleftrightarrow p[\![v']\!]q \Longleftrightarrow \text{the head symbol of } N_1|_q \text{ is } g$$

and

$$p[\![\phi]\!]q \Longleftrightarrow p[\![\phi']\!]q \Longleftrightarrow \text{the head symbol of } N_2|_q \text{ is } g.$$

Finally, the result follows from noting that $N_1 = N_2$ from the confluence of the σ-calculus. □

B.2. On σ-disjoint sets (Section 4.2)

In this section, we show that the descendant of a σ-disjoint set is, once again, σ-disjoint.

Lemma 36 (*Descendants of a σ-disjoint set*). *Let M be a term and S be a σ-disjoint set in M. Suppose $\phi : M \twoheadrightarrow_\sigma N$. Let S' be the set of ϕ descendants of positions in S in N. Then S' is a σ-disjoint set in N, i.e. all ϕ descendants of positions in S are once again disjoint and, moreover, they are not σ-nested.*

Proof. Let S be a σ-disjoint set in M and suppose $M \xrightarrow{u}_\sigma N$ (the general result follows by induction on the length of the derivation). The proof is by induction on the (length of the) position where the rewrite step takes place. The interesting cases are a subset of those in which the rewrite step takes place at the root.

(1) $f(P_1, \ldots, P_n)[s] \to f(P_1[s], \ldots, P_n[s])$. If $S = \{p\}$ then the u descendants of p are a σ-disjoint set in N (note that if $p \in s$ then p shall have n descendants). Otherwise, consider any two positions $p, q \in S$ with $p \neq q$. Then
   - either, there are indices i, j with $1 \leqslant i, j \leqslant n$ such that $p \in P_i$ and $q \in P_j$,
   - or, $p, q \in s$.

   In both cases their u descendants are disjoint and not σ-nested. Note that it is not possible that $p \in P_i$ for some $1 \leqslant i \leqslant n$ and $q \in s$ since S is not σ-nested.

(2) $P[s][t] \to P[s \circ t]$. If $S = \{p\}$ then either $p \in P$, or $p \in s$ or $p \in t$. These cases are seen to hold. Otherwise, consider any two positions $p, q \in S$ with $p \neq q$. Then it must be that $p, q \in P$, or $p, q \in s$ or $p, q \in t$, in all cases S' is seen to be a σ-disjoint set in N.

(3) $1[id] \to 1$ or $\uparrow \circ id \to \uparrow$. These cases hold trivially since $S = \emptyset$ and there are no u descendants to consider.

(4) $1[P \cdot s] \to P$. If $S = \{p\}$ then either $p \in P$, or $p \in s$ and the result follows since p has at most one u descendant. Otherwise, consider any two positions $p, q \in S$ with $p \neq q$. Then either $p, q \in P$ or $p, q \in s$ or $p \in P$ and $q \in s$, and the result holds as above.

We say $p \in SPos(M)$ is a *substitution position* in M iff there is a prefix q of p such that the head symbol of $M|_q$ is either, a "$\circ$" or "$\cdot$"-symbol or, is the substitution operator ($\bullet[\bullet]$) and $q.2 \leqslant p$. A *non-substitution position* is a symbol position that is not a substitution position. □

Corollary 37 (*Descendants of non-substitution symbol positions*). *Let M be a term and $p \in SPos(M)$ a non-substitution symbol position in M. Suppose $\phi : M \twoheadrightarrow_\sigma N$. Then there is a unique $q \in SPos(N)$ such that $p[\![\phi]\!]q$ and, moreover, q is a non-substitution symbol position.*

Proof. Follows from the proof of Lemma 36 by taking $S = \{p\}$ and noting that duplication of p is not possible. □

B.3. σ-Nesting Captures Dynamic Nesting (Section 4.2)

In this section, we show that the notion of σ-nesting captures "completely" all situations in which the σ-calculus nests disjoint symbol positions.

A symbol position $p \in SPos(M)$ *dynamically nests* a symbol position $q \in SPos(M)$ in M if there is a σ derivation $\phi : M \twoheadrightarrow_\sigma N$ such that $p[\![\phi]\!]p'$, $q[\![\phi]\!]q'$ and $p' < q'$. We show that two disjoint symbol positions are dynamically nested if and only if they are σ-nested (Proposition 41). This is the main result of the section. Before doing so however, we prove some auxiliary results. The first two lemmas determine what the descendants of nested positions look like (Lemma 38) and what the descendants of σ-nested positions look like (Lemma 39), respectively. The final auxiliary result provides an "invariant" condition required for the proof of the main result of this section.

Lemma 38 (*One-step descendants of nested symbol positions*). *Let $p, q \in SPos(M)$ with $p < q$. Let $M \xrightarrow{u}_\sigma N$ and let $p[\![u]\!]p'$ and $q[\![u]\!]q'$ in N. Then either*
(1) *$p' < q'$ or,*
(2) *$\{p', q'\}$ is a σ-disjoint set in N.*

Proof. By induction on the (length of the) position where the rewrite step takes place. If the rewrite step is at an internal position, then the result is either obvious or follows from the induction hypothesis. Therefore, we concentrate on the case where the rewrite step is at the root position.

(1) $f(M_1, \ldots, M_n)[s] \to f(M_1[s], \ldots, M_n[s])$. If $p, q \in M_i$ for some $1 \leqslant i \leqslant n$ then p', q' are unique and $p' < q'$. If $p = 1$ and $q \in M_i$ with $1 \leqslant i \leqslant n$ then $p' = \varepsilon < q'$. Finally, if $p, q \in s$ then

$$\begin{array}{ll} p' < q' & \text{if } i < p' \text{ and } i < q' \text{ for some } 1 \leqslant i \leqslant n, \\ \{p', q'\}\ \sigma\text{-disjoint} & \text{otherwise.} \end{array}$$

(2) $\xi(M_1, \ldots, M_n)[s] \to \xi(P_1[1 \cdot (s \circ \uparrow)], \ldots, P_n[1 \cdot (s \circ \uparrow)])$, as in the previous case.
(3) $P[s][t] \to P[s \circ t]$. Then either $p, q \in P$, or $p, q \in s$, or $p, q \in t$. In each case p', q' are unique and $p' < q'$.
(4) $1[id] \to 1$ or $\uparrow \circ id \to \uparrow$. These cases hold trivially since there are no symbol positions in M.
(5) $1[P \cdot s] \to P$. Then $p, q \in P$ and p', q' are unique, and $p' < q'$.
(6) $id \circ s \to s$. Then $p, q \in s$ and p', q' are unique, and $p' < q'$.
(7) $\uparrow \circ (P \cdot s) \to s$. Analogous to the previous case.

(8) $(s_1 \circ s_2) \circ s_3 \to s_1 \circ (s_2 \circ s_3)$. Then either $p, q \in s_1$, or $p, q \in s_2$ or $p, q \in s_3$. In all cases p', q' are unique and $p' < q'$.
(9) $(P \cdot s) \circ t \to P[t] \cdot (s \circ t)$. If $p, q \in P$ or $p, q \in s$ then p', q' are unique and $p' < q'$. Otherwise $p, q \in t$ and we reason as follows:

$$\begin{array}{ll} p' < q' & \text{if } 1.2 < p' \text{ and } 1.2 < q', \\ p' < q' & \text{if } 2.2 < p' \text{ and } 2.2 < q', \\ \{p', q'\}\ \sigma\text{-disjoint} & \text{otherwise.} \end{array}$$

□

Lemma 39 (*One-step descendants of σ-nested symbol positions*). *Let $p, q \in SPos(M)$ with p σ-nested with q. Let $M \xrightarrow{u}_{\sigma} N$ and let $p[\![u]\!]p'$ and $q[\![u]\!]q'$ in N. Then one of the following holds*:
(1) *either*, $p' < q'$,
(2) *or*, $\{p', q'\}$ *is a σ-disjoint set in N*,
(3) *or*, p' *is σ-nested with* q' *in N*.

Proof. By induction on the (length of the) position where the rewrite step takes place.
- rewrite step at the root position.
  (1) $f(M_1, \ldots, M_n)[s] \to f(M_1[s], \ldots, M_n[s])$. Then
    - If $p, q \in M_i$ with $1 \leqslant i \leqslant n$ then p', q' are unique and p' is σ-nested with q'.
    - If $p, q \in s$ then one of the following cases holds:

$$\begin{array}{ll} p' \ \sigma\text{-nested with } q' & \text{if } i < p' \text{ and } i < q' \text{ for some } 1 \leqslant i \leqslant n, \\ \{p', q'\}\ \sigma\text{-disjoint} & \text{otherwise.} \end{array}$$

    - If $p = 1$ and $q \in s$ then $p' = \varepsilon < q'$ for all $q' \in q[\![u]\!]$.
    - If $p \in M_i$ for some $1 \leqslant i \leqslant n$ and $q \in s$ then one of the following cases holds:

$$\begin{array}{ll} p' \ \sigma\text{-nested with } q' & \text{if } i < q', \\ \{p', q'\}\ \sigma\text{-disjoint} & \text{otherwise.} \end{array}$$

  (2) $\xi(M_1, \ldots, M_n)[s] \to \xi(M_1[1 \cdot (s \circ \uparrow)], \ldots, M_n[1 \cdot (s \circ \uparrow)])$, as in the previous case.
  (3) $P[s][t] \to P[s \circ t]$. If $p, q \in P$, or $p, q \in s$, or $p, q \in t$ then p', q' are unique and p' is σ-nested with q'. If $p \in M$ and $q \in s$ or $q \in t$ then the same holds. If $p \in s$ and $q \in t$ then also p', q' are unique and p' is σ-nested with q'. No other cases are possible.
  (4) $1[id] \to 1$ or $\uparrow \circ id \to \uparrow$. These cases hold trivially since there are no symbol positions in M.
  (5) $1[P \cdot s] \to P$. Then $p, q \in P$ and p', q' are unique, and p' is σ-nested with q'.
  (6) $id \circ s \to s$ or $\uparrow \circ (P \cdot s) \to s$. Then $p, q \in s$ and p', q' are unique, and p' σ-nested with q'.
  (7) $(s_1 \circ s_2) \circ s_3 \to s_1 \circ (s_2 \circ s_3)$. Then p', q' are unique and p' is σ-nested with q'.
  (8) $(P \cdot s) \circ t \to P[t] \cdot (s \circ t)$. Then
    - If $p, q \in P$ or $p, q \in s$ then p', q' are unique and p' is σ-nested with q'.

- If $p \in P$ and $q \in t$ (the case $p \in s$ and $q \in t$ is analogous) then we reason as follows:

$$\begin{array}{ll} p' \ \sigma\text{-nested with } q' & \text{if } 1.2 < q', \\ \{p', q'\} \ \sigma\text{-disjoint} & \text{otherwise.} \end{array}$$

- If $p, q \in t$ then we reason as follows:

$$\begin{array}{ll} p' \ \sigma\text{-nested with } q' & \text{if } 1.2 < p' \text{ and } 1.2 < q', \\ p' \ \sigma\text{-nested with } q' & \text{if } 2.2 < p' \text{ and } 2.2 < q', \\ \{p', q'\} \ \sigma\text{-disjoint} & \text{otherwise.} \end{array}$$

- rewrite step at an internal position.
  (1) $M = f(M_1, \ldots, M_n)$ and $M_i \xrightarrow{u}_{\sigma} M_i'$ for some $1 \leqslant i \leqslant n$. Then either $p, q \in M_j$ with $j \neq i$ and the result is direct, or $p, q \in M_i$ and we may use the induction hypothesis.
  (2) $M = \xi(M_1, \ldots, M_n)$, as in the previous case.
  (3) $M = P[s]$. Then if $p, q \in P$ or $p, q \in s$ we conclude directly or we apply the induction hypothesis as above. Otherwise $p \in P$ and $q \in s$ in which case p' is σ-nested with q' for all $p' \in p[\![u]\!]$ and $q' \in q[\![u]\!]$.
  (4) $M = P \cdot s$. We use the induction hypothesis.
  (5) $M = s \circ t$. Analogous to the case $M = P[s]$. □

Lemma 40. *Let $p, q \in SPos(M)$. If p is σ-nested with q in M then there exists a rewrite step $M \xrightarrow{u}_{\sigma} N$ and positions $p', q' \in SPos(N)$ such that $p[\![u]\!]p'$, $q[\![u]\!]q'$ and,*
(1) *either, $p' < q'$,*
(2) *or, p' is σ-nested with q' in N.*

Proof. We proceed by case analysis. By hypothesis $p = o.1.p_1$ and $q = o.2.q_1$ and we have two cases to consider:

- $M|_o = M_1[s]$. We shall consider all possible cases for M_1.
  (1) $M_1 = M_2[s']$. Then $M|_o \rightarrow_{Clos} M_2[s' \circ s]$ and the unique descendants of p and q are σ-nested in N.
  (2) $M_1 = f(M_1, \ldots, M_n)$ (the case $M_1 = \xi(M_1, \ldots, M_n)$ is similar). Then $M|_o \rightarrow_{Func_f} f(M_1[s], \ldots, M_n[s])$. If p is in M_i then we may choose q' such that the result holds, namely the descendant that lies in the copy of s embracing M_i. If $p = \varepsilon$ then for any q' such that $q[\![u]\!]q'$ we have $p' < q'$.

  Note that there are no further cases to consider for M_1.
- $M|_o = s_1 \circ s_2$. We consider all possible cases for s_1.
  (1) $s_1 = s_3 \circ s_4$. Then $M|_o \rightarrow_{Ass} s_3 \circ (s_4 \circ s_2)$ and the unique descendants of p and q are σ-nested in N.
  (2) $s_1 = M_1 \cdot s_3$. Then $M|_o \rightarrow_{Map} M_1[s_2] \cdot (s_3 \circ s_2)$. Then p shall have a unique descendant. As for q' we select the occurrence which is more convenient.

  Note that there are no further cases to consider for s_1. □

Proposition 41. *Let $\mathcal{R}_\sigma^{\mathsf{ES}}$ be an implementation of an ERS$_{\mathrm{db}}$ $\mathcal{R}$. Let $p, q \in SPos(M)$*

p dynamically nests q in M iff p is σ-nested with q in M.

Proof. ($\Rightarrow$) Let p and q be disjoint symbol positions in M and suppose there is a σ derivation $\phi : M \twoheadrightarrow_\sigma N$ such that $p[\![\phi]\!]p'$, $q[\![\phi]\!]q'$ and $p' < q'$. Then by Lemma 36 $\{p, q\}$ is not a σ-disjoint set. Therefore, one of two situations arises: either, p is σ-nested with q in M, or q is σ-nested with p in M. By Lemma 39 (as extended to the many step rewriting relation) and the fact that $p' < q'$, it is not possible for q to be σ-nested with p in M. Hence we conclude.

($\Leftarrow$) Suppose p is σ-nested with q in M. We show that there exists a derivation $\phi : M \twoheadrightarrow_\sigma N$ such that $p[\![\phi]\!]p'$, $q[\![\phi]\!]q'$ and $p' < q'$. We use induction on $\xrightarrow{+}_\sigma$ since σ is strongly normalising [11,32,36]. Let $M \xrightarrow{u}_\sigma M'$ by applying Lemma 40. Then $p[\![u]\!]p''$, $q[\![u]\!]q''$, and

(1) either, $p'' < q''$ and by taking $p' = p''$ and $q' = q''$ we are done,
(2) or, p'' σ-nested with q''. Then we apply the induction hypothesis. □

B.4. Correspondence lemma (Section 4.2)

If $p = p'.i$ is a position in M where i is some natural number, then the *predecessor* of p is given by $pred(p, M) \stackrel{\text{def}}{=} p'$. A *rod* in M is a pair (p, q) s.t.

- $p, q \in SPos(M)$,
- $p < q$, and
- for every $o \in Pos(M)$ such that $p < o < q$, either $o \in SPos(M)$, or $M|_o = M'[s]$ and $o.1 \leqslant q$.

We write $p <_S q$ for the rod (p, q). For example, in the term

$$(3\,4)[(\lambda\lambda 1 \cdot id) \circ (\lambda 2 \cdot \uparrow)])\,(\lambda 1)$$

we have $\varepsilon <_S 1.1$ and $1.2.1.1 <_S 1.2.1.1.1$, but $\varepsilon <_S 1.2$ is not a rod.

The intuitive idea is that σ rewriting preserves the structure of $\mathcal{R}^{\mathsf{ES}}$ redexes. Since the patterns of these redexes may be described by rods, a basic result required for the Correspondence Lemma (Lemma 26) is that σ rewriting preserves rods. This is precisely the statement of the following lemma.

Lemma 42 (*Rod preservation*). *Let $p, q \in SPos(M)$ and $M \xrightarrow{u}_\sigma N$. Suppose $p <_S q$ and $p[\![u]\!]p'$.*

(1) *Then there is a unique $q' \in SPos(N)$ such that $q[\![u]\!]q'$ and $p' <_S q'$.*
(2) *Moreover, if $p.i \leqslant q$ then $p'.i \leqslant q'$.*

Proof. By induction on the (length of) the position where the u-rewrite step takes place.

- The rewrite step takes place at the root. We consider the following subcases:
  (1) $f(M_1, \ldots, M_n)[s] \to f(M_1[s], \ldots, M_n[s])$. If $p, q \in M_i$ for some $1 \leqslant i \leqslant n$ then p', q' are unique and $p' <_S q'$. If $p = 1$ and $q \in M_i$ with $1 \leqslant i \leqslant n$ then $p' = \varepsilon <_S q'$. Finally, if $p, q \in s$ then suppose $i < p'$ for some $1 \leqslant i \leqslant n$. Take the unique q' such that $i < q'$. Then $p' <_S q'$.
  (2) $\xi(M_1, \ldots, M_n)[s] \to \xi(M_1[1 \cdot (s \circ \uparrow)], \ldots, M_n[1 \cdot (s \circ \uparrow)])$. As in the previous case.

(3) $P[s][t] \to P[s \circ t]$. Then either $p, q \in P$, or $p, q \in s$, or $p, q \in t$. In each case p', q' are unique and $p' <_S q'$.
(4) $1[id] \to 1$ or $\uparrow \circ id \to \uparrow$. These cases hold trivially since there are no symbol positions in M.
(5) $1[P \cdot s] \to P$. Then $p, q \in P$ and p', q' are unique, and $p' <_S q'$.
(6) $id \circ s \to s$ or $\uparrow \circ (P \cdot s) \to s$. Then $p, q \in s$ and p', q' are unique, and $p' <_S q'$.
(7) $(s_1 \circ s_2) \circ s_3 \to s_1 \circ (s_2 \circ s_3)$. Then either $p, q \in s_1$, or $p, q \in s_2$ or $p, q \in s_3$. In all cases p', q' are unique and $p' <_S q'$.
(8) $(P \cdot s) \circ t \to P[t] \cdot (s \circ t)$. If $p, q \in P$ or $p, q \in s$ then p', q' are unique and $p' <_S q'$. Otherwise $p, q \in t$ and suppose $1.2 < p'$ (the case $2.2 < p'$ is handled similarly). Then take the unique q' such that $p' <_S q'$.

- rewrite-step at an internal position.
(1) $M = f(M_1, \ldots, M_n)$ and $M_i \xrightarrow{u}_\sigma M_i'$ for some $1 \leqslant i \leqslant n$. If $p, q \in M_j$ with $i \neq j$ then the result is direct, if $i = j$ then we use the induction hypothesis. Otherwise $p = \varepsilon$. If $q \in M_j$ with $i \neq j$ the result is direct. Suppose therefore that $q \in M_i$ hence $q = i.o$. Note that o is a non-substitution symbol position in M_i since $p <_S q$. By Corollary 37 there is a unique o' u descendant of o and moreover o' is a non-substitution symbol position. Then $p' = \varepsilon <_S q' = i.o'$.
(2) $M = \xi(M_1, \ldots, M_n)$. As in the previous case.
(3) $M = P[s]$, $M = P \cdot s$ or $M = s \circ t$. Then either $p, q \in P$ or $p, q \in s$ or $p, q \in t$ and the result is direct or follows from the induction hypothesis. □

Lemma 26 (*Correspondence lemma*). *Let $\mathcal{R}_\sigma^{\mathsf{ES}}$ be an implementation of an ERS_{db} $\mathcal{R}$. Suppose $\phi : M \rightarrowtail_\sigma N$, let $p \in SPos(M)$ be the position of an (L, R) redex r with $(L, R) \in \mathcal{R}^{\mathsf{ES}}$, and $p[\![\phi]\!]p'$. Then p' is the position of an (L, R) redex r' in N, i.e. $r[\![\phi]\!]r'$. Moreover, if p is not inside the body of a substitution, then r' is the unique correspondent of r.*

Proof. We shall show that the pattern of r descends to $N|_{p'}$. Since the LHS of a $\mathcal{R}^{\mathsf{ES}}$ rewrite rule does not contain substitution symbols, it suffices to show that for each pair of positions $o, pred(o, M) \in SPos(M)$ in the pattern of r (thus $p \leqslant pred(o, M) < o$) we have

$$pred(o, M)[\![\phi]\!]q' \text{ implies } q' = pred(o', N), \tag{B.3}$$

where by Lemma 42 q' is the unique descendent of $pred(o, M)$ below p', and o' is the unique descendent of o strictly below p'. This situation is summarised in Fig. B.1.

This suffices since we may then traverse the pattern of r in a breadth-first manner and reconstruct, with the aid Remark 35, the pattern in N, hence r' shall be an instance of the same rewrite rule as r.

In order to prove (B.3) let us proceed by contradiction. By Lemma 42 on rod preservation we know that $q' <_S o'$, and hence that $q' < o'$. Suppose, moreover, that there is a position q'' in N such that $q' < q'' < o'$. This is illustrated in Fig. B.2.

Since N is a σ-normal form then N is a pure term and hence $q'' \in SPos(N)$. Now since function and binder symbols may not be created by σ there is a (unique) position $q \in SPos(M)$ such that $q[\![\phi]\!]q''$. By case analysis we shall see that this is not possible.

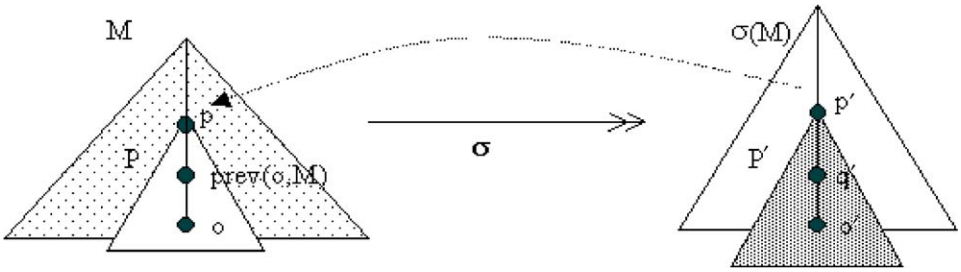


Fig. B.1. Pattern descendence.

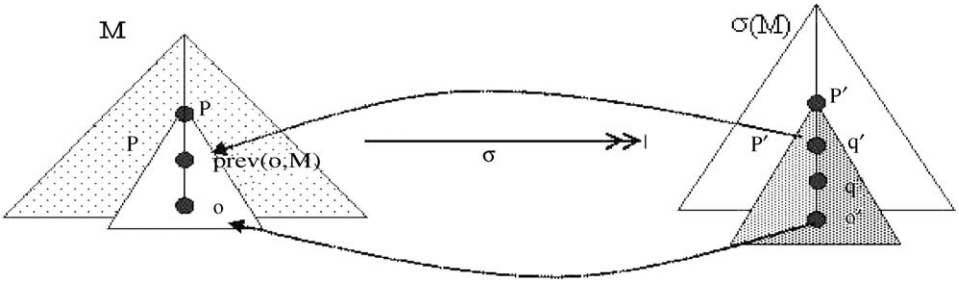


Fig. B.2. Pattern descendence. Impossible situation.

(1) $o < q$. Impossible since by Lemma 38 we would have $o' < q''$ or $\{o', q''\}$ σ-disjoint (hence $o' \| q''$), both of which contradict our knowledge that $q'' < o'$.
(2) $q < o$. Note that $q \neq pred(o, M)$ by the disjointness property. So then $q < pred(o, M)$ and by Lemma 38 we would have $q'' < q'$ or $\{q'', q'\}$ σ-disjoint (hence $q'' \| q'$), which is not possible since $q' < q''$.
(3) $q \| o$. First observe the following fact which is a direct consequence of the definition of σ-nested symbol positions (Definition 30): If $p, q \in SPos(M)$, p σ-nested with q and $pred(q, M) \in SPos(M)$, then p σ-nested with $pred(q, M)$. Second, by Proposition 41 q is σ-nested with o and by the first observation q is σ-nested with $pred(o, M)$. Then by Lemma 39 three situations may arise:
   (a) either, $q'' < q'$, which is not possible.
   (b) or, q'' σ-nested with q'. Then by definition of σ-nested q' must be a substitution position and this together with the fact that N is a pure term yields a contradiction.
   (c) or, $\{q'', q'\}$ σ-disjoint in N, hence $q'' \| q'$ and we arrive at a contradiction.

The assertion that if p is not inside the body of a substitution, then s is the unique correspondent of r, follows from Corollary 37. □

B.5. Ancestors of non-substitution positions (Section 4.3)

We recall from Section 2 that if s, r are redexes with source M, then we say *s nests r*, written $s < r$, if the position of s is a prefix of the position of r in M; if neither s nests r nor r nests s, then they are said to be *disjoint* in M and we write $s \| r$.

Lemma 43 (*Ancestor of non-substitution position*). *Let* $p, q \in SPos(M)$ *with* p *a non-substitution symbol position, suppose* $M \xrightarrow{u}_{\mathcal{R}_\sigma^{\mathsf{ES}}} N$, *and* $p[\![u]\!]p'$, $q[\![u]\!]q'$, *and* $q' < p'$. *Then* $q < p$.

Proof. Note that the condition that p be a non-substitution symbol is required as the following example illustrates: $(\lambda P)[\lambda 1 \cdot id] \to_{Lam} \lambda(P[1 \cdot ((\lambda 1 \cdot id) \circ \uparrow)])$, where $p = 2.1$ and $q = 1$ (hence $q' = \varepsilon$ and $p' = 1.2.2.1.1$).

Let us verify that $q < p$ is the only possible situation. Note that by Lemma 36 (descendants of a σ-disjoint set) it must be the case that $p \neq q$. So three further situations are possible.

- $p \| q$. Let us consider three further subcases. Since they are exhaustive and none of them are possible we arrive at a contradiction.
  (1) $\{p, q\}$ is a σ-disjoint set in M. Then by Lemma 36 we must have that $\{p', q'\}$ is a σ-disjoint set in N, which is not possible since $q' < p'$.
  (2) p is σ-nested with q in M. We reach a contradiction by Lemma 39.
  (3) q is σ-nested with p in M. We reach a contradiction since if q is σ-nested with p in M then p must be a substitution position.
- $p < q$. By Lemma 38 it follows that either $p' < q'$ or $\{p', q'\}$ is a σ-disjoint set in N. Neither case is possible since $q' < p'$.
- $q < p$. By Lemma 38 it follows that $q' < p'$. Note that $\{p', q'\}$ σ-disjoint in N is not possible.

Note that the condition that p be a non-substitution symbol is required as the following example illustrates: $(\lambda P)[\lambda 1 \cdot id] \to_{Lam} \lambda(P[1 \cdot ((\lambda 1 \cdot id) \circ \uparrow)])$, where $p = 2.1$ and $q = 1$ (hence $q' = \varepsilon$ and $p' = 1.2.2.1.1$).

This result may be extended to finite derivations in $\mathcal{R}_\sigma^{\mathsf{ES}}$ by noting that the unique σ descendant of a non-substitution symbol position is once again a non-substitution symbol position. □

B.6. Lévy-equivalent derivations project to Lévy-equivalent derivations (Section 4.4)

We recall from Section 2 that if s, r are redexes with source M, then we say s *nests* r, written $s < r$, if the position of s is a prefix of the position of r in M; if neither s nests r nor r nests s, then they are said to be *disjoint* in M and we write $s \| r$. Before proving the main result, we prove that non-overlapping redexes have non-overlapping correspondents.

Lemma 44 (*Correspondents of σ-nested and nested redexes*). *Let* $\mathcal{R}_\sigma^{\mathsf{ES}}$ *be an implementation of an* ERS_{db} $\mathcal{R}$ *and* r, u *be* $\mathcal{R}^{\mathsf{ES}}$ *redexes in* M. *Let* $\phi : M \rightarrowtail_\sigma N$, $r[\![\phi]\!]r'$, $u[\![\phi]\!]u'$. *If* r *is* σ*-nested with* u *in* M *or* r *nests* u *in* M:
(1) *either*, r' *nests* u',
(2) *or*, r' *is disjoint with* u'.

Proof. We shall consider the case in which r is σ-nested with u in M; the other one may be dealt with similarly.

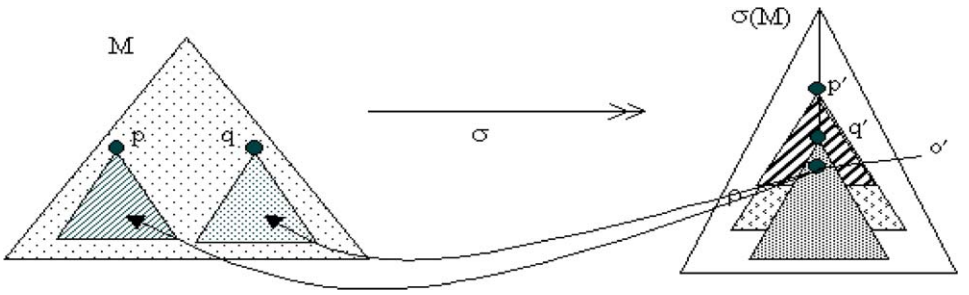


Fig. B.3. Non-overlapping redexes may not have overlapping correspondents.

Let p (resp. q) be the position of the head symbol of r (resp. u) in M. Suppose $p[\![\phi]\!]p'$ and $q[\![\phi]\!]q'$. By Lemma 39 (as extended to the many step rewriting relation) either $p' < q'$, or $\{p', q'\}$ is a σ-disjoint set in N, or p' is σ-nested with q' in N. Since N is a σ-normal form, the last case is not possible. If $\{p', q'\}$ is a σ-disjoint set in N, then r' and u' are disjoint and we are done. Otherwise, suppose $p' < q'$. We are left to verify that r' and u' do not overlap.

Suppose that u' overlaps r', i.e., there is a position o' in the pattern of r' such that o' is also a position in the pattern of u'. Then we have $p' < q' \leqslant o'$, as Fig. B.3 illustrates. Since $r[\![\phi]\!]r'$ there must be a position o_1 in the pattern of r such that $o_1[\![\phi]\!]o'$. Likewise, there must be a position o_2 in the pattern of u such that $o_2[\![\phi]\!]o'$. But this contradicts the fact that the ancestor relation on function and binder symbols induced by σ rewriting is a (partial) function. Therefore, r' nests u'. □

Lemma 45. *Let χ and ϕ be $\mathcal{R}_\sigma^{\mathsf{ES}}$ derivations. If $\chi \Rightarrow \phi$ then $\sigma(\chi) \equiv \sigma(\phi)$.*

Proof. Let $\chi : M \xrightarrow{r} N_1 \xrightarrow{u'} N$ for $\{r, u\}$ $\mathcal{R}_\sigma^{\mathsf{ES}}$ redexes in M such that r does not nest u, and u' is the (unique) r residual of u. It suffices to show that the claim holds for the following two cases:

- *Case* 1: if $\chi \diamond \phi$ for $\phi = u; r'$, then $\sigma(\chi) \equiv \sigma(\phi)$, and
- *Case* 2: if $\chi \triangleright \phi$ for $\phi = u; \phi'$, then $\sigma(\chi) \equiv \sigma(\phi)$.

Our analysis depends on whether r and u are σ or $\mathcal{R}^{\mathsf{ES}}$ redexes in M and shall distinguish cases 1 and 2 as needed.

- In either case, if r and u are σ redexes then the result holds trivially by completeness of σ and reflexiveness of $\equiv$: $e_{\sigma(M)} \equiv e_{\sigma(M)}$.
- Suppose u is a σ redex and r a $\mathcal{R}^{\mathsf{ES}}$ redex (the viceversa case is analogous). Then
  $$\sigma(\chi) : \sigma(M) \twoheadrightarrow_{\mathcal{R}} \sigma(N_1) = \sigma(N) \quad \text{and} \quad \sigma(\phi) : \sigma(M) = \sigma(N_2) \twoheadrightarrow_{\mathcal{R}} \sigma(N').$$
  By parametricity of σ the correspondents of r in $\sigma(N_2)$ are the same as those in $\sigma(M)$. Furthermore, the Correspondence lemma together with Lemma 36 on descendants of a σ-disjoint set of symbol positions determine that any two developments of the correspondents of r and r', respectively, shall yield equivalent derivations modulo $\simeq$.

- Suppose both u and r are $\mathcal{R}^{\mathsf{ES}}$ redexes in M. Here we distinguish the two subcases:
  - Reversible permutation (Case 1). Suppose $\{r, u\}$ are disjoint redexes in M. Then if the correspondents of $\{r, u\}$ via σ are disjoint in $\sigma(M)$ we may simply develop them and obtain equivalent derivations modulo $\simeq$, as above. Otherwise, by Lemma 36 on descendants of a σ-disjoint set of positions we may assume that r is σ-nested with u.

    Let $S = \{r_1, \ldots, r_n\}$ be the set of (pairwise disjoint by Lemma 36) correspondents of r in $\sigma(M)$. Then by Lemma 44 each correspondent u' of u is either disjoint with all redexes in S or is nested by some (one) redex in S. Finally, note that the set of correspondents of u in $\sigma(M)$ are pairwise disjoint too by Lemma 36.

    Thus we may construct the standardisation $\sigma(\chi) \Leftarrow \sigma(\phi)$, where $\sigma(\chi)$ rewrites all r_is in some order and then rewrites all the correspondents of u in some order, and $\sigma(\phi)$ rewrites all correspondents of u in some order, and then all the (unique) correspondents of the r_is in some order.
  - Irreversible permutation (Case 2). Suppose u nests r. Let $S = \{u_1, \ldots, u_n\}$ be the set of (disjoint by Lemma 36) correspondents of u in $\sigma(M)$. Then by Lemma 44 each correspondent of r in $\sigma(M)$ is either disjoint with all redexes in S or is nested by some (one) redex in S. Finally, note that the set of correspondents of r in $\sigma(M)$ is pairwise disjoint too.

    Thus we may construct the standardisation $\sigma(\chi) \Rightarrow \sigma(\phi)$, where $\sigma(\chi)$ rewrites all the correspondents of r in some order and then rewrites all the (unique correspondents of the) u_is in some order, and $\sigma(\phi)$ rewrites all the u_is in some order and then all the (correspondents of) r in some order. □

References

[1] M. Abadi, L. Cardelli, P.-L. Curien, J.-J. Lévy, Explicit substitutions, J. Funct. Programming 4 (1) (1991) 375–416.

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

[3] H.P. Barendregt, The Lambda Calculus: its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics, Vol. 103, North-Holland, Amsterdam, revised edition, 1984.

[4] Z. Benaissa, D. Briaud, P. Lescanne, J. Rouyer-Degli, λv, a calculus of explicit substitutions which preserves strong normalisation, J. Funct. Programming 6 (5) (1996) 699–722.

[5] I. Bethke, J.W. Klop, R. de Vrijer, Descendents and origins in term rewriting, Inform. and Comput. 159 (1–2) (2000) 59–124.

[6] E. Bonelli, Term rewriting and explicit substitutions, Ph.D. Thesis, Université de Paris Sud, November 2001.

[7] E. Bonelli, Normalisation for higher-order calculi with explicit substitutions, in: Proceedings of FOSSACS'03, Lecture Notes in Computer Science, Warsaw, Poland, Springer, Berlin, 2003.

[8] E. Bonelli, D. Kesner, A. Ríos, A de Bruijn notation for higher-order rewriting, in: Proceedings of the 11th RTA, Lecture Notes in Computer Science, Vol. 1833, Springer, Berlin, 2000.

[9] E. Bonelli, D. Kesner, A. Ríos, From higher-order to first-order rewriting, in: Proceedings of the 12th RTA, Lecture Notes in Computer Science, Vol. 2051, Springer, Berlin, 2001.

[10] G. Boudol, Computational semantics of trm rewrite systems, in: M. Nivat, J.C. Reynolds (Eds.), Algebraic Methods in Semantics, Cambridge University Press, Cambridge, 1985.

[11] P.-L. Curien, T. Hardin, A. Ríos, Strong normalisation of substitutions, in: Proceedings of Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Vol. 629, Springer, Berlin, 1992, pp. 209–217.

[12] R. David, B. Guillaume, A λ-calculus with explicit weakening and explicit substitutions, Math. Struct. Comput. Sci. 11 (1) (2001).

[13] N.G. de Bruijn, Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation with application to the church-rosser theorem, Indag. Math. 5 (35) (1972).
[14] N. Dershowitz, J.-P. Jouannaud, Rewrite systems, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Vol. B, North-Holland, Amsterdam, 1990, pp. 243–309.
[15] J. Glauert, R. Kennaway, Z. Khasidashvili, Stable results and relative normalisation, J. Logic Comput. 10 (3) (2000).
[16] Th. Hardin, L. Maranget, P. Pagano, Functional back-ends within the lambda-sigma calculus, in: Proceedings of the International Conference on Functional Programming, SIGPLAN Notices, Vol. (31)6, ACM Press, New York, 1996.
[17] G. Huet, J.-J. Lévy, Computations in orthogonal rewriting systems, in: J.L. Lassez, G.D. Plotkin (Eds.), Computational Logic; Essays in Honor of Alan Robinson, MIT Press, Cambridge, MA, 1991, pp. 394–443.
[18] F. Kamareddine, A. Ríos, A λ-calculus à la de Bruijn with explicit substitutions, in: Proceedings of the International Symposium on Programming Language Implementation and Logic Programming (PLILP), Lecture Notes in Computer Science, Vol. 982, Springer, Berlin, 1995.
[19] Z. Khasidashvili, Optimal normalisation in orthogonal term rewrite systems, in: Proceedings of the 5th International Conference on Rewriting Techniques and Applications, Lecture Notes in Computer Science, Vol. 690, Springer, Berlin, 1993, pp. 243–258.
[20] J.W. Klop, Combinatory reduction systems, Ph.D. Thesis, Mathematical Centre Tracts, Vol. 127, CWI, Amsterdam, 1980.
[21] J.W. Klop, Term rewriting systems, Handbook Logic Comput. Sci. 2 (1992) 1–116.
[22] J.W. Klop, V. van Oostrom, F. van Raamsdonk, Combinatory reduction systems: introduction and survey, Theoret. Comput. Sci. 121 (1–2) (1993) 279–308.
[23] J.-J. Lévy, Réductions correctes et optimales dans le lambda-calcul, Ph.D. Thesis, Université Paris VII, 1978.
[24] L. Maranget, La stratégie paresseuse, Ph.D. Thesis, Université Paris VII, 1992.
[25] P.-A. Melliès, Typed λ-calculi with explicit substitutions may not terminate, in: Proceedings of Typed Lambda Calculi and Applications, Lecture Notes in Computer Science, Vol. 902, Springer, Berlin, 1995.
[26] P.-A. Melliès, Description Abstraite des Systèmes de Réécriture, Ph.D. Thesis, Université Paris VII, 1996.
[27] P.-A. Melliès, Axiomatic rewriting theory II: the $\lambda\sigma$-calculus enjoys finite normalisation cones, J. Logic Comput. 10 (3) (2000) 461–487.
[28] P.-A. Melliès, Axiomatic Rewriting Theory I: An Axiomatic Standardisation Theorem, Prépublication de l'équipe PPS 13, Université Paris VII, December 2002.
[29] P.-A. Melliès, Residual theory revisited, in: Proceedings of RTA02, Lecture Notes in Computer Science, Vol. 2378, Springer, Berlin, 2002.
[30] D. Miller, A logic programming language with lambda-abstraction, function variables, and simple unification, in: P. Schroeder-Heister (Ed.), Proceedings of the International Workshop on Extensions of Logic Programming, FRG, 1989, Lecture Notes in Artificial Intelligence, Vol. 475, Springer, Berlin, December 1991.
[31] T. Nipkow, Higher-order critical pairs, in: Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press, Silver Spring, MD, July 1991.
[32] A. Ríos, Contribution à l'étude des λ-calculs avec substitutions explicites. Ph.D. Thesis, Université de Paris VII, 1993.
[33] V. van Oostrom, Confluence for abstract and higher-order rewriting, Ph.D. thesis, Vrije University, 1994.
[34] V. van Oostrom, FD à la Melliès, 1997, Obtainable by ftp at ftp://www.phil.uu.nl/õostrom/publication/rewriting.html.
[35] V. van Oostrom, Normalisation in weakly orthogonal rewriting, in: Proceedings of the 10th International Conference on Rewriting Techniques and Applications, Lecture Notes in Computer Science, Vol. 1631, Springer, Berlin, 1999.
[36] H. Zantema, Termination of term rewriting by semantic labelling, Fund. Inform. 24 (1995).