Title	Perpetuality and Uniform Normalization in Orthogonal Rewrite Systems
Author(s)	Khasidashvili, Zurab; Ogawa, Mizuhito; Oostrom, Vincent van
Citation	Information and Computation, 164(1): 118-151
Issue Date	2001-01-10
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/5032
Rights	NOTICE: This is the author's version of a work accepted for publication by Academic Press(Elsevier). Zurab Khasidashvili, Mizuhito Ogawa, Vincent van Oostrom, Information and Computation, 164(1), 2001, 118-151, http://dx.doi.org/10.1006/inco.2000.2888
Description	

Perpetuality and Uniform Normalization in Orthogonal Rewrite Systems

Zurab Khasidashvili
Department of Mathematics and Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel
E-mail: khasidz@cs.biu.ac.il

and

Mizuhito Ogawa
NTT Communication Science Laboratories, 3-1 Morinosato-Wakamiya Atsugi Kanagawa 243-0198, Japan
E-mail: mizuhito@theory.brl.ntt.co.jp

and

Vincent van Oostrom
Department of Philosophy, Utrecht University, P.O. Box 80089, 3508 TB Utrecht, The Netherlands
and
CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
E-mail: oostrom@phil.uu.nl

Communicated by Mariangiola Dezani-Ciancaglini

Received August 13, 1998; revised August 3, 1999; accepted September 15, 1999

We study perpetuality of reduction steps, as well as perpetuality of redexes, in orthogonal rewrite systems.

A perpetual step is a reduction step which retains the possibility of infinite reductions. A perpetual redex is a redex which, when put into an arbitrary context, yields a perpetual step. We generalize and refine existing criteria for the perpetuality of reduction steps and redexes in orthogonal Term Rewriting Systems and the λ-calculus due to Bergstra and Klop, and others.

We first introduce *Context-sensitive Conditional Expression Reduction Systems* (CCERSs) and define a concept of orthogonality (which implies confluence) for them. In particular, several important λ-calculi and their extensions and restrictions can naturally be embedded into orthogonal CCERSs. We then define a perpetual reduction strategy which enables one to construct *minimal* (w.r.t. Lévy's *permutation* ordering on reductions) infinite reductions in orthogonal fully-extended CCERSs.

Using the properties of the minimal perpetual strategy, we prove

1. perpetuality of any reduction step that does not erase *potentially infinite* arguments, which are arguments that may become, via substitution, infinite after a number of *outside* steps, and

2. perpetuality (in every context) of any *safe* redex, which is a redex whose substitution instances may discard infinite arguments only when the corresponding contracta remain infinite.

We prove both these perpetuality criteria for orthogonal fully-extended CCERSs and then specialize and apply them to restricted λ-calculi, demonstrating their usefulness. In particular, we prove the equivalence of weak and strong normalization (which equivalence is here called *uniform normalization*) for various restricted λ-calculi, most of which cannot be derived from previously known perpetuality criteria.

1. INTRODUCTION

The main objective of this paper is to study sufficient conditions for *uniform normalization*. Here a term t is uniformly normalizing, UN for short, if either it does not have any normal form (t is not weakly normalizing), or all reductions starting from t are finite, (t is strongly normalizing). We study UN for both first- and higher-order orthogonal term rewrite systems, where a rewrite system is said to be UN if each of its terms is so.

Interest in the criteria for UN arises, for example, in the proofs of strong normalization of typed λ-calculi, since these criteria are related to the work on reducing strong normalization proofs to proving weak normalization [50, 37, 23, 70, 17, 31, 24, 25, 65, 73, 49]. Furthermore, the question: 'Which classes of terms are UN ?' is posed by Böhm and Intrigila [11] in connection with finding UN solutions to fixed point equations, and with the representability of partial recursive functions by UN terms only, in the λ-calculus.[1] A useful UN subclass of λ-terms has recently been identified by Møller Neergaard and Sørensen [49].

Let us call a term t an ∞-*term* if it has an infinite reduction. Furthermore, we call a reduction step $t \to s$ and the corresponding contracted redex-occurrence *perpetual* if s is an ∞-term if t is so. A redex is called perpetual if its occurrence in every context (and the corresponding reduction step) is perpetual. It is easy to see that a rewriting system is UN iff all of its reduction steps are perpetual iff all of its redexes are perpetual. Studying uniform normalization therefore reduces to studying the perpetuality of redexes and reduction steps, which has been studied quite extensively. The classical results in this direction are Church's *Conservation Theorem for the* λ_I*-calculus* [13], stating that the λ_I-calculus is UN, and the *Conservation Theorem* (for the λ_K-calculus) due to Barendregt, Bergstra, Klop and Volken [7, 5], stating that β_I-redexes are perpetual in the λ-calculus. Bergstra and Klop [8] gave a necessary and sufficient criterion for the perpetuality of β_K-redexes. Klop [37] generalized Church's Theorem to non-erasing orthogonal Combinatory Reduction Systems (CRSs) by showing that those systems are UN, and Khasidashvili [31, 32] generalized the Conservation Theorem to orthogonal Expression Reduction Systems (ERSs) by proving that all non-erasing redexes are perpetual in orthogonal fully-extended ERSs.[2]

For orthogonal Term Rewriting Systems (TRSs), Klop [38] obtained a very powerful perpetuality criterion in terms of *critical* steps (or critical redex-occurrences). These are steps that are not perpetual, i.e., they reduce ∞-terms to SN terms. Klop showed that any critical step (contracting a redex-occurrence u) must erase an argument of u possessing an infinite reduction. This is not true for orthogonal higher-order rewrite systems, because substitutions (from the outside) into the arguments of u may occur during rewrite steps and such substitutions may turn a SN argument of u into an ∞-term. However, we show that (1) a critical step $t \xrightarrow{u} s$ must necessarily erase a *potentially infinite* argument, i.e., an argument that would become an ∞-(sub)term after a number of (*passive*, i.e., performed in the context of u) steps in t. From this we derive another criterion stating (2) perpetuality of *safe* redexes (in every context), which is similar to the perpetuality criterion for β_K-redexes [8]. These two criteria are the main results of this paper, and we will demonstrate their usefulness in applications.

To unify our results with the ones already in the literature for different orthogonal rewrite systems, we first introduce a framework of *Context-sensitive Conditional Expression Reduction Systems* (CCERSs). This framework provides a format for higher-order rewriting which extends ERSs [27] by allowing restrictions on term formation, on arguments of redexes, and on the contexts in which the redexes can be contracted. Various interesting typed λ-calculi (such as the simply typed λ-calculus [6], its extension with pairing [68], and system **F** [6]) and λ-calculi with specific reduction strategies (such as the call-by-value λ-calculus [60]) can be directly encoded as CCERSs (see also [39]). After demonstrating the expressiveness of CCERSs, we will focus our attention on orthogonal CCERSs, present a concept of

[1] Uniform normalization is called *strong normalization* in [11].

[2] The restriction to full extendedness was missing in [31]; full extendedness simply means that no rules are subject to occur conditions like the one in the η-rule.

orthogonality for CCERSs, and prove the standard results for orthogonal CCERSs (the Finite Developments Theorem [FD], confluence, etc.). Further, by necessity, we will restrict our attention to *fully-extended* orthogonal CCERSs; roughly, in fully-extended CCERSs, an erasing step cannot turn a non-admissible redex into an admissible one.

To prove our perpetuality criteria, we will first generalize, from term rewriting and the λ-calculus to orthogonal fully-extended CCERSs, the *constricting* perpetual strategies discovered independently by Plaisted [59], Gramlich [16], Sørensen [63], and Melliès [47]. These strategies specify a construction of infinite reductions (whenever possible) such that all steps are performed in some smallest ∞-subterm. Our strategy is slightly more general than the constricting ones (i.e., it specifies a set of redexes from which any one can be selected for contraction), and can be restricted so that resulting reduction sequences become constricting. The restricted strategy allows for simple and concise proofs of our perpetuality criteria. We will also show that constricting perpetual reductions are minimal w.r.t. Lévy's *permutation ordering* on reductions in orthogonal rewriting systems [44, 22].

Even though our criteria are simple and intuitive, they are strong tools in proving strong normalization from weak normalization in orthogonal (typed or type-free) rewrite systems. We will show that all known related criteria [13, 7, 8, 37, 38, 31], except the one in [21], can be obtained as special cases. We will also demonstrate that uniform normalization for a number of variations of β-reduction (most of which cannot be derived from previously known perpetuality criteria) [60, 17, 11, 20, 41] is an immediate consequence of our criteria. ERSs are similar to the Klop's CRSs [37] and we claim that all our results are valid for orthogonal fully-extended CRSs as well (see [61] for a detailed comparison of various forms of higher-order rewriting). We will demonstrate, however, that our results cannot be extended to higher-order rewriting systems where function variables can be bound [71, 52, 57], since already the Conservation Theorem fails for these systems.

The paper is organized as follows: In Section 2, we introduce CCERSs and show how several rewrite and transition systems can be encoded as CCERSs. In Section 3, we prove some standard results for orthogonal CCERSs. In Section 4, we study properties of an extension of existing constricting perpetual strategies, and in Section 5, we use these properties to obtain our perpetuality criteria for orthogonal fully-extended CCERSs. Section 6 gives a number of applications, and Section 7 concludes the paper.

The main results of this paper have been published previously in [33, 35].

2. CONTEXT-SENSITIVE CONDITIONAL ERSS

A term rewriting system is a pair consisting of an alphabet and a set of rewrite rules. The alphabet is used *freely* to generate the terms and the rewrite rules can be applied in any *surroundings* (context), generating the rewrite relation. In the first-order case one speaks of TRSs, while in the higher-order case there are several conceptually similar, but notationally often quite different, proposals. The first general higher order format was introduced long ago by Klop [37] under the name of *Combinatory Reduction Systems* (CRSs). Since then, several other interesting formalisms have been introduced [27, 71, 52, 45, 57]. Restricted rewriting systems with substitutions were first studied by Pkhakadze [58] and Aczel [2]. See van Raamsdonk [61] for a detailed comparison of various forms of higher-order rewriting.

It is often of interest to have the possibility of putting restrictions on the generation of either the terms or the rewrite relation or both. For example, many typed lambda calculi (such as the simply typed λ-calculus and the system **F** [6]) can be viewed as untyped lambda calculi with restrictions on the formation of *terms*. (See [39] for an encoding of the system **F** as a *substructure CRS*.) On the other hand, many rewrite strategies are naturally expressed by restricting the application of the *rewrite rules*. The call-by-value strategy in λ-calculus [60], for example, can be specified by restricting the second *argument* of the β-rule to values. In general, restricting arguments gives rise to so-called *conditional* ERSs (cf. [8]). The leftmost-outermost strategy can be specified by restricting the *context* in which the β-rule may be applied. We will call the latter kind of rules in which contexts are restricted *context-sensitive*.[3] We will now introduce *CCERSs* which allow all three kinds of restriction.

2.1. The syntax of CCERSs

CCERSs are an extension of ERSs, which are based on the syntax of Pkhakadze [58]. Terms in CCERSs are built from the alphabet just like they are in the first-order case. The symbols having binding power (like the λ in λ-calculus and the $\int$ in integrals) require some binding variables and terms as arguments, as specified by their *arity*. *Scope indicators* are used to specify which variables have binding power in which arguments. 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 $\int x(s,t,f(x))$ by using an operator sign $\int$ of arity $(1,3)$ and scope indicator (3).

Metaterms will be used to write rewrite rules. They are constructed from *metavariables* and meta-expressions for substitutions, called *metasubstitutions*. Instantiation of metavariables in metaterms yields terms. Metavariables play the rôle of variables in the TRS rules and of function variables in other formats of higher-order rewriting such as Higher-Order TRSs (HOTRSs) [71], Higher-Order Rewrite Systems (HRS) [52], and Higher-Order Rewriting Systems (HORSs) [57]. Unlike the function variables in HOTRSs, HRSs, and HORSs, however, metavariables *cannot* be bound.

Definition 2.1. Let Σ be an *alphabet* comprising infinitely many *variables*, denoted by x, y, $z, \ldots$, and *symbols* (*signs*). A symbol σ can be either a *function symbol* (*simple operator*) having an *arity* $n \in \mathcal{N}$ or an *operator sign* (*quantifier sign*) having *arity* $(m,n) \in \mathcal{N}^+ \times \mathcal{N}^+$. If it is an operator sign it needs to be supplied with m *binding* variables $x_1,\ldots,x_m$ to form a *quantifier* (*compound operator*) $\sigma x_1 \ldots x_m$, and it also has a *scope indicator* specifying in which of the n arguments it has binding power.[4] *Terms* t, s, e, o are constructed from variables, function symbols,

[3]The distinction between 'conditional' and 'context-sensitive' is, however, more historical than conceptual.

[4]Scope indicators can be avoided at the expense of side conditions of the form $x \notin FV(s)$. In this case, in order to avoid unintended bindings, such conditions must be imposed on construction of (admissible) terms rather than on the usage of rewrite rules.

and quantifiers in the usual first-order way, respecting (the second component of the) arities. A predicate AT on terms specifies which terms are *admissible.*

Metaterms are constructed like terms, but also allowing *metavariables* A, $B, \ldots$ and *metasubstitutions* $(t_1/x_1, \ldots, t_n/x_n)t_0$, where each t_i is an arbitrary metaterm and the x_i have a binding effect in t_0. Metaterms without metasubstitutions are called *simple.* An *assignment* θ maps each metavariable to a term. The application of θ to a metaterm t is written $t\theta$ and is obtained from t by replacing metavariables with their values under θ and by replacing metasubstitutions $(t_1/x_1, \ldots, t_n/x_n)t_0$, in right to left order, with the result of substitution of terms $t_1,\ldots,t_n$ for free occurrences of $x_1,\ldots,x_n$ in t_0. 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.[5] Unless otherwise specified, the default renaming relation is the total binary relation on variables (a partial renaming relation may be useful for conditional systems).

The specification of a CCERS consists of an alphabet (generating a set of terms possibly restricted by the predicate AT as specified above), and a set of rules (generating the rewrite relation possibly restricted by *admissibility* predicates AA and AC as specified below). The predicate AT can be used to express sorting and typing constraints, since sets of admissible terms allowed for arguments of an operator can be seen as terms of certain sorts or types. The predicates AA and AC impose restrictions respectively on arguments of (admissible) redexes and on the contexts in which they can be contracted.

The CCERS syntax is very close to the syntax of the λ-calculus. Those already familiar with the λ-calculus may therefore find ERSs easier to understand than CRSs, although the differences between the two are 'semantically' insignificant. See also [61]. 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 terms. The η-rule is written as $\lambda xAp(A, x) \to A$, where for any assignment $\theta \in AA(\eta)$, $x \notin FV(A\theta)$ (the set of free, i.e., unbound, variables of $A\theta$); otherwise an x occurring free in $A\theta$ and therefore bound in $\lambda xAp(A\theta, x)$ would become free. A rule like $f(A) \to \exists x(A)$ is also allowed, but in that case the assignment θ with $x \in A\theta$ is not allowed. Such a collision between free and bound variables cannot arise when assignments are restricted by the condition $(*)$, described below.

Familiar rules for defining existential quantifier $\exists x$ and the quantifier $\exists!x$ (there exists exactly one x) are written as $\exists x(A) \to (\tau x(A)/x)A$ and $\exists!x(A) \to \exists x(A) \wedge \forall x\forall y(A \wedge (y/x)A \Rightarrow x = y)$, respectively. For the assignment associating $x = 5$ to the metavariable A, these rules generate rewrite steps $\exists x(x = 5) \to \tau x(x = 5) = 5$ and $\exists!x(x = 5) \to \exists x(x = 5) \wedge \forall x\forall y(x = 5 \wedge y = 5) \Rightarrow x = y)$. In general, evaluation of a reduction step may involve execution of a number of substitutions

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

corresponding to the metasubstitutions in the right-hand-side of the rule. This will be explained by examples in the next section.

DEFINITION 2.2. A *Context-sensitive Conditional Expression Reduction System* (CCERS) 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., metaterms possibly containing 'free' metavariables but not containing free variables).

Furthermore, each rule r has a set of *admissible assignments* $AA(r)$ which, to prevent confusion of variable bindings, must satisfy the following *variable-capture-freeness* condition:

(*) 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 occurrences are.

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 by renaming of bound variables), and $s\theta$ is the *contractum* of $t\theta$. We call R *simple* if the right-hand sides of R-rules are simple metaterms. We call redexes that are instances of the same rule *weakly similar.*

Furthermore, each pair (r, θ) with $r \in R$ and $\theta \in AA(r)$ has a set $AC(r, \theta)$ of *admissible contexts* such that if a context $C[\,]$ is admissible for (r, θ) and o is the contractum of $u = r\theta$ according to r, then $C[u] \to C[o]$ is an R-reduction step. In this case, u is *admissible* for r in the term $C[u]$. We require that the set of admissible terms be closed under reduction. We also require that admissibility of terms, assignments, and contexts be closed under the renaming of bound variables.[6]

We call a CCERS *context-free*, or simply a *Conditional Expression Reduction System* (CERS), if every term is admissible, if every context is admissible for any redex, if the rules $r : t \to s$ are such that t is a simple metaterm and is not a metavariable, and if each metavariable that occurs in s also occurs in t. Moreover if for any rule $r \in R$, $AA(r)$ is the maximal set of variable-capture-free assignments, then we call the CERS an *unconditional* Expression Reduction System, or simply an Expression Reduction System (ERS).[7]

Note that in CCERSs (but not in CERSs or ERSs) we allow metavariable-rules like $\eta^{-1} : A \to \lambda x Ap(A, x)$ and metavariable-introduction-rules like $f(A) \to g(A, B)$, which are usually excluded a priori. This is useful only when the system is conditional. Like in the η-rule,the requirement $(*)$ forces $x \notin FV(A\theta)$ for every $\theta \in AA(\eta^{-1})$.

Let $r : t \to s$ be a rule in a CCERS R and let θ be admissible for r. Subterms of a redex $v = t\theta$ that correspond to the metavariables in t are the *arguments* of v, and the rest of v is the *pattern* of v (hence the binding variables of the quantifiers occurring in the pattern also belong to the pattern). Subterms of v whose head symbols are in its pattern are called the *pattern-subterms* of v. The pattern of the right-hand side of a simple CCERS rule is defined similarly.

[6]Closure of admissibility of contexts under the renaming of bound variables may need some clarification: We mean that if u is admissible in $C[u]$ and if $C'[u']$ is its variant (obtained by a renaming of bound variables in $C[u]$), then u' must be a redex admissible for $C'[\,]$.

[7]The renaming relation for ERSs is total.

Notation We use a, b, c, d for constants, use t, s, e, o for terms and metaterms, use u, v, w for redexes, and use N, P, Q for reductions (i.e., reduction paths). We write $s \subseteq t$ if s is a subterm (occurrence) 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 (sequence) from t to s, write $P : t \twoheadrightarrow$ if P may be infinite, and write $P : t \twoheadrightarrow \infty$ if P is infinite (i.e, of the length ω). For finite P, $P + Q$ denotes the concatenation of P and Q.

Below, when we refer to terms and redexes, we will always mean admissible terms and admissible redexes except that are explicitly mentioned.

2.2. Expressive power of CCERSs

To avoid a significant deviation from the main theme, how to encode conditional TRSs [9] and reduction strategies as CCERSs is described in this subsection only very briefly. For more details refer to Khasidashvili and van Oostrom [34] where, for example, encodings of Hilbert- and Gentzen-style proof systems into CCERSs are also given. An encoding of the π-calculus into a CCERS is given in Appendix A.1.

Conditional TRSs

Conditional term rewriting systems (CTRSs) were introduced by Bergstra and Klop [9]. Their conditional rules have the form $t_1 = s_1 \wedge \cdots \wedge t_n = s_n \Rightarrow t \to s$, where s_i and t_i may contain variables in t and s. According to such a rule, $t\theta$ can be rewritten to $s\theta$ if all the equations $s_i\theta = t_i\theta$ are satisfied. CTRSs were classified depending on how satisfaction is defined ('=' can be interpreted as $\twoheadrightarrow$, $\leftrightarrow^*$, etc.) As Bergstra and Klop remark this can be generalized by allowing for arbitrary predicates on the variables as conditions (cf. also [14, 67]).

Clearly, all these CTRSs are context-free CCERSs since they allow conditions on the arguments but not on the context of rewrite rules. For this reason results for them are sometimes a special case of general results holding for all CCERSs. In particular, *stable* CTRSs for which the unconditional version is orthogonal as defined in [9] are orthogonal in our sense (to be defined in Subsection 3) and so are confluent.

Encoding of strategies

In the literature a strategy for a rewriting system (R, Σ) is often defined as a map $F{:}\,Ter(\Sigma) \to Ter(\Sigma)$, such that $t \to F(t)$ if t is not a normal form, and $t = F(t)$ otherwise (e.g., [5]). Such strategies are deterministic and do not specify the way in which to obtain $F(t)$ from t.

The first thing to take into account here is that in a term there may be disjoint redex occurrences yielding the same result if reduced. For example, take simply the TRS $R = \{f(x) \to a,\ b \to b\}$ and the term $t = g(b, f(b))$. Then t is rewritten to itself when either the first or the second occurrence of b in it is rewritten (using the second rule). The leftmost b is *essential* (i.e., contributes to the normal form) [28], whereas the rightmost b is not. Here our knowing that a strategy F rewrites t to t is not enough to tell us whether F rewrites an essential redex in t or an inessential one. Similarly, $I(Ix)$ can be β-reduced in one step to Ix, where $I = \lambda x.x$, but the information $I(Ix) \to Ix$ is not enough to determine whether the outermost redex has been contracted or the innermost one (the effect that contraction of different redexes yields the same result is called a 'syntactic accident' [43]). So a strategy should specify which redex occurrence must be contracted.

The second thing to take into account is that a redex occurrence can be an instance of more than one rule. That is, $LHS(r_1)\theta_1 = u = LHS(r_2)\theta_2$ for some rules r_1 and r_2 and some assignments $\theta_1 \in AA(r_1)$ and $\theta_2 \in AA(r_2)$. And the contracta of the different redexes can be the same, which shows that even knowing the occurrence of the redex may not be sufficient for knowing which rule has been applied. For example, consider the rules for parallel *or*:

$$or(true, x) \to true, or(x, true) \to true.$$

Then $or(true, true) \to true$ by applying either of the two rules. So a strategy should specify which rule must be applied.

Finally, although for orthogonal ERSs the result of a reduction step from some term t is uniquely determined by the redex occurrence and the rule to be applied, this need not be the case in general. For example, applying the (variable-introducing, hence non-orthogonal) rule $a \to A$ to the term a in the empty context may lead to any result, depending on the assignment to A.

Thus we prefer to view a strategy as a set F of triples $(r, \theta, C[\,])$ specifying that rule $r : t \to s \in R$ can be used with assignment θ in context $C[\,]$ to rewrite $C[t\theta]$ to $C[s\theta]$.[8] Thus a strategy F may be non-deterministic in that the redex to be contracted in a term t can be selected from a possibly non-singleton set of redexes of t specified by F. To a strategy F one can associate a CCERS R_F encoding exactly the same information by taking $\theta, C[\,]$ admissible for r iff $(r, \theta, C[\,]) \in F$. Obviously, this also holds the other way around; that is, every CCERS can be viewed as a strategy for its unconditional version.

Note that the set of terms $und(F)$ on which a strategy F (considered as a set of triples) is undefined need not coincide with the set of normal forms. Indeed, many strategies halt once they reach terms to a set of *values* (e.g., head normal forms or weak head normal forms in the λ-calculus), or if a *deadlock* situation arises; see [42] for a number of such strategies. So our definition provides for such strategies, except the information about which terms from $und(F)$ are values (and which correspond to a deadlock situation) must be added explicitly.

3. ORTHOGONAL CCERSS

In this section, we introduce a suitable concept of orthogonality for CCERSs, prove confluence for them, and illustrate how this result can be used for proving confluence for restricted λ-calculi. We then recall some results concerning the *similarity* of redexes [31] in orthogonal CCERSs. Finally, we present a new proof of the existence of external redexes [22] in every reducible term in an orthogonal fully-extended CCERS. The results concerning the similarity of redexes and external redexes will be used later on to study the perpetuality of redexes in orthogonal fully-extended CCERSs.

[8] Note that an ordinary strategy F can be directly encoded by associating the set $\{(r : t \to s, \theta, C[\,]) \mid r \in R, C[s\theta] = F(C[t\theta])\}$ to it.

3.1. Orthogonality and confluence

The idea of orthogonality is that contraction of a redex does not destroy other redexes (in whatever way) but instead leaves a number of their residuals. A prerequisite for the definition of residual is the concept of *descendant*, also called *trace*, which allows the tracing of subterms along a reduction. Whereas this concept is pretty simple in the first-order case, CCERSs may exhibit complex behaviour due to the possibility of nested metasubstitutions in the right-hand sides of rules, thereby complicating the definition of descendants. A standard technique in higher-order rewriting [37] (illustrated below on examples) is to *decompose* or *refine* each rewrite step into two parts: a *TRS*-part in which the left-hand side is replaced by the right-hand side without evaluating the (meta)substitutions, and a *substitution*-part in which the delayed substitutions are evaluated. To express substitution we use the S-reduction rules

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

where S^{n+1} is the *operator sign of substitution* with arity $(n, n+1)$ and scope indicator $(n+1)$ and where $x_1, \dots, x_n$ and $A_1, \dots, A_n, A_0$ are pairwise distinct variables and metavariables. (We assume that the CCERS does not contain symbols S^{n+1}; it can of course contain a renamed variant of S-rules. The collection of all substitution rules, renamed or not, is an ERS itself.) Thus S^{n+1} binds only in the last argument. One can think of S-redexes as (simultaneous) let-expressions.

Thus the descendant relation of a rewrite step can be obtained by composing the descendant relation of the TRS-step and the descendant relations of the S-reduction steps. All known concepts of descendants agree in the cases when the subterm $s \subseteq t$ which is to be traced during a step $t \overset{u}{\to} o$ is (1) in an argument of the contracted redex u, (2) properly contains u, or (3) does not overlap with u. The concepts differ when s is a pattern-subterm (i.e., when s is in the contracted redex u but is not in any of its arguments), in which case we define the contractum of u to be the descendant of s. According to many definitions, however, s does not have a u-descendant (*descendant* is often used as a synonym of *residual*, which it is not). In the case of TRSs, our definition coincides with Boudol's [12] and differs slightly from Klop's [38]: according to Klop's definition the descendants of a contracted redex, as well as of any of its pattern-subterms, are all subterms whose head-symbols are within the pattern of the contractum.

We first explain our descendant concept by using examples. Consider a TRS-step $t = f(g(a)) \to h(b) = s$ performed according to the rule $f(g(x)) \to h(b)$. The descendant of both pattern-subterms $f(g(a))$ and $g(a)$ of t in s is $h(b)$[9] and a does not have a descendant in s. The refinement of a β-step $t = Ap(\lambda x(Ap(x,x)), z) \to_\beta Ap(z,z) = e$ would be $t = Ap(\lambda x(Ap(x,x)), z) \to_{\beta_f} o = S^2xzAp(x,x) \to_S Ap(z,z) = e$: the descendant of both t and $\lambda x(Ap(x,x))$ after the *TRS*-step is the contractum $S^2xzAp(x,x)$, the descendants of $Ap(x,x), z \subseteq t$ are the respective subterms $Ap(x,x), z \subseteq o$, the descendant of both $o = S^2xzAp(x,x)$ and $Ap(x,x)$ after the substitution step is the contractum e, and the descendants of $z \subseteq o$, as well as of the bound occurrence of x in $Ap(x,x)$, are the occurrences of z in e.

[9]According to Klop's definition, the occurrence of b in $h(b)$ is also a descendant for both $f(g(a))$ and $g(a)$.

This definition by example can be formalized using *paths* to refer to subterm positions in a term t: Paths, denoted by ϕ, ψ, ζ, ξ, are strings of integers: the empty string ε refers to the top-position (i.e., the term t itself) and if a path $i_1, \ldots, i_k$ refers to a subterm $\sigma x_1 \ldots x_m(t_1, \ldots, t_n)$ of t, then $i_1, \ldots, i_k, i_{k+1}$ is again a path for each $1 \leq i_{k+1} \leq n$ which refers to the subterm $t_{i_{k+1}}$ of t; $\preceq$ denotes the prefix ordering on paths. (The binding variables in a quantifier are considered to be at the same position as the quantifier symbol itself. They therefore can be ignored because they are not subterms.)

DEFINITION 3.1. Let t be a term in a simple CCERS R (so the refinement of an R-step coincides with the R-step itself), let $r : t' \to s' \in R$, let u be an (admissible) r-redex in t occurring at a position ϕ, let $t \xrightarrow{u} s$, and let o be a subterm of t at a position ψ.

1. If ϕ and ψ are disjoint (i.e, neither $\phi \preceq \psi$ nor $\psi \preceq \phi$), then the descendant of o is the subterm of s at the same position ψ;

2. If $\psi \preceq \phi$, then again the descendant of o is the subterm of s at the same position ψ;

3. If $\psi = \phi \cdot \zeta$ where ζ is a nonvariable position in the left-hand-side t' of r ($\cdot$ is the concatenation operation on paths), then the descendant of o is the subterm of s at the position ϕ (i.e., is the contractum of u);

4. If $\psi = \phi \cdot \zeta_i \cdot \xi$ where ζ_i is the position of the ith-from-the-left variable occurrence in t', then the descendants (0 or more) of o are the subterms in s at all positions $\psi_j = \phi \cdot \zeta^i_j \cdot \xi$, $1 \leq j \leq k_i$, where $\zeta^i_1, \ldots, \zeta^i_{k_i}$ are the positions of all occurrences of the same variable in the right-hand-side s' of r.

DEFINITION 3.2. Let $S^{n+1}x_1 \ldots x_n\, t_1 \ldots t_n t_0$ be an S-redex in a term t at a position ϕ in a CCERS, let $t \xrightarrow{u}_S s$, and let o be a subterm of t at a position ψ.

1. If ϕ and ψ are disjoint, then the descendant of o is the subterm of s at the same position ψ;

2. If $\psi \preceq \phi$, then again the descendant of o is the subterm of s at the same position ψ;

3. If $\psi = \phi \cdot n + 1 \cdot \xi$ (i.e., $o \subseteq t_0$), then the descendant of o is the subterm in s at position $\phi \cdot \xi$.

4. If $\psi = \phi \cdot i \cdot \xi$ where $1 \leq i \leq n$, then the descendants (0 or more) of o are the subterms in s at all positions $\psi_j = \phi \cdot \zeta^i_j \cdot \xi$, $1 \leq j \leq k_i$, where $\zeta^i_1, \ldots, \zeta^i_{k_i}$ are the positions of all occurrences of x_i in t_0.

To illustrate further the third and the fourth cases of Definition 3.2, consider the S-reduction step $t = Sxf(a)g(x) \to_S g(f(a)) = s$. Then the descendant of $x \subseteq t$ is $f(a) \subseteq s$, and the descendant of $g(x) \subseteq t$ is s. The descendants of $f(a), a \subseteq t$ are the occurrences $f(a), a \subseteq s$, respectively.

The *descendant* concept extends by transitivity to arbitrary reductions consisting of TRS-steps and S-reduction steps. 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 inverse of the descendant relation. The descendant concept allows us to define residuals:

DEFINITION 3.3. Let $t \xrightarrow{u} s$ be in a CCERS R, let $v \subseteq t$ be an admissible redex, and let $w \in s$ be a u-descendant of v. We call w a *u-residual* of v if (a) the patterns of u and v do not overlap (i.e., the pattern-occurrences do not share an occurrence of a symbol in t), (b) w is a redex weakly similar to v (see Definition 2.2), and (c) w is admissible. (So u itself does not have *u-residuals* in s.) The concept of a *residual* of redexes extends naturally to arbitrary reductions. A redex in s is called a *new* redex or a *created* redex if it is not a residual of a redex in t. The *predecessor* relation is inverse to that of residual.

DEFINITION 3.4. We call a CCERS *orthogonal* if:

- the left-hand sides of rules are not single metavariables,
- the left-hand side of any rule is a simple metaterm and its metavariables contain those of the right-hand side, and
- all the descendants of an admissible redex u in a term t under the contraction of any other admissible redex $v \subseteq t$ are residuals of u.

The second condition ensures that rules exhibit deterministic behaviour when they can be applied. The last condition is the counterpart of the *subject reduction property* in typed λ-calculi [6]. For example, consider the rules $a \to b$ and $f(A) \to A$ with the admissible assignment $A\theta = a$. The descendant $f(b)$ of the redex $f(a)$ after contraction of a is not a redex because the assignment $A\theta = b$ is not admissible. Hence the system is not orthogonal.

DEFINITION 3.5. Reductions starting from the same term are called *co-initial*. Recall that co-initial reductions $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$ are *weakly equivalent* or *Hindley-equivalent* [5], written $P \approx_H Q$, if $s = e$ and the residuals of any redex of t under P and under Q are the same redexes in s. Furthermore, P and Q are *strictly equivalent* [26], written $P \approx_{st} Q$, if $s = e$ and the descendants of any subterm of t under P and under Q are the same subterms in s.

Using these equivalences and the above definition of residuals, we can easily infer *strong* [43, 22] and *strict* [26] forms of the Church-Rosser property for CCERSs.

A standard method of proving the *strong* version of CR is one using FD and the fact that any pair of redexes u, v in a term *strongly commute*: $u + v/u \approx_H v + u/v$ [43]; that latter property will be called *strong local confluence*.[10] Indeed, as in orthogonal TRSs [22], the λ-calculus [44, 5], orthogonal CRSs [37], and orthogonal

[10] FD is often referred to the stronger property that all developments of a set of redexes in a term are terminating and all complete developments of the same set of redexes are Hindley-equivalent. This stronger version follows easily from the weaker version (i.e., termination of all developments) and the strong commutativity of co-initial steps.

HRSs [54], one can in orthogonal CCERSs use FD and strong commutativity to define for any co-initial reductions P and Q the *residual of P under Q*, written P/Q. We write $P \trianglelefteq_L Q$ if $P/Q = \emptyset$ ($\trianglelefteq_L$ is the *Lévy-embedding* relation); P and Q are called *Lévy-equivalent* or *permutation-equivalent* (written $P \approx_L Q$) if $P \trianglelefteq_L Q$ and $Q \trianglelefteq_L P$. It follows from the definition of / that if $P+P'$ and $Q+Q'$ are co-initial finite reductions in an orthogonal CCERS, then $(P+P')/Q \approx_L P/Q + P'/(Q/P)$ and $P/(Q+Q') \approx_L (P/Q)/Q'$. This is all well known and we do not give more details. The strong Church-Rosser theorem then states that, for any co-initial finite reductions P and Q in an orthogonal ERS, $P \sqcup Q \approx_L Q \sqcup P$, where $P \sqcup Q$ means $P + Q/P$. The *Strict Church-Rosser* theorem states that, for any co-initial finite reductions P and Q in an orthogonal ERS, $P \sqcup Q \approx_{st} Q \sqcup P$. (Thus, $P \approx_L Q$ implies $P \approx_{st} Q$.) Like the strong CR property, the strict CR property follows from FD and the following *strict local confluence* property: any two co-initial steps u, v *strictly commute*: $u \sqcup v \approx_{st} v \sqcup u$.

Since developments in CCERSs are obtained by restricting developments in ERSs, and the latter are a special case of developments in PRSs [61] which are finite [56], we obtain the following result.

THEOREM 3.1. **(Finite Developments)** *All developments of a term t in an orthogonal CCERS R eventually terminate.*

Using this theorem and the last condition in the definition of orthogonality, the next theorem follows from some abstract theory of residuals.

THEOREM 3.2. *Let P and Q be any co-initial finite reductions in an orthogonal CCERS R. Then*

*(1)***(Strong Church-Rosser)** $P \sqcup Q \approx_L Q \sqcup P$.
*(2)***(Strict Church-Rosser)** $P \sqcup Q \approx_{st} sQ \sqcup P$.

The λ-calculus [5] is the prime example of an ERS. If one restricts term formation in it, one arrives at a large class of typed lambda calculi. Since the rewrite relation in these calculi is not restricted in any way and typed terms are closed under β-reduction,[11] these CCERSs are orthogonal, hence confluent. In Appendix A.2 we demonstrate how the above confluence result can be used to prove confluence for the call-by-need λ-calculus of Ariola et al. [4].

An emerging class of context-sensitive conditional ERSs is the class of λ-calculi with restricted expansion rules like $\bar{\eta}$ (see e.g. [3]). These calculi are not orthogonal, but their confluence can be shown by modifying the confluence diagrams arising from FD for the corresponding unconditional expansion rules.

3.2. Similarity of redexes

The idea of *similarity* of redexes [29, 31] u and v is that u and v are weakly similar – that is, they match the same rewrite rule – and quantifiers in the pattern

[11] Proving this *subject reduction* property is sometimes nontrivial.

of u and v bind 'similarly' in the corresponding arguments. For example, recall that a β-redex $Ap(\lambda xt, s)$ is an *I-redex* if $x \in FV(t)$ and is a *K-redex* otherwise. Then all I-redexes are similar and all K-redexes are similar, but no I-redexes are similar to a K-redex. Consequently, for any pair of corresponding arguments of u and v, either both are erased after contraction of u and v or none is.

A redex in a CCERS has the form $u = C[t_1, \ldots, t_n]$, where C is the pattern and $t_1, \ldots, t_n$ are the arguments. Sometimes we will write u as $u = C[\overline{x_1}t_1, \ldots, \overline{x_n}t_n]$, where $\overline{x_i} = \{x_{i_1}, \ldots, x_{i_{n_i}}\}$ is the subset of binding variables of C such that t_i is in the scope of an occurrence of each x_{i_j}, $j = 1, \ldots, n_i$. Let us call the maximal subsequence $j_1, \ldots, j_k$ of $1, \ldots, n$ such that $t_{j_1}, \ldots, t_{j_k}$ have u-descendants the *main sequence* of u (or the *u-main sequence*), call $t_{j_1}, \ldots, t_{j_k}$ the (u-)*main arguments*, and call the remaining arguments (u)-*erased*. Further, call u *erasing* if $k < n$ and *non-erasing* otherwise.

Now the similarity of redexes can be defined as follows: weakly similar redexes $u = C[\overline{x_1}t_1, \ldots, \overline{x_n}t_n]$ and $v = C[\overline{x_1}s_1, \ldots, \overline{x_n}s_n]$ are *similar* if, for any $1 \le i \le n$, $\overline{x_i} \cap FV(t_i) = \overline{x_i} \cap FV(s_i)$. For example, consider the rule $\sigma x(A, B) \to (\sigma x(f(A), A)/x)B$. Then the redexes $u = \sigma x(x, y)$ and $v = \sigma x(f(x), y)$ are similar, while $w = \sigma x(y, y)$ is not similar to any of them since $x \notin FV(y)$. However, note that the second arguments of all the redexes u, v and w are main and the first arguments are erased. In this paper it is more convenient to use a slightly relaxed concept of similarity, written $\sim$, such that $u \sim v \sim w$:

DEFINITION 3.6. We write $u \sim v$ if the main sequences of u and v coincide and for any main argument t_i of u, $\overline{x_i} \cap FV(t_i) = \overline{x_i} \cap FV(s_i)$.

The following lemma implies in particular that, indeed, if u and v are similar, then $u \sim v$, and that $\sim$ is an equivalence relation. Because its proof involves properties of essentiality not needed elsewhere in this paper, we omit the proof and instead refer to previous work [31]. The lemma is quite intuitive anyway: it shows that only pattern-bindings (i.e., bindings from inside the pattern) of free variables in *main* arguments of a redex are relevant for the erasure of its arguments.

Below, θ will not only denote assignments but will also denote substitutions assigning terms to variables; when we write $o' = o\theta$ for a substitution θ, we assume that no free variables of the substituted subterms become bound in o' (i.e., we rename bound variables in o when necessary).

LEMMA 3.1. *Let $u = C[\overline{x_1}t_1, \ldots, \overline{x_n}t_n]$ and $v = C[\overline{x_1}s_1, \ldots, \overline{x_n}s_n]$ be weakly similar redexes, and let for any main argument s_i of v, $\overline{x_i} \cap FV(t_i) \subseteq \overline{x_i} \cap FV(s_i)$. Then the main sequence of u is a subset of the main sequence of v.*

COROLLARY 3.1. *Let $u = C[\overline{x_1}t_1, \ldots, \overline{x_n}t_n]$ and $v = C[\overline{x_1}s_1, \ldots, \overline{x_n}s_n]$ be weakly similar redexes, and let for any main argument s_i of v, $\overline{x_i} \cap FV(t_i) = \overline{x_i} \cap FV(s_i)$. Then $u \sim v$. In particular, if $u = v\theta$, then $u \sim v$.*

3.3. External redexes

In this subsection we will show that every reducible term in an orthogonal *fully-extended* (see Definition 3.7) CCERS has an *external* redex. External redexes for orthogonal TRSs were introduced by Huet and Lévy [22], who also proved the existence of external redexes in every reducible term. Both the original definition of external redexes and the existence proof are quite lengthy.

With our concept of descendant, external redexes can be defined as redexes whose descendants can never occur inside the arguments of other redexes. Any external redex is trivially outermost, but an outermost redex is not necessarily external. Contracting a redex *disjoint from it*, may cause its residual to be non-outermost. For example, consider the orthogonal TRS $\{f(x,b) \to c, a \to b\}$. The first a in $f(a,a)$ is outermost but not external; contracting the second a (which is disjoint from it) creates the redex $f(a,b)$ having the residual of the first a as argument. The second a is external.

In an ERS, there may be another reason why an outermost redex need not be external. Contracting a redex *in one of its argument, may cause its residual to be non-outermost. This already shows up in the $\lambda\beta\eta$-calculus. Let $I = \lambda x.x$ and $K = \lambda xy.x$, as usual [5]. The redex $u = I(KIx)$ in $\lambda x.I(KIx)x$ is outermost but not external; contracting the redex KIx in its argument creates the η-redex $\lambda x.IIx$ having the residual II of u as argument. This example can be readily encoded as an orthogonal ERS. We will see later that because of rules like η which test for the absence of variables in subterms (occur check!) even the conservation theorem fails for orthogonal CCERSs in general. To rule out such rules, following [55, 19], we introduce the concept of* full extendedness *for CCERSs:*

Definition 3.7. We call a CCERS R *fully-extended* iff for any step $t \xrightarrow{u} s$ in R and any occurrence $w \subseteq t$ of an instance of the left-hand-side (of a rule $r \in R$) such that:

(a) the patterns of w and u in t do not overlap, and
(b) w has a u-descendant $w' \in s$ that is a redex,

w is an admissible redex in t weakly similar to w'.

Now we can easily generalize the proof of existence of external redexes in [28] from orthogonal TRSs to fully-extended orthogonal CCERSs.

Definition 3.8. Let $P : t \twoheadrightarrow o$ in an orthogonal fully-extended CCERS. A subterm $s \subseteq t$ is *P-external* if no descendants of s along P appear inside redex-arguments and is *P-internal* otherwise. A subterm $s \subseteq t$ is *external* if s is *P-external* for any finite reduction $Q : t \twoheadrightarrow$; otherwise s is *internal*.[12]

Consider the λ-term $t = \Omega((\lambda xy.xy)I(Ix))$, where I is as defined above and $\Omega = (\lambda x.xx)(\lambda x.xx)$, and consider the β-reduction $P : t \xrightarrow{v} \Omega((\lambda y.Iy)(Ix)) \xrightarrow{w} \Omega(I(Ix))$ contracting the redexes $v = (\lambda xy.xy)I$ and $w = (\lambda y.Iy)(Ix)$. Then the redexes $\Omega, v \subseteq t$ are P-external, whereas the redex $Ix \subseteq t$ is P-internal (since after

[12] In [28], an external (resp. P-external) redex is called *unabsorbed* (P-*unabsorbed*).

the step v the residual of $Ix \subseteq t$ is inside an argument the created redex w). Note that for the outermost redexes $\Omega, I(Ix) \subseteq s$, there are P-external redexes $\Omega, v \subseteq t$ such that the unique P-descendant of $\Omega \subseteq t$ overlaps the pattern of $\Omega \subseteq s$ and the unique P-descendant of v overlaps the pattern of $I(Ix) \subseteq s$. Note also that $Ix \subseteq t$ may be $Q \sqcup P$-internal even if it is Q-external. For instance, consider a reduction Q which contracts the occurrences of Ω a finite number of times. These intuitions are formalized in the following three lemmas and are then used to prove the existence of external redexes in reducible terms.

LEMMA 3.2. *Let $P : t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} \ldots \xrightarrow{u_{n-1}} t_n$ in an orthogonal fully-extended CCERS. Then for any outermost redex $v \subseteq t_n$ there is a P-external redex $u \subseteq t_0$ whose unique P-descendant $s \subseteq t_n$ overlaps the pattern of v (i.e., either $v \subseteq s$ or $s = e$ for some proper pattern-subterm e of v.)*

Proof. By induction on $|P|$. If $|P| = 0$ the result is obvious. Suppose $|P| > 0$ and let $P = P' + u_{n-1}$.

(a) Assume first that v is a residual of a redex $v' \subseteq t_{n-1}$. Let $v^* = v'$ if $v' \not\subseteq u_{n-1}$ and let $v^* = u_{n-1}$ otherwise. By full extendedness, since v is outermost, v^* is outermost. By the induction hypothesis there is a P'-external redex $u \subseteq t_0$ whose unique P'-descendant $s' \subseteq t_{n-1}$ satisfies either $v^* \subseteq s'$ or $s' = e'$ for some proper pattern-subterm e' of v^*. Since u is P'-external, s' has a unique descendant s in t_n. If $v^* \subseteq s'$ it is easy to see $v \subseteq s$. Otherwise $e' = s'$ and we consider two cases:

1. $v^* = v'$. Since the patterns of the redexes v' and u_{n-1} do not overlap (by orthogonality), s is a pattern-subterm of v.
2. $v^* = u_{n-1}$. Since the descendant of each pattern-subterm of u_{n-1} is the contractum of u_{n-1}, $v \subseteq s$.

Therefore u is P-external.

(b) Assume now that u_{n-1} creates v. By full extendedness, the contractum of u_{n-1} overlaps the pattern of v. Since v is outermost, u_{n-1} is outermost. By the induction hypothesis there is a P'-external redex $u \subseteq t_0$ such that its unique descendant $s' \subseteq t_{n-1}$ satisfies either $u_{n-1} \subseteq s'$ or $e' = s'$ for some proper pattern-subterm e' of u_{n-1}. Since u is P'-external, s' has a unique descendant s in t_n. Since the descendant of each pattern-subterm of u_{n-1} is the contractum of u_{n-1}, s contains the contractum of u_{n-1}. Thus s overlaps the pattern of v. Therefore u is P-external. ∎

LEMMA 3.3. *Let $P : t \twoheadrightarrow s$ be in an orthogonal fully-extended CCERS. If t is reducible, there is a P-external redex u in t.*

Proof. If $|P| = 0$ or $|P| > 0$ and s is not a normal form, then the lemma follows immediately from Lemma 3.2. Otherwise, let $P : t \overset{P'}{\twoheadrightarrow} s' \xrightarrow{v} s$. Since s is a normal form, v is outermost. By Lemma 3.2 there is a P'-external redex $u \subseteq t$ whose unique descendant in s' overlaps the pattern of v. Since s has no redexes, u is P-external. ∎

LEMMA 3.4. *Let $P : t \twoheadrightarrow s$ and $Q : t \twoheadrightarrow e$ be in an orthogonal fully-extended CCERS. If u is P-internal, then it is $Q \sqcup P$-internal.*

Proof. By induction on $|Q|$. It is enough to consider the case when $|Q| = 1$; the rest follows from the induction hypothesis. So let $Q = w$ for a redex w in t. Furthermore, let $P = P^* + v^*$. Without loss of generality we can assume that u is P^*-external, so v^* creates a redex v that contains the unique P-descendant o of u in its argument.

(a) Assume first that o does not have a w/P-descendant. By Theorem 3.2 u does not have $w \sqcup P$-descendants. Hence u is $w \sqcup P$-internal (otherwise its descendants cannot be erased).

(b) Assume now that o has a w/P-descendant o'. Since w/P contracts only residuals of w and v is a new redex, v has a residual v' that contains o' in its argument. By Theorem 3.2 o' is also a $w \sqcup P$-descendant of u. Hence u is $w \sqcup P$-internal. ∎

THEOREM 3.3. *Every reducible term in an orthogonal fully-extended CCERS has an external redex.*

Proof. Assume that for any outermost redex $u_i \subseteq t$ there is a finite reduction P_i such that u_i is P_i-internal ($i = 1, \ldots, k$). Then by Lemma 3.4 all redexes u_i are P-internal for $P = P_1 \sqcup \ldots \sqcup P_k$. But this is impossible by Lemma 3.3. ∎

4. A MINIMAL PERPETUAL STRATEGY

In this section we introduce a perpetual strategy F_m^∞ for orthogonal fully-extended CCERSs by generalizing the *constricting* perpetual strategies in the literature [59, 63, 16, 47, 62]. We also study properties of F_m^∞ that are used in the next section to obtain new criteria for the perpetuality of redexes and of redex occurrences in orthogonal fully-extended CCERSs. A recent survey on perpetual reductions in the λ-calculus and its extensions can be found in [66, 62].

For convenience we have collected the definitions of all related perpetual strategies in Appendix A.3. To unify the notation we follow [66, 62] and use F_1 and F_3 to denote the perpetual strategies of Bergstra and Klop [8] and Sørensen [63], respectively. And we use F_z to denote the zoom-in strategy of Melliès [47].

Let us first fix the terminology. Recall that a term t is called *weakly normalizing* (WN), written $WN(t)$, if it is reducible to a *normal form* (i.e., a term without a redex), and t is called *strongly normalizing* (SN), written $SN(t)$, if it does not possess an infinite reduction. We call t an ∞-*term* (written ∞t), if $\neg SN(t)$. Clearly, for any term t, $SN(t) \Rightarrow WN(t)$. If the converse is also true, then we call t *uniformly normalizing* (UN). So a UN term t either does not have a normal form or all reductions from t eventually terminate. Correspondingly, a rewrite system R is called WN, SN, or UN if all terms in R are WN, SN, or UN, respectively.

Following [8, 38], we call a rewrite step $t \xrightarrow{u} s$, as well as the redex-occurrence $u \subseteq t$, *perpetual* if $\infty t \Rightarrow \infty s$. Otherwise we call them *critical.* We call a redex (not an occurrence) *perpetual* iff its occurrence in every (admissible) context is perpetual. A *perpetual strategy* in an orthogonal fully-extended CCERS is a (partial) function on terms which in any reducible term selects a perpetual redex-occurrence; the orthogonality of the CCERS implies that the redex-occurrence uniquely determines the rewrite rule (and the corresponding admissible assignment) according to which the redex is to be contracted.

DEFINITION 4.1. Let $P : t \twoheadrightarrow$ and $s \subseteq t$. Reduction P is *internal* to s if it contracts redexes only in (the descendant of) s. (The contracted redexes in P need not be *proper* subterms of s.)

DEFINITION 4.2. (1) Let t be an ∞-term in an orthogonal fully-extended CCERS and let $s \subseteq t$ be a smallest subterm of t such that $\infty(s)$ (i.e., such that every proper subterm $e \subset s$ is SN). Then we call s a *minimal perpetual subterm* of t, and call any external redex of s (such a redex exists by Theorem 3.3) a *minimal perpetual redex* of t.

(2) Let F_m^∞ be a one-step strategy that contracts a minimal perpetual redex in t if ∞t and otherwise contracts any redex. Then we call F_m^∞ a *minimal perpetual strategy*. We call F_m^∞ *constricting* if for any F_m^∞-reduction $P : t_0 \stackrel{u_0}{\to} t_1 \stackrel{u_1}{\to} \dots$ (i.e., any reduction constructed using F_m^∞) starting from an ∞-term t_0 and for any i, $P_i^* : t_i \stackrel{u_i}{\to} t_{i+1} \stackrel{u_{i+1}}{\to} \dots$ is internal to s_i, where $s_i \subseteq t_i$ is the minimal perpetual subterm containing u_i. Constricting minimal perpetual strategies will be denoted F_{cm}^∞.

Recall that, according to Gramlich [16, Remark 3.3.7], a reduction in a TRS is called constricting if it has the form

$$C_0[s_0] \stackrel{u_0}{\to} C_0[C_1[s_1]] \stackrel{u_1}{\to} C_0[C_1[C_2[s_2]]] \stackrel{u_2}{\to} \dots$$

where s_i are minimal perpetual subterms and $u_i \subseteq s_i$. Hence any F_{cm}^∞-reduction is constricting (according to Gramlich). Plaisted [59] constructs a constricting perpetual strategy (for TRSs) that in each step contracts a perpetual redex of the leftmost (innermost) minimal perpetual subterm.[13] Sørensen's β-reduction strategy F_3 [63, 62], as well as Melliès' *zoom-in* β-strategy F_z, produce constricting reductions (on ∞-terms) and are special cases of F_m^∞. Specifically, F_z is obtained from F_{cm}^∞ if in each step the leftmost redex of a minimal perpetual subterm is contracted (the leftmost redex in any λ-term is external); and F_3 is a special case of F_z. The perpetual strategy F_2 [62] is not zoom-in but is constricting. Note that F_m^∞ is not in general a computable strategy, since SN is already undecidable in orthogonal TRSs [38]; the strategies F_1, F_2, F_3, and F_z are not computable either. These four strategies all produce standard reductions.

LEMMA 4.1. *Let t be an ∞-term in an orthogonal fully-extended CCERS, let $s \subseteq t$ be a minimal perpetual subterm of t, and let $P : t \twoheadrightarrow \infty$ be internal to s. Then exactly one residual of any external redex u of s is contracted in P.*

[13] As noted by Gramlich [16], Plaisted's original definition of 'constricting' is not correct because any infinite reduction becomes constricting.

Proof. Let $t = C[s]$ and $s = C'[s_1, \ldots, u, \ldots, s_n]$, where C' consists of the symbols on the path from the top of s to u (the context C' can be empty, in which case $s = u$). If, on the contrary, P does not contract a residual of u, then every step of P takes place either in one of the s_i or in the arguments of u (since u is external in s). Hence at least one of these subterms has an infinite reduction – a contradiction, since s is a minimal perpetual subterm. Since u is external, P cannot duplicate its residuals; hence P contracts exactly one residual of u. ■

The following theorem justifies the terminology 'minimal perpetual redex'.

THEOREM 4.1. *F_m^∞ is a perpetual strategy in any orthogonal fully-extended CCERS.*

Proof. Suppose $\infty(t_0)$, let s_0 be a minimal perpetual subterm of t_0, and let $u \subseteq s_0$ be a minimal perpetual redex. Let $P : t_0 \stackrel{u_0}{\to} t_1 \stackrel{u_1}{\to} t_2 \twoheadrightarrow \infty$ be internal to s_0. By Lemma 4.1 exactly one residual of u, say u_i, is contracted in P. Let $P_{i+1} : t_0 \stackrel{u_0}{\to} t_1 \stackrel{u_1}{\to} \ldots \stackrel{u_i}{\to} t_{i+1}$ and $P^*_{i+1} : t_{i+1} \stackrel{u_{i+1}}{\to} t_{i+2} \twoheadrightarrow \infty$ (i.e., $P : t_0 \stackrel{P_i}{\twoheadrightarrow} t_i \stackrel{u_i}{\to} t_{i+1} \stackrel{P^*_{i+1}}{\twoheadrightarrow}$). Since P_i and u are co-initial, $u + P_i/u \approx_L P_i + u/P_i = P_i + u_i = P_{i+1}$ by Theorem 3.2, hence $P = P_{i+1} + P^*_{i+1} \approx_L u + P_i/u + P^*_{i+1}$. That is, u is a perpetual redex-occurrence. Hence F_m^∞ is perpetual. ■

DEFINITION 4.3. F_m^∞ is the *leftmost* minimal perpetual strategy, denoted F_{lm}^∞, if in each term it contracts the leftmost minimal perpetual redex. (See Definition A.5 for the definition of F_{lm}^∞ for the case of the λ-calculus.)

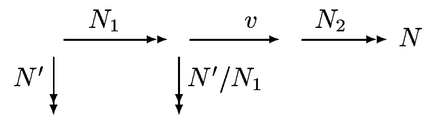
THEOREM 4.2. *F_{lm}^∞ is a constricting strategy in any orthogonal fully-extended CCERS.*

Proof. Let $P : t_0 \stackrel{u_0}{\to} t_1 \stackrel{u_1}{\to} t_2 \twoheadrightarrow \infty$ be a leftmost minimal perpetual reduction, and let $s_i \subseteq t_i$ be the leftmost minimal perpetual subterm of t_i. Since by Theorem 4.1 u_i is perpetual for the term s_i, the descendant of s_i is an ∞-term and thus contains s_{i+1}, and it is immediate that P is constricting. ■

Although we do not use it in the following, it is interesting to note that the constricting perpetual reductions are minimal w.r.t. Lévy's embedding relation $\trianglelefteq_L$. Hence the term *minimal*.

The relations $\trianglelefteq_L, \approx_L$, and $/$ (defined in Section 3) are extended to possibly infinite co-initial reductions N, N' as follows. $N \trianglelefteq_L N'$, or equivalently, $N/N' = \emptyset$ if for any redex v contracted in N, say $N = N_1 + v + N_2$, $v/(N'/N_1) = \emptyset$ (see the diagram below); and $N \approx_L N'$ iff $N \trianglelefteq_L N'$ and $N' \trianglelefteq_L N$. Here, for any infinite P, $u/P = \emptyset$ if $u/P' = \emptyset$ for some finite initial part P' of P. And P/Q is defined only for finite Q as the reduction whose initial parts are residuals of initial parts of P

under Q.



THEOREM 4.3. *Let $P : t_0 \xrightarrow{u_0} t_1 \xrightarrow{u_1} t_2 \twoheadrightarrow \infty$ be a constricting minimal perpetual reduction in an orthogonal fully-extended CCERS and let $Q : t_0 \twoheadrightarrow \infty$ be any infinite reduction such that $Q \trianglelefteq_L P$. Then $Q \approx_L P$.*

Proof. Since P is constricting, there is a minimal perpetual subterm $s_0 \subseteq t_0$ such that P is internal to s_0. Since $Q \trianglelefteq_L P$, Q is internal to s_0 as well. By the construction, u_0 is an external redex in s_0, and by Lemma 4.1 exactly one residual u' of u_0 is contracted in Q. So let $Q : t_0 \overset{Q_j}{\twoheadrightarrow} t'_j \xrightarrow{u'} t'_{j+1} \overset{Q^*_{j+1}}{\twoheadrightarrow} \infty$. Then $Q \approx_L u_0 + Q_j/u_0 + Q^*_{j+1}$, and obviously $u_0 \trianglelefteq_L Q$. Similarly, since P is constricting, for any finite initial part P' of P, $P' \trianglelefteq_L Q$, and therefore $P \trianglelefteq_L Q$. Thus $Q \approx_L P$. ■

5. TWO CHARACTERIZATIONS OF CRITICAL REDEXES

In this section we give an intuitive characterization of critical redex occurrences for orthogonal fully-extended CCERSs, generalizing Klop's characterization of critical redex occurrences for orthogonal TRSs [38], and derive from it a characterization of perpetual redexes similar to Bergstra and Klop's perpetuality criterion for β-redexes [8]. Our proofs are surprisingly simple, yet the results are rather general and useful in applications. We need three simple lemmas first.

LEMMA 5.1. *Let $t \xrightarrow{u} s$ be in a CCERS, let $o \subseteq t$ be either in an argument of u or not overlapping with u, and let $o' \subseteq s$ be a u-descendant of o. Then $o' = o\theta$ for some substitution θ. Moreover, if o is a redex, then so is o' and $o \sim o'$.*

Proof. Since u can be decomposed as a TRS-step followed by a number of substitution steps, it is enough to consider the cases when u is a TRS step and when it is an S-reduction step. If u is a TRS-step, or is an S-reduction step and o is not in its last argument, then o and o' coincide (hence $o \sim o'$ when o is a redex). Otherwise, $o' = o\theta$ for some substitution θ, and if o is a redex, we have again $o \sim o'$ by orthogonality and Corollary 3.1, since free variables of the substituted subterms cannot be bound in $o\theta$ (by the variable convention). ■

LEMMA 5.2. *Let s be a minimal perpetual subterm of t, in an orthogonal fully-extended CCERS, and let $P : t \twoheadrightarrow \infty$ be internal to s. Then P has the form $P = t \twoheadrightarrow o \xrightarrow{u} e \twoheadrightarrow \infty$, where u is the descendant of s in o (i.e., a descendant of s necessarily becomes a redex and is contracted in P).*

Proof. If P did not contract descendants of s, then infinitely many steps of P would be contracted in at least one of the proper subterms of s, and this would contradict the minimality of s. ■

LEMMA 5.3. *In an orthogonal fully-extended CCERS, let $P = u + P'$ be a constricting minimal perpetual reduction starting from t, and let u be in an argument o of a redex $v \subseteq t$. Then P is internal to o.*

Proof. Let $s \subseteq t$ be the minimal perpetual subterm containing u. By definition of minimal perpetual reductions, u is an external redex of s; hence s does not contain v. Since P is constricting, it is internal to s, and orthogonality and Lemma 5.2 tell us that s cannot overlap the pattern of v. The lemma follows. ■

DEFINITION 5.1. (1) Let $P : t_0 \stackrel{u_0}{\to} t_1 \stackrel{u_1}{\to} \dots \stackrel{u_{k-1}}{\to} t_k$, be in an orthogonal CCERS, and let $s_0, s_1, \dots, s_k$ be a chain of descendants of s_0 along P (i.e, s_{i+1} is a u_i-descendant of $s_i \subseteq t_i$). Then, following [8], we call P *passive* w.r.t. $s_0, s_1, \dots, s_k$ if the pattern of u_i does not overlap s_i (s_i may be in an argument of u_i or be disjoint from u_i) for $0 \le i < k$, and we call s_k a *passive* descendant of s_0. By Lemma 5.1, $s_k = s\theta$ for some substitution θ, which we call a *passive substitution*, or *P-substitution* (w.r.t. $s_0, s_1, \dots, s_k$).

(2) Let t be a term in an orthogonal fully-extended CCERS and let $s \subseteq t$. We call s a *potentially infinite* subterm of t if s has a passive descendant s' s.t. $\infty(s')$. (Thus $\infty(s\theta)$ for some passive substitution θ.)

THEOREM 5.1. *Let t be an ∞-term and let $t \stackrel{v}{\to} s$ be a critical step in an orthogonal fully-extended CCERS. Then v erases a potentially infinite argument o (thus $\infty(o\theta)$ for some passive substitution θ).*

Proof. Let $P : t = t_0 \stackrel{u_0}{\to} t_1 \stackrel{u_1}{\to} t_2 \twoheadrightarrow \infty$ be a constricting minimal perpetual reduction, which exists by Theorem 4.1 and Theorem 4.2. Since v is critical, $SN(s)$; hence P/v is finite. Let j be the minimal number such that $u_j/V_j = \emptyset$ and $u_j \notin V_j$, where $V_j = v/P_j$ and $P_j : t \twoheadrightarrow t_j$ is the initial part of P with j steps. (Below, V_j will denote both the corresponding set of residuals of v and its complete development.) By the Finite Developments theorem, no tails of P can contract only residuals of v; and since P/v is finite, such a j exists.

$$
\begin{array}{ccccccccc}
t = t_0 & \twoheadrightarrow & t_l & \twoheadrightarrow & t_j & \stackrel{u_j}{\longrightarrow} & t_{j+1} & \twoheadrightarrow & P \\
v\downarrow & & V_l \downarrow & & V_j \downarrow & & \downarrow & & \\
s = s_0 & \twoheadrightarrow & s_l & \twoheadrightarrow & s_j & \underset{\emptyset}{\twoheadrightarrow} & s_{j+1} & \underset{\emptyset}{\twoheadrightarrow} & P/v
\end{array}
$$

Since $u_j/V_j = \emptyset$ and $u_j \notin V_j$, there is a redex $v' \in V_j$ whose residual is contracted in V_j and erases (the residuals of) u_j. Since V_j consists of (possibly nested) residuals of a single redex $v \subseteq t_0$, the quantifiers in the pattern of v' cannot bind variables inside arguments of other redexes in V_j. Therefore, by Corollary 3.1, v' is similar to its residual contracted in V_j, and hence $u_j/v' = \emptyset$, implying that v' erases its argument o', say the m-th from the left, containing u_j. By Lemma 5.3, the tail $P_j^* : t_j \twoheadrightarrow \infty$ of P is internal to o'.

Let $v_i \subseteq t_i$ be the predecessors of v' along P_j (so $v_0 = v$ and $v_j = v'$; note that a redex can have at most one predecessor), and let o_i be the m-th argument of v_i (thus $o' = o_j$). Note that $u_i \neq v_i$ because v_i has residuals. Let l be the minimal number such that u_l is in an argument of v_l (such an l exists because u_j is in an argument of v_j). Then, by Lemma 5.3, all the remaining steps of P are in the same argument of v_l and it must be the m-th argument o_l of v_l (thus $\infty(o_l)$); but v' erases its m-th argument, implying by Corollary 3.1 that v_l also erases its m-th argument o_l. Furthermore, by the choice of l, no steps of P are contracted inside v_i for $0 \leq i < l$; thus v_l is a passive descendant of v, and o_l is a passive descendant of o_0. Hence, by Lemma 5.1 $v \sim v_l$. Thus v erases a potentially infinite argument o_0 (since $\infty(o_l)$), and we are done. ■

Note in the above theorem that if the orthogonal fully-extended CCERS is an orthogonal TRS, a potentially infinite argument is actually an ∞-term (since passive descendants are all identical), implying Klop's perpetuality lemma [38]. O'Donnell's [53] lemma, stating that any term from which an innermost reduction is normalizing is strongly normalizing, is an immediate consequence of Klop's Lemma.

COROLLARY 5.1. *Any redex whose erased arguments are closed SN terms is perpetual in orthogonal fully-extended CCERSs.*

Proof. Immediate, since closed SN terms cannot be potentially infinite subterms. ■

Note that Theorem 5.1 implies a general (although not computable) perpetual strategy: simply contract a redex u in the term t whose erased arguments (if any) are not potentially infinite w.r.t. at least one ∞-subterm $s \subseteq t$ (although the erased arguments of u may be potentially infinite w.r.t. t). It is easy to see that the perpetual strategy F^∞ of Barendregt et al. [7, 5] and, in general, the *limit* perpetual strategy F^∞_{lim} of Khasidashvili [30, 31, 32] are special cases, since these strategies contract redexes whose arguments are in normal form and no (sub)terms can be substituted in the descendants of these arguments. The strategy F^∞_m (and hence the strategies F_3 and F_z), as well as the strategies F_1 and F_2, are also special cases of the above general perpetual strategy.

We conclude this section with a characterization of the perpetuality of erasing redexes, a characterization similar to the perpetuality criterion of β_K-redexes that was given by Bergstra and Klop [8].

Below, a substitution θ will be called SN iff $SN(x\theta)$ for every variable x.

DEFINITION 5.2. We call a redex u *safe* (respectively, *SN-safe*) if it is non-erasing or if it is erasing and for any (resp. SN-) substitution θ, if $u\theta$ erases an ∞-argument, then the contractum of $u\theta$ is an ∞-term. (Note that, by Corollary 3.1, u is erasing iff $u\theta$ is, for any θ, erasing.)

THEOREM 5.2. *In an orthogonal fully-extended CCERS R, any safe redex v is perpetual.*

Proof. Assume on the contrary that there is a context $C[\,]$ such that $t = C[v] \to s$ is a critical step. Let l be the minimal number such that, for some constricting minimal perpetual reduction $P : t = t_0 \overset{u_0}{\to} t_1 \overset{u_1}{\to} t_2 \twoheadrightarrow \infty$, the tail $P_l^* : t_l \twoheadrightarrow \infty$ of P is in an erased argument of a residual of v. Such an l exists by the proof of Theorem 5.1 (in the notation of that theorem, P_l^* is in an erased argument of $v_l \subseteq t_l$). Let v_l be the outermost of the redexes in t_l which contain u_l (and therefore, P_l^*) in an erased argument o_l, say the m-th from the left (thus $\infty(o_l)$). By the proof of Theorem 5.1, the m-th argument o of v is v-erased, $o_l = o\theta$, and $v_l = v\theta$ for some passive substitution θ.

We want to prove that the safety of v implies $\infty(s_l)$, hence $\infty(s)$, contradicting the assumption that $t \overset{v}{\to} s$ is critical (see the diagram for Theorem 5.1). By the Finite Developments theorem, we can assume that s_l is obtained from t_l by contracting (some of) the redexes in V_l in the following order: (a) contract redexes in V_l disjoint from v_l; (b) contract redexes in V_l that are in the main arguments of v_l; (c) contract the residual v_l^* of v_l; (d) contract the remaining redexes, i.e., those containing v_l in a main (by the choice of v_l) argument. Since the contractions (a) and (b) do not affect o_l, v_l^* erases an ∞-argument. (Recall from the proof of Theorem 5.1 that redexes in V_l are similar to their residuals contracted in any development of V_l.) Since $v_l = v\theta$ and redexes in (b) are in the substitution part of v_l, $v_l^* = v\theta^*$ for some substitution θ^*; hence its contractum e is infinite by the safety of v. By the choice of v_l, e has a descendant e' in s_l after the contractions (d). By the following diagram (where t_l^0 is obtained from t_l by the steps (a), (b) and (c); $w_0 + w_1 + \ldots$ is an infinite reduction of $e \subseteq t_l^0$; $U_0^{(d)}$ is the set of residuals of redexes in (d); and $U_i^{(d)}$ are respective residuals of $U_0^{(d)}$), $\infty(e)$ implies $\infty(e')$. Indeed, if $e_i \subseteq t_l^i$ is the descendant of e in t_l^i, then all $U_i^{(d)}$-descendants of e_i in s_l^i are disjoint and identical to e_i, and $s_l^i \twoheadrightarrow s_l^{i+1}$ contracts exactly one residual of w_i in every $U_i^{(d)}$-descendant of e_i (the latter are also descendants of $e' \subseteq s_l$). Hence $\infty(s_l)$ – a contradiction.

$$
\begin{array}{ccccccccc}
 & t_l^0 & \xrightarrow{w_0} & t_l^1 & \xrightarrow{w_1} & t_l^2 & \xrightarrow{w_2} & t_l^3 & \twoheadrightarrow \\
 & U_0^{(d)} \downarrow & & U_1^{(d)} \downarrow & & U_2^{(d)} \downarrow & & U_3^{(d)} \downarrow & \\
s_l = & s_l^0 & \underset{+}{\twoheadrightarrow} & s_l^1 & \underset{+}{\twoheadrightarrow} & s_l^2 & \underset{+}{\twoheadrightarrow} & s_l^3 & \underset{+}{\twoheadrightarrow}
\end{array}
$$

■

Møller Neergaard and Sørensen [49] give a different proof of perpetuality of safe K-redexes in the λ-calculus (safe K-redexes are called there *good*).

The following example demonstrates that non-erasing steps need not be perpetual in orthogonal CCERSs in general, that is, the restriction to fully-extended CCERSs is necessary:

EXAMPLE 5.1. Consider the ERS with rules:

$$
\begin{aligned}
\lambda x(A, B) &\to (B/x)A \\
\kappa yz(A) &\to (a/z)A \\
e(A, B) &\to c
\end{aligned}
$$

$$f(a) \;\rightarrow\; f(a)$$

where λ is a partial quantifier symbol binding only in its first argument, and $y \notin FV(A\theta)$ for any assignment θ admissible for the κ-rule. Consider the term $s = \kappa yz(\lambda x(e(x,y), f(z)))$. Note that s is not a redex (yet) due to the occurrence of y. On the one hand, contracting the e-redex yields an infinite reduction

$$s \rightarrow \kappa yz(\lambda x(c, f(z))) \rightarrow \lambda x(c, f(a)) \rightarrow \ldots$$

On the other hand, contracting the (non-erasing) λ-redex yields

$$s \rightarrow \kappa yz(e(f(z), y)) \rightarrow \kappa yz(c) \rightarrow c$$

as only, and strongly normalizing, reduction. Hence the λ-step is non-erasing but critical.

6. APPLICATIONS

We now give a number of applications demonstrating the power and usefulness of our perpetuality criteria. In some of the examples we will use the conventional λ-calculus notation [5], and by the *argument* of a β-redex $(\lambda x.s)o$ we will mean its second argument o.

6.1. The restricted orthogonal λ-calculi

Let us call an *orthogonal restricted λ-calculus* (ORLC) a calculus that is obtained from the λ-calculus by restricting the term set and the β-rule (by some conditions on arguments and contexts) and that is an orthogonal fully-extended CCERS. Examples include the λ_I-calculus, the call-by-value λ-calculus [60], and a large class of typed λ-calculi.

If R is an ORLC, then in the proofs of Theorem 5.1 and Theorem 5.2, the P_l-substitution (and in general, any passive substitution along a constricting perpetual reduction) is SN. This can be proved in a way similar to the one used to prove the Bergstra-Klop criterion (see [8, Proposition 2.8]), since in the terminology of [8] and in the notation of Theorem 5.1 and Theorem 5.2:

- P_l is SN-substituting (meaning that the arguments of contracted β-redexes are SN). This is immediate from the minimality of P_l.
- P_l is *simple* (meaning that no subterms can be substituted in the subterms substituted during the previous steps). This follows immediately from externality, w.r.t. the chosen minimal perpetual subterm, of minimal perpetual redexes (P_l is standard).

Hence, we have the following two corollaries. The first one is a perpetuality criterion for redex-occurrences and can be seen as a refinement of the Bergstra-Klop criterion [8] in that it takes into account passive substitutions that can be generated by the context. The second corollary is simply an extension of the Bergstra-Klop criterion (in the case of β-redexes, the converse statement is much easier to prove, see [8]).

COROLLARY 6.1. *Let t be an ∞-term and let $t \xrightarrow{v} s$ be a critical step in an ORLC. Then v erases a potentially infinite argument o such that $\infty(o\theta)$ for some passive SN-substitution θ.*

COROLLARY 6.2. *In an ORLC, any SN-safe redex v is perpetual.*

For the case of the λ-calculus, a different proof of Corollary 6.2 was published by Xi [72]. A simple proof of the Bergstra-Klop criterion, one that uses the strategy F_2 and thus is closely related to our proof was given by van Raamsdonk et al. [62] (that proof was obtained independently). Honsell and Lenisa [21] derive a strengthened version of the Bergstra-Klop criterion using semantical methods. They show that β-redexes that are safe w.r.t. *closed NF-substitutions* are also perpetual (closed *NF*-substitutions instantiate variables by closed normal forms). This criterion cannot be derived (at least, directly) from the above corollaries.

Note that these corollaries are not valid for orthogonal fully-extended CCERSs in general since, unlike the passive substitutions in an ORLC, the passive substitutions along constricting perpetual reductions in orthogonal fully-extended CCERSs need not be SN: Let $R = S \cup \{\sigma x AB \rightarrow Sx\omega(A/x)B,\ E(A) \rightarrow a\}$ where $\omega = \lambda x.Ap(x,x)$. Then the step $\sigma x Ap(x,x)E(x) \rightarrow \sigma x Ap(x,x)a$ is SN-safe (since it erases only a variable) but is critical as can be seen from the following diagram, of which the bottom part is the only reduction starting from $\sigma x Ap(x,x)a$:

$$
\begin{array}{ccccccccc}
\sigma x Ap(x,x)E(x) & \xrightarrow[\sigma]{} & SxwE(Ap(x,x)) & \xrightarrow[S]{} & E(Ap(w,w)) & \xrightarrow[\beta]{} & E(Ap(w,w)) & \xrightarrow[\beta]{} & \\
E\downarrow & & E\downarrow & & E\downarrow & & E\downarrow & & \\
\sigma x Ap(x,x)a & \xrightarrow[\sigma]{} & Sxwa & \xrightarrow[S]{} & a & \xrightarrow[\emptyset]{} & a & \xrightarrow[\emptyset]{} &
\end{array}
$$

6.2. Plotkin's call-by-value λ-calculus

To investigate the relation between the λ-calculus and ISWIM language of Landin [40], Plotkin [60] introduced the *call-by-value* λ-calculus λ_V. This calculus restricts the λ-calculus by allowing the contraction of redexes whose arguments are *values*, i.e., either abstractions $\lambda x.t$ or variables (we assume that there are no δ-rules in the calculus). Let the *lazy* call-by-value λ-calculus λ_{LV} be obtained from λ_V by allowing only call-by-value redexes that are not in the scope of a λ-occurrence (λ_{LV} is enough for computing values in λ_V, see Corollary 1 in [60]). Then it follows from Corollary 6.1, as well as from Corollary 6.2, that any λ_{LV}-redex is perpetual; hence λ_{LV} is UN. Indeed, let $v = (\lambda x.s)o$ be a λ_{LV}-redex. Then if o is a variable it is immediate that v cannot be critical and that if o is an abstraction any of its instances is an abstraction too and hence is a λ_{LV}-normal form. This is not surprising, however, because λ_{LV}-redexes are disjoint[14] and there is no duplication or erasure of (admissible) redexes.

[14] if u, v are redexes in a term t and $u = (\lambda x.e)o$, then $v \notin e$ because of the main λ of u, and $v \notin o$ since o is either a variable or an abstraction; orthogonality of λ_{LV} follows from a similar argument.

6.3. De Groote's β_{IS}-reduction

De Groote [17] introduced β_S-reduction on λ-terms by the following rule:

$$\beta_S : ((\lambda x.M)N)O \to (\lambda x.MO)N,$$

where $x \notin FV(M,O)$. He proved that the β_{IS}-calculus is UN. Clearly, this is an immediate corollary of Theorem 5.1 since the β_S- and β_I-rules are non-erasing (note that these rules do not conflict because of the conditions on bound variables). Using this result, de Groote proves strong normalization of a number of typed λ-calculi.

6.4. Böhm and Intrigila's λ-δ_k-calculus

Böhm and Intrigila [11] introduced the λ-δ_k-calculus in order to study UN solutions to fixed point equations, in the $\lambda\eta$-calculus. Since the K-redexes are the reason for the failure of uniform normalization in the $\lambda(\eta)$-calculus, Böhm and Intrigila define a 'restricted' K-combinator δ_K by the following rule:

$$\delta_K AB \to A,$$

where B can be instantiated to closed λ-δ_k-normal forms (possibly containing δ_K constants; such a reduction is still well defined). λ-δ_k-terms are λ_I-terms with the constant δ_K. Böhm and Intrigila show that the λ-δ_k-calculus is UN.

Whereas the η-rule is not fully-extended on the set of all (possibly erasing) terms, it is fully-extended on the restricted set of (non-erasing) λ-δ_k-terms. However, UN does not follow from Corollary 5.1 since λ-δ_k-calculus violates the orthogonality assumption. It is only weakly orthogonal since there are the usual (trivial) critical pairs between the β- and η-rule. We believe to have shown that Corollary 5.1 can be generalized to weakly orthogonal fully-extended CCERSs, which would yield UN of λ-δ_k-calculus, but we leave this to future work.

6.5. Honsell and Lenisa's β_{N^o}- and β_{KN}-calcului

Motivated by a semantical study of the λ_I- and λ_V-calculi, Honsell and Lenisa [20] and Lenisa [41] defined similar reductions, β_{N^o}- and β_{KN}-reductions, respectively, on λ-terms by the following rules:

$$\beta_{N^o}: \;\; (\lambda x.A)B \to (B/x)A,$$

where $\theta \in AA(\beta_{N^0})$ iff $\theta(B)$ is a closed β-normal form, and

$$\beta_{KN}: \;\; (\lambda x.A)B \to (B/x)A,$$

where $\theta \in AA(\beta_{KN})$ iff either $x \in FV(A\theta)$ or $\theta(B)$ is a variable or a closed β-normal form. We have immediately from Corollary 5.1 and Corollary 6.2, respectively, that β_{N^o} and β_{KN} are UN. Note however that these conclusions do not follow (at least, without an extra argument) from Bergstra and Klop's or Honsell and Lenisa's characterizations of perpetual β_K-redexes [8, 21], since $\beta_{N^o}, \beta_{KN} \subset \beta$ but not vice versa. (If t has an infinite β_{N^o}-reduction and $t \xrightarrow{u} s$ is a β_{N^o}-step, then the Bergstra-Klop and Honsell-Lenisa criteria imply the existence of an infinite *β-reduction* starting from s, not the existence of an infinite β_{N^o}-reduction, and similarly for β_{KN}.) In [20], semantical proofs of UN for β_{N^o} and β_{KN} are given.

7. CONCLUDING REMARKS

We have introduced (orthogonal fully-extended) Context-sensitive Conditional Expression Reduction Systems in which several (typed or untyped) λ-calculi can be expressed straightforwardly. Furthermore, we have obtained two powerful criteria for the perpetuality of redexes in orthogonal fully-extended CCERSs and have demonstrated their usefulness in applications.

As stated above, we claim that our results are also valid for Klop's orthogonal fully-extended *substructure* CRSs [39].

Intuitively this is the case since both ERSs and CRSs are essentially second-order frameworks, i.e., abstractions over metavariables are not allowed. We will now present an example showing that allowing abstractions on function variables, as is possible in Nipkow's higher-order rewriting systems [52], renders the Conservation Theorem invalid. The example exhibits a non-erasing step which is not perpetual.

EXAMPLE 7.1. Consider the higher-order rewrite system with rules:

$$\begin{array}{lll} f(\lambda yz.F(\lambda x.y(x), z)) & \to_f & F(\lambda x.c, \Omega) \\ app(abs(\lambda x.F(x)), S) & \to_{beta} & F(S) \end{array}$$

where the first rule contains a function variable (y) as argument to a free variable (F), the second rule is the usual [46] higher-order rendering of the β-rule from λ-calculus, and $\Omega = app(abs(\lambda x.app(x, x)), abs(\lambda x.app(x, x)))$. Then

$$f(\lambda yz.app(abs(\lambda x.y(x)), z)) \to_{beta} f(\lambda yz.y(z))$$

is non-erasing but critical. This can be seen from the following diagram, of which the bottom part is the only reduction starting from $f(\lambda yz.y(z))$.

$$\begin{array}{ccccccc} f(\lambda yz.app(abs(\lambda x.y(x)), z)) & \xrightarrow[f]{} & app(abs(\lambda x.c), \Omega) & \xrightarrow[beta]{\Omega} & app(abs(\lambda x.c), \Omega) & \xrightarrow[beta]{\Omega} & \cdots \\ beta \downarrow & & beta \downarrow & & beta \downarrow & & \\ f(\lambda yz.y(z)) & \xrightarrow[f]{} & c & \xrightarrow[\emptyset]{} & c & \xrightarrow[\emptyset]{} & \cdots \end{array}$$

The point of the example is that, unlike in the ERS- or CRS-case, in HRSs a substitution inside (caused by contracting a redex outside) a non-erasing redex can turn it into an erasing one.

There are several interesting directions for further research. One is to try to lift the orthogonality requirement somewhat, e.g. to weakly orthogonal systems or to calculi with explicit substitutions. Another is to try to find a higher-order analogue of our results (circumventing the counterexample above).

ACKNOWLEDGMENT

We thank the referees for constructive comments.

REFERENCES

1. Abramsky, S., Gabbay, D., and Maibaume, T. Eds. (1992), "Handbook of Logic in Computer Science," Vol. II, Oxford Univ. Press, London.

2. Aczel, P. (1978), "A general Church-Rosser theorem," Preprint, University of Manchester.
3. Akama, Y. (1993), On Mints' reduction for ccc-calculus, *in* [10, pp. 1–12].
4. Ariola, Z. M., Felleisen, M., Maraist, J., Odersky, M., and Wadler P. (1995), A call-by-need lambda calculus, *in* "Conference Record of the Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages," pp. 233–246, Assoc. Comput. Mach., New York.
5. Barendregt H.P. (1984), "The Lambda Calculus, its Syntax and Semantics," 2nd (rev.) ed., North-Holland, Amsterdam.
6. Barendregt H. P. (1992), Lambda calculi with types, *in* [1, pp. 117–309].
7. Barendregt, H. P., Bergstra, J., Klop, J. W., and Volken H. (1976), Some notes on lambda-reduction, *in* "Degrees, reductions, and representability in the lambda calculus," pp. 13–53, Technical Report Preprint 22, Department of mathematics, University of Utrecht.
8. Bergstra, J. A. and Klop, J. W. (1982), Strong normalization and perpetual reductions in the lambda calculus, *J. Inform. Process. Cybernet.* **18**, 403–417.
9. Bergstra, J. A. and Klop, J. W. (1986), Conditional rewrite rules: confluence and termination, *J. Comp. Sys. Sci.* **32**(3), 323–362.
10. Bezem, M. and Groote, J. F. (Eds.) (1993), "Typed Lambda Calculus and Applications," Lecture Notes in Computer Science, Vol. 664, Springer-Verlag, Berlin/New York.
11. Böhm, C. and Intrigila, B. (1994), The ant-lion paradigm for strong normalization, *Inform. and Comp.* **114**(1), 30–49.
12. Boudol, G. (1985), Computational semantics of term rewriting systems, *in* "Algebraic methods in semantics" (M. Nivat and J. C. Reynolds, Eds.), pp. 169–236, Cambridge Univ. Press, Cambridge, UK.
13. Church, A. (1941), The Calculi of Lambda-Conversion, Princeton Univ. Press, Princeton, NJ.
14. Dershowitz, N., Okada, M., and Sivakumar G. (1988), Canonical conditional rewrite systems, *in* "9th International Conference on Automated Deduction" (Lusk, E. and Overbeek, R., Eds.), pp. 538–549, Lecture Notes in Computer Science, Vol. 310, Springer-Verlag, Berlin/New York.
15. Ganzinger, H. (Ed.) (1996), "Rewriting Techniques and Applications," Lecture Notes in Computer Science, Vol. 1103, Springer-Verlag, Berlin/New York.
16. Gramlich, B. (1996), Termination and confluence properties of structured rewrite systems, Ph.D. Thesis, Universität Kaiserslautern.
17. De Groote, P. (1993), The conservation theorem revisited, *in* [10, pp. 163–178].
18. De Groote, P. and Hindley, J. R. (Eds.) (1997), "Typed Lambda Calculus and Applications," Lecture Notes in Computer Science, Vol. 1210, Springer-Verlag, Berlin/New York.
19. Hanus, M. and Prehofer, C. (1996), Higher-order narrowing with definitional trees, *in* [15, pp. 138–152].
20. Honsell, F. and Lenisa, M. (1993), Some results on the full abstraction problem for restricted lambda calculi, *in* "Mathematical Foundations of Computer Science 1993" (A. M. Borzyszkowski and S. Sokolowski, Eds.), pp. 84–104, Lecture Notes in Computer Science, Vol. 711, Springer-Verlag, Berlin/New York.
21. Honsell, F. and Lenisa, M. (1999), Semantical analysis of perpetual strategies in λ-calculus, *Theoret. Comp. Sci.* **212**(1–2), 183–209.
22. Huet, G. and Lévy, J.-J. (1991), Computations in orthogonal rewriting systems, *in* "Computational Logic, Essays in Honor of Alan Robinson" (J.-L. Lassez and G. Plotkin, Eds.), pp. 394–443, MIT Press, Cambridge, MA.
23. Karr, M. (1985), "Delayability" in proofs of strong normalizability in the typed λ-calculus, *in* "Mathematical Foundations of Computer Software" (H. Ehrig, C. Floyd, M. Nivat, and J. Tatcher, Eds.), pp. 208–222, Lecture Notes in Computer Science, Vol. 185, Springer-Verlag, Berlin/New York.
24. Kfoury, A. J. and Wells, J. (1995), New notions of reduction and non-semantic proofs of strong β normalization in typed λ-calculi, *in* "Logic in Computer Science" (D. Kozen, Ed.), pp. 311–321, IEEE Computer Society Press, Los Alamitos, CA.
25. Kfoury, A. J. and Wells, J. (1995), "Addendum to 'New notions of reduction and non-semantic proofs of β-strong normalization in typed λ-calculi' ". Report 95–007, Boston University Computer Science Department.

26. Khasidashvili, Z. (1990), β-reductions and β-developments of λ-terms with the least number of steps, *in* "COLOG-88" (P. Martin-Löf and G. Mints, Eds.), pp. 105–111, Lecture Notes in Computer Science, Vol. 417, Springer-Verlag, Berlin/New York.

27. Khasidashvili, Z. (1992), "The Church-Rosser theorem in orthogonal combinatory reduction systems," Report 1825, INRIA Rocquencourt.

28. Khasidashvili, Z. (1993), Optimal normalization in orthogonal term rewriting systems, *in* [36, pp. 243–258].

29. Khasidashvili, Z. (1994), On higher order recursive program schemes, *in* "Trees in Algebra and Programming - CAAP'94"(S. Tison, Ed.), pp. 172–186, Lecture Notes in Computer Science, Vol. 787, Springer-Verlag, Berlin/New York.

30. Khasidashvili, Z. (1994), Perpetuality and strong normalization in orthogonal term rewriting systems, *in* "STACS 94" (P. Enjalbert, E. W. Mayr, and K. W. Wagner, Eds.), pp. 163–174, Lecture Notes in Computer Science, Vol. 775, Springer-Verlag, Berlin/New York.

31. Khasidashvili, Z. (1994), The longest perpetual reductions in orthogonal expression reduction systems, *in* [51, pp. 191–203].

32. Khasidashvili, Z. (1997), On longest perpetual reductions in orthogonal expression reduction systems, Submitted for publication.

33. Khasidashvili, Z. and van Oostrom, V. (1995), Context-sensitive conditional expression reduction systems, *in* "SEGRAGRA 1995" (A. Corradini and U. Montanari, Eds.), Electronic Notes in Theoretical Computer Science, Vol.2, Elsevier Science Publishers B.V., Amsterdam.

34. Khasidashvili, Z. and van Oostrom, V. (1995), Context-sensitive conditional rewrite systems, Report SYS–C95–06, University of East Anglia.

35. Khasidashvili, Z. and Ogawa, M. (1997), Perpetuality and uniform normalization, *in* "Algebraic and Logic Programming" (M. Hanus, J. Heering, and K. Meinke, Eds.), pp. 240–255. Lecture Notes in Computer Science, Vol. 1298, Springer-Verlag, Berlin/New York.

36. Kirchner, C. (Ed.) (1993), "Rewriting Techniques and Applications," Lecture Notes in Computer Science, Vol. 690, Springer-Verlag, Berlin/New York.

37. Klop, J. W. (1980), "Combinatory Reduction Systems," Ph.D. Thesis, Utrecht University; CWI Tracts no. 127, Amsterdam.

38. Klop, J. W. (1992), Term rewriting systems, *in* [1, pp. 1–116].

39. Klop, J. W., van Oostrom, V., and van Raamsdonk, F. (1993), Combinatory reduction systems: introduction and survey, *Theoret. Comp. Sci.* **121**(1–2), 279–308.

40. Landin, P. J., The mechanical evaluation of expressions, *Computer Journal* **6**(4), 308–320.

41. Lenisa, M. (1997), Semantic techniques for deriving coinductive characterizations of observational equivalences for λ-calculi, *in* [18, pp. 248–266].

42. Lenisa, M. (1997), A uniform syntactical method for proving coinduction principles in λ-calculi, *in* "TAPSOFT'97: Theory and Practice of Software Development" (M. Bidoit and M. Dauchet, Eds.), pp. 309–320, Lecture Notes in Computer Science, Vol. 1214, Springer-Verlag, Berlin/New York.

43. Lévy, J.-J. (1978), "Réductions correctes et optimales dans le lambda-calcul," Ph.D. Thesis, Université Paris VII.

44. Lévy, J.-J. (1980), Optimal reductions in the Lambda-calculus, *in* "To H. B. Curry: Essays on Combinatory Logic, Lambda-calculus and Formalism" (J.R. Hindley and J.P. Seldin, Eds.), pp. 159–192, Academic Press Limited, London.

45. Loría-Sáenz, C. A. (1993), "A theoretical framework for reasoning about program construction based on extensions of rewrite systems," Ph.D. Thesis, Universität Kaiserslautern.

46. Mayr, R. and Nipkow, T. (1998), Higher-order rewrite systems and their confluence, *Theoret. Comp. Sci.* **192**, 3–29.

47. Melliès, P.-A. (1996), "Description abstraite des systèmes de réécriture," Ph.D. Thesis, Université Paris VII.

48. Milner, R. (1992), Functions as processes, *J. Math. Struct. Comp. Sci.* **2**(2), 119–141.

49. Møller Neergaard, P. and Sørensen, M. H. (1999), Conservation and uniform normalization in lambda calculi with erasing reductions, Submitted for publication.

50. Nederpelt, R. P. (1973), "Strong normalization for a typed lambda-calculus with lambda structured types," Ph.D. Thesis, Technische Hogeschool Eindhoven.

51. Nerode, A. and Matiyasevich, Yu. V. (Eds.) (1994), "Logical Foundations of Computer Science," Lecture Notes in Computer Science, Vol. 813, Springer-Verlag, Berlin/New York.

52. Nipkow, T. (1993), Orthogonal higher-order rewrite systems are confluent, *in* [10, pp. 306–317].

53. O'Donnell, M. J. (1977), "Computing in Systems Described by Equations," Lecture Notes in Computer Science, Vol. 58, Springer-Verlag, Berlin/New York.

54. Van Oostrom, V. (1994), Confluence for abstract and higher-order rewriting, Ph.D. Thesis, Vrije Universiteit, Amsterdam.

55. Van Oostrom, V. (1996), Higher-Order Families, [15, pp. 392–407].

56. Van Oostrom, V. (1997), Finite Family Developments, *in* "Rewriting Techniques and Applications" (H. Comon, Ed.), pp. 308–322, Lecture Notes in Computer Science, Vol. 1232, Springer-Verlag, Berlin/New York.

57. Van Oostrom, V. and van Raamsdonk, F. (1994), Weak orthogonality implies confluence: the higher-order case, *in* [51, pp. 379–392].

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

59. Plaisted, D. A. (1993), Polynomial time termination and constraint satisfaction tests, *in* [36, pp. 405–420].

60. Plotkin, G. (1975), Call-by-name, call-by-value and the λ-calculus, *Theoret. Comp. Sci.* **1**, 125–159.

61. Van Raamsdonk, F. (1996), "Confluence and normalisation for higher-order rewriting," Ph.D. Thesis, Vrije Universiteit, Amsterdam.

62. Van Raamsdonk, F., Severi, P., Sørensen, M. H., and Xi H. (1999), Perpetual reductions in λ-calculus, *Inform. and Comp.* **149**(2), 173–229.

63. Sørensen, M. H. (1998), Properties of infinite reduction paths in untyped λ-calculus, *in* "Tbilisi Symposium on Logic, Language and Computation, Selected papers" (J. Ginzburg, Z. Khasidashvili, J.J. Lévy, C. Vogel, and E. Vallduví, Eds.), pp. 353–367, SiLLI Publications, CSLI, Stanford.

64. Sørensen, M. H. (1996), Effective longest and infinite reduction paths in untyped λ-calculus. *in* "Trees in Algebra and Programming - CAAP'96" (H. Kirchner, Ed.), pp. 287–301, Lecture Notes in Computer Science, Vol. 1059, Springer-Verlag, Berlin/New York.

65. Sørensen, M. H. (1997), Strong Normalization from weak normalization in typed λ-calculi, *Inform. and Comp.* **133**(1), 35–71.

66. Sørensen, M. H. (1997), "Normalization in λ-calculus and type theory," Ph.D.Thesis, Københavns Universitet.

67. Toyama, Y. (1988), Confluent term rewriting systems with membership conditions, *in* "Conditional Term Rewriting Systems" (S. Kaplan and J.-P. Jouannaud, Eds.), pp. 128–141, Lecture Notes in Computer Science, Vol. 308, Springer-Verlag, Berlin/New York.

68. Troelstra, T. A and Schwichtenberg, H. (1996), "Basic Proof Theory," Cambridge Tracts in Theoretical Computer Science, Vol. 43, Cambridge Univ. Press, Cambridge, UK.

69. De Vrijer, R. C. (1987), Exactly estimating functionals and strong normalization, *Koninklijke Nederlandse Akademie van Wetenschappen* **90**(4). Also appeared in *Indag. Math.* **49**, 479–493.

70. De Vrijer, R. C. (1987), "Surjective pairing and strong normalization: two themes in lambda calculus," Ph.D. Thesis, Universiteit van Amsterdam.

71. Wolfram, D. (1993), "The Causal Theory of Types," Cambridge Tracts in Theoretical Computer Science, Vol. 21, Cambridge Univ. Press, Cambridge, UK.

72. Xi, H. (1996), "An induction measure on λ-terms and its applications," Research Report 96–192, Department of Mathematical Sciences, Carnegie Mellon University.

73. Xi, H. (1997), Weak and strong beta normalisations in typed λ-calculi, *in* [18, pp. 390–404].

APPENDIX

A.1. ENCODING OF THE π-CALCULUS AS A CCERS

In this section we will encode as a CCERS the version of the π-calculus described by Milner [48] . Recall that the π-calculus agents P, Q, ... are defined as follows:

$$P ::= \overline{x}y.P \mid x(y).P \mid 0 \mid P|P \mid !P \mid (x)P$$

Basic interaction is generated from the rule

$$x(y).P|\overline{x}z.Q \to [z/y]P|Q$$

by closing under unguarded contexts and working modulo structural congruence (see [48]).

A CCERS (Σ_π, R_π) can be associated to the π-calculus as follows. The alphabet Σ_π consists of the function symbols $0, !, |, O$ with respective arities $0, 1, 2, 3$ and the quantifier symbols I and R with arities $(1, 2)$ and $(1, 1)$. I binds only in its last argument. The map $[\,]$ transforms π-terms into terms in $Ter(\Sigma_\pi)$. The only non-obvious cases are input, output, and restriction:

$$[x(y).P] = Iy(x, [P])\,;\ [\overline{x}z.Q] = O(x, z, [Q])\,;\ [(x)P] = Rx([P])$$

Combining the transformation $[\,]$ with the closing under unguarded contexts and the structural congruence leads to rules R_π of the form

$$C_1[Iy(X, P)] \,|\, C_2[O(X, Z, Q)] \to C_1[(Z/y)P] \,|\, C_2[Q], \text{ where}$$

1. P, Q, X, Z are metavariables, and admissible assignments for X, Z are variables.
2. The indicated subterms must be unguarded in $C_1[\,]$ and $C_2[\,]$ and not in the scope of RX (among the symbols above them can occur only the operators $|$, $!$ and Rx with $x \neq X$).
3. For any redex only (all) unguarded contexts are admissible.

The 'critical pairs' for the interaction rule are obviously preserved by the translation, so R_π is not orthogonal. Nevertheless, we expect results like the following: for the standard translation of the λ-calculus into the π-calculus, the corresponding subcalculus R_π is orthogonal and hence confluent modulo the structural congruence.

A.2. CONFLUENCE FOR A CALL-BY-NEED λ-CALCULUS

We will show that the *call-by-need λ-calculus* introduced and studied by Ariola et al. [4] is an orthogonal CCERS. Terms in this calculus are ordinary λ-terms possibly containing let expressions, but the rewrite rules have conditions on them as follows. Define the syntactic categories by the following grammar:

$$\begin{aligned}
M &::= x \mid MM \mid \lambda x.M \mid \mathsf{let}\ x = M\ \mathsf{in}\ M \\
V &::= \lambda x.M \\
A &::= V \mid \mathsf{let}\ x = M\ \mathsf{in}\ A \\
E &::= [\,] \mid EM \mid \mathsf{let}\ x = M\ \mathsf{in}\ E \mid \mathsf{let}\ x = E\ \mathsf{in}\ E[x]
\end{aligned}$$

The rules are the following:

$$
\begin{aligned}
(\lambda x.M)M' &\rightarrow \mathsf{let}\ x = M'\ \mathsf{in}\ M \\
\mathsf{let}\ x = V\ \mathsf{in}\ E[x] &\rightarrow \mathsf{let}\ x = V\ \mathsf{in}\ E[V] \\
(\mathsf{let}\ x = M\ \mathsf{in}\ A)M' &\rightarrow \mathsf{let}\ x = M\ \mathsf{in}\ AM' \\
\mathsf{let}\ x = (\mathsf{let}\ y = M\ \mathsf{in}\ A)\ \ \mathsf{in}\ E[x] &\rightarrow \mathsf{let}\ y = M\ \mathsf{in}\ \mathsf{let}\ x = A\ \mathsf{in}\ E[x]
\end{aligned}
$$

the rewrite relation $\rightarrow_s$ is obtained from these rules by allowing arbitrary contexts.

By case analysis we show that each of the syntactic categories is closed under $\rightarrow_s$ and that there are no overlaps between the rules, so the system is an orthogonal conditional ERS.

- M is obviously $\rightarrow_s$-closed and contains V, A, and $E[y]$ for every y.
- V is $\rightarrow_s$-closed by the previous item and the fact that no root-steps are possible.
- A is $\rightarrow_s$-closed since V is (by the previous item) and we can see that $\mathsf{let}\ x = M\ \mathsf{in}\ A$ is closed by considering (root-)overlaps with the four rewrite rules.

1. Root-overlap with the first or third rule is syntactically not possible.

2. To show that root-overlap with the second rule and fourth rule is not possible it suffices to show that no elements in A are of the form $E[y]$ for any y, which we prove by induction on the definition of A:

   (i) $V \cap E[y] = \emptyset$ since $E[y] \not\equiv \lambda x.M$.

   (ii) $(\mathsf{let}\ x = M\ \mathsf{in}\ A) \not\equiv E[y]$ since

   a. $(\mathsf{let}\ x = M\ \mathsf{in}\ A) \not\equiv y$,

   b. $(\mathsf{let}\ x = M\ \mathsf{in}\ A) \not\equiv E[y]N$,

   c. $(\mathsf{let}\ x = M\ \mathsf{in}\ A) \not\equiv (\mathsf{let}\ z = N\ \mathsf{in}\ E[y])$ by induction hypothesis, and

   d. $(\mathsf{let}\ x = M\ \mathsf{in}\ A) \not\equiv (\mathsf{let}\ z = E[y]\ \mathsf{in}\ E[z])$ by induction hypothesis.

- $E[y]$ is shown to be $\rightarrow_s$-closed by induction on the definition of E.

1. y is a normal form.

2. $E[y]M$ cannot be root-rewritten because $E[y] \not\equiv \lambda x.N$ (first rule) and $E[y] \cap A = \emptyset$ (third rule). $E[y]$ and M are $\rightarrow_s$-closed by hypothesis.

3. $\mathsf{let}\ x = M\ \mathsf{in}\ E[y]$ cannot be root-rewritten (since $x \not\equiv y$ in the second and fourth rules), and M and $E[y]$ are $\rightarrow_s$-closed by hypothesis.

4. $\mathsf{let}\ x = E[y]\ \mathsf{in}\ F[x]$ cannot be root-rewritten because V and $E[y]$ are disjoint (second rule), and A and $E[y]$ are disjoint (fourth rule). Both $E[y]$ and $F[x]$ are $\rightarrow_s$-closed by hypothesis.

Because of the $\rightarrow_s$-closedness of the syntactic categories, to show orthogonality we need only to check for possible 'critical pairs' between the rules. One easily confirms that there are no such pairs by using the earlier observation that $E[y] \cap A = \emptyset$ (which avoids the possibility of a conflict between the third and fourth rules).

A.3. PERPETUAL STRATEGIES

In this appendix we collect definitions of all perpetual strategies mentioned in the body of the paper.

Perpetual strategies on λ-terms will be defined by induction on the structure of terms not in β-normal form, and the redex chosen by a strategy for contraction will be indicated here by underlining. SN_β (resp. NF_β) will denote the set of strongly β-normalizing λ-terms (resp. the set of λ-terms in β-normal form). $\overline{t}$ will denote a sequence of λ-terms $t_1, \cdots, t_n$ and $\overline{t} \in S$ will denote $t_i \in S$ for each i.

DEFINITION A.1. ([7]) The β-reduction strategy F_∞ is defined as follows:

$$\begin{array}{lcll} F_\infty(x\overline{t}s\overline{o}) & = & x\overline{t}F_\infty(s)\overline{o} & \text{if } \overline{t} \in NF_\beta, s \notin NF_\beta \\ F_\infty(\lambda x.t) & = & \lambda x.F_\infty(t) & \\ F_\infty((\lambda x.t)s\overline{o}) & = & (\lambda x.t)F_\infty(s)\overline{o} & \text{if } x \notin FV(t),\ s \notin NF_\beta \\ F_\infty((\lambda x.t)s\overline{o}) & = & \underline{(\lambda x.t)s}\overline{o} & \text{if } x \in FV(t) \text{ or } s \in NF_\beta \end{array}$$

DEFINITION A.2. ([8]) The β-reduction strategy F_1 (called F by Bergstra and Klop [8]) is defined as follows:

$$\begin{array}{lcll} F_1(x\overline{t}s\overline{o}) & = & x\overline{t}F_1(s)\overline{o} & \text{if } \overline{t} \in NF_\beta, s \notin NF_\beta \\ F_1(\lambda x.t) & = & \lambda x.F_1(t) & \\ F_1((\lambda x.t)s\overline{o}) & = & (\lambda x.t)F_1(s)\overline{o} & \text{if } s \notin SN_\beta \\ F_1((\lambda x.t)s\overline{o}) & = & \underline{(\lambda x.t)s}\overline{o} & \text{if } s \in SN_\beta \end{array}$$

DEFINITION A.3. ([62]) The β-reduction strategy F_2 is defined as follows:

$$\begin{array}{lcll} F_2(x\overline{t}s\overline{o}) & = & x\overline{t}F_2(s)\overline{o} & \text{if } \overline{t} \in SN_\beta, s \notin SN_\beta \\ F_2(\lambda x.t) & = & \lambda x.F_2(t) & \\ F_2((\lambda x.t)\overline{o}) & = & (\lambda x.F_2(t))\overline{o} & \text{if } t \notin SN_\beta \\ F_2((\lambda x.t)s\overline{o}) & = & (\lambda x.t)F_2(s)\overline{o} & \text{if } t \in SN_\beta, s \notin SN_\beta \\ F_2((\lambda x.t)s\overline{o}) & = & \underline{(\lambda x.t)s}\overline{o} & \text{if } t, s \in SN_\beta \end{array}$$

DEFINITION A.4. ([63, 62]) The β-reduction strategy F_3 is defined as follows:

$$\begin{array}{lcll} F_3(x\overline{t}s\overline{o}) & = & x\overline{t}F_3(s)\overline{o} & \text{if } \overline{t} \in SN_\beta, s \notin SN_\beta \\ F_3(\lambda x.t) & = & \lambda x.F_3(t) & \\ F_3((\lambda x.t)\overline{o}) & = & (\lambda x.F_3(t))\overline{o} & \text{if } t \notin SN_\beta \\ F_3((\lambda x.t)\overline{o}s\overline{e}) & = & (\lambda x.t)\overline{o}F_3(s)\overline{e} & \text{if } t, \overline{o} \in SN_\beta, s \notin SN_\beta \\ F_3((\lambda x.t)s\overline{o}) & = & \underline{(\lambda x.t)s}\overline{o} & \text{if } t, s, \overline{o} \in SN_\beta \end{array}$$

DEFINITION A.5. The β-reduction strategy F_{lm}^∞ is defined as follows:

$$
\begin{array}{lcll}
F^{\infty}_{lm}(x\bar{t}s\bar{o}) & = & x\bar{t}F^{\infty}_{lm}(s)\bar{o} & \text{if } \bar{t} \in SN_\beta, s \notin SN_\beta \\
F^{\infty}_{lm}(\lambda x.t) & = & \lambda x.F^{\infty}_{lm}(t) & \\
F^{\infty}_{lm}((\lambda x.t)\bar{o}) & = & (\lambda x.F^{\infty}_{lm}(t))\bar{o} & \text{if } t \notin SN_\beta \\
F^{\infty}_{lm}((\lambda x.t)\bar{o}s\bar{e}) & = & (\lambda x.t)\bar{o}F^{\infty}_{lm}(s)\bar{e} & \text{if } t, \bar{o}, (\lambda x.t)\bar{o} \in SN_\beta,\ s \notin SN_\beta \\
F^{\infty}_{lm}((\lambda x.t)s\overline{oe}) & = & \underline{(\lambda x.t)s}\overline{oe} & \text{if } t, s, \bar{o} \in SN_\beta,\ (\lambda x.t)s\bar{o} \notin SN_\beta
\end{array}
$$

DEFINITION A.6. ([30, 31, 32]) The *limit* strategy F^{∞}_{lim} in an orthogonal fully-extended CCERS is defined as follows:

1. Let u_l be a redex in a term t defined as follows: choose an external redex u_1 in t; choose an erased argument s_1 of u_1 that is not in normal form (if any); choose in s_1 an external redex u_2, and so on as long as possible. Let $u_1, s_1, u_2, \ldots, u_l$ be such a sequence. The redex u_l is called a *limit redex* of t.

2. We call a strategy *limit*, noted F^{∞}_{lim}, if in any term not in normal form it selects a limit redex. (Note that by Theorem 3.3 in any term not in normal form there is a limit redex.)