Relative Normalization in Orthogonal Expression Reduction Systems

John Glauert and Zurab Khasidashvili

School of Information Systems, UEA
Norwich NR4 7TJ England
jrwg@sys.uea.ac.uk, zurab@sys.uea.ac.uk *

Abstract. We study reductions in orthogonal (left-linear and non-ambiguous) Expression Reduction Systems, a formalism for Term Rewriting Systems with bound variables and substitutions. To generalise the normalization theory of Huet and Lévy, we introduce the notion of *neededness* with respect to a set of reductions Π or a set of terms $\mathcal{S}$ so that each existing notion of neededness can be given by specifying Π or $\mathcal{S}$. We imposed natural conditions on $\mathcal{S}$, called *stability*, that are sufficient and necessary for each term not in $\mathcal{S}$-normal form (i.e., not in $\mathcal{S}$) to have at least one $\mathcal{S}$-needed redex, and repeated contraction of $\mathcal{S}$-needed redexes in a term t to lead to an $\mathcal{S}$-normal form of t whenever there is one. Our relative neededness notion is based on tracing *(open) components*, which are occurrences of contexts not containing *any* bound variable, rather than tracing redexes or subterms.

1 Introduction

Since a normalizable term, in a rewriting system, may have an infinite reduction, it is important to have a *normalizing* strategy which enables one to construct reductions to normal form. It is well known that the leftmost-outermost strategy is normalizing in the λ-calculus. For Orthogonal Term Rewriting Systems (OTRSs), a general normalizing strategy, called the *needed* strategy, was found by Huet and Lévy in [HuLé91]. The needed strategy always contracts a *needed* redex – a redex whose residual is to be contracted in any reduction to normal form. Huet and Lévy showed that any term t not in normal form has a needed redex, and that repeated contraction of needed redexes in t leads to its normal form whenever there is one; we refer to it as the *Normalization Theorem.* They also defined the class of *strongly sequential* OTRSs where a needed redex can efficiently be found in any term.

Barendregt et al. [BKKS87] generalized the concept of neededness to the λ-calculus. They studied neededness not only w.r.t. normal forms, but also w.r.t. head-normal forms – a redex is *head-needed* if its residuals are contracted in each reduction to a head-normal form. The authors proved correctness of the two

* This work was supported by the Engineering and Physical Sciences Research Council of Great Britain under grant GR/H 41300

needed strategies for computing normal forms and head-normal forms, respectively. Khasidashvili defined a similar normalizing strategy, called the *essential* strategy, for the λ-calculus [Kha88] and OTRSs [Kha93]. The strategy contracts *essential* redexes – the redexes that have *descendants* under any reduction. The notion of descendant is a refinement of that of *residual* – the descendant of a contracted redex is its contractum, while it does not have residuals. This refined notion allows for much simpler proofs of correctness of the essential strategy in OTRSs and the λ-calculus, which generalize straightforwardly to all Orthogonal Expression Reduction Systems (OERSs). Kennaway and Sleep [KeSl89] used a generalization of Lévy's labelling for the λ-calculus [Lév78] to adapt the proof from [BKKS87] to the case of Klop's OCRSs [Klo80], which can also be applied to OERSs. Khasidashvili [Kha94] showed that in *Persistent* OERSs, where redex-creation is limited, one can find *all* needed redexes in any term. Gardner [Gar94] described a *complete* way of encoding neededness information using a type assignment system in the sense that using the principal type of a term one can find all the needed redexes in it (the principal type cannot be found efficiently, as one might expect). Antoy et al. [AEH94] designed a needed narrowing strategy.

In [Mar92], Maranget introduced a different notion of neededness, where a redex u is needed if it has a residual under any reduction that does not contract the residuals of u. This neededness notion makes sense also for terms that do not have a normal form, and coincides with the notion of essentiality [Kha93] (essentiality makes sense for all subterms, not only for redexes). In [Mar92], Maranget studied also a strategy that computes a (in fact, the 'minimal' in some sense) weak head-normal form of a term in an OTRS. Normalization w.r.t. another interesting set of 'normal forms', that of constructor head-normal forms in constructor OTRSs, is studied by Nöcker [Nök94].

A question arises naturally: what are the properties that a set of terms must possess in order for the neededness theory of Huet and Lévy still to make sense? The main contribution of this paper is to provide a solution to that question. We introduce the notion of *neededness* w.r.t. a set of reductions Π or a set of terms $\mathcal{S}$ so that each existing notion of neededness can be given by specifying Π or $\mathcal{S}$. For example, *Huet&Lévy-neededness* is neededness w.r.t. the set of normal forms, *Maranget-neededness* is neededness w.r.t. all fair reductions, *head-neededness* is neededness w.r.t. the set of head-normal forms, etc. We impose natural conditions on $\mathcal{S}$, called *stability*, that are sufficient and necessary for each term not in $\mathcal{S}$-normal form (i.e., not in $\mathcal{S}$) to have at least one $\mathcal{S}$-needed redex, and repeated contraction of $\mathcal{S}$-needed redexes in a term t to lead to an $\mathcal{S}$-normal form of t whenever there is one.

A set $\mathcal{S}$ of terms is stable if it is *closed under parallel moves*: for any $t \notin \mathcal{S}$, any $P : t \twoheadrightarrow o \in \mathcal{S}$, and any $Q : t \twoheadrightarrow e$, the final term of P/Q, the residual of P under Q, is in $\mathcal{S}$; and is *closed under unneeded expansion*: for any $e \overset{u}{\to} o$ such that $e \notin \mathcal{S}$ and $o \in \mathcal{S}$, a residual of u is contracted in any reduction from e to a term in $\mathcal{S}$. We present a counterexample to show that the $\mathcal{S}$-needed strategy is not *hypernormalizing* for every stable $\mathcal{S}$, i.e., an $\mathcal{S}$-normalizable term may possess a reduction which never reaches a term in $\mathcal{S}$ even though $\mathcal{S}$-needed redexes are contracted infinitely many times, but $\mathcal{S}$-unneeded redexes are contracted as well. Therefore, *multistep* $\mathcal{S}$-needed reductions need not be $\mathcal{S}$-normalizing. This is because a 'non-standard' situation – when a component under an $\mathcal{S}$-unneeded component is $\mathcal{S}$-needed – may occur for some, call it *irregular* stable sets $\mathcal{S}$. However, if $\mathcal{S}$ is a regular stable set, then the $\mathcal{S}$-needed strategy is again hypernormalizing, and the multistep $\mathcal{S}$-needed strategy is normalizing.

Our relative neededness notion is based on tracing *(open) components*, which are occurrences of contexts not containing *any* bound variable, rather than tracing redexes or subterms. We therefore introduce notions of *descendant* and *residual* for components that are invariant under Lévy-equivalence. A component of a term $t \notin \mathcal{S}$ is called $\mathcal{S}$*-needed* if at least one descendant of it is 'involved' in any $\mathcal{S}$-normalizing reduction; a redex is $\mathcal{S}$*-needed* if so is its pattern. Besides generality, this approach to defining the neededness notion via components is crucial from a technical point of view, because components of a term in an OERS enjoy the same 'disjointness' property that subterms of a term in an OTRS possesses: residuals of disjoint components of a term in an OERS remain disjoint, and this allows for simpler proofs.

The rest of the paper is organized as follows. In the next section, we review Expression Reduction Systems (ERS), a formalism for higher order rewriting that we use here [Kha90, Kha92]; define the descendant relation for components, and show that it is invariant under Lévy-equivalence. Section 3 establishes equivalence of Maranget's neededness and our essentiality for OERSs. In section 4, we introduce the relative notion of neededness. In section 5, we sketch some properties of the labelling system of Kennaway&Sleep [KeSl89] for OERSs needed to define a *family-relation* among redexes. We prove correctness of the $\mathcal{S}$-needed strategy for finding terms of $\mathcal{S}$, for all stable $\mathcal{S}$, in section 6. The conclusions appear in section 7.

2 Orthogonal Expression Reduction Systems

Klop introduced *Combinatory Reduction Systems* (CRSs) in [Klo80] to provide a uniform framework for reductions with substitutions (also referred to as higher-order rewriting) as in the λ-calculus [Bar84]. Restricted rewriting systems with substitutions were first studied in Pkhakadze [Pkh77] and Aczel [Acz78]. Several interesting formalisms have been introduced later [Nip93, Wol93, OR94]. We refer to Klop et al. [KOR93] and van Oostrom [Oos94] for a survey. Here we use a system of higher order rewriting, *Expression Reduction Systems* (ERSs), defined in Khasidashvili [Kha90, Kha92] (ERSs are called CRSs in [Kha92]); the present formulation is simpler.

Definition 2.1 Let Σ be an *alphabet*, comprising *variables*, denoted by x, y, z; *function symbols*, also called *simple operators*; and *operator signs* or *quantifier signs*. Each function symbol has an *arity* $k \in N$, and each operator sign σ has an *arity* (m, n) with $m, n \neq 0$ such that, for any sequence $x_1, \ldots, x_m$ of pairwise distinct variables, $\sigma x_1 \dots x_m$ is a *compound operator* or a *quantifier* with *arity* n. Occurrences of $x_1, \dots, x_m$ in $\sigma x_1 \dots x_m$ are called *binding variables*. Each quantifier $\sigma x_1 \dots x_m$, as well as the corresponding quantifier sign σ and binding variables $x_1 \dots x_m$, has a *scope indicator* $(k_1, \dots, k_l)$ to specify the arguments in which $\sigma x_1 \dots x_m$ binds all free occurrences of $x_1, \dots, x_m$. *Terms* are constructed from variables using functions and quantifiers in the usual way.

Metaterms are constructed similarly from *terms* and *metavariables* $A, B, \dots$, which range over terms. In addition, *metasubstitutions*, expressions of the form $(t_1/x_1, \dots, t_n/x_n)t_0$, with t_j arbitrary metaterms, are allowed, where the *scope* of each x_i is t_0. Metaterms without metasubstitutions are *simple metaterms*. An *assignment* maps each metavariable to a term over Σ. If t is a metaterm and θ is an assignment, then the *θ-instance $t\theta$* of t is the term obtained from t by replacing metavariables with their values under θ, and by replacing metasubstitutions $(t_1/x_1, \dots, t_n/x_n)t_0$, in right to left order, with the result of substitution of terms $t_1, \dots, t_n$ for free occurrences of $x_1, \dots, x_n$ in t_0.

For example, a β-redex in the λ-calculus appears as $Ap(\lambda x\ t, s)$, where Ap is a function symbol of arity 2, and λ is an operator sign of arity (1,1) and scope indicator (1). Integrals such as $\int_s^t f(x)\,dx$ can be represented as $\int x\, s\, t\, f(x)$ using an operator sign $\int$ of arity (1,3) and scope indicator (3).

Definition 2.2 An *Expression Reduction System* (ERS) is a pair (Σ, R), where Σ is an *alphabet*, described in Definition 2.1, and R is a set of *rewrite rules* $r : t \to s$, where t and s are closed metaterms (i.e., no free variables) such that t is a simple metaterm and is not a metavariable, and each metavariable that occurs in s occurs also in t.

Further, each rule r has a set of *admissible assignments* $AA(r)$ which, in order to prevent undesirable confusion of variable bindings, must satisfy the condition that:

(a) for any assignment $\theta \in AA(r)$, any metavariable A occurring in t or s, and any variable $x \in FV(A\theta)$, either every occurrence of A in r is in the scope of some binding occurrence of x in r, or no occurrence is.

For any $\theta \in AA(r)$, $t\theta$ is an *r-redex* or an *R-redex*, and $s\theta$ is the *contractum* of $t\theta$. We call R *simple* if right-hand sides of R-rules are simple metaterms.

Our syntax is similar to that of Klop's CRSs [Klo80], but is closer to the syntax of the λ-calculus and of First Order Logic. For example, the β-rule is written as $Ap(\lambda x A, B) \to (B/x)A$, where A and B can be instantiated by any terms; the η-rule is written as $\lambda x(Ax) \to A$ which requires that an assignment θ is admissible iff $x \notin (A\theta)$, otherwise an x occurring in $A\theta$ and therefore bound in $\lambda x(A\theta x)$ would become free. A rule like $f(A) \to \exists x(A)$ is also allowed, but an assignment θ with $x \in A\theta$ is not. The recursor rule from [AsLa93] is written as $\mu(\lambda x A) \to (\mu(\lambda x A)/x)A$. $\exists x A \to (\tau x(A)/x)A$ and $\exists! x A \to \exists x A \wedge \forall x \forall y(A \wedge (y/x)A \Rightarrow x = y)$ are rules corresponding to familiar definitions.

Below we restrict ourselves to the case of non-conditional ERSs, i.e., ERSs where an assignment is admissible iff the condition (a) of Definition 2.2 is satisfied. We ignore questions relating to renaming of bound variables. As usual, a rewrite step consists of replacement of a redex by its contractum. Note that the use of metavariables in rewrite rules is not really necessary – free variables can be used instead, as in TRSs. We will indeed do so at least when giving TRS examples.

To express substitution, we use the S-reduction rules

$$S^{n+1}x_1 \ldots x_n A_1 \ldots A_n A_0 \to (A_1/x_1, \ldots, A_n/x_n)A_0, \quad n = 1, 2, \ldots,$$

where S^{n+1} is the *operator sign of substitution* with arity $(n, n+1)$ and scope indicator $(n+1)$, and $x_1, \ldots, x_n$ and $A_1, \ldots, A_n, A_0$ are pairwise distinct variables and metavariables. Thus S^{n+1} binds free variables only in the last argument. The difference with β-rules is that S-reductions can only perform β-developments of λ-terms [Kha92].

Notation 2.1 We use a, b, c, d for constants, t, s, e, o for terms and metaterms, u, v, w for redexes, and N, P, Q for reductions. We write $s \subseteq t$ if s is a subterm of t. A one-step reduction in which a redex $u \subseteq t$ is contracted is written as $t \xrightarrow{u} s$ or $t \to s$ or just u. We write $P : t \twoheadrightarrow s$ or $t \overset{P}{\twoheadrightarrow} s$ if P denotes a reduction of t to s. $|P|$ denotes the length of P. $P + Q$ denotes the concatenation of P and Q.

Let $r : t \to s$ be a rule in an ERS R and let $\theta \in AA(r)$. Subterms of a redex $v = t\theta$ that correspond to metavariables of t are the *arguments* of v, and the rest is the *pattern* of v. Subterms of v rooted in the pattern are called the *pattern-subterms* of v. If R is a simple ERS, then arguments, pattern, and pattern-subterms are defined analogously in the contractum $s\theta$ of v.

We now recall briefly the definition of *descendant* of subterms as introduced in [Kha88, Kha93, Kha92] for the λ-calculus, TRSs, and ERSs, respectively. First, we need to split an ERS R into a *TRS-part* R_f and the *substitution-part* S. For any ERS R, which we assume does not contain symbols S^{n+1}, R_f is the ERS obtained from R by adding symbols S^{n+1} in the alphabet and by replacing in right-hand sides of the rules all metasubstitutions of the form $(t_1/x_1, \ldots, t_n/x_n)t_0$ by $S^{n+1}x_1 \ldots x_n t_1 \ldots t_n t_0$, respectively. For example, the β_f rule would be $Ap(\lambda x A, B) \to S^2 x B A$. If R is simple, then $R_{fS} =_{def} R_f =_{def} R$. Otherwise $R_{fS} =_{def} R_f \cup S$. For each step $C[t\theta] \xrightarrow{u} C[s\theta]$ in R there is a reduction $P : C[t\theta] \to_{R_f} C[s'\theta] \twoheadrightarrow_S C[s\theta]$ in R_{fS}, where $C[s'\theta] \twoheadrightarrow_S C[s\theta]$ is the rightmost innermost normalizing S-reduction. We call P the *refinement* of u. The notion of *refinement* generalizes to R-reductions with 0 or more steps.

Let $t \xrightarrow{u} s$ be an R_f-reduction step and let e be the contractum of u in s. For each argument o of u there are 0 or more arguments of e. We will call them *u-descendants* of o. We refer to the i-th (from the left, $i > 0$) descendant of o also as the (u, i)-*descendant* of o. Correspondingly, subterms of o have 0 or more *descendants*. By definition, the *descendant*, referred to also as the $(u, *)$-*descendant*, of each pattern-subterm of u is e. It is clear what is to be meant by the *descendant* of a subterm $s' \subseteq t$ that is not in u. We call it also the $(u, *)$-*descendant* of s'. In an S-reduction step $C[S^{n+1}x_1 \ldots x_n\, t_1 \ldots t_n t_0] \xrightarrow{u} C[(t_1/x_1, \ldots, t_n/x_n)t_0]$, the argument t_i and its subterms have the same number of descendants as the number of free occurrences of x_i in t_0; the i-th descendant is referred to as the

(u,i)-*descendant*. Every subterm of t_0 has exactly one descendant, the $(u,0)$-*descendent* (in particular, the $(u,0)$-descendants of free occurrences of $x_1,\ldots,x_n$ in t_0 are the substituted subterms). The descendant or $(u,*)$-*descendent* of the contracted redex u itself is its contractum. The pairs (u,i) and $(u,*)$ are called the *indexes* of corresponding descendants. The descendants of all redexes except the contracted one are called *residuals*.

The notions of *descendant* and *residual* extend by transitivity to arbitrary R_{fS}-reductions; *indexes* of descendants and residuals are sequences of indexes of immediate (i.e., under one step) descendants and residuals in the chain leading from the initial to the final subterm. If P is an R-reduction, then P descendants are defined to be the descendants under the refinement of P. The *ancestor* relation is the converse of the descendant relation.

We call a *component* an occurrence of a context that does not contain *any* bound variables; that is, neither variables bound from above in the term, nor variables for which the binder is in the component, belong to the component. Since we also consider occurrences of the empty context $[\,]$, which has an arity 1, we will think of a component as a pair $(context, path)$, where the path characterizes the *position* of the component in the term (usually, a position is a chain of natural numbers). Thus an *empty component* or *empty occurrence* is a pair $([\,], path)$. If terms are represented by trees, then the empty occurrence $([\,], path)$ can be seen as the *connection* at the *top* of the symbol at the position *path*.

Obviously, a component C can be considered as its *corresponding subterm* (the subterm rooted at the position of C), with some subterms (the *arguments* of C), removed. In particular, a subterm s with itself removed becomes the empty occurrence at the position of s. We say that an empty occurrence $([\,], path)$ in a term t is *in* a subterm or component e in t if *path* is a *non-top* position of e. We use C and the letters s,t,e,o used for terms to denote components as well. We write $s \sqsubseteq t$ if s is a component of t.

The concept of *descendent* can be extended to components in the following way:

Definition 2.3 Let R be an ERS, C be a component of a term t in R_{fS}, let $s = C[s_1,\ldots,s_n]$ be the corresponding subterm of C in t, let $t\overset{u}{\to}t'$ in R_{fS}, and let $o' \subseteq t'$ be the contractum of u. The pattern of u will be denoted by $pat(u)$ and the pattern of the contractum by $cpat(u)$. We define u-*descendants* of C by considering all relative positions of u and C.

(1) $C \cap u = \emptyset$ (so if C is an empty component, C is not in u). Then the $(u,*)$-descendant of C is the $(u,*)$-descendant of s with the $(u,*)$-descendants of s_j removed. (If C is an empty occurrence $([\,], path)$, then its $(u,*)$-descendant is the same pair.)

(2) C is the empty occurrence at the top of an argument o of u. Then the (u,i)-descendant of C is the empty occurrence at the top-position of the (u,i)-descendant of o, if the latter exists; if o doesn't have u-descendants, then C doesn't have u-descendants either. (Thus, the (u,i)-descendant of C is the (u,i)-descendant of its corresponding subterm $s=o$ with itself removed.)

(3) u is an R_f-redex and C is in an argument of u. Then the (u,i)-descendant of C is the (u,i)-descendant of s with the (u,i)-descendants of s_j removed $(i \geq 1)$.

(4) u is an S-redex and C is in an argument of u. Then the (u,i)-descendant of C is the (u,i)-descendant of s with the (u,i)-descendants of s_j removed $(i \geq 0)$.

(5) u is an R_f-redex and $pat(u)$ is in C. Then the $(u,*)$-*descendant* of C is the $(u,*)$-descendant of s with the descendants of s_j removed (see Figure 1). In particular, if C is *collapsed*, i.e. $pat(u) = C$ and $cpat(u) = \emptyset$, then the $(u,*)$-descendant of C is the empty occurrence at the position of o'.

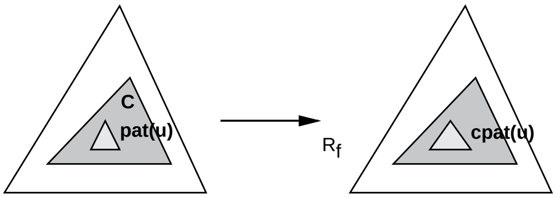


Fig. 1.

(6) u is an S-redex, say with two arguments (for simplicity) $u = S^2x\,e\,o$, with the top in C, and let $s = C'[s_1,\ldots,S^2x\,C_e[s_k,\ldots,s_l]\,C_o[s_{l+1},\ldots,s_m],\ldots,s_n]$, with $C = C'[\ldots,S^2x\,C_e[\;]\,C_o[\;],\ldots]$, $e = C_e[s_k,\ldots,s_l]$, and $o = C_o[s_{l+1},\ldots,s_m]$. Then, the $(u,*)$-*descendants* of C are the $(u,*)$-descendant of s with the descendants of the subterms s_j removed, and the descendants of C_e, as defined in (4) (see Figure 2, where $k = l = 1$ and $l+1 = m = 2$). Note that if $C = Pat(u) = S$, then C_e and C_o are empty components, C is *collapsed*, and its descendants are the empty occurrences at the top positions of the descendants of u and its arguments.

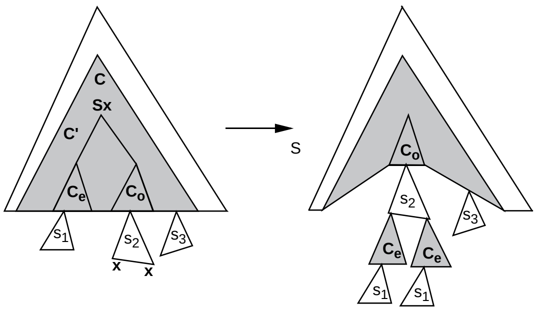


Fig. 2.

(7) u is an R_f-redex, $pat(u)$ and C partially overlap (i.e., neither contains another), and the top of u is (not necessarily strictly) below the top of C. Then the $(u,*)$-descendant of C is the $(u,*)$-descendant of s with the descendants of the arguments of u that do not overlap with C and the descendants of s_i that do not overlap with $pat(u)$ removed (see Figure 3). In addition, if the top symbol of o' doesn't belong to the (above) $(u,*)$-descendant of C, then the empty component at the top of o' is also a $(u,*)$-descendant of C.

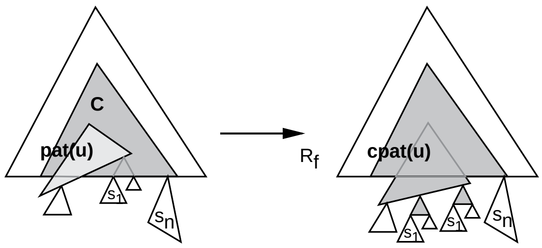


Fig. 3.

(8) u is an R_f-redex, $pat(u)$ and C partially overlap, and the top of C is below the top of u. Then the $(u,*)$-descendant of C is the $(u,*)$-descendant of s with the descendants of s_i that do not overlap with $pat(u)$ and the descendants of the arguments of u that do not overlap with C removed (see Figure 4). In addition, if the top symbol of o' doesn't belong to the (above) $(u,*)$-descendant of C, then the empty component at the top of o' is also a $(u,*)$-descendant of C.

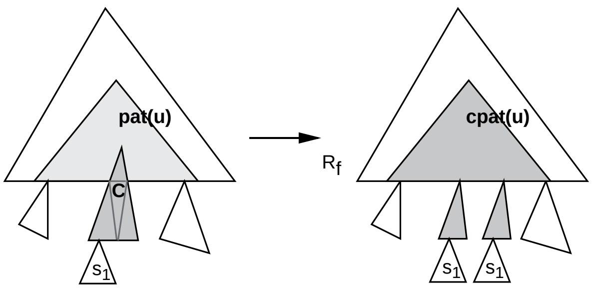


Fig. 4.

(9) u is an R_f-redex and $pat(u)$ contains C (C may be an empty component). Then the $(u,*)$-descendant of C is the contractum-pattern of u (the latter may also be empty) (see Figure 5).

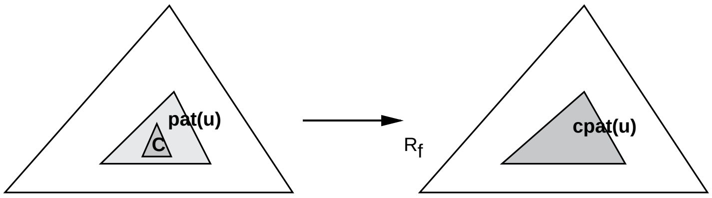


Fig. 5.

The notion of descendant for components generalises by transitivity to all R_{fS} reductions and via refinements to all R-reductions.

In [Klo92], Klop also introduced a notion of *descendant of (an occurrence of) a symbol* for the case of TRSs. According to Klop's definition, descendants of every symbol in the pattern of the contracted redex are *all* symbols in the contractum pattern. We extend this also to empty occurrences in the pattern of the contracted redex. If the rule is collapsing, i.e., the right-hand side is a variable, then Klop defines the descendant of pattern-symbols of the contracted redex to be the topmost symbol of the contractum. We define the descendant of the pattern-symbols in that case to be the empty occurrence at the position of the contractum. We can define the notion of descendant of a symbol in the same spirit for S-reduction steps by declaring that the descendants of the top S-symbol of a contracted S-redex u are the empty occurrences at the positions of the descendant of u and the descendants of its arguments. We define the descendants of other symbols, in particular of bound variables, to be the top-symbols of the descendants of the corresponding subterms. This gives us a definition of descendant for symbols for all ERSs. Now, it is not difficult to check that descendants of a component are composed of the 'corresponding' descendants of its symbols and occurrences, and similarly for the descendants of subterms. Note that, according to Klop's definition, the descendants of a pattern-subterm o are all subterms rooted in the contractum-pattern, not just the contractum of the redex, as in our definition; so descendants of o need not be composed of descendants of the symbols in o, which is less natural.

Definition 2.4 Co-initial reductions $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$ are called *Hindley-equivalent*, written $P \approx_H Q$, if $s = e$ and the residuals of a redex of t under P and Q are the same occurrences. We call P and Q respectively *Klop-equivalent, strictly equivalent* [Kha92], or *strictly* equivalent*, written $P \approx_K Q$, $P \approx_{st} Q$, or $P \approx^*_{st} Q$, if $s = e$ and P-descendants and Q-descendants of respectively any symbol, subterm, or component of t are the same occurrences in s and e.

Definition 2.5 A rewrite rule $t \to s$ in an ERS R is *left-linear* if t is linear, i.e., no metavariable occurs more than once in t. R is *left-linear* if each rule in R is so. $R = \{r_i \,|\, i \in I\}$ is *non-ambiguous* or *non-overlapping* if in no term redex-patterns can overlap, i.e., if r_i-redex u contains an r_j-redex u' and $i \neq j$, then u' is in an argument of u, and the same holds if $i = j$ and u' is a proper subterm of u. R is *orthogonal* (OERS) if it is left-linear and non-overlapping.

As in the the λ-calculus [Bar84], for any co-initial reductions P and Q, one can define in OERSs the notion of *residual of P under Q*, written P/Q, due to Lévy [Lév80], via the notion of *development* of a set of redexes in a term. We write $P \trianglelefteq Q$ if $P/Q = \emptyset$ ($\trianglelefteq$ is the *Lévy-embedding* relation); P and Q are called *Lévy-equivalent* or *permutation-equivalent* (written $P \approx_L Q$) if $P \trianglelefteq Q$ and $Q \trianglelefteq P$. It follows immediately from the definition of / that if P and Q are co-initial reductions in an OERS, then $(P + P')/Q \approx_L P/Q + P'/(Q/P)$ and $P/(Q + Q') \approx_L (P/Q)/Q'$.

Theorem 2.1 Let P and Q be co-initial reductions in an OERS R. Then:
(1) ($CR(res)$**: Church-Rosser for residuals**) $P + Q/P \approx_H Q + P/Q$.
(2) ($CR(sub)$**: Church-Rosser for subterms**) $P + Q/P \approx_{st} Q + P/Q$.
(3) ($CR(sym)$**: Church-Rosser for symbols**) $P + Q/P \approx_K Q + P/Q$.
(4) ($CR(com)$**: Church-Rosser for components**) $P + Q/P \approx^*_{st} Q + P/Q$.

Proof. (1) is proved in [Klo80]. (2) is obtained in [Kha92]. The proof of (3) is routine (it is enough to consider the case when $|P| = |Q| = 1$). (4) follows both from (3) and (2), since descendants of a component can be defined both via descendants of symbols and via descendants of subterms. (Note that (2) can also be derived from (3) for the same reason.)

3 Neededness, Essentiality and Unabsorbedness

In this section, we recall Huet&Lévy and Maranget's notion of neededness, and relate them to the notion of essentiality, in OERSs. We also prove existence of an essential redex in any term, in an OERS, not in normal form.

Definition 3.1 A redex u in t is *Huet&Lévy-needed* [HuLé91, Lév80] if in each reduction of t to normal form (if any) at least one residual of u is contracted; u is *Maranget-needed* [Mar92] if u has at least one residual under any reduction starting from t that does not contract residuals of u.

Definition 3.2 A subterm s in t is *essential* (written $ES(s,t)$) if s has at least one descendant under any reduction starting from t and is *inessential* (written $IE(s,t)$) otherwise [Kha93, Kha88].

Definition 3.3 A subterm s of a term t is *unabsorbed in a reduction $P : t \twoheadrightarrow e$* if none of the descendants of s appear in redex-arguments of terms in P, and is *absorbed in P* otherwise; s is *unabsorbed in t* if it is unabsorbed in any reduction starting from t and *absorbed in t* otherwise [Kha93].

Remark 3.1 It is easy to see that a redex $u \subseteq t$ is unabsorbed iff u is *external* [HuLé91] in t. Clearly, unabsorbedness implies essentiality, and Huet&Levy- and Maranget-neededness coincide for normalizable terms. □

Definition 3.4 Let $P : t \twoheadrightarrow s$ and o be a subterm or a component in t. Then we say that P *deletes* o if o doesn't have P-descendants.

Definition 3.5 (1) Let $P : t \to t_1 \to \ldots$ and $u \subseteq t_i$. Then u is called *erased* in P if there is $j > i$ such that u does not have residuals in t_j. P is *fair* if each redex in any t_i is erased in P [Klo92].

(2) We call a reduction P starting from t *strictly cofinal* if for any $Q : t \twoheadrightarrow e$ there is an initial part $P' : t \twoheadrightarrow s$ of P, and a reduction $Q' : e \twoheadrightarrow s$ such that $P' \approx_{st} Q + Q'$.

Lemma 3.1 Any strictly cofinal reduction P starting from t deletes all inessential subterms of t.

Proof. Immediate from Definitions 3.2 and 3.5.

For example, fair reductions are strictly cofinal: Klop's proof of cofinality of fair reductions (Theorem 12.3 in [Klo80]) can be modified to a proof of strict cofinality of fair reductions by using $CR(sub)$ (Theorem 2.1.(2)) instead of the CR theorem.

The following lemma from [Kha94] follows from $CR(sub)$ (Theorem 2.1.(2)); the proofs are same as for OTRS [Kha93].

Lemma 3.2 (1) Let $P : t \twoheadrightarrow t'$ and $s \subseteq t$. Then $IE(s, t)$ iff all P-descendants of s are inessential in t'.

(2) Let $t \overset{u}{\to} t'$ and $e \subseteq s \subseteq t$. Then any u-descendant of e is contained in some u-descendant of s.

(3) Let $e \subseteq s \subseteq t$ and $ES(e, t)$. Then $ES(s, t)$.

(4) Let s be a pattern-subterm of a redex $u \subseteq t$. Then $ES(u, t)$ iff $ES(s, t)$.

Notation We write $t = (t_1//s_1, \ldots, t_n//s_n)s$ if $s_1, \ldots, s_n$ are disjoint subterms in s and t is obtained from s by replacing them with $t_1, \ldots, t_n$, respectively.

Definition 3.6 Let $t = (t_1//s_1, \ldots, t_n//s_n)s$ in an OERS R and let $P : s = e_0 \overset{v_0}{\to} e_1 \overset{v_1}{\to} \ldots$ be an R_{fS}-reduction. We define the reduction $P\|(t) : t = o_0 \overset{u_0}{\to} o_1 \overset{u_1}{\to} \ldots$ as follows.

(1) If v_0 is an R_f-redex and its pattern does not overlap with $s_1, \ldots, s_n$, then u_0 is the corresponding subterm of v_0 in $t = o_0$.

(2) If v_0 is an R_f-redex that is not inside the subterms $s_1, \ldots, s_n$ and its pattern does overlap with some of $s_1, \ldots, s_n$, then $u_0 = \emptyset$.

(3) If v_0 is an S-redex that is outside the replaced subterms, then u_0 is the corresponding S-redex in t_0.

(4) If v_0 is in some of subterms $s_1, \ldots, s_n$, then $u_0 = \emptyset$.

In the first case, o_1 is obtained from e_1 by replacing the descendants of $s_1, \ldots, s_n$ with the corresponding descendants of $t_1, \ldots, t_n$, respectively; in the second case, o_1 is obtained from e_1 by replacing the descendant of v_0 by the descendant of its corresponding subterm in o_0, and by replacing the descendants of the subterms s_i that do not overlap with v_0 by the corresponding descendants of t_i; in the third case, o_1 is obtained from e_1 by replacing outermost descendants of $s_1, \ldots, s_n$ with the corresponding descendants of $t_1, \ldots, t_n$; in the fourth case, o_1 is obtained from e_1 by replacing the descendants of $s_1, \ldots, s_n$ with the corresponding descendants of $t_1, \ldots, t_n$, respectively. Thus, in o_1 we can choose the redex u_1 analogously, and so on. (Note that $P\|(t)$ depends not only on P and t, but also on the choice of $s_1, \ldots, s_k$, but the notation does not give rise to ambiguity.)

Lemma 3.3 Let $s_1, \ldots, s_n$ be inessential in s, in an OERS R, and let $t = (t_1//s_1, \ldots, t_n//s_n)s$. Then $t_1, \ldots, t_n$ are inessential in t.

Proof. We show by induction on $|Q|$ that if an R_{fS}-reduction $Q : s \twoheadrightarrow o$ deletes $s_1, \ldots, s_n$, then $Q\|(t)$ deletes $t_1, \ldots, t_n$; such a Q exists by Lemma 3.1. Let $Q = v + Q'$, $s \xrightarrow{v} e$, and $s'_1, \ldots, s'_m$ be all the v-descendants of $s_1, \ldots, s_n$. By Lemma 3.2.(1), s'_i are inessential in e. By Definition 3.6, if $v\|(t) : t \to o$, then o is obtained from e by replacing some inessential subterms that contain $s'_1, \ldots, s'_m$, and all the descendants of $t_1, \ldots, t_n$ also are in the replaced subterms of o. By the induction assumption, the replaced subterms in o are inessential. Hence, by Lemma 3.2.(3), all the descendants of $t_1, \ldots, t_n$ in o are inessential and, by Lemma 3.2.(1), $t_1, \ldots, t_n$ are inessential in t.

It follows immediately from Definition 3.6 and the proof of Lemma 3.3 that replacement of inessential subterms in a term does not effect its normal form.

Corollary 3.1 Any term not in normal form, in an OERS, contains an essential redex.

Proof. If all redexes in t were inessential, their replacement by fresh variables would yield a term in normal form containing inessential subterms, a contradiction.

Existence of an unabsorbed redex in any term not in normal form can be proved exactly as in OTRSs [Kha93] (the proof does not use the proof of Corollary 3.1).

Proposition 3.1 Let t be a term in an OERS R and let u be a redex in t. Then u is essential in t iff it is Maranget-needed in t.

Proof. ($\Leftarrow$) Let $IE(u, t)$. Further, let $FV(u) = \{x_1, \ldots, x_n\}$, let f be a fresh n-ary function symbol that does not occur in the left-hand sides of rewrite rules (we can safely add such a symbol to the alphabet, if necessary), and let $s = (f(x_1, \ldots, x_n)//u)t$. By Lemma 3.3, $IE(f(x_1, \ldots, x_n), s)$, i.e., there is some reduction P starting from s such that $f(x_1, \ldots, x_n)$ does not have P-descendants. Let $r : f(x_1, \ldots, x_n) \to u$ be a rule and $Q : s \to t$ be the r-reduction step. Obviously, $R \cup \{r\}$ is orthogonal. Hence $P + Q/P \approx_{st} Q + P/Q$ and therefore u does not have P/Q-descendants. But P/Q does not contract the residuals of u. Thus u is not Maranget-needed. ($\Rightarrow$) From Definitions 3.2 and 3.1.

4 Relative Notions of Neededness

In this section, we introduce notions of neededness relative to a set of reductions Π and to a set of terms $\mathcal{S}$. We show how all existing notions of neededness can be obtained by specifying Π or $\mathcal{S}$; $\mathcal{S}$-neededness is also a special case of Π-neededness. We introduce *stability* of a set of terms, in an OERS, and show that if $\mathcal{S}$ is not stable, contraction of $\mathcal{S}$-needed redexes in a term t need not terminate at a term in $\mathcal{S}$ even if t can be reduced to a term in $\mathcal{S}$. It is the aim of the last section to show that if $\mathcal{S}$ is stable, then a $\mathcal{S}$-needed strategy is $\mathcal{S}$-normalizing.

Definition 4.1 (1) We call a reduction P starting from a term t in an OERS R *external* to a component $e \sqsubseteq t$ if there is no redex executed in P whose pattern overlaps with a descendant of e (e can be empty). We call P *external* to a redex $u \subseteq t$ if P is external to $pat(u)$, i.e., if P doesn't contract the residuals of u.

(2) Let Π be a set of reductions. We call $e \sqsubseteq t$ *Π-needed* if there is no $P \in \Pi$ starting from t that is external to e, and call it *Π-unneeded* otherwise.

(3) We call $e \sqsubseteq t$ *P-(un)needed* if it is $\langle P \rangle_L$-(un)needed, where $\langle P \rangle_L$ is the set of all reductions Lévy-equivalent to P.

(4) Let $\mathcal{S}$ be a set of terms in R, and let $\Pi_{\mathcal{S}}$ be the set of *$\mathcal{S}$-normalizing* reductions, i.e., reductions that end at a term in $\mathcal{S}$. We call $e \sqsubseteq t$ *$\mathcal{S}$-(un)needed* if it is *$\Pi_{\mathcal{S}}$-(un)needed*.

(5) If $o \subseteq t$, then we call o ($P, \Pi, \mathcal{S}$-)*unneeded* (*needed*) if so is $Int(o)$, the component obtained from the subterm o by removing *all* bound variables. However, if not otherwise stated, we say that a redex $u \subseteq t$ (which is a subterm) is ($P, \Pi, \mathcal{S}$-)*unneeded* (*needed*) if so is its pattern.

(6) We say that $P : t \twoheadrightarrow o$ *$\mathcal{S}$-suppresses* $e \sqsubseteq t$ if P is $\mathcal{S}$-normalizing and is external to e. We say that P *$\mathcal{S}$-suppresses* $u \subseteq t$ if it $\mathcal{S}$-suppresses $pat(u)$, i.e., is $\mathcal{S}$-normalizing and is external to u. (Obviously, a redex or component is $\mathcal{S}$-unneeded iff it is $\mathcal{S}$-suppressed by some $\mathcal{S}$-normalizing reduction.)

We write $NE_{(P,\Pi,M)}(e, t)$ ($UN(e, t)$) if e is (P, Π, M)-needed (unneeded).

If $o \sqsubseteq t$ and $P : t \twoheadrightarrow s$ is external to o, then we call the descendants of o also the *residuals* of o. A component that does overlap with the pattern of a contracted redex does not have residuals. Note that if $u \subseteq t$, then $P : t \twoheadrightarrow s$ is external to $pat(u)$ iff P does not contract the residuals of u, because orthogonality of the system implies that if the pattern $pat(v)$ of a redex v contracted in P does overlap with a residual of $pat(u)$, then $pat(u) = pat(v)$. Hence u is $\mathcal{S}$-needed iff at least one residual of it is contracted in each reduction from t to a term in $\mathcal{S}$ (the intended notion of neededness). Obviously, any redex in a term that is not $\mathcal{S}$-normalizable is $\mathcal{S}$-needed; we call such redexes *trivially $\mathcal{S}$-needed*.

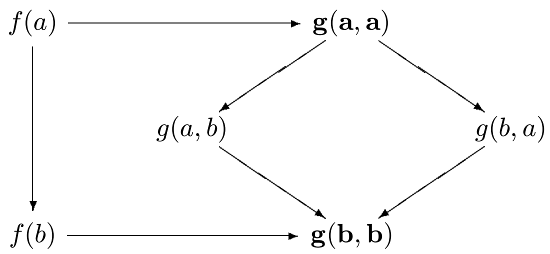
The following definition introduces the property of sets of terms for which it is possible to generalise the Normalization Theorem:

Definition 4.2 We call a set $\mathcal{S}$ of terms *stable* if:

(a) $\mathcal{S}$ is *closed under parallel moves*: for any $t \notin \mathcal{S}$, any $P : t \twoheadrightarrow o \in \mathcal{S}$, and any $Q : t \twoheadrightarrow e$, the final term of P/Q is in $\mathcal{S}$; and

(b) $\mathcal{S}$ is *closed under unneeded expansion*: for any $e \stackrel{u}{\to} o$ such that $e \notin \mathcal{S}$ and $o \in \mathcal{S}$, u is $\mathcal{S}$-needed.

Remark 4.1 Of course, a set closed under reduction is closed under parallel moves as well. But a set closed under parallel moves, even if closed under unneeded expansion, need not be closed under reduction. Indeed, consider $R = \{f(x) \to g(x, x),\ a \to b\}$, and take $\mathcal{S} = \{g(a, a), g(b, b)\}$. The only one-step $\mathcal{S}$-normalizing reductions are $g(a, b) \to g(b, b)$, $g(b, a) \to g(b, b)$, $f(a) \to g(a, a)$, and $f(b) \to g(b, b)$. Therefore, one can check that $\mathcal{S}$ is closed under unneeded expansion. Also, $\mathcal{S}$ is closed under parallel moves, since the right-bottom term $g(b, b)$ in the diagram below, which is the only non-trivial diagram to be checked, is in $\mathcal{S}$. However, $\mathcal{S}$ is not closed under reduction, since, e.g., $g(a, a) \to g(b, a)$, $g(a, a) \in \mathcal{S}$, but $g(b, a) \notin \mathcal{S}$. Note that the second occurrence of a in $g(a, a)$ is $\mathcal{S}$-unneeded, but its residual in $g(b, a)$ is $\mathcal{S}$-needed.

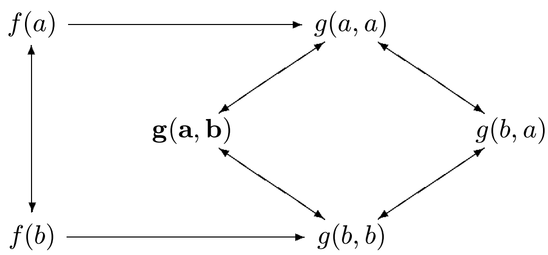


□

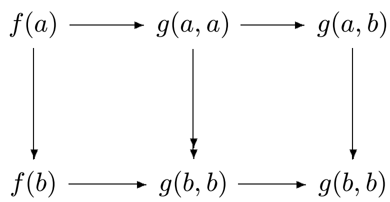
The most appealing examples of stable sets, for an OERS, are the set of normal forms [HuLé91], the set of head-normal forms [BKKS87], the set of weak-head-normal forms (a partial result is in [Mar92]), and the set of constructor-head-normal forms for constructor TRSs [Nök94]. The sets of terms having (resp. not having) (head-, constructor-head-) normal forms are stable as well. The graph G_s of a term s (which consists of terms to which s is reducible) is closed under reduction, but need not be closed under unneeded expansion. For example, the graph $G_{I(x)} = \{I(x),\ x\}$ of $I(x)$ is closed under reduction but is not closed under unneeded expansion: $I(I(x))$ can be reduced to $I(x)$ by reducing either I-redex (according to the rule $I(x) \to x$). Hence *none* of the redexes in $I(I(x))$ are $\mathcal{S}$-needed. Thus the closure of $\mathcal{S}$ under unneeded expansion is a necessary condition for the normalization theorem.

We say that a set $\mathcal{S}$ of terms is *closed under ($\mathcal{S}$)-normalization* if any reduct of every $\mathcal{S}$-normalizable term is still $\mathcal{S}$-normalizable. Obviously, sets closed under parallel moves are closed under normalization as well. Even if $\mathcal{S}$ is closed under unneeded expansion, closure of $\mathcal{S}$ under normalization is also necessary for the normalization theorem to be valid for $\mathcal{S}$. Indeed, consider $R = \{f(x) \to g(x,x),\ a \to b\}$, take $\mathcal{S} = \{g(a,b)\}$, and take $t = f(a)$. Then $t \to g(a,a) \to g(a,b)$ is an $\mathcal{S}$-needed $\mathcal{S}$-normalizing reduction, while after the $\mathcal{S}$-needed step $t \to f(b)$, the term $f(b)$ is not $\mathcal{S}$-normalizable any more (the only redex in $f(b)$ is only trivially $\mathcal{S}$-needed). However, the following example shows that closure of $\mathcal{S}$ under normalization (even in combination with closure of $\mathcal{S}$ under unneeded expansion) is not enough; closure of $\mathcal{S}$ under parallel moves is necessary.

Example 4.1 Let $R = \{f(x) \to g(x,x),\ a \to b,\ b \to a\}$ and $\mathcal{S} = \{g(a,b)\}$. Since the reduction preserves the height of a term and the property to be a ground term, only the terms in the following diagram are $\mathcal{S}$-normalizable.



Therefore, it is clear from the diagram that $\mathcal{S}$ is closed under normalization. It is easy to see that, in $f(a)$ and $f(b)$, all the redexes are $\mathcal{S}$-needed; hence $f(a) \to f(b) \to f(a) \to \ldots$ is an infinite $\mathcal{S}$-needed reduction that never reaches $\mathcal{S}$ (there are many others). One can check that $\mathcal{S}$ is closed under unneeded expansion. Thus the reason for the failure of the normalization theorem is that, as it can be seen from the following diagram, $\mathcal{S}$ is not closed under parallel moves.



□

Proposition 4.1 A redex $u \subseteq t$ is Maranget-needed iff it is needed w.r.t. the set of all fair reductions starting from t.

Proof. ($\Rightarrow$) Let u be Maranget-needed in t. Then any fair P starting from t should contract a residual of u (to erase it). ($\Leftarrow$) If u is not Maranget-needed, i.e., there is a reduction $Q : t \twoheadrightarrow e$ in which u is erased and that does not contract the residuals of u, then there is a reduction Q' such that $Q^* = Q + Q'$ is fair and obviously u is not Q^*-needed, a contradiction.

Proposition 4.2 A redex $u \subseteq t$ is essential iff u (or $pat(u)$) is needed w.r.t. the set of all fair reduction starting from t.

Proof. An immediate corollary of Proposition 3.1 and Proposition 4.1.

Lemma 4.1 Let $P : t \twoheadrightarrow s$ be external to $e \sqsubseteq t$, and let $o \subseteq t$ be the subterm corresponding to e. Then any descendant of o along P is the subterm corresponding to some P-descendant of e.

Proof. Immediate from Definition 2.3, since P is external to e.

Proposition 4.3 A subterm $s \subseteq t$ is inessential iff there is a reduction that is external to $Int(s)$ and deletes it.

Proof. ($\Leftarrow$) Immediate from Lemma 4.1. ($\Rightarrow$) Let $x_1, \ldots, x_n$ be the list of occurrences of bound variables in s from left to right, let f be a fresh n-ary function symbol not occurring in left-hand sides of rewrite rules, and let $t^* = (f(x_1, \ldots, x_n)//s)t$. Since $IE(s,t)$, there is Q starting from t that deletes s. Therefore, it follows from Definition 3.6 that $P = ((Q\|t^*)\|t)$ is external to $Int(s)$ and deletes it.

5 A Labelling for OERSs

In Kennaway&Sleep [KeSl89] a labelling is introduced for OCRSs, based on the labelling system of Klop [Klo80], which is in turn a generalization of the labelling system for the λ-calculus introduced by Lévy [Lév78]. Each *label* of Kennaway&Sleep [KeSl89] is a tuple of labels, built up from a set of *base* labels. For any OERS R, terms in the corresponding labelled OERS R^L are those of R where each subterm has one or more labels, represented as a *string* of labels. A labelling of a term is *initial* if all its subterms are labelled by different base labels. The *signature* of a labelled term is the tuple of all its labels, from left to right. Rules of R^L are those of R where pattern-symbols in left-hand sides are labelled by a string of labels except for the head-symbol, which has just one label (a string of length one). Each subterm (including metavariables) in the right-hand side of a rule bears the signature of the corresponding left-hand side. Further, a *redex-index* of a redex is the maximal depth of nesting in the labels of the corresponding left-hand side of the rule. The *index* $Ind(P)$ of a reduction P is the maximal redex-index of redexes contracted in it.

The crucial properties of the labelling are given by the following propositions.

Proposition 5.1 [KeSl89] If a step $t \xrightarrow{u} s$ in an OERS R creates a redex $v \subseteq s$, then, for any labelling t^l of t, the corresponding step $t^l \xrightarrow{u^{l'}} s^{l''}$ in the corresponding labelled OERS R^L creates a redex v^{l^*} whose label l^* contains the label l' of u. Thus $Ind(u^{l'}) < Ind(v^{l^*})$. If $w \subseteq s^{l''}$ is a residual of a redex $w' \subseteq t^l$, then w and w' have the same labels, thus $Ind(w) = Ind(w')$.

Corollary 5.1 [KeSl89] Let P and Q be co-initial reductions such that P creates a redex u and Q does not contract residuals of any redex of t having a residual contracted in P. Then the redexes in $u/(Q/P)$ are created by P/Q and Q/P is external to u.

Proposition 5.2 [Klo80, Lév78] Any reduction in which only redexes with a bounded redex-index are contracted is terminating.

Remark 5.1 The above propositions are obtained for OCRS, but it is straightforward to carry them over OERSs. □

Definition 5.1 (1) For any co-initial reductions P and Q, the redex Qv in the final term of Q (read as *v with history Q*) is called a *copy* of a redex Pu if $P \unlhd Q$, i.e., $P + Q/P \approx_L Q$, and v is a Q/P-residual of u; the *zig-zag* relation $\simeq_z$ is the symmetric and transitive closure of the copy relation [Lév80]. A *family* relation is an equivalence relation among redexes with histories containing the zig-zag relation.

(2) For any co-initial reductions P and Q, the redexes Qv and Pu are in the same *labelling-family* if for any initial labelling of the initial term of P and Q, they bear the same labels.

Proposition 5.1 implies that the labelling-family relation is indeed a family relation. As pointed out in [AsLa93], for OERSs in general the zig-zag and labelling family relations do not coincide. Below by family we always mean the labelling-family.

6 The Relative Normalization Theorem

In this section, we present a uniform proof of correctness of the needed strategy that works for all stable sets of 'normal forms'. Our proof is different from all known proofs because properties of needed and unneeded components are different in the general case (the main difference is that a component under an unneeded component may be needed). However, the termination argument we use is the same as in [KeSl89] and in [Mar92], and is based on Proposition 5.2. The main idea and a proof in the same spirit is already in [Lév80].

Below in this section $\mathcal{S}$ always denotes a stable set of terms.

Lemma 6.1 Let $t \overset{w}{\to} s$, $v \subseteq t$, $o \sqsubseteq t$, and let $pat(v) \cap o = \emptyset$. Further, let $v' \subseteq s$ be a w-residual of v and $o' \sqsubseteq s$ be a w-descendant of o. Then $pat(v') \cap o' = \emptyset$.

Proof. Immediate from Definition 2.3.

Corollary 6.1 Let F be a set of redexes in t, and let every redex $v \in F$ be external to $s \sqsubseteq t$. Then any development of F is external to s.

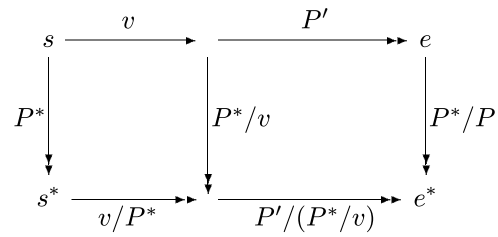
Lemma 6.2 Let $t \overset{v}{\to} s$, $P : t \twoheadrightarrow o$, $e \sqsubseteq t$, and v be external to e. Then v/P is external to every P-descendant of e.

Proof. By Lemma 6.1, every P-residual of v is externa to each P-descendant of e, and the lemma follows from Corollary 6.1.

Lemma 6.3 (1) Let $s_1, \ldots, s_n \sqsubseteq s$ be disjoint, let $P : s \twoheadrightarrow e \neq \emptyset$ be external to $s_1, \ldots, s_n$, let $P^* : s \twoheadrightarrow s^*$, let $s_1^*, \ldots, s_m^*$ be all P^*-descendants of $s_1, \ldots, s_n$ in s^*, and let $Q = P/P^* : s^* \twoheadrightarrow e^*$. Then Q is external to $s_1^*, \ldots, s_m^*$.

(2) If P $\mathcal{S}$-suppresses $s_1, \ldots, s_n$, then Q $\mathcal{S}$-suppresses $s_1^*, \ldots, s_m^*$.

Proof. (1) By induction on $|P|$. Let $P = v + P'$, let $s'_1, \ldots, s'_l$ be v-descendants of $s_1, \ldots, s_n$, and let $s_1'^*, \ldots, s_k'^*$ be P^*/v-descendants of $s'_1, \ldots, s'_l$. By $CR(com)$ (Theorem 2.1.(4)), $s_1'^*, \ldots, s_k'^*$ are v/P^*-descendants of $s_1^*, \ldots, s_m^*$. By the induction assumption, $P'/(P^*/v)$ is external to $s_1'^*, \ldots, s_k'^*$. But by Lemma 6.2 v/P^* is external to $s_1^*, \ldots, s_m^*$; hence $P/P^* = v/P^* + P'/(P^*/v)$ is external to $s_1^*, \ldots, s_m^*$.



(2) By (1) and closure of $\mathcal{S}$ under parallel moves.

Corollary 6.2 (1) Descendants of $\mathcal{S}$-unneeded redexes of $t \notin \mathcal{S}$ remain $\mathcal{S}$-unneeded.

(2) Residuals of $\mathcal{S}$-unneeded redexes of $t \notin \mathcal{S}$ remain $\mathcal{S}$-unneeded.

Lemma 6.4 (1) Let $t \xrightarrow{u} t'$ and $e \sqsubseteq s \sqsubseteq t$. Then any u-descendant of e is contained in some u-descendant of s.

(2) Let $e \sqsubseteq s \sqsubseteq t$ and $NE_{\mathcal{S}}(e, t)$. Then $NE_{\mathcal{S}}(s, t)$.

(3) Let $u \subseteq t$ and let $s \sqsubseteq pat(u)$. Then $NE_{\mathcal{S}}(u, t)$ iff $NE_{\mathcal{S}}(s, t)$.

Proof. (1) By Definition 2.3.

(2) By (1) and Definition 4.1.

(3) From Definition 4.1, since a reduction $\mathcal{S}$-suppresses s iff it $\mathcal{S}$-suppresses u (orthogonality of the system implies that any redex whose pattern contains a symbol from a residual of $pat(u)$ coincides with $pat(u)$ and hence contains a symbol from a residual of s as well).

Note that if a component $s \sqsubseteq t$ is below $o \sqsubseteq t$, then $UN_{\mathcal{S}}(o, t)$ does not necessarily imply $UN_{\mathcal{S}}(s, t)$, although the inessentiality of the subterm corresponding to o implies that of the subterm corresponding to e (Lemma 3.2.(3)). Take for example $R = \{f(x) \to g(x),\ a \to b\}$, and take for $\mathcal{S}$ the set of terms not containing occurrences of a. Then $\mathcal{S}$ is stable, a is $\mathcal{S}$-needed in $f(a)$, but $f(a)$ is not.

Lemma 6.5 Let $t \notin \mathcal{S}$, $t \xrightarrow{u} t'$, $UN_{\mathcal{S}}(u,t)$, and let $u' \subseteq t'$ be a u-new redex. Then $UN_{\mathcal{S}}(u',t')$.

Proof. $UN_{\mathcal{S}}(u,t)$ implies existence of $P : t \twoheadrightarrow e$ that $\mathcal{S}$-suppresses $pat(u)$; thus P is external to u. By Corollary 5.1, P/u is external to u'. Also, P/u is $\mathcal{S}$-normalizing since $\mathcal{S}$ is closed under parallel moves. Hence u' is $\mathcal{S}$-unneeded.

We call $P : t_0 \to t_1 \to \ldots$ *$\mathcal{S}$-needed* if it contracts only $\mathcal{S}$-needed redexes.

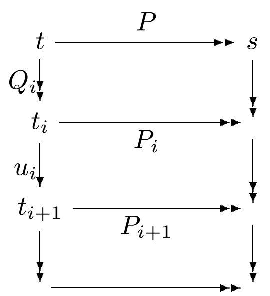
Theorem 6.1 (**Relative Normalization**) Let $\mathcal{S}$ be a stable set of terms in an OERS R.

(1) Any $\mathcal{S}$-normalizable term $t \notin \mathcal{S}$ in R contains an $\mathcal{S}$-needed redex.

(2) If $t \notin \mathcal{S}$ is $\mathcal{S}$-normalizable, then any $\mathcal{S}$-needed reduction starting from t eventually ends at a term in $\mathcal{S}$.

Proof. (1) Let $P : t \twoheadrightarrow s \xrightarrow{u} e$ be an $\mathcal{S}$-normalizing reduction that doesn't contain terms in $\mathcal{S}$ except for e. By the stability of $\mathcal{S}$, u is $\mathcal{S}$-needed. By Corollary 6.2.(2) and Lemma 6.5, it is either created by or is a residual of an $\mathcal{S}$-needed redex in s, and (1) follows by repeating the argument.

(2) Let $P : t \twoheadrightarrow s$ be an $\mathcal{S}$-normalizing reduction that doesn't contain terms in $\mathcal{S}$ except for e, and let $Q : t \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots$ be an $\mathcal{S}$-needed reduction. Further, let $Q_i : t \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots \xrightarrow{u_{i-1}} t_i$ and $P_i = P/Q_i$ $(i \geq 1)$. By Proposition 5.1, $Ind(P_i) \leq Ind(P)$. Since Q is $\mathcal{S}$-needed and P_i is $\mathcal{S}$-normalizing (by the closure of $\mathcal{S}$ under parallel moves), at least one residual of u_i is contracted in P_i. Therefore, again by Proposition 5.1, $Ind(u_i) \leq Ind(P_i)$. Hence $Ind(Q) \leq Ind(P)$ and Q is terminating by Proposition 5.2.



Lemma 6.6 Let t be $\mathcal{S}$-normalizable, let $t \xrightarrow{u} s$, $e \sqsubseteq t$, $NE_{\mathcal{S}}(e,t)$, and $pat(u) \cap e = \emptyset$. Then e has at least one $\mathcal{S}$-needed u-residual in s. In particular, any $\mathcal{S}$-needed redex $v \subseteq t$ different from u has an $\mathcal{S}$-needed residual.

Proof. Let $P : s \twoheadrightarrow o$ be an $\mathcal{S}$-needed $\mathcal{S}$-normalizing reduction; there is one by Theorem 6.1. Then if all u-residuals of e were $\mathcal{S}$-unneeded, P would $\mathcal{S}$-suppress them, and $u + P$ would $\mathcal{S}$-suppress e, a contradiction.

We call a stable set $\mathcal{S}$ *regular* if $\mathcal{S}$-unneeded redexes cannot duplicate $\mathcal{S}$-needed ones. One can show using Lemma 6.6 (e.g., as in [KeSl89] or in [Kha88]) that, for any regular stable $\mathcal{S}$, the $\mathcal{S}$-needed strategy is *$\mathcal{S}$-hypernormalizing*. That is, a term is $\mathcal{S}$-normalizing iff it does not have a reduction which contracts infinitely many $\mathcal{S}$-needed redexes. However, this is not the case for some irregular stable $\mathcal{S}$. Indeed, consider the OTRS $R = \{f(x) \to h(f(x), f(x)),\ a \to b\}$ and take for $\mathcal{S}$ the set of terms not containing occurrences of a. Then the reduction $f(a) \to h(f(a), f(a)) \to h(f(b), f(a)) \to h(f(b), h(f(a), f(a))) \to h(f(b), h(f(b), f(a))) \to \ldots$ contracts infinitely many $\mathcal{S}$-needed redexes, while $f(a) \to f(b)$ is $\mathcal{S}$-normalizing. This example shows also that multistep $\mathcal{S}$-needed reductions need not be $\mathcal{S}$-normalizing — just omit in the above reduction the initial step and group each pair of consecutive steps as a single multistep. Recall that multistep needed reductions are normalizing in the λ-calculus [Lév80]. The same holds for all regular stable $\mathcal{S}$; this follows immediately from hypernormalization of the $\mathcal{S}$-needed strategy for such $\mathcal{S}$.

7 Conclusions and Future Work

We have introduced a relative notion of neededness and proved the Relative Normalization Theorem in OERSs. We expect that this and other results of this paper can be proved for other higher-order rewriting systems too. Analogous questions arise for other strategies. For example, how can one construct reductions that avoid head-normal forms? Besides strong sequentiality for normal forms studied in Huet&Lévy [HuLé91], strong sequentiality is studied w.r.t. head-normal forms in Kennaway [Ken94]. Investigation of relative strong sequentiality and related to it strictness analysis (see e.g., [Nök94]) seems also as an interesting topic for future research. In forthcoming papers, we extend the theory of relative normalization in two directions: we study *minimal* and *optimal* relative normalization in OERSs, and study relative normalization in an abstract setting (in *Deterministic Residual Structures* and in *Family Structures*).

Acknowledgements

We thank J. R. Kennaway, F. van Raamsdonk, and M. R. Sleep for useful discussions, and J.-J. Lévy, L. Maranget, and P.-A. Melliès for help in overcoming difficulties in an early version of the paper. The use of empty descendants in the definition of descendants of components was suggested by L. Maranget. Some of the diagrams were drawn using P. Taylor's diagram package.

References

[Acz78] Aczel P. A general Church-Rosser theorem. Preprint, University of Manchester, 1978.

[AEH94] Antoy S., Echahed R., Hanus M. A needed narrowing strategy. In: Proc. of the 21^{st} ACM Symposium on Principles of Programming Languages, POPL'94, Portland, Oregon, 1994.

[AsLa93] Asperti A., Laneve C. Interaction Systems I: The theory of optimal reductions. Mathematical Structures in Computer Science, vol. 11, Cambridge University Press, 1993, p. 1-48.

[Bar84] Barendregt H. P. The Lambda Calculus, its Syntax and Semantics. North-Holland, 1984.

[BKKS87] Barendregt H. P., Kennaway J. R., Klop J. W., Sleep M. R. Needed Reduction and spine strategies for the lambda calculus. Information and Computation, v. 75, no. 3, 1987, p. 191-231.

[DeJo90] Dershowitz N., Jouannaud J.-P. Rewrite Systems. In: J. van Leeuwen ed. Handbook of Theoretical Computer Science, Chapter 6, vol. B, 1990, p. 243-320.

[Gar94] Gardner P. Discovering needed reductions using type theory. In: Proc. of the 2^{nd} International Symposium on Theoretical Aspects of Computer Software, TACS'94, Springer LNCS, v. 789, M. Hagiya, J. C. Mitchell, eds. Sendai, 1994, p. 555-574.

[HuLé91] Huet G., Lévy J.-J. Computations in Orthogonal Rewriting Systems. In: Computational Logic, Essays in Honor of Alan Robinson, J.-L. Lassez and G. Plotkin, eds. MIT Press, 1991.

[Ken94] Kennaway J. R. A conflict between call-by-need computation and parallelism. Workshop on conditional (and typed) term rewriting systems, Jerusalem, 1994.

[KeSl89] Kennaway J. R., Sleep M. R. Neededness is hypernormalizing in regular combinatory reduction systems. Preprint, School of Information Systems, University of East Anglia, Norwich, 1989.

[Kha88] Khasidashvili Z. β-reductions and β-developments of λ-terms with the least number of steps. In: Proc. of the International Conference on Computer Logic COLOG'88, Tallinn 1988, Springer LNCS, v. 417, P. Martin-Löf and G. Mints, eds. 1990, p. 105–111.

[Kha90] Khasidashvili Z. Expression Reduction Systems. Proceedings of I. Vekua Institute of Applied Mathematics of Tbilisi State University, vol. 36, 1990, p. 200-220.

[Kha92] Khasidashvili Z. The Church-Rosser theorem in Orthogonal Combinatory Reduction Systems. Report 1825, INRIA Rocquencourt, 1992.

[Kha93] Khasidashvili Z. Optimal normalization in orthogonal term rewriting systems. In: Proc. of the 5^{th} International Conference on Rewriting Techniques and Applications, RTA'93, Springer LNCS, vol. 690, C. Kirchner, ed. Montreal, 1993, p. 243-258.

[Kha94] Khasidashvili Z. On higher order recursive program schemes. In: Proc. of the 19^{th} International Colloquium on Trees in Algebra and Programming, CAAP'94, Springer LNCS, vol. 787, S. Tison, ed. Edinburgh, 1994, p. 172-186.

[Klo80] Klop J. W. Combinatory Reduction Systems. Mathematical Centre Tracts n. 127, CWI, Amsterdam, 1980.

[Klo92] Klop J. W. Term Rewriting Systems. In: S. Abramsky, D. Gabbay, and T. Maibaum eds. Handbook of Logic in Computer Science, vol. II, Oxford University Press, 1992, p. 1-116.

[KOR93] Klop J. W., van Oostrom V., van Raamsdonk F. Combinatory reduction systems: introduction and survey. In: To Corrado Böhm, J. of Theoretical Computer Science 121, 1993, p. 279-308.

[Lév78] Lévy J.-J. Réductions correctes et optimales dans le lambda-calcul, Thèse de l'Université de Paris VII, 1978.

[Lév80] Lévy J.-J. Optimal reductions in the Lambda-calculus. In: To H. B. Curry: Essays on Combinatory Logic, Lambda-calculus and Formalism, Hindley J. R., Seldin J. P. eds, Academic Press, 1980, p. 159-192.

[Mar92] Maranget L. La stratégie paresseuse. Thèse de l'Université de Paris VII, 1992.

[Nip93] Nipkow T. Orthogonal higher-order rewrite systems are confluent. In: Proc. of the 1^{st} International Conference on Typed Lambda Calculus and Applications, TLCA'93, Springer LNCS, vol. 664, Bazem M., Groote J.F., eds. Utrecht, 1993, p. 306-317.

[Nök94] Nöcker E. Efficient Functional Programming. Compilation and Programming Techniques. Ph. D. Thesis, Katholic University of Nijmegen, 1994.

[Oos94] Van Oostrom V. Confluence for Abstract and Higher-Order Rewriting. Ph. D. Thesis, Free University of Amsterdam, 1994.

[OR94] Van Oostrom V., van Raamsdonk F. Weak orthogonality implies confluence: the higher-order case. In: Proc. of the 3^{rd} International Conference on Logical Foundations of Computer Science, 'Logic at St. Petersburg', LFCS'94, Springer LNCS, vol. 813, Narode A., Matiyasevich Yu. V. eds. St. Petersburg, 1994. p. 379-392.

[Pkh77] Pkhakadze Sh. Some problems of the Notation Theory (in Russian). Proceedings of I. Vekua Institute of Applied Mathematics of Tbilisi State University, Tbilisi, 1977.

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