Stable Results and Relative Normalization

John Glauert, Richard Kennaway
School of Information Systems, UEA
Norwich NR4 7TJ England
J.Glauert@uea.ac.uk, R.Kennaway@uea.ac.uk

Zurab Khasidashvili
Department of Mathematics and Computer Science
Bar-Ilan University, Ramat-Gan 52900, Israel
Khasidz@cs.biu.ac.il *

Abstract

In orthogonal expression reduction systems, a common generalization of term rewriting and λ-calculus, we extend the concepts of normalization and needed reduction by considering, instead of the set of normal forms, a set S of "results". When S satisfies some simple axioms, we prove the corresponding generalizations of some fundamental theorems: the existence of needed redexes, that needed reduction is normalizing, the existence of minimal normalizing reductions, and the optimality theorem.

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 [10].

Normalization by Needed Reduction

For Orthogonal Term Rewriting Systems (OTRSs), a general normalizing strategy, called the *needed* strategy, was found by Huet and Lévy in [18]. The needed strategy always contracts a *needed* redex – a redex with at least one contracted *residual* in every reduction to normal form. Huet and Lévy show 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 this as the *Normalization Theorem.* They also define the class of *strongly sequential* OTRSs where a needed redex can be found efficiently in any term.

*Work undertaken at UEA with the partial support of the Engineering and Physical Sciences Research Council of Great Britain under grant GR/H 41300.

Extending the concept of Neededness

Barendregt et al. [7] generalize the concept of neededness to the λ-calculus. They study 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. They prove correctness of the two needed strategies for computing normal forms and head-normal forms, respectively. Middeldorp [40] studies normalization w.r.t. root-stable terms (which are terms that cannot be rewritten to a redex). Normalization w.r.t. another interesting set of 'normal forms', that of constructor head-normal forms in constructor OTRSs, is studied by Nöcker [41].

The normalization by neededness theory has been extended in other directions too, of which we mention a few. Khasidashvili defined the *essential* strategy for the λ-calculus [22], OTRSs [25], and *Expression Reduction Systems* (OERSs) [26]. This strategy contracts *essential* redexes – the redexes that have *descendants* under every reduction. The notion of descendant is a refinement of *residual* – the descendant of a contracted redex is its contractum, while it does not have residuals.[1] Essentiality makes sense for all subterms, not only redexes. In [35], Maranget introduces a different notion of neededness, where a redex u is needed if it has a residual under any reduction that does not contract residuals of u. This neededness notion makes sense even for terms that do not have a normal form, and coincides with the notion of essentiality.

Sekar and Ramakrishnan [43] study a normalizing strategy which proceeds by *multisteps*[2], each of which contracts a *necessary* set of redexes. Khasidashvili [26] shows that in Higher Order Recursive Program Schemes one can find all needed redexes in any term, implying decidability of weak and strong normalization. Khasidashvili and Piperno [28] designed an algorithm for statically finding all inessential subterms (i.e., the garbage) in simply typeable λ-terms. Gardner [11] described a complete way of encoding neededness information, for the case of the λ-calculus, 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. Antoy et al. [2] designed a needed narrowing strategy. Kennaway et al. [20] studied needed strategies for infinitary OTRSs. Van Oostrom [47] extended the essential strategy to weakly orthogonal systems, and Boudol [9] and Melliès [39] generalized the needed strategy to (not necessarily confluent) non-orthogonal systems. A different approach to normalization is developed by Kennaway [19] and by Antoy and Middeldorp [3].

The concept of Relative Neededness

Since in the practice of Functional Programming one is interested in terms of different shapes (weak-head-normal forms, constructor head-normal forms, etc.) as the results of computation, it is natural to ask what properties a set of terms must possess in order for the neededness theory of Huet and Lévy still to make sense. Here we provide a comprehensive solution to that question.

We introduce the notion of *neededness* w.r.t. a set of reductions Π so that each existing notion of neededness can be given by specifying Π. Usually it is convenient to consider a set of terms $\mathcal{S}$, and the induced set of reductions $\Pi_{\mathcal{S}}$ ending at a term

[1] Note that descendants and residuals are sometimes treated as synonymous in the literature.
[2] A multistep contracts a set of redexes simultaneously.

in $\mathcal{S}$. For example, *Huet and Lévy-neededness* [18] is neededness w.r.t. the set NF of normal forms, *Maranget-neededness* [35] is neededness w.r.t. all fair reductions, *head-neededness* [7] is neededness w.r.t. the set of head-normal forms, *root-neededness* [40] is neededness w.r.t. the set of *root-stable* forms, etc.

We impose a natural condition on $\mathcal{S}$, *stability*, which guarantees that each term not in $\mathcal{S}$-normal form (i.e., not in $\mathcal{S}$) has at least one $\mathcal{S}$-needed redex, such that repeated contraction of $\mathcal{S}$-needed redexes in a term t will 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 unneeded expansion*: for any $e \xrightarrow{u} o$ 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 $\mathcal{S}$; and is *closed under reduction*: for any $t \to o$ where $t \in \mathcal{S}$, then $o \in \mathcal{S}$. [3]

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 contracting both $\mathcal{S}$-needed and $\mathcal{S}$-unneeded redexes which never reaches a term in $\mathcal{S}$ even though $\mathcal{S}$-needed redexes are contracted infinitely many times. Therefore, *multistep $\mathcal{S}$-needed* reductions, where every multistep contracts at least one $\mathcal{S}$-needed redex, need not be $\mathcal{S}$-normalizing. This is because a 'non-standard' situation may arise in what we will call *irregular* stable sets $\mathcal{S}$, where $\mathcal{S}$-unneeded redexes may contain $\mathcal{S}$-needed ones. However, for regular stable sets $\mathcal{S}$, the $\mathcal{S}$-needed strategy is always hypernormalizing, and the multistep $\mathcal{S}$-needed strategy is normalizing.

Although we claim that our definition of stability is the most natural that has been proposed, it is possible to weaken the conditions for stability and still retain the concept of relative neededness. For example, an alternative definition is explored in [13, 12] where a stable set $\mathcal{S}$ is closed under unneeded expansion and *closed under parallel moves*: for any $t \notin \mathcal{S}$, any $P : t \twoheadrightarrow o \in \mathcal{S}$, and any $Q : t \twoheadrightarrow e$ not containing terms in $\mathcal{S}$, the final term of P/Q, the *residual* of P under Q, is in $\mathcal{S}$.

Minimal and Optimal Relative Normalization

We further develop a theory of *minimal* and *optimal* reduction in the framework of relative normalization, and establish a relationship between them. While normal forms are unique in an OERS, a term may have many $\mathcal{S}$-normal forms. A reduction $P : t \twoheadrightarrow s$ with $t \notin \mathcal{S}$ and $s \in \mathcal{S}$ is said to be *$\mathcal{S}$-minimal* if it does no more work than any other $\mathcal{S}$-normalizing reduction $Q : t \twoheadrightarrow e$, i.e., the residual P/Q of P under Q is empty. The final term in the $\mathcal{S}$-minimal reduction is said to be a *minimal* $\mathcal{S}$-normal form.

Minimal $\mathcal{S}$-normal forms are useful to compute since any other $\mathcal{S}$-normal form is accessible from the minimal one. Further, strategies computing partial results (such as head-normal-forms (hnfs) and weak hnfs, in the λ-calculus) usually compute minimal reductions, and it is natural to ask whether optimality can be achieved while retaining minimality. The prime example is the leftmost outermost strategy computing the so called 'principal' hnf and whnf of a λ-term, and used in constructions of Böhm [5] trees and Lévy-Longo [30, 33] (or *lazy*) trees, respectively. These trees represent the values of the term according to different semantics – Böhm semantics and lazy semantics,

[3]When the rewrite system is non-duplicating and non-erasing, our concept of stability coincides with the concept of stability in event structures [50].

respectively. Clearly this property of minimality is not useful for full normal forms, but full normal forms are rarely used in the practice of functional programming.

Our research on minimal $\mathcal{S}$-normalizing reductions was inspired by a result of Maranget [35], stating that *standard* reductions are minimal among reductions computing a 'stable prefix' of a given term. The earliest minimality result we are aware of was obtained by Berry and Lévy in [8], where existence of minimal reductions was shown for any finite or infinite approximation of a possibly infinite value of a term, for Recursive Program Schemes. Minimal reductions were used to design optimal reductions, both finite and infinite, and minimality and optimality of *outermost complete family-reductions* were shown. Minimal and optimal reductions were studied also in arbitrary interpretations, not only for term (or Herbrand) models.

Here we restrict ourselves to finite reductions only, and study only syntactic properties. We show that, for any stable and regular $\mathcal{S}$, any $\mathcal{S}$-normalizable term not yet in $\mathcal{S}$ possesses an $\mathcal{S}$-needed $\mathcal{S}$-*external* redex, and repeated contraction of such redexes gives $\mathcal{S}$-minimal $\mathcal{S}$-normalizing reductions. These redexes play the role of *standard* redexes w.r.t. $\mathcal{S}$, since they are $\mathcal{S}$-needed and cannot be duplicated. We show that an $\mathcal{S}$-normalizing reduction is $\mathcal{S}$-minimal iff it contracts $\mathcal{S}$-*erased* redexes, i.e., the redexes that do not have residuals under any $\mathcal{S}$-normalizing reduction. We show also that $\mathcal{S}$-minimal reductions need not exist if $\mathcal{S}$ is stable but is not regular.

Our study of optimal normalization w.r.t. stable sets $\mathcal{S}$ is a generalization of Lévy's optimality theory [32], developed for the λ-calculus. That is, we consider *multistep* reductions contracting a number of redexes in the same *family* in parallel, and consider optimality w.r.t. the number of such multisteps. This is chosen because Barendregt et al [6] showed that no one-step optimal recursive β-reduction strategy exists for the λ-calculus. We introduce a suitable labelling system, allowing us to define a family relation in OERSs. The generalization is then rather straightforward. We show that complete $\mathcal{S}$-needed family-reductions, which contract all members of a family containing an $\mathcal{S}$-needed redex in a multistep, are optimal.

It is easy to see that $\mathcal{S}$-needed complete family reductions, though optimal, need not be $\mathcal{S}$-minimal, because they may contract $\mathcal{S}$-unneeded redexes that are not $\mathcal{S}$-erased. It is tempting to think that contracting only the $\mathcal{S}$-needed redexes of $\mathcal{S}$-needed families could yield $\mathcal{S}$-optimal reductions that are $\mathcal{S}$-minimal at the same time. We show however that this is not the case either in the λ-calculus or in OTRSs.

Overview

In the next section, we review Expression Reduction Systems [23, 24, 27]. In Section 3, we introduce the relative notion of neededness. In Section 4, we sketch some properties of our labelling system for OERSs needed to define a family relation among redexes. We prove correctness of the $\mathcal{S}$-needed strategy for computing terms of $\mathcal{S}$, for all stable $\mathcal{S}$, in Section 5, and prove hypernormalization of the $\mathcal{S}$-needed strategy w.r.t. regular stable sets $\mathcal{S}$ in Section 6. In Section 7, we study $\mathcal{S}$-minimal reductions for regular stable sets $\mathcal{S}$. In Section 8 we establish a Relative Standardization Theorem. In Section 9, we prove the Relative Optimality Theorem. Finally, in Section 10, we relate relative optimal and minimal reductions. The conclusions appear in Section 11.

The concepts introduced in Section 3 build on [13], but here the definitions use only the residual concept while the earlier definitions were based on a concept of descendants for subterms and components. The stability concept is simpler, and hence slightly less general. The labelling system in Section 4 extends unpublished work by Kennaway and Sleep [21]. The proofs in Sections 5 and 6 simplify analogous proofs published as [13], and the remaining results are published in [14]. The results were first reported in [12].

Our results here apply to a general class of higher-order rewriting systems. The reader unacquainted with these can read our results in terms of the special cases of λ-calculusand term rewrite systems. For the benefit of those familiar with some form of higher-order rewriting, we describe in the next section the particular formalism that we use, for there are several in the literature to choose from.

2 Orthogonal Expression Reduction Systems

Klop introduced *Combinatory Reduction Systems* (CRSs) in [29] to provide a uniform framework for reductions with substitutions (also referred to as higher order rewriting) as in the λ-calculus [5] and its extensions. Restricted rewriting systems with substitutions were first studied in Pkhakadze [42] and Aczel [1]. Several interesting formalisms have been introduced later [24, 51, 36, 48]. We refer to van Raamsdonk [49] for a survey.

Expression Reduction Systems

Here we use *Expression Reduction Systems* (ERSs), defined in [24] (under the name of CRSs). The present formulation follows [27] and is simpler.

Definition 1 Let Σ be an *alphabet* comprising *variables* $x, y, z, \ldots$; *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, $\sigma x_1 \ldots x_m$ is a *compound operator* or a *quantifier* with *arity* n. Occurrences of $x_1, \ldots, x_m$ in $\sigma x_1 \ldots x_m$ are called *binding variables*. Each quantifier sign σ, as well as any corresponding quantifier $\sigma x_1 \ldots x_m$ and binding variables $x_1 \ldots x_m$, have a *scope indicator* $(k_1, \ldots, k_l)$ to specify the arguments in which $\sigma x_1 \ldots x_m$ binds all free occurrences of $x_1, \ldots, x_m$. *Terms* are constructed from variables by using functions and quantifiers in the usual way: variables are terms, and if $t_1, \ldots, t_n$ are terms and δ is an n-ary (simple or compound) operator, then $\delta(t_1, \ldots, t_n)$ is a term too.

Metaterms are constructed similarly from *terms* and *metavariables* $A, B, \ldots$ that range over terms, but with an extra operation: if $t_0, \ldots, t_n$ are metaterms, then so is $(t_1/x_1, \ldots, t_n/x_n)t_0$, also called a *metasubstitution*, 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 $(t_1/x_1, \ldots, t_n/x_n)t_0$ with the result of substitution of terms $t_1, \ldots, t_n$ for free occurrences of $x_1, \ldots, x_n$ in t_0. Because metasubstitutions can be nested, we specify that multiple substitutions are done innermost first, because it is clear that this process terminates. In fact, they can be done in any order, and will always give the same result, but there is no need to depend on this or prove it. The substitution operation may involve a *renaming* of bound variables to avoid collision, and we assume that the set of variables in Σ comes equipped with an equivalence relation, called renaming, such that any equivalence class of variables is infinite. We also assume that any variable can be renamed by any other variable in the corresponding equivalence class.[4] Unless otherwise specified, the default renaming relation is the total binary relation on variables (a partial renaming relation may be useful for conditional systems).

For example, a β-redex in the λ-calculus appears as $Ap(\lambda x\ t, s)$ in our notation, 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 A *Conditional Expression Reduction System* (CERS) is a pair (Σ, R), where Σ is an *alphabet* described in Definition 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.

Furthermore, each rule $r : t \to s$ has a set of *admissible assignments* $AA(r)$ which, in order to prevent undesirable confusion of variable bindings, must satisfy the following condition of being *variable-capture-free*:

[vcf] 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 so is any *variant* of $t\theta$ obtained from it by renaming of bound variables), and $s\theta$ is the *contractum* of $t\theta$.

If for any rule $r \in R$, $AA(r)$ is the maximal set of variable-capture free assignments, then the CERS is called an *unconditional* Expression Reduction System, or simply an Expression Reduction System (ERS).

For the sake of simplicity, we will restrict ourselves to *variable-capture-free ERSs*, which are ERSs in which all assignments are admissible for all rules. The results (and proofs) are valid for the much larger class of *fully-extended context-sensitive CERSs* [27]. Roughly, full extendedness means that an erasing step cannot turn a non-redex instance of a left-hand side of a rule into a redex. For example, the $\lambda\eta$-calculus is not fully extended because an erasing beta step can create an η-redex above the contractum. It was first observed by van Oostrom that the full extendedness restriction is necessary for many important results in the theory of higher-order rewriting, and since then this restriction has been adopted in many recent works [49, 45, 17, 46, 47, 27]. It is no surprise that the restriction is needed here.

We ignore questions relating to renaming of bound variables. As usual, a rewrite step consists of replacement of a redex by its contractum. Subterms of a redex corresponding to metavariables are *arguments* of the redex, and the rest is its *pattern*. Note that the use of metavariables in rewrite rules of ERSs is not really necessary – free variables can be used instead, as in TRSs, since (free) variables in TRS rules play the role of metavariables in ERS rules. We will indeed do so at least when giving TRS examples.

[4] An equivalence class of variables can, for example, be the set of variables of the same type in a typed language.

Examples

Our syntax is similar to that of Klop's CRSs [29], but has the attraction that it is closer to the syntax of the λ-calculus. For example, the β-rule is written as $Ap(\lambda xA, B) \to (B/x)A$, where A and B can be instantiated by any term; 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 μ-recursor rule is written as $\mu(\lambda xA) \to (\mu(\lambda xA)/x)A$. $\exists xA \to (\tau x(A)/x)A$ and $\exists !xA \to \exists xA \wedge \forall x \forall y(A \wedge (y/x)A \Rightarrow x = y)$ are rules corresponding to familiar definitions.

Notation 3 We use a, b, c, d for constants, t, s, e, o for terms, 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 contracting a redex $u \subseteq t$ is written as $t \xrightarrow{u} s$ or $t \to s$ or just u. We write $P : t \twoheadrightarrow s$ if P denotes a reduction of t to s. $P + Q$ denotes the concatenation of P and Q. We write $U \subseteq t$ if U is a set of redexes in t.

Orthogonality, Residuals, and Lévy-equivalence

The definition of *orthogonality* in ERSs is similar to the case of CRSs: all the rules are left-linear and redex-patterns never overlap [29]. The *residual* relation on redexes in ERSs is defined as a combination of the residual relations in TRSs and the λ-calculus, since any ERS step can be decomposed into a TRSs step followed by a number of substitution steps. Since the residual concept is familiar both in TRSs and the λ-calculus we do not reintroduce the concept here for ERSs, and instead refer to [24, 27] for more details. *Developments* of sets of redexes $U \subseteq t$ are defined in ERSs as usual [5].

As in the case of the λ-calculus [5] or OCRSs, for co-initial reductions P and Q, one can define in OERSs the notion of *residual of P under Q*, written P/Q, using Klop's method of commutative diagrams [29]. Klop's method is equivalent to Lévy's original definition of the residual relation in the λ-calculus [31, 32]; the latter is more 'algebraic' in nature and uses multisteps rather than complete developments.

We write $P \trianglelefteq Q$ if $P/Q = \emptyset$, where $\trianglelefteq$ is the *Lévy-embedding* relation. P and Q are called *Lévy-equivalent*[5], written $P \approx_L Q$, if $P \trianglelefteq Q$ and $Q \trianglelefteq P$. It follows from the definition of / that if P and Q are co-initial reductions in an OERS, then $(P + P')/Q = P/Q + P'/(Q/P)$ and $P/(Q + Q') = (P/Q)/Q'$.

We will often be interested in residuals of single redexes written u/P where u is a single-step reduction contracting u, a redex in the initial term of P. For example, $P \approx_L Q$ for finite P and Q implies that P and Q end at the same term and that $u/P = u/Q$ for all redexes, u, in the initial term.

The *Strong Church-Rosser (confluence)* property is established for OERSs in [24, 27]; the Finite Developments Theorem [5, 29] is proved first, from which strong confluence follows by a standard argument. Strong confluence for other higher-order rewriting formats are obtained, among others, in [29, 48, 44, 37].

[5] or *strongly-equivalent*, or *permutation-equivalent*

Theorem 4 (**Finite Developments**) All complete developments of a set of redexes in a term t, in an OERS, end at the same term s, and the residuals in s of any redex in t along any complete development are the same.

Theorem 5 (**Strong Church-Rosser**) For any co-initial reductions P and Q in an OERS, $P + Q/P \approx_L Q + P/Q$.

3 Stability and 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 $\mathcal{S}$; all existing notions of neededness can be obtained by specifying Π or $\mathcal{S}$; $\mathcal{S}$-neededness is a special case of Π-neededness. We introduce *stability* of a set of terms in an OERS. It is the aim of the Section 5 to show that if $\mathcal{S}$ is stable, then a $\mathcal{S}$-needed strategy is $\mathcal{S}$-normalizing.

Relative Neededness

Definition 6 (1) $P : t \twoheadrightarrow o$ is *external* to a redex $u \subseteq t$ if P does not contract any residual of u.

(2) $P : t \twoheadrightarrow o$ is *external* to a set of redexes $U \subseteq t$ if P is external to all $u \in U$.

(3) P *$\mathcal{S}$-suppresses* u if P is $\mathcal{S}$-normalizing and is external to u.

(4) P *erases* u if $u/P = \emptyset$. Note that contracting a redex erases it.

(5) P *discards* u if P is external to u and erases it.

Definition 7 (1) Let Π be a set of reductions in an OERS R. We call a redex $u \subseteq t$ Π-*needed* if in every reduction in Π starting from t at least one of its residuals is contracted (i.e., no reduction in Π is external to u) and call it Π-*unneeded* otherwise.

(2) Let $\mathcal{S}$ be a set of terms in an OERS R. Let $\Pi_{\mathcal{S}}$ be the set of all reductions that end at a term in $\mathcal{S}$. These will be denoted *$\mathcal{S}$-normalizing* reductions. A term t is denoted *$\mathcal{S}$-normalizing* if some $\mathcal{S}$-normalizing reduction starts at t. We call a redex u *$\mathcal{S}$-needed*, written $NE_{\mathcal{S}}(u,t)$, if it is $\Pi_{\mathcal{S}}$-needed, and call it $\mathcal{S}$-unneeded, written $UN_{\mathcal{S}}(u,t)$, otherwise.

Thus Huet and Lévy neededness coincides with neededness w.r.t. the set of normal forms. It is easy to see that Maranget's notion of neededness [35], where a redex is needed if it has a residual along any reduction external to it, coincides with neededness w.r.t. the set of fair reductions [12]. (Recall that a reduction is fair if all redexes in any of its terms are erased in it.) Obviously, any redex in a term that is not $\mathcal{S}$-normalizable is $\mathcal{S}$-needed; we call such redexes *trivially $\mathcal{S}$-needed*.

Stability

Definition 8 A set $\mathcal{S}$ of terms is *stable* if:

(a) $\mathcal{S}$ is *closed under reduction*: if $t \in \mathcal{S}$ and $t \to s$, then $s \in \mathcal{S}$; and

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

The most appealing examples of stable sets in an OERS are normal forms [18], head-normal forms [7], weak-head-normal forms, constructor-head-normal forms for constructor TRSs [41], and root-stable forms (terms that cannot be rewritten to a redex) [40].

Such sets are closed under reduction, which seems a natural condition for stability. Closure under unneeded expansion is less intuitive, but without it there will be $\mathcal{S}$-normalizable terms with no $\mathcal{S}$-needed redex. For example, consider $R = \{I(x) \to x\}$, and $\mathcal{S}$ that contains $I(t)$, but not $I(I(t))$, for some term t. There is no $\mathcal{S}$-needed redex in $I(I(t))$ as it can be reduced to $I(t)$ by reducing either I redex .

Weaker Stability Conditions

A weaker definition of stability is studied in [13] where a set $\mathcal{S}$ of terms is said to be stable if it is closed under unneeded expansion and is *closed under parallel moves*:

Definition 9 A set $\mathcal{S}$ is *closed under parallel moves* if for all $t \notin \mathcal{S}$, and $P : t \twoheadrightarrow o \in \mathcal{S}$, then for any $Q : t \twoheadrightarrow e$ that contains no term in $\mathcal{S}$, the final term of P/Q is in $\mathcal{S}$.

Clearly, closure under reduction implies closure under parallel moves. The $\mathcal{S}$-needed strategy can also be proven to be $\mathcal{S}$-normalizing under these weaker conditions. This rather technical definition ensures that if unneeded steps are taken when a normalizing needed reduction exists, it is still possible to reach a term in $\mathcal{S}$ by completing the residual of the normalizing reduction.

4 A Labelling for OERSs

We now introduce a labelling system for ERSs which will be used to establish termination of certain reductions and in section 9 to define the concept of a redex *family.* The labelling is a modification of one used by Klop [29] for CRSs.

This labelling system applies to all orthogonal ERSs (with one very minor restriction). Readers interested only in OTRSs and λ-calculusmay skip this section, as labelling systems due to Maranget [34] and Lévy [30, 31] suffice for these systems respectively. The concepts from this section that we use later are those of the label and the index of a redex, *lab*(u) and *Ind*(u).

Fix some OERS R. For technical reasons, we assume that R does not contain any rules whose left-hand side consists of just an operator applied to variables and metavariables. This is not a substantial restriction, since if the system includes a rule of the form $f(x, y, z) \to \ldots$, we can replace the symbol $f(\ldots)$ wherever it occurs by $g(f(\ldots))$, where g is a new unary function symbol. The new system will behave in all essential respects like the old and will satisfy the restriction. A similar restriction is adopted in [29].

Let there be a set $\mathcal{L}_0$ of *atomic labels*, and two sets $\mathcal{C}$ and $\mathcal{C}'$ of *label constructors* in $1-1$ correspondence with the set of subterms of all the right-hand sides of the rules of the OERS, and in $1-1$ correspondence with the rules of the OERS, respectively. The arity of each constructor in $\mathcal{C}'$ is the number of nodes in the left hand side of the rule which it is associated with, excluding variable and metavariable nodes and the root. And the arity of each constructor in $\mathcal{C}$ is 1. The set $\mathcal{L}$ of labels is the set of terms freely generated by $\mathcal{L}_0$ and $\mathcal{C} \cup \mathcal{C}'$. The *height* of a label is its height considered as a tree.

We add all members of $\mathcal{L}$ to the OERS as new unary function symbols. By a *labelled term* we mean any term which results from this extension of the signature. Every labelled term t determines an unlabelled term $U(t)$ by simply dropping the labels — that is, by replacing every subterm of the form $l(t')$ by t', where l is any label. A labelling of a term is *initial* if all its subterms are labelled by different atomic labels.

Although for technical reasons we have introduced labels as new function symbols, in writing terms down we shall indicate labels by superscripts. A term $\alpha(\beta(\gamma(t)))$ would be written $t^{\gamma\beta\alpha}$. This suggests an alternative way of looking at labels, closer to the way Lévy and Klop use them: the labels are considered to be annotations attached to the nodes of the syntax tree of the unlabelled term.

We now construct the set of *labelled rules*. Given any rule $s \rightarrow t$ of the unlabelled system, and labellings f of s and f' of t, $s^f \rightarrow t^{f'}$ is a labelled rule provided that:

- f does not label the root of s, nor any occurrence of a metavariable.
- f' attaches the label $c(c'(l_1, \ldots, l_n))$ to each subterm of t, other than metasubstitutions, where c is the label constructor corresponding to t, c' is the constructor corresponding to the rule, and $l_1, \ldots, l_n$ are the labels present in $f(s)$, listed in depth-first left-to-right order.

The choice of depth-first left-to-right ordering is not significant for the theory; it is merely a convenient standard choice. Note that f' is completely determined by f, s, and t. All nodes of $t^{f'}$ have distinct labels, and different labellings of s give different labellings of t. The reason for the restriction we made earlier is to ensure that every left-hand side has at least one place to attach a label to.

The set of labelled rules constitutes the system $\mathcal{L}(R)$. This is clearly an OERS.

As an example, consider the beta rule of lambda calculus. In OERS notation, and writing the application operator explicitly, it is:

$$Ap((\lambda x(A)), B) \rightarrow (B/x)A$$

An example of a redex is

$$Ap((\lambda y(Ap(y, y))), (\lambda z(z)))$$

A labelled version of this redex is:

$$(Ap((\lambda y(Ap(y^\alpha, y^\beta))^\gamma)^\delta, (\lambda z(z)^\epsilon)^\zeta))^\eta$$

This contains a redex of the labelled rule:

$$Ap((\lambda x(A))^{\delta}, B) \to (B^{c(g(\delta))}/x)A^{d(g(\delta))}$$

where g is the label constructor for the β-rule) and it reduces to:

$$((Ap(((\lambda z(z^{\epsilon}))^{\zeta})^{c(g(\delta))}, ((\lambda z(z^{\epsilon}))^{\zeta})^{c(g(\delta))}))^{d(g(\delta))})^{\eta}.$$

For a redex u in a labelled term, we define $lab(u)$ to be $c'(l_1, \ldots, l_n)$, where c' is the constructor for the rule for u and $(l_1, \ldots, l_n)$ is the tuple of labels in the left-hand side of the labelled rule for u. We call this the *label* of the redex. The *index* $Ind(u)$ of u is the height of $lab(u)$. The *index* $Ind(P)$ of a reduction P is the maximal index of the redexes contracted in it.

This definition of labelling differs slightly from that of Klop [29]. Klop combines multiple labels on a single term into a single compound label, and thus for his definitions a term t^{α} is not a subterm of $t^{\alpha\beta}$, whereas for us the term t^{α} is a subterm of $(t^{\alpha})^{\beta}$. He also represents compound labels as flat strings, instead of structured terms, using multiple underlining where we use multiple nesting. In addition, our use of label constructor symbols provides a greater generality which allows us simultaneously to treat several variants of labelled rewrite systems uniformly. (For example, in the case of the λ-calculus, there are two constructors in $\mathcal{C}$ corresponding to underlining and overlining [30, 31]. In the case of Interaction Systems [4] the constructors in $\mathcal{C}$ replace addresses in right-hand sides of rules. In the case of Klop's labelling system [29], all constructors are 'empty'. Redex-labels occurring as the arguments in right-hand sides of rewrite rules correspond to (constrained) labelled patterns in Maranget's labelling system for TRSs [34]. And in his labelling system for HRSs, van Oostrom [45] uses numbers where we use constructors in $\mathcal{C}$, and he uses rules where we use label constructors in $\mathcal{C}'$.) However, our definitions are sufficiently close that Klop's results for labelled systems carry over to the present setting.

The crucial properties of labelled reduction are given by the following propositions.

Proposition 10 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^* strictly 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 $Ind(w) = Ind(w')$.

Proof Easy from the definition of labelling. Note that this is where we require the restriction on labelled OERSs, that each left-hand side contain more than one function symbol.

Corollary 11 In an OERS, let P and Q be co-initial reductions such that P creates a redex u and Q is external to every redex of t having a residual contracted in P. Then every member of $u/(Q/P)$ is created by P/Q, and Q/P is external to u.

Proposition 12 In a labelled OERS, every reduction for which there is a bound on the indexes of the redexes it reduces is terminating.

Proof See [29, 37, 46].

$$
\begin{tikzcd}
t_0 \arrow[r, "v_0 = P_1"] \arrow[d, "Q_0"'] & t_1 \arrow[r, two heads] \arrow[d, "Q_1"'] & t_{n-1} \arrow[r, "v_{n-1}"] \arrow[d, "Q_{n-1}"'] & t_n \arrow[d, "Q_n"] \\
o_0 \arrow[r, two heads, "P'_1"'] & o_1 \arrow[r, two heads] & o_{n-1} \arrow[r, two heads, "P'_n"'] & o_n
\end{tikzcd}
$$

Figure 1: Lemma 13.

5 The Relative Normalization Theorem

In this section, we present a uniform proof of correctness of the needed strategy that works for all stable sets $\mathcal{S}$ of 'normal forms'. Our proof differs from all known proofs because properties of needed and unneeded redexes are different when arbitrary stable sets are considered. The main difference is that $\mathcal{S}$-unneeded redexes may replicate $\mathcal{S}$-needed ones. However, the termination argument we use is the same as in [21] and in [35], and is based on Proposition 12. The main idea and a proof in the same spirit is already in [32]. As in most earlier proofs, we use the fact that residuals of unneeded redexes remain unneeded, and that unneeded steps cannot create needed redexes. However, in order to prove that every needed redex has at least one needed residual after contracting any other redex, we need first to show the existence of a needed normalizing reduction.

In the rest of this section $\mathcal{S}$ always denotes a stable set of terms.

Lemma 13 Let $P : t_0 \xrightarrow{v_0} t_1 \xrightarrow{v_1} \ldots \to t_n$ be external to $U = \{u_1, \ldots, u_n\} \subseteq t_0$, and let $Q_0 : t_0 \twoheadrightarrow o_0$. Then $P' = P/Q_0$ is external to $U' = U/Q_0$. If P is $\mathcal{S}$-normalizing, then so is P'.

Proof Let $P_i : t_0 \xrightarrow{v_0} t_1 \xrightarrow{v_1} \ldots \to t_i$, $Q_i = Q_0/P_i$, and $P'_{i+1} = v_i/Q_i$, $0 \leq i < n$ (see Figure 1). Since P is external to U, we have for each i that $v_i \notin U/P_i$. Therefore, $v_i/Q_i \cap U/(P_i + Q_i) = \emptyset$ (since the residuals of different redexes are different). Thus, by Theorem 5, $v_i/Q_i \cap U/(Q_0 + P'_1 + \ldots + P'_i) = \emptyset$. Hence, P'_{i+1} is external to $U'/(P'_1 + \ldots + P'_i)$. This means that P' is external to U'. If P is $\mathcal{S}$-normalizing, then so is P', by the closure of $\mathcal{S}$ under reduction.

Corollary 14 For stable $\mathcal{S}$, residuals of $\mathcal{S}$-unneeded redexes are $\mathcal{S}$-unneeded.

Lemma 15 Let $t \notin \mathcal{S}$, $t \xrightarrow{u} t'$, $UN_{\mathcal{S}}(u, t)$, and let $u' \subseteq t'$ be a created redex. Then $UN_{\mathcal{S}}(u', t')$. (Unneeded redexes cannot create needed redexes.)

Proof $UN_{\mathcal{S}}(u, t)$ implies existence of $P : t \twoheadrightarrow e$ that $\mathcal{S}$-suppresses u; thus P is external to u. By Corollary 11, P/u is external to u'. Also, P/u is $\mathcal{S}$-normalizing since $\mathcal{S}$ is closed under reduction. Hence u' is $\mathcal{S}$-unneeded.

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

$$\begin{tikzcd}
t \arrow[r, two heads, "P"] \arrow[d, two heads, "Q_i"'] & s \arrow[d, two heads] \\
t_i \arrow[r, two heads, "P_i"'] \arrow[d, "u_i"'] & {} \arrow[d, two heads] \\
t_{i+1} \arrow[r, two heads, "P_{i+1}"'] \arrow[d, two heads] & {} \arrow[d, two heads] \\
{} \arrow[r, two heads] & {}
\end{tikzcd}$$

Figure 2: Theorem 16.

Relative Normalization

Theorem 16 (**Relative Normalization**) Let $\mathcal{S}$ be a stable set of terms in an OERS R.

(1) Every $\mathcal{S}$-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 reaches a term in $\mathcal{S}$.

Proof See Figure 2.

(1) Let $P : t \twoheadrightarrow s \xrightarrow{u} e$ be an $\mathcal{S}$-normalizing reduction that does not contain terms in $\mathcal{S}$ except for e. By the stability of $\mathcal{S}$, u is $\mathcal{S}$-needed. By Corollary 14 and Lemma 15, either it is a residual of an $\mathcal{S}$-needed redex in t, or it is created by an earlier step v of $t \twoheadrightarrow s$ where v is $\mathcal{S}$-needed. In the latter case the argument may be repeated.

(2) Let $P : t \twoheadrightarrow e$ be an $\mathcal{S}$-normalizing reduction that does not 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 $P_i = P/Q_i$, $i \geq 1$ (see the figure). Let R^L be the corresponding labelled OERS of R, let t have an initial labelling, and assume that P and Q are reductions in R^L. By Proposition 10, $\mathit{Ind}(P_i) \leq \mathit{Ind}(P)$. Since Q is $\mathcal{S}$-needed and P_i is $\mathcal{S}$-normalizing (by the closure of $\mathcal{S}$ under reduction), at least one residual of u_i is contracted in P_i. Therefore, again by Proposition 10, $\mathit{Ind}(u_i) \leq \mathit{Ind}(P_i)$. Hence $\mathit{Ind}(Q) \leq \mathit{Ind}(P)$ and Q is terminating by Proposition 12.

Notation $t\downarrow_{\mathcal{S}}$ denotes that t is $\mathcal{S}$-normalizable, i.e., reducible to a term in $\mathcal{S}$.

Lemma 17 Let $t\downarrow_{\mathcal{S}}$ and let $t \xrightarrow{u} s$. Then 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 16. Then if all u-residuals of v were $\mathcal{S}$-unneeded, P would $\mathcal{S}$-suppress them, and $u + P$ would $\mathcal{S}$-suppress v, a contradiction.

We conclude this section by summarizing the neededness properties of residuals and created redexes, established in Corollary 14, Lemma 15, and Lemma 17, in the following proposition.

Proposition 18 (1) Residuals of $\mathcal{S}$-unneeded redexes in a term $t \notin \mathcal{S}$ remain $\mathcal{S}$-unneeded.

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

(3) Let $t \downarrow_{\mathcal{S}}$, $t \xrightarrow{u} s$, $v \subseteq t$ be $\mathcal{S}$-needed, and $v \neq u$. Then v has an $\mathcal{S}$-needed residual in s.

6 The Relative Hypernormalization Theorem

In this section, we prove the Relative Hypernormalization Theorem for all *regular* stable sets of final terms in OERSs, and demonstrate that the theorem fails for some irregular stable sets. Thus this theorem refines the Relative Normalization Theorem, but is less general as it is restricted to regular stable sets of results only. However, regularity is not a restriction from a practical point of view, as all the sets of results that have been used in practical programming languages are regular. Moreover, it allows for much simpler proofs. We do not need to consider termination of reductions with bounded index (Proposition 12), as the Finite Developments Theorem is enough. We do not use the Standardization Theorem [18, 29] either.

Failure of Relative Hypernormalization for some stable sets

For the case of normal forms, the needed strategy is *hypernormalizing*, that is, reductions starting from a normalizable term that contract finite sequences of unneeded steps in addition to needed redexes are still normalizing.

The following example shows that this need not be the case for all stable sets $\mathcal{S}$:

Example 19 Consider $R = \{f(x) \to h(x, f(x)),\ a \to b\}$ and take for $\mathcal{S}$ the set of terms not containing occurrences of a. Then the reduction $f(a) \to h(a, f(a)) \to h(b, f(a)) \to h(b, h(a, f(a))) \to h(b, h(b, f(a))) \to \ldots$ contracts infinitely many $\mathcal{S}$-needed redexes, while the reduction $f(a) \to f(b)$ is $\mathcal{S}$-normalizing. This example shows also that multistep $\mathcal{S}$-needed reductions need not be $\mathcal{S}$-normalizing: group each pair of consecutive steps in the reduction above as a multistep.

Regular Stable Sets

For some stable $\mathcal{S}$, $\mathcal{S}$-unneeded redexes may contain $\mathcal{S}$-needed ones. Consider the simpler example $R = \{f(x) \to g(x),\ a \to b\}$, where $\mathcal{S}$ is the set of terms not containing occurrences of a: we observe that a is $\mathcal{S}$-needed in $f(a)$, but $f(a)$ is not.

Nesting of $\mathcal{S}$-needed redexes within $\mathcal{S}$-unneeded ones need not cause problems; the failure of hypernormalization arises from duplication of $\mathcal{S}$-needed redexes by $\mathcal{S}$-unneeded ones. Hence we introduce the following definition:

Definition 20 We call a stable set $\mathcal{R}$ *regular* if, for any $t \notin \mathcal{R}$, $\mathcal{R}$-unneeded redexes cannot duplicate $\mathcal{R}$-needed ones.

Recall from [32] that multistep needed reductions are normalizing in the λ-calculus. The same holds for all regular stable $\mathcal{R}$; this follows immediately from hypernormalization of the $\mathcal{R}$-needed strategy, which we will prove in the rest of the section.

Canonical forms for Relative Reductions

The proof uses the ability to transform any reduction into an equivalent form where $\mathcal{R}$-needed steps precede $\mathcal{R}$-unneeded steps, while conserving the number of $\mathcal{R}$-needed steps.

Definition 21 We call P *$\mathcal{S}$-quasi-needed* if $\mathit{NE}_{\mathcal{S}}|P| = \infty$, where $\mathit{NE}_{\mathcal{S}}|P|$ denotes the number of $\mathcal{S}$-needed steps in P. We call P *$\mathcal{S}$-semi-needed* if it can be expressed as $P = P_N + P_U$, where P_N is $\mathcal{S}$-needed and P_U is $\mathcal{S}$-unneeded. In the latter case, we call P_N the *$\mathcal{S}$-needed part* of P, and call P_U the *$\mathcal{S}$-unneeded part* of P.

In the following definition, we describe an algorithm that, for a *regular* stable $\mathcal{R}$ in an OERS, transforms any finite reduction P into an $\mathcal{R}$-semi-needed reduction Lévy-equivalent to P, and any $\mathcal{R}$-quasi-needed reduction Q into an infinite $\mathcal{R}$-needed reduction. The regularity of $\mathcal{R}$ is essential for termination of the transformation process, and for preservation of $\mathcal{R}$-quasi-neededness of the reduction under transformation. The latter property is crucial for our method of proving the Relative Hypernormalization Theorem.

Definition 22 Let Red be the set of all reductions in an OERS R, let $\mathcal{R}$ be a regular stable set of terms in R, and let Red^* be the set of $\mathcal{R}$-semi-needed reductions. We will define a function $K : \mathit{Red} \to \mathit{Red}^*$. For any reduction P, with finite $\mathcal{R}$-needed steps, we will denote the $\mathcal{R}$-needed part $K_N(P)$ and the $\mathcal{R}$-unneeded part $K_U(P)$ so that $K(P) = K_N(P) + K_U(P)$. K is defined as follows:

(a) Let P be a finite reduction. $K(P)$ is the $\mathcal{R}$-semi-needed reduction obtained from P as follows: find in P the leftmost subreduction $P_1 : t \xrightarrow{u} s \xrightarrow{v} o$ such that $\mathit{UN}_{\mathcal{R}}(u, t)$ and $\mathit{NE}_{\mathcal{R}}(v, s)$. Let $P = P_0 + P_1 + P_2$. By Proposition 18.(2), v is a residual of a redex $v' \subseteq t$, which is $\mathcal{R}$-needed by Proposition 18.(1). Since $\mathcal{R}$ is regular, v is the only residual of v', hence P_1 and $P_1' = v' + u/v'$ are both complete developments of the set $u, v' \subseteq t$. Now replace P_1 by P_1' in P. Transform the obtained reduction P' in the same way, and so on, as long as possible. Obviously, the number of $\mathcal{R}$-unneeded steps in P' preceding v' is less than the number preceding v in P, and the number of $\mathcal{R}$-needed steps in P and P' coincide. Therefore, the procedure terminates. The result is $K(P)$. Obviously, $P \approx_L K(P)$, and $K(P) \in \mathit{Red}^*$.

(b) Let $\mathit{NE}_{\mathcal{R}}|P| < \infty$ ($|P| = \infty$ is possible). P can be expressed as $P = P_1 + P_2$, where P_2 is $\mathcal{R}$-unneeded and the last step in P_1 is $\mathcal{R}$-needed. Then we take $K(P) = K(P_1) + P_2$, which is possible since $P_1 \approx_L K(P_1)$. Obviously, $K(P) \in \mathit{Red}^*$ and $P \approx_L K(P)$.

(c) Let $\mathit{NE}_{\mathcal{R}}|P| = \infty$. Suppose that P_i is the initial part of P with a length i and $Q_i = K_N(P_i)$. Let $|Q_i| = m_i$. We define $K(P)$ as the reduction whose prefix of length m_i is given by Q_i ($i = 0, 1, \ldots$). It follows from (a)-(b) that if $i < j$,

$$\begin{tikzcd}
t_0 \arrow[r, "u_0"] \arrow[d, "u = U_0"'] & t_1 \arrow[r, "u_1"] \arrow[d, "U_1"'] & t_2 \arrow[r, ->>] \arrow[d, "U_2"'] & {} \\
s_0 \arrow[r, ->>, "u_0/U_0"'] & s_1 \arrow[r, ->>, "u_1/U_1"'] & s_2 \arrow[r, ->>] & {}
\end{tikzcd}$$

Figure 3: Lemma 24.

then Q_i is a prefix of Q_j, so $K(P)$ is defined consistently. Obviously, $K(P)$ is $\mathcal{R}$-needed and hence $K(P) \in Red^*$.

Lemma 23 Let P be a finite or infinite reduction in an OERS, and let $\mathcal{R}$ be regular.

(1) If P ends at a term in $\mathcal{R}$, then $K_N(P)$ ends at a term in $\mathcal{R}$ as well.

(2) If $NE_{\mathcal{R}}|P| = \infty$, then $K(P)$ is $\mathcal{R}$-needed, and is infinite.

Proof (1) follows from the stability of $\mathcal{R}$ — $K(P)$ is $\mathcal{R}$-semi-needed, it ends at $\mathcal{R}$, and the step of $K(P)$ entering $\mathcal{R}$ is $\mathcal{R}$-needed.

(2) follows immediately from Definition 22.

Lemma 24 Let t_0 have an $\mathcal{R}$-quasi-needed reduction and $t_0 \xrightarrow{u} s_0$. Then s_0 also has an $\mathcal{R}$-quasi-needed reduction.

Proof By Lemma 23, t_0 has an infinite $\mathcal{R}$-needed reduction $P : t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots$. Let $U_i = u/(u_0 + \ldots + u_{i-1})$, $i = 0, 1, \ldots$. We can construct Figure 3. It follows from finiteness of developments that there are infinitely many numbers k such that $u_k \notin U_k$ (otherwise there should be a number m such that $t_m \xrightarrow{u_m} t_{m+1} \xrightarrow{u_{m+1}} \ldots$ is an infinite U_m-development). By Proposition 18.(3), $u_k \notin U_k$ and $NE_{\mathcal{R}}(u_k, t_k)$ imply that u_k has at least one $\mathcal{R}$-needed U_k-residual in s_k, i.e. u_k/U_k contains at least one $\mathcal{R}$-needed step. Hence P/u is $\mathcal{R}$-quasi-needed.

Relative Hypernormalization

Theorem 25 (**Relative Hypernormalization**) Let $\mathcal{R}$ be a regular stable set of terms in an OERS R, and let $t \notin \mathcal{R}$ be a term in R. Then t has an $\mathcal{R}$-normal form iff it does not possess a reduction in which infinitely many times $\mathcal{R}$-needed redexes are contracted.

Proof ($\Rightarrow$) Let $t \overset{P}{\twoheadrightarrow} s \in \mathcal{R}$. Suppose on the contrary that there is an $\mathcal{R}$-quasi-needed Q starting from t. Then by Lemma 24 Q/P is $\mathcal{R}$-quasi-needed as well. By closure of $\mathcal{R}$ under reduction, Q/P is in $\mathcal{R}$ – a contradiction, since terms in $\mathcal{R}$ do not contain $\mathcal{R}$-needed redexes. ($\Leftarrow$) By Theorem 16.(1), one can repeatedly contract $\mathcal{R}$-needed redexes in t, unless a term in $\mathcal{R}$ is reached; the latter is inevitable since t does not have an infinite $\mathcal{R}$-needed reduction.

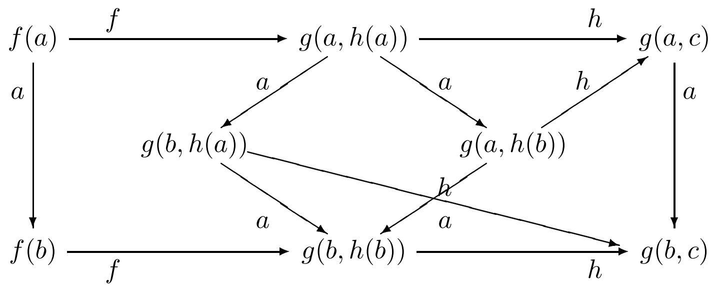

Figure 4: Example 27.

7 Minimal Relative Normalization

The theory of *minimal* reduction in the framework of relative normalization is based on reducing no more redexes than necessary when computing $\mathcal{S}$-normal forms. A reduction $P : t \twoheadrightarrow s$ with $t \notin \mathcal{S}$ and $s \in \mathcal{S}$ is $\mathcal{S}$-minimal if $P \trianglelefteq Q$ for all $\mathcal{S}$-normalizing $Q : t \twoheadrightarrow e$. Minimal $\mathcal{S}$-normal forms, such as s above, are useful to compute since any other $\mathcal{S}$-normal form is accessible from them.

We define $\mathcal{S}$-*external*, *persistently* $\mathcal{S}$-*needed*, and $\mathcal{S}$-*erased* redexes. We show that each class is a strict subset of the next when $\mathcal{S}$ is regular, and use such redexes to build $\mathcal{S}$-minimal reductions for regular $\mathcal{S}$. We show that $\mathcal{S}$-minimal reductions need not exist for irregular stable $\mathcal{S}$.

Persistently $\mathcal{S}$-needed and $\mathcal{S}$-erased redexes

Definition 26 (1) We call $u \subseteq t$ *persistently* $\mathcal{S}$-*needed* if all residuals of u are $\mathcal{S}$-needed.

(2) We call $u \subseteq t$ $\mathcal{S}$-*erased* if u does not have a residual under any $\mathcal{S}$-normalizing reduction.

(3) We call a reduction $\mathcal{S}$-*erased* if it only contracts $\mathcal{S}$-erased redexes.

Note that $\mathcal{S}$-erased redexes need not be $\mathcal{S}$-needed (e.g., when $\mathcal{S}$ is the set of normal forms and the OERS has an erasing rule, say $f(x) \to a$). The following example illustrates the introduced concepts using a simple OTRS.

Example 27 Consider an OTRS $R = \{f(x) \to g(x, h(x)),\ h(x) \to c,\ a \to b\}$ and a term (redex) $u = f(a)$, whose reduction graph is illustrated in Figure 4. Consider the following sets of terms in the reduction graph of u, in R: the set $\mathcal{S}_1 = \{g(b, c)\}$ of normal forms; the set $\mathcal{S}_2 = \{g(b, h(a)), g(b, h(b)), g(b, c)\}$ of terms not containing a redex on the left-spine, i.e. not containing a redex with its head symbol on the left-spine, when the term is considered as a tree; the set $\mathcal{S}_3 = \{f(b),\ g(b, h(b)),\ g(b, c)\}$ of terms not containing occurrences of a; and the set $\mathcal{S}_4 = \{g(a, h(b)),\ g(a, c),\ f(b),\ g(b, h(b)),\ g(b, c)\}$ of terms not containing a on the right-spine. Then $\mathcal{S}_1$ and $\mathcal{S}_2$ are regular stable sets; $\mathcal{S}_3$ is stable but not regular (since $\mathcal{S}_3$-unneeded redex u duplicates the $\mathcal{S}_3$-needed redex a); and $\mathcal{S}_4$ is not stable. Then, for the two redexes u and a in $u = f(a)$, we have the following:

1. u is $\mathcal{S}_1$-needed, persistently $\mathcal{S}_1$-needed, and $\mathcal{S}_1$-erased. $a \subseteq u$ is $\mathcal{S}_1$-needed but not persistently $\mathcal{S}_1$-needed, since the second residual of a in $g(a, h(a))$ is $\mathcal{S}_1$-unneeded. Nevertheless, a is $\mathcal{S}_1$-erased.
2. u is $\mathcal{S}_2$-needed, persistently $\mathcal{S}_2$-needed, and $\mathcal{S}_2$-erased. $a \subseteq u$ is $\mathcal{S}_2$-needed but not persistently $\mathcal{S}_2$-needed. a is not $\mathcal{S}_2$-erased as it has a residual along the $\mathcal{S}_2$-normalizing reduction $u \to g(a, h(a)) \to g(b, h(a))$.
3. u is neither (persistently) $\mathcal{S}_3$-needed nor $\mathcal{S}_3$-erased. $a \subseteq u$ is $\mathcal{S}_3$-needed but not persistently $\mathcal{S}_3$-needed (since the second residual of a in $g(a, h(a))$ is $\mathcal{S}_3$-unneeded); still, a is $\mathcal{S}_3$-erased.
4. Both u and a are neither (persistently) $\mathcal{S}_4$-needed nor $\mathcal{S}_4$-erased.

Lemma 28 Every persistently $\mathcal{S}$-needed redex is $\mathcal{S}$-erased, but an $\mathcal{S}$-erased redex, even if $\mathcal{S}$-needed, need not be persistently $\mathcal{S}$-needed.

Proof ($\Rightarrow$) Let $u \subseteq t$ be persistently $\mathcal{S}$-needed, and let $P : t \twoheadrightarrow s$ be $\mathcal{S}$-normalizing. If u/P was not empty, then every $u' \in u/P$ (the set of P-residuals of u) would be $\mathcal{S}$-needed, which is not possible since $s \in \mathcal{S}$. ($\Leftarrow$) From Example 27 (cases 1 and 3).

Minimal Relative Reduction

Definition 29 We call $P : t \twoheadrightarrow s$ $\mathcal{S}$-*minimal*[6] if it is $\mathcal{S}$-normalizing and $P \trianglelefteq Q$ for any $\mathcal{S}$-normalizing $Q : t \twoheadrightarrow o$. When P is $\mathcal{S}$-minimal, we call s a *minimal* $\mathcal{S}$-normal form of t.

It follows immediately from Definition 29 that if $t \downarrow_{\mathcal{S}}$ and $t \notin \mathcal{S}$, then t has no more than one minimal $\mathcal{S}$-normal form s. For any other $\mathcal{S}$-normal form e of t, it holds that $s \twoheadrightarrow e$. Note that the latter property of minimal $\mathcal{S}$-normal forms cannot be taken as the definition, because in that case an $\mathcal{S}$-normalizable term could have many minimal $\mathcal{S}$-normal forms, due for example to a cycle in $\mathcal{S}$, and some of them may require more reduction to be reached than others. For example, take $R = \{a \to b,\, b \to a,\, f(x) \to x\}$ and regular stable set $\mathcal{S} = \{a, b\}$. The term $t = f(a)$ has two $\mathcal{S}$-normal forms, a and b, and each is accessible from the other. However, any reduction from t to b must contract the $\mathcal{S}$-unneeded redex a and therefore no reduction from t to b can be considered as $\mathcal{S}$-minimal.

Lemma 30 Every $\mathcal{S}$-erased $\mathcal{S}$-normalizing reduction is $\mathcal{S}$-minimal.

Proof Let $P : t_0 \stackrel{u_0}{\to} t_1 \to \ldots \to t_n$ be an $\mathcal{S}$-erased $\mathcal{S}$-normalizing reduction, let $P_i : t_0 \stackrel{u_0}{\to} \ldots \to t_i$, and let $Q : t_0 \twoheadrightarrow o \in \mathcal{S}$. By stability of $\mathcal{S}$, $Q_i = Q/P_i$ is $\mathcal{S}$-normalizing. Since u_i is $\mathcal{S}$-erased and Q_i is $\mathcal{S}$-normalizing, $u_i/Q_i = \emptyset$. Hence $P/Q = \emptyset$, i.e., P is $\mathcal{S}$-minimal.

[6] We prefer to use minimal rather than *least* or *smallest*.

$$
\begin{tikzcd}
f(a) \arrow[rr] \arrow[dd] & & g(a,a) \arrow[dl] \arrow[dr] & \\
& g(a,b) \arrow[dr] & & \boxed{g(b,a)} \arrow[dl] \\
\boxed{f(b)} \arrow[rr] & & \boxed{g(b,b)} &
\end{tikzcd}
$$

Figure 5: Example 31.

Existence of Minimal $\mathcal{S}$-normal forms

In the study of $\mathcal{S}$-minimal reductions below, we will restrict ourselves to *regular* stable $\mathcal{S}$. The reason is that, as shown by the following example, an $\mathcal{S}$-normalizable term need not have an $\mathcal{S}$-minimal reduction when $\mathcal{S}$ is irregular.

Example 31 Consider $R = \{f(x) \to g(x,x),\ a \to b\}$, with term $t = f(a)$, and let $\mathcal{S} = \{g(b,a), f(b), g(b,b)\}$, the set of terms not containing a on the left spine. The reduction graph of $f(a)$ is given in Figure 5, with the members of $\mathcal{S}$ emphasized. Obviously, $\mathcal{S}$ is closed under unneeded expansion, because the only $\mathcal{S}$-needed redex in a term $s \notin \mathcal{S}$ is the leftmost occurrence of a in it, and $\mathcal{S}$ is closed under reduction. $\mathcal{S}$ is not regular, because the outermost redex in t is $\mathcal{S}$-unneeded, but duplicates the innermost one which *is* $\mathcal{S}$-needed.

There are three candidates for $\mathcal{S}$-minimal reductions starting from t: $P : f(a) \to f(b)$ and $Q : f(a) \to g(a,a) \to g(b,a)$ and $N : f(a) \to g(a,a) \to g(a,b) \to g(b,b)$. There are two more reductions that continue P and Q, but they clearly cannot be $\mathcal{S}$-minimal. We have $P \not\trianglelefteq Q$ as $P/Q = g(b,a) \to g(b,b)$, $Q \not\trianglelefteq P$ as $Q/P = f(b) \to g(b,b)$, and $N \not\trianglelefteq P$ as, again, $N/P = f(b) \to g(b,b)$. Hence none of the reductions is $\mathcal{S}$-minimal.

$\mathcal{S}$-external redexes

We go on to show that, when $\mathcal{S}$ is regular, an $\mathcal{S}$-normalizing reduction is $\mathcal{S}$-minimal iff it is $\mathcal{S}$-erased, i.e. contracts only $\mathcal{S}$-erased redexes. However, $\mathcal{S}$-erased reductions need not be $\mathcal{S}$-needed, and hence need not be $\mathcal{S}$-normalizing, and again for regular $\mathcal{S}$, we show existence of $\mathcal{S}$-external $\mathcal{S}$-normalizing reductions, which are $\mathcal{S}$-needed $\mathcal{S}$-minimal reductions.

Definition 32 Let $U \subseteq t$. We call P an *U-reduction* if it contracts only residuals of redexes from U and created redexes; we call such redexes *U-redexes*. Below $U(t)$ will denote the set of all redexes of t, and $U_{\mathcal{S}}(t)$ will denote the set of $\mathcal{S}$-needed redexes of t.

Definition 33 (1) Let $U \subseteq t$. We call a redex $u \subseteq t$ *U-external* (in t) if $u \in U$ and, for any U-reduction P, none of the residuals of u along P appear in arguments of U-redexes.

(2) We call $u \subseteq t$ *external in t* if it is $U(t)$-external, and *$\mathcal{S}$-external* if it is $U_{\mathcal{S}}(t)$-external. (Thus any $\mathcal{S}$-external redex is necessarily $\mathcal{S}$-needed.) We call a reduction P *$\mathcal{S}$-external* if each redex contracted in it is.

Example 34 Consider an OTRS $R = \{a \to c,\ b \to b',\ f(c, x) \to c'\}$, and take the term $t = g(f(a, b), a)$. Both occurrences of a in t are external in t, while b is not external in t: we have $t \to g(f(c, b), a) = s$, and the residual of b in s is in an argument of the created redex $f(c, b)$. If $U \subseteq t$ contains two redexes – the first occurrence of a in t and the redex $b \subseteq t$, then only the first $a \subseteq t$ is U-external in t. If the set of terms not having a left-spine redex is taken for $\mathcal{S}$, then the first a is the only $\mathcal{S}$-external redex in t (it is the only $\mathcal{S}$-needed redex too).

It is shown in [18, 25, 27] that any term t not in normal form contains an external redex (such redexes are called *unabsorbed* in [25, 13]). Now, if one ignores all redexes in t except those in $U \subseteq t$, it follows that, for any $U \neq \emptyset$, U contains an U-external redex. And by taking $U_{\mathcal{S}}(t)$ for U ($U_{\mathcal{S}}(t) \neq \emptyset$ by Theorem 16), we obtain the following proposition:

Proposition 35 Every term $t \notin \mathcal{S}$ contains an ($\mathcal{S}$-needed) $\mathcal{S}$-external redex.

Lemma 36 If a redex $u \subseteq t$ is $\mathcal{R}$-external, then it need not be external in t, but it cannot be replicated and is persistently $\mathcal{R}$-needed.

Proof Let $P : t \twoheadrightarrow o$, not necessarily an $U_{\mathcal{R}}(t)$-reduction. By Proposition 18.(3), it is enough to show that if a residual u' of u can appear inside an $\mathcal{R}$-needed redex $w' \neq u'$, then w' cannot replicate u'; therefore u has at most one residual in any term of P. Suppose, on the contrary, that there is $P : t \twoheadrightarrow o$ such that a residual u' of u is inside an $\mathcal{R}$-needed redex w' such that w' replicates u'; and assume that P is a shortest such a reduction, i.e., u has exactly one residual in every term in P. By Lemma 23, there are $\mathcal{R}$-needed P' and $\mathcal{R}$-unneeded P'' such that $P \approx_L P' + P''$. Since u' and w' are $\mathcal{R}$-needed and P'' is $\mathcal{R}$-unneeded, it follows from Proposition 18.(2) that there are $\mathcal{R}$-needed u'' and w'' in the final term of P' such that u' and w' are the only residuals of u'' and w'', respectively. Since u is $\mathcal{R}$-external, $u'' \notin w''$. Hence u'' has exactly one w''-residual, say u^*. By Theorem 5, $w'' + P''/w''$ replicates u'', since w' replicates u'. Thus P''/w'' replicates u^* – a contradiction, since P''/w'' is $\mathcal{R}$-unneeded by Proposition 18.(1), and $\mathcal{R}$ is regular.

Note that if $\mathcal{S}$ is irregular, then an $\mathcal{S}$-external redex $u \subseteq t$ need not be persistently $\mathcal{S}$-needed or $\mathcal{S}$-erased. Indeed, take R, $\mathcal{S}$, and Q as in Example 31. Then a in t is $\mathcal{S}$-needed, so is its leftmost residual in $g(a, a)$, but the rightmost residual is $\mathcal{S}$-unneeded, and $a/Q = \emptyset$. Hence $a \subseteq t$ is not persistently $\mathcal{S}$-needed or $\mathcal{S}$-erased. But $a \subseteq t$ is $\mathcal{S}$-external, since the only $U_{\mathcal{S}}(t)$-reduction is $N : f(a) \to f(b)$, and a is $U_{\mathcal{S}}(t)$-external in N.

Characterizing $\mathcal{R}$-minimal reductions

Proposition 37 An $\mathcal{R}$-normalizing reduction is $\mathcal{R}$-minimal iff it is $\mathcal{R}$-erased.

Proof ($\Leftarrow$) From Lemma 30. ($\Rightarrow$) Let $P : t_0 \xrightarrow{u_0} t_1 \to \ldots \to t_n$ be $\mathcal{R}$-minimal, and let $Q : t_0 \twoheadrightarrow o$ be $\mathcal{R}$-external, hence $\mathcal{R}$-erased by Lemmas 36 and 28, $\mathcal{R}$-normalizing reduction; Q exists by Proposition 35. Further, let $P_i : t_0 \xrightarrow{u_0} \ldots \to t_i$ and let $Q_i = Q/P_i$. Since Q is $\mathcal{R}$-erased, so is Q_i, and Q_i is $\mathcal{R}$-normalizing by the closure of $\mathcal{R}$ under reduction. Hence Q_i is $\mathcal{R}$-minimal by Lemma 30. Since P is $\mathcal{R}$-minimal too, $u_i/Q_i = \emptyset$ for every i. But for every $\mathcal{R}$-normalizing reduction $Q'_i : t_i \twoheadrightarrow o_i$, it holds that $Q_i \trianglelefteq Q'_i$ (since Q_i is $\mathcal{R}$-minimal). Hence $u_i/Q'_i = \emptyset$, i.e., u_i is $\mathcal{R}$-erased, and P is $\mathcal{R}$-erased too.

Minimal Relative Normalization

Theorem 38 (**Minimal Relative Normalization**) Let $\mathcal{R}$ be a regular stable set of terms in an OERS, and let $t \downarrow_{\mathcal{R}}$ where $t \notin \mathcal{R}$. Then repeated contraction of $\mathcal{R}$-needed $\mathcal{R}$-erased redexes in t yields an $\mathcal{R}$-minimal $\mathcal{R}$-normalizing reduction, even if a finite number of $\mathcal{R}$-unneeded $\mathcal{R}$-erased, and only such, redexes are also contracted. In particular, any $t \downarrow_{\mathcal{R}}$ where $t \notin \mathcal{R}$ has an $\mathcal{R}$-external $\mathcal{R}$-minimal reduction, which is $\mathcal{R}$-needed.

Proof By Proposition 35, any $t \downarrow_{\mathcal{R}}$ where $t \notin \mathcal{R}$ has an $\mathcal{R}$-external redex, which is $\mathcal{R}$-needed and $\mathcal{R}$-erased by Lemma 36 and Lemma 28. It remains to apply Theorem 16 and Proposition 37.

8 The Relative Standardization Theorem

Recall that a reduction is *standard* if redexes in it are contracted in left-to-right outside-in order [5, 18, 29]. Maranget proved in [35] that standard reductions are minimal among reductions computing a 'stable prefix' of a term, in an OTRS. Note that for regular $\mathcal{R}$, in general, $\mathcal{R}$-normalizing standard reductions need not be $\mathcal{R}$-needed, according to the definitions of [5, 18, 29], or according to [16], where left-to-right order of contracted redexes is not required.

Take for example $R = \{f(x) \to g(x,x),\ a \to b\}$, and take for $\mathcal{R}$ the set of terms not containing a redex on the right-spine; then $\mathcal{R}$ is regular, $f(a) \to g(a,a) \to g(b,a) \to g(b,b)$ is standard and $\mathcal{R}$-normalizing, but the second step is $\mathcal{R}$-unneeded and the final term is not a minimal $\mathcal{R}$-normal form.

Therefore, we should require that relative standard reductions are $\mathcal{R}$-minimal as in the following definition:

Definition 39 We call an $\mathcal{R}$-normalizing reduction $\mathcal{R}$-*standard* if it is outside-in and $\mathcal{R}$-minimal.

It is not difficult to check that $\mathcal{R}$-external $\mathcal{R}$-normalizing reductions are then $\mathcal{R}$-standard (see [12]), and the left-to-right order of contraction of $\mathcal{R}$-external redexes can also be arranged. Hence we have the following Relative Standardization Theorem:

Relative Standardization

Theorem 40 (**Relative Standardization**) Let $\mathcal{R}$ be a regular stable set of terms in an OERS, and let $t \downarrow_{\mathcal{R}}$ where $t \notin \mathcal{R}$. Then t has an $\mathcal{R}$-standard $\mathcal{R}$-normalizing reduction. In particular, $\mathcal{R}$-external $\mathcal{R}$-normalizing reductions are $\mathcal{R}$-standard.

9 The Relative Optimality Theorem

In this section, we generalize Lévy's Optimality Theorem to the case of all stable sets of normal forms, in OERSs. The family concept we use is based on the labelling system of Section 4.

Definition 41 For co-initial reductions $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow o$, redexes $u \in s$ and $v \in o$ with *histories* P and Q, written Pu and Qv, are in the same *(labelling-)family* if for any initial labelling of t, they bear the same labels.

Definition 42 A multistep reduction $P : t_0 \overset{U_0}{\twoheadrightarrow} t_1 \overset{U_1}{\twoheadrightarrow} \dots \twoheadrightarrow t_n$ is called a *family-reduction* if each U_i is a set of redexes belonging to the same family ($t_i \overset{U_i}{\twoheadrightarrow} t_{i+1}$ is the multistep corresponding to complete developments of U_i). $\|P\|$ will denote the number of multisteps in P. The family-reduction P is *complete* if each U_i is a maximal set of redexes of t_i belonging to the same family. A family-reduction P is called $\mathcal{S}$*-needed* if each U_i contains at least one $\mathcal{S}$-needed redex.

Notation: Below *FAM*(P) will denote the set of families (whose member redexes are) contracted in P. *Card*(*FAM*(P)) will denote the number of families in *FAM*(P).

Lemma 43 Every family is contracted at most once in a complete family-reduction.

Proof Let $P_n : t_0 \overset{U_0}{\twoheadrightarrow} t_1 \overset{U_1}{\twoheadrightarrow} \dots \overset{U_{n-1}}{\twoheadrightarrow} t_n$ be a complete family-reduction. We show by induction on $n = \|P\|$ that $(a)_n$: all families contracted in P_n are different; and $(b)_n$: there is no redex in t_n whose family has been contracted in P_n. So let us assume that P_n is a labelled reduction with t_0 having an initial labelling. The case $n = 0$ is clear. Further, $(a)_n$ follows immediately from $(a)_{n-1}$ and $(b)_{n-1}$. Again by $(a)_{n-1}$ and $(b)_{n-1}$, and by completeness of P_n, all redexes in t_n that are residuals of redexes of t_{n-1} are in families that have not been contracted before. The label of any new redex in t_n must contain $\mathit{lab}(U_{n-1})$. A redex with the same label cannot occur in $t_0, \dots, t_{n-1}$ since t_0 has an initial labelling and, by the induction assumption, $\mathit{lab}(U_{n-1}) \neq \mathit{lab}(U_i)$, for all $i < n-1$. Thus $(b)_n$ is also valid.

Relative Optimality

Theorem 44 (**Relative Optimality**) Let $\mathcal{S}$ be a stable set of terms in an OERS R, and let $t \downarrow_{\mathcal{S}}$. Then any $\mathcal{S}$-needed complete family-reduction starting from t is $\mathcal{S}$-optimal in the sense that it reaches an $\mathcal{S}$-normal form of t in a minimal number of family-reduction steps.

Proof Similar to the proof of the optimality theorem for the λ-calculus in [32]. Let $P : t \twoheadrightarrow s$ be an $\mathcal{S}$-normalizing family-reduction and $Q : t \overset{U_0}{\twoheadrightarrow} t_1 \overset{U_1}{\twoheadrightarrow} \dots$ be an $\mathcal{S}$-needed complete family-reduction. Further, let $Q_i : t \overset{U_0}{\twoheadrightarrow} t_1 \overset{U_1}{\twoheadrightarrow} \dots \twoheadrightarrow t_i$ and $P_i = P/Q_i$. By Proposition 10, *FAM*(P_i) $\subseteq$ *FAM*(P). Since Q is $\mathcal{S}$-needed, at least one residual of some $u_i \in U_i$ is contracted in P_i (P_i is $\mathcal{S}$-normalizing by the closure of $\mathcal{S}$ under reduction). Hence, again by Proposition 10, *FAM*(Q) $\subseteq$ *FAM*(P). Hence, by Lemma 43, $\|Q\| = \mathit{Card}(\mathit{FAM}(Q)) \leq \mathit{Card}(\mathit{FAM}(P)) \leq \|P\|$.

10 Relative optimal versus minimal reductions

It is easy to see that any $\mathcal{R}$-needed family-reduction that in each step contracts all the $\mathcal{R}$-needed redexes of some family, but does not necessarily contract its $\mathcal{R}$-unneeded members, is still optimal. We will call this an $\mathcal{R}$-needed *semi-complete* family-reduction. It follows from Proposition 37 that such a reduction is $\mathcal{R}$-minimal as well iff every $\mathcal{R}$-needed redex contracted in it is $\mathcal{R}$-erased.

For example, consider $R = \{g(x) \to f(x,x),\ a \to b\}$, where $\mathcal{R}$ is the set of terms not containing left-spine redexes. Then $g(a) \to f(a,a) \to f(b,a)$ is both $\mathcal{R}$-minimal and a $\mathcal{R}$-optimal semi-complete family-reduction, whereas complete family reductions would lead to $f(b,b)$ which is not the minimal $\mathcal{R}$-normal form.

However, the following example shows that a term in an OERS need not possess an $\mathcal{R}$-minimal $\mathcal{R}$-optimal semi-complete family-reduction:

Example 45 For the OERS $R = \{\sigma x AB \to \delta x(((A/x)B/x)A),\ f(A) \to g(A,A)\}$, let $\mathcal{R}$ be the set of terms not containing left-spine redexes. Then $\mathcal{R}$ is closed under unneeded expansion because for any $t \notin \mathcal{R}$ such that $t \xrightarrow{u} s \in \mathcal{R}$, u must be the outermost left-spine redex. Also, $\mathcal{R}$ is closed under reduction – no redex can be created or put on the left-spine without contracting a left-spine redex, which does not exist in terms from $\mathcal{R}$. Thus $\mathcal{R}$ is stable. $\mathcal{R}$ is moreover regular, since if there is u such that $UN_{\mathcal{R}}(u,t)$, then the reduction $P : t \twoheadrightarrow s \in \mathcal{R}$ that contracts outermost left-spine redexes is $\mathcal{R}$-needed, and is external to u, and each residual of u along P that is placed on the left-spine is discarded by contraction of a left-spine redex above it. But so do the residuals of redexes that are in u, and hence they cannot be $\mathcal{R}$-needed. That is, no $\mathcal{R}$-unneeded redex can contain (or duplicate) a $\mathcal{R}$-needed one.

Now let us consider the term $t = \sigma x(f(x), x)$. There are two $\mathcal{R}$-needed semi-complete family-reductions starting from t:

$$P : t \to \delta x(f(f(x))) \twoheadrightarrow \delta x(g(g(x,x), g(x,x)))$$

(since both occurrences of f in $\delta x(f(f(x)))$ are $\mathcal{R}$-needed) and

$$Q : t \to \sigma x(g(x,x), x) \to \delta x(g(g(x,x), g(x,x))),$$

but neither reaches the $\mathcal{R}$-minimal $\mathcal{R}$-normal form $\delta x(g(g(x,x), f(x)))$ of t, obtainable by the reduction $t \to \delta x(f(f(x))) \to \delta x(g(f(x), f(x))) \to \delta x(g(g(x,x), f(x)))$.

The reason why the above counterexample works is that redexes of the same family are nested. However the following examples show that, even if there is a 'hierarchy' of nesting of redexes of different families, the term still need not have $\mathcal{R}$-minimal $\mathcal{R}$-optimal family-reductions.

Example 46 Consider the OTRS $R = \{f(x) \to g(x,x),\ g(b,x) \to h(x,x),\ a \to b\}$, and again take for $\mathcal{R}$ the set of terms not containing left-spine redexes (e.g. $\{h(b,a), h(b,b)\}$). The reduction graph of $f(a)$ is illustrated in Figure 6, with members of $\mathcal{R}$ emphasized.

By the same argument as in Example 45, we can show that $\mathcal{R}$ is a regular stable set. Now $P : f(a) \to g(a,a) \to g(b,a) \to h(a,a) \to h(b,a)$ is an $\mathcal{R}$-minimal reduction, but $h(b,a)$ is not reachable by an $\mathcal{R}$-needed semi-complete family reduction. If the

$$\begin{tikzcd}
f(a) \arrow[rr] \arrow[dd] & & g(a,a) \arrow[dl] \arrow[dr] & & \\
& g(a,b) \arrow[dr] & & g(b,a) \arrow[dl] \arrow[r] & h(a,a) \arrow[dl] \arrow[d] \\
f(b) \arrow[rr] & & g(b,b) \arrow[dr] & \boxed{h(b,a)} \arrow[d] & h(a,b) \arrow[dl] \\
& & & \boxed{h(b,b)} &
\end{tikzcd}$$

Figure 6: Example 46.

first step reduces a then we reach the $\mathcal{R}$-normal form $h(b,b)$ which is not $\mathcal{R}$-minimal. Hence, in order to reduce $f(a)$ to $h(b,a)$, one should delay contraction of the $\mathcal{R}$-needed occurrences of a (which all belong to the same family). So $f(a) \to g(a,a)$ must be the first step. In $g(a,a)$, both occurrences of a are $\mathcal{R}$-needed, but their contraction makes $h(b,a)$ unreachable. Thus there is no $\mathcal{R}$-minimal reduction that is $\mathcal{R}$-optimal at the same time.

Example 47 Take for $\mathcal{R}$ the set of λ-terms in head-normal form, which is regular, and take $t = (\lambda x.xx)u$, where $u = (\lambda y.\lambda z.zvz)w$, and y, z, v and w are different variables. Then $P : t \to uu \to (\lambda z.zvz)u \to uvu \to (\lambda z.zvz)vu \to vvvu = e$ is an $\mathcal{R}$-minimal reduction. In order to reach e from t by a semi-complete $\mathcal{R}$-needed family reduction, one should delay contraction of $\mathcal{R}$-needed redexes in the family of u. So the outermost redex in t must be contracted first. In the obtained term $o = uu$, both occurrences of u are $\mathcal{R}$-needed, and their contraction would make e unreachable – there is no occurrence of w in $(\lambda z.zvz)(\lambda z.zvz)$.

Obviously, if reductions are non-duplicating, then every $\mathcal{R}$-optimal reduction is $\mathcal{R}$-minimal too, and every $\mathcal{R}$-needed $\mathcal{R}$-normalizing reduction is both $\mathcal{R}$-minimal and $\mathcal{R}$-optimal at the same time. This suggests that there may not be conflict between minimality and optimality when graph rewriting is concerned.

11 Conclusions

We have investigated properties which a set $\mathcal{S}$ of 'final terms' or '(partial) results' should possess in order for the normalization-by-neededness theory still to make sense. We introduced appropriate notions of neededness, defined stability and regularity of sets of terms, and proved a Normalization Theorem relative to stable sets of 'normal forms', and a Hypernormalization Theorem relative to regular stable sets of 'normal forms'. We have also studied minimal and optimal normalization relative to regular stable sets $\mathcal{R}$ of final terms, and have showed that $\mathcal{R}$-normalizing reductions that are both minimal and optimal need not exist for any $\mathcal{R}$-normalizable term t, despite the fact that t possesses minimal as well as optimal $\mathcal{R}$-normalizing reductions.

The abstract content of the relative normalization and optimality results has been analyzed by introducing *stable Deterministic Residual Structures (SDRSs)* and *Deterministic Family Structures*, respectively [15]. SDRSs axiomatize the residual relation in orthogonal systems, and DFSs are SDRSs with an axiomatized family relation. And our minimality results hold in such SDRSs where every set U of redexes in a term has an element that has at most one residual under any U-reduction (for example, U-external redexes in OERSs are such). Since this work was first presented, Melliès [38] has also established minimality results for his axiomatic rewrite systems.

Acknowledgments

We thank J.-J. Lévy, L. Maranget, P.-A. Melliès, V. van Oostrom, F. van Raamsdonk, and M. R. Sleep for useful comments and discussions. The referees provided valuable suggestions. The diagrams were drawn using P. Taylor's diagram package.

References

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

[2] S. Antoy, R. Echahed, and M. Hanus. A needed narrowing strategy. In 21^{st} *ACM Symposium on Principles of Programming Languages, POPL'94*, pages 268–279. ACM, 1994.

[3] S. Antoy and A. Middeldorp. A sequential reduction strategy. *Theoretical Computer Science*, 165(1):75–95, 1996.

[4] A. Asperti and C. Laneve. Interaction Systems I: The theory of optimal reductions. *Mathematical Structures in Computer Science*, 11:1–48, 1993.

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

[6] H. P. Barendregt, J. Bergstra, J. W. Klop, and H. Volken. Degrees, reductions and representability in the λ-calculus. Technical Report 22, University of Utrecht, 1976.

[7] H. P. Barendregt, J. R. Kennaway, J. W. Klop, and M. R. Sleep. Needed reduction and spine strategies for the lambda calculus. *Information and Computation*, 75(3):191–231, 1987.

[8] G. Berry and J.-J. Lévy. Minimal and optimal computations of recursive programs. *Journal of the ACM*, 26:148–175, 1979.

[9] G. Boudol. Computational semantics of term rewriting systems. In M. Nivat and J. C. Reynolds, editors, *Algebraic Methods in Semantics*, pages 169–236. Cambridge University Press, 1985.

[10] H. B. Curry and R. Feys. *Combinatory Logic*, volume 1. North-Holland, 1958.

[11] P. Gardner. Discovering needed reductions using type theory. In M. Hagiya and J. C. Mitchell, editors, 2^{nd} *International Symposium on Theoretical Aspects of Computer Software, TACS'94*, volume 789 of *LNCS*, pages 555–574. Springer-Verlag, 1994.

[12] J. R. W. Glauert and Z. Khasidashvili. Minimal and optimal relative normalization in expression reduction systems. Technical Report SYS-C94-06, UEA, Norwich, UK, 1994.

[13] J. R. W. Glauert and Z. Khasidashvili. On relative normalization in orthogonal expression reduction systems. In N. Dershowitz and N. Lindenstrauss, editors, 4^{th} *International Workshop on Conditional (and Typed) Term Rewriting Systems, CTRS'94*, volume 968 of *LNCS*, pages 144–165. Springer-Verlag, 1994.

[14] J. R. W. Glauert and Z. Khasidashvili. Minimal relative normalization in orthogonal expression reduction systems. In V. Chandru, editor, 16^{th} *International Conference on Foundations of Software Technology and Theoretical Computer Science, FST&TCS'96*, volume 1180 of *LNCS*, pages 238–249, 1996.

[15] J. R. W. Glauert and Z. Khasidashvili. Relative normalization in deterministic residual structures. In H. Kirchner, editor, 19^{th} *International Colloquium on Trees in Algebra and Programming, CAAP'96*, volume 1059 of *LNCS*, pages 180–195. Springer-Verlag, 1996.

[16] G. Gonthier, J.-J. Lévy, and P.-A. Melliès. An abstract standardisation theorem. In 7^{th} *Annual IEEE Symposium on Logic in Computer Science, LICS'92*, pages 72–81. The IEEE Computer Society Press, 1992.

[17] M. Hanus and C. Prehofer. Higher-order narrowing with definitional trees. In H. Ganzinger, editor, 7^{th} *International Conference on Rewriting Techniques and Applications, RTA'96*, volume 1103 of *LNCS*, pages 138–152. Springer-Verlag, 1996.

[18] G. Huet and J.-J. Lévy. Computations in orthogonal rewriting systems. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic, Essays in Honor of Alan Robinson*. MIT Press, 1991.

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

[20] J. R. Kennaway, J. W. Klop, M. R. Sleep, and F.-J. de Vries. Transfinite reductions in orthogonal term rewriting. *Information and Computation*, 119(1):18–38, 1995.

[21] J. R. Kennaway and M. R. Sleep. Neededness is hypernormalizing in regular combinatory reduction systems. Technical report, UEA, Norwich, UK, 1989.

[22] Z. Khasidashvili. β-reductions and β-developments of λ-terms with the least number of steps. In P. Martin-Löf and G. Mints, editors, *International Conference on Computer Logic, COLOG'88*, volume 417 of *LNCS*, pages 105–111. Springer-Verlag, 1990.

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

[24] Z. Khasidashvili. The Church-Rosser theorem in orthogonal combinatory reduction systems. Technical Report 1825, INRIA Rocquencourt, 1992.

[25] Z. Khasidashvili. Optimal normalization in orthogonal term rewriting systems. In C. Kirchner, editor, 5^{th} *International Conference on Rewriting Techniques and Applications, RTA'93, Montreal*, volume 690 of *LNCS*, pages 243–258. Springer-Verlag, 1993.

[26] Z. Khasidashvili. On higher order recursive program schemes. In S. Tison, editor, 19^{th} *International Colloquium on Trees in Algebra and Programming, CAAP'94, Edinburgh*, volume 787 of *LNCS*, pages 172–186. Springer-Verlag, 1994.

[27] Z. Khasidashvili, M. Ogawa, and V. van Oostrom. Perpetuality and uniform normalization in orthogonal rewrite systems. *Information and Computation*, To appear. Available from `http://www.brl.ntt.co.jp/people/mizuhito/papers/TRS.html`.

[28] Z. Khasidashvili and A. Piperno. Normalization of typeable terms by superdevelopments. In *Proceedings of CSL'98*, volume 1584 of *LNCS*, pages 260–282. Springer-Verlag, 1999.

[29] J. W. Klop. *Combinatory Reduction Systems*, volume 127 of *Mathematical Centre Tracts*. CWI, Amsterdam, 1980.

[30] J.-J. Lévy. An algebraic interpretation of the $\lambda\beta K$-calculus; and an application of a labelled lambda calculus. *Theoretical Computer Science*, 2:97–114, 1976.

[31] J.-J. Lévy. *Réductions Correctes et Optimales dans le Lambda-Calcul*. Thèse de l'Université de Paris VII, 1978.

[32] J.-J. Lévy. Optimal reductions in the lambda-calculus. In J. R. Hindley and J. P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda-calculus and Formalism*, pages 159–192. Academic Press, 1980.

[33] G. Longo. Set theoretic models of lambda calculus: Theories, expansions and isomorphisms. *Annals of Pure and Applied Logic*, 24:53–188, 1983.

[34] L. Maranget. Optimal derivations in weak lambda-calculi and in orthogonal term rewriting systems. In 17^{th} *ACM Symposium on Principles of Programming Languages, POPL'91*, pages 255–269. ACM, 1991.

[35] L. Maranget. *La Stratégie Paresseuse*. Thèse de l'Université de Paris VII, 1992.

[36] R. Mayr and T. Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192:3–29, 1998.

[37] P.-A. Melliès. *Description Abstraite des Systèmes de Réécriture*. Thèse de l'Université Paris VII, 1996.

[38] P.-A. Melliès. A stability theorem for rewriting theory. In *Proceedings of LICS'98*, 1998.

[39] P.-A. Melliès. Axiomatic rewriting theory ii: The $\lambda\sigma$-calculus enjoys finite normalisation cones. In *These proceedings*, 2000.

[40] A. Middeldorp. Call by need computations to root-stable form. In 24^{th} *ACM Symposium on Principles of Programming Languages, POPL'97*, pages 94–105. ACM, 1997.

[41] E. Nöcker. *Efficient Functional Programming: Compilation and Programming Techniques*. PhD thesis, University of Nijmegen, 1994.

[42] Sh. Pkhakadze. Some problems of the notation theory (in Russian). In *Proceedings of I. Vekua Institute of Applied Mathematics of Tbilisi State University*, Tbilisi, 1977.

[43] R. C. Sekar and I. V. Ramakrishnan. Programming in equational logic: Beyond strong sequentiality. *Information and Computation*, 104(1):78–109, 1993.

[44] V. van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Free University, Amsterdam, 1994.

[45] V. van Oostrom. Higher order families. In H. Ganzinger, editor, 7^{th} *International Conference on Rewriting Techniques and Applications, RTA'96*, volume 1103 of *LNCS*, pages 392–407. Springer-Verlag, 1996.

[46] V. van Oostrom. Finite family developments. In H. Common, editor, 8^{th} *International Conference on Rewriting Techniques and Applications, RTA'97*, volume 1232 of *LNCS*, pages 308–322. Springer-Verlag, 1997.

[47] V. van Oostrom. Normalisation in weakly orthogonal rewriting. In 10^{th} *International Conference on Rewriting Techniques and Applications, RTA'99*, volume 1631 of *LNCS*, pages 60–74. Springer-Verlag, 1999.

[48] V. van Oostrom and F. van Raamsdonk. Weak orthogonality implies confluence: The higher-order case. In A. Narode and Yu. V. Matiyasevich, editors, 3^{rd} *International Conference on Logical Foundations of Computer Science, LFCS'94*, volume 813 of *LNCS*, pages 379–392. Springer-Verlag, 1994.

[49] F. van Raamsdonk. *Confluence and Normalisation for Higher-Order Rewriting*. PhD thesis, Free University, Amsterdam, 1996.

[50] G. Winskel. An introduction to event structures. In *Linear Time, Branching Time and Partial Order in Logics and Models of Concurrency*, volume 354 of *LNCS*, pages 364–397. Springer-Verlag, 1989.

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