

Document downloaded from:

<http://hdl.handle.net/10251/50899>

This paper must be cited as:

Lucas Alba, S.; Meseguer, J. (2014). Strong and weak operational termination of order-sorted rewrite theories. En *Rewriting Logic and Its Applications*. Springer Verlag (Germany). 178-194. doi:10.1007/978-3-319-12904-4_10.



The final publication is available at

http://link.springer.com/chapter/10.1007/978-3-319-12904-4_10

Copyright Springer Verlag (Germany)

Strong and Weak Operational Termination of Order-Sorted Rewrite Theories^{*}

Salvador Lucas^{1,2} and José Meseguer²

¹ DSIC, Universitat Politècnica de València, Spain

² CS Dept. University of Illinois at Urbana-Champaign, IL, USA

Abstract. This paper presents several new results on conditional term rewriting within the general framework of order-sorted rewrite theories (OSRTs) which contains the more restricted framework of conditional term rewriting systems (CTRSs) as a special case. The results uncover some subtle issues about conditional termination. We first of all generalize a previous known result characterizing the operational termination of a CTRS by the quasi-decreasing ordering notion to a similar result for OSRTs. Second, we point out that the notions of *irreducible* term and of *normal form*, which coincide for unsorted rewriting are *totally different* for conditional rewriting and formally characterize that difference. We then define the notion of a *weakly operationally terminating* (or *weakly normalizing*) OSRT, give several evaluation mechanisms to compute normal forms in such theories, and investigate general conditions under which the rewriting-based operational semantics and the initial algebra semantics of a confluent OSRT coincide thanks to a notion of *canonical term algebra*. Finally, we investigate appropriate conditions and proof methods to ensure good executability properties of an OSRT for computing normal forms.

Keywords: Conditional term rewriting, strong and weak operational termination, irreducible terms, normalized terms, rewriting logic, Maude.

1 Introduction

This paper presents several new contributions to conditional term rewriting and to the semantics of declarative, rewriting-based languages. The key notion is that of an *Order-Sorted Rewrite Theory* (OSRT) $\mathcal{R} = (\Sigma, B, R)$, where (Σ, B) is an order-sorted equational theory [10] with equational axioms B , and R is a collection of rewrite rules with oriented conditions of the form: $\ell \rightarrow r \Leftarrow s_1 \rightarrow t_1, \dots, s_n \rightarrow t_n$, which are applied *modulo* B . All the results are *in particular* new results for *Conditional Term Rewriting Systems* (CTRSs);

^{*} Research partially supported by NSF grant CNS 13-19109. Salvador Lucas' research was developed during a sabbatical year at the CS Dept. of the UIUC and was also partially supported by Spanish MECD grant PRX12/00214, MINECO project TIN2010-21062-C02-02, and GV grant BEST/2014/026 and project PROMETEO/2011/052.

that is, for order-sorted rewrite theories of the form $\mathcal{R} = (\Sigma, \emptyset, R)$, with Σ having a *single* sort. The greater generality of OSRTs is not a caprice, but an absolute necessity for making formal specification and declarative programming practical and expressive.

Our contributions consist in asking and providing detailed answers to the following, innocent-sounding questions:

1. Can the operational termination of OSRTs be characterized in terms of orders?
2. What is the right notion of *normal form* for an OSRT?
3. What is the right notion of *weak operational termination* for an OSRT?
4. Under what conditions can OSRTs be used as *declarative programs* having a well-behaved semantics? And how can we *evaluate* such programs?
5. Under what conditions does an OSRT have a *canonical term algebra* that can be effectively computed and that provides a *complete agreement* between the operational semantics of the OSRT as a functional program, and its mathematical, initial algebra semantics? How can some of these executability conditions be checked in practice?

Surprisingly enough, some of these questions seem to never have been asked. At best, the issues involved seem to have remained implicit as not well-understood, anomalous features in the literature. Consider, for example, question (2) above, which asks about the notion of normal form. For unconditional term rewriting the notion is absolutely clear and unproblematic: a normal form is a term t that is *irreducible*, that is, such that there is no t' with $t \rightarrow t'$. For an OSRT, and in particular for a CTRS, the notion of normal form is actually highly problematic. The big problem is that for an OSRT there can be terms t that are irreducible in the above sense, i.e., there is no t' with $t \rightarrow t'$, but such that when we give t to a rewrite engine for evaluation such an engine loops! For a trivial example, consider the single conditional rewrite rule $a \rightarrow b \Leftarrow a \rightarrow c$. Since the rewrite relation defined by this conditional rule is the empty set, the constant a is trivially irreducible; but the proof tree associated to the normalization of a using the CTRS inference system is *infinite* [7], and a rewrite engine that tries to evaluate a will loop when trying to satisfy the rule's condition.³ Therefore, calling a a *normal form* is a very bad joke, since, intuitively, a term is considered to be a normal form if it is “fully normalized,” that is, if it is the *result* of fully evaluating some input term by rewriting. Our answer to this puzzle is to introduce a precise distinction (fully articulated in the paper) between irreducible terms and normal forms: every normal form is irreducible, but, as the above example shows, not every irreducible term is a normal form. We call an OSRT *normal* iff every irreducible term is a normal form, and call it *abnormal* otherwise. Abnormal theories, like the one above, are hopeless for executability purposes and should be viewed as monsters in the menagerie of CTRSs and OSRTs.

³ For this trivial example one could find ways for an engine to detect *this* looping; but undecidability of termination makes a general loop-detecting engine an oxymoron.

Termination is quite a subtle issue for OSRTs in general and CTRSs in particular. Many notions have been proposed (see e.g., [12]), but it is by now well-understood that the most satisfactory notion from a computational point of view is that of *operational termination* [7] (more on this later). Here we ask and answer two questions, further developing this notion. The first is question (1) above. For the case of deterministic 3-CTRS we proved in [7] that operational termination is *equivalent* to the order-based notion of *quasi-decreasingness*. In Section 3 we generalize this result to a similar result characterizing operational termination of OSRTs in terms of an (axiom-compatible) term ordering.

A second, related question, seemingly not previously addressed in the literature, is question (3), which could be rephrased as follows: what is the right notion of weak termination/normalization for OSRTs? As further explained in Section 4, there are in fact two possible notions, a computationally ill-behaved one (*weak termination*: every term has a terminating rewrite sequence ending in an irreducible term), and a computationally well-behaved one (*weak operational termination*: every term has a normal form).

The notions of normal OSRT and of weak operational termination are closely related to another question, namely, question (4), on executability conditions for declarative, conditional rule-based programs, and on their evaluation methods, i.e., their operational semantics. Interestingly enough, as we explain in Section 4, there are *several* evaluation methods, which become increasingly efficient as we impose further conditions on the OSRT which we use as our program.

For *functional* programs specified by an OSRT, the issue is not just one of having good executability conditions, but actually of *correctness*. More precisely, of *semantic agreement* between an abstract *initial algebra semantics* when the rules are viewed as equations, and an *operational semantics* based on rewriting, where the computed *values* —that is, the normal forms— give rise to a very intuitive algebra, the *canonical term algebra*, which under the assumptions of confluence, coherence, sort-decreasingness and operational termination is *isomorphic* to the initial algebra of the specification. Question (5) above asks, essentially: what is the *non plus ultra* in terms of generality to maintain this isomorphism and keeping an exact agreement between mathematical and operational semantics? That is, what are the right conditions for this semantic agreement when we drop the operational termination condition? This is also answered in Section 4, relating the answers to associated evaluation methods to compute normal forms. Last but not least, in Sections 4 and 5 we investigate appropriate *conditions* and *proof methods* to ensure that a theory has good executability properties such as being normal, and evaluation to normal form defining a total recursive function.

2 Preliminaries

Order-Sorted Algebra. We summarize here material from [4, 10] on order-sorted algebra. We start with a partially ordered set (S, \leq) of *sorts*, where $s \leq s'$ is interpreted as *subsort inclusion*. The *connected components* of (S, \leq) are the equivalence classes $[s]$ corresponding to the least equivalence relation \equiv_{\leq}

containing \leq . We also define $[s] = \{s' \in S \mid s' \leq s\}$, i.e., the sorts in S which are smaller than or equal to s . When $[s]$ has an upper bound, we denote it by $\top_{[s]}$. An order-sorted signature (Σ, S, \leq) consists of a poset of sorts (S, \leq) and a $S^* \times S$ -indexed family of sets $\Sigma = \{\Sigma_{w,s}\}_{(w,s) \in S^* \times S}$, which are *function symbols* with a given string of argument sorts and a result sort. If $f \in \Sigma_{s_1 \dots s_n, s}$, then we display the function symbol f as $f : s_1 \dots s_n \longrightarrow s$. This is called a *rank declaration* for symbol f . Some of these symbols f can be *subsort-overloaded*, i.e., they can have several rank declarations related in the \leq ordering [4]. Constant symbols, however, have only one rank declaration. To avoid ambiguous terms, we assume that Σ is *sensible*, meaning that if $f : s_1 \dots s_n \rightarrow s$ and $f : s'_1 \dots s'_n \rightarrow s'$ are such that $[s_i] = [s'_i]$, $1 \leq i \leq n$, then $[s] = [s']$. Throughout this paper, Σ will always be assumed *sensible*.

Given an S -sorted set $\mathcal{X} = \{\mathcal{X}_s \mid s \in S\}$ of *mutually disjoint* sets of variables, the set $\mathcal{T}_\Sigma(\mathcal{X})_s$ of terms of sort s is the least set such that (i) $\mathcal{X}_s \subseteq \mathcal{T}_\Sigma(\mathcal{X})_s$, (ii) $\mathcal{T}_\Sigma(\mathcal{X})_s \supseteq \mathcal{T}_\Sigma(\mathcal{X})_{s'}$, and (iii) for each $f : s_1 \dots s_n \longrightarrow s$ and $t_i \in \mathcal{T}_\Sigma(\mathcal{X})_{s_i}$, $1 \leq i \leq n$, $f(t_1, \dots, t_n) \in \mathcal{T}_\Sigma(\mathcal{X})_s$. The assumption that Σ is sensible ensures that if $[s] \neq [s']$, then $\mathcal{T}_\Sigma(\mathcal{X})_{[s]} \cap \mathcal{T}_\Sigma(\mathcal{X})_{[s']} = \emptyset$.

The set $\mathcal{T}_\Sigma(\mathcal{X})$ of order-sorted terms is $\mathcal{T}_\Sigma(\mathcal{X}) = \cup_{s \in S} \mathcal{T}_\Sigma(\mathcal{X})_s$. The family $\{\mathcal{T}_\Sigma(\mathcal{X})_s\}_{s \in S}$ together with the operations $f : (t_1, \dots, t_n) \mapsto f(t_1, \dots, t_n)$ define an order-sorted Σ -algebra called the *free algebra* on \mathcal{X} and denoted $\mathcal{T}_\Sigma(\mathcal{X})$. When $\mathcal{X} = \emptyset$, $\mathcal{T}_\Sigma = \mathcal{T}_\Sigma(\emptyset)$ denotes the *initial algebra*. An element of any set $\mathcal{T}_\Sigma(\mathcal{X})_s$ is called a *well-formed term*. A simple syntactic condition on (Σ, S, \leq) called *preregularity* [4] ensures that each well-formed term t has always a *least sort* possible among all sorts in S , which is denoted $LS(t)$. An order-sorted substitution σ is an S -sorted mapping $\sigma = \{\sigma : \mathcal{X}_s \rightarrow \mathcal{T}_\Sigma(\mathcal{X})_s\}_{s \in S}$ from variables to terms. The application of an OS-substitution σ to t (denoted $\sigma(t)$) consists of simultaneously replacing the variables occurring in t by a term according to the mapping σ . A specialization ν is an injective OS-substitution that maps a variable x of sort s to a variable x' of sort $s' \leq s$.

Order-Sorted Rewrite Theories. An (order-sorted) rewrite rule is an ordered pair (l, r) , written $l \rightarrow r$, with $l, r \in \mathcal{T}_\Sigma(\mathcal{X})$, and $LS(l) \equiv_{\leq} LS(r)$. An *order-sorted conditional rewrite theory* (OSRT) is a triple $\mathcal{R} = (\Sigma, B, R)$, where Σ is an order-sorted signature, B is a set of Σ -equations, and R is a collection of conditional rewrite rules with oriented conditions of the form $\ell \rightarrow r \Leftarrow s_1 \rightarrow t_1, \dots, s_n \rightarrow t_n$, where $\ell \rightarrow r$ and the $s_i \rightarrow t_i$ are order-sorted rewrite rules (with $\ell \notin \mathcal{X}_s$ for all $s \in S$), and where the conditions $s_i \rightarrow t_i$ are intended to express the reachability of (instances of) t_i from (instances of) s_i . Throughout this paper the equations $(u = v) \in B$ are assumed to be: (i) *regular* (i.e., $\text{Var}(u) = \text{Var}(v)$), (ii) *linear* (i.e., no repeated variables in either u or v); (iii) there is a B -matching algorithm; and (iv) *sort-preserving* (i.e., for each substitution θ , $LS(\theta(u)) = LS(\theta(v))$). Examples of axioms B satisfying (i)–(iii) include combinations of associativity and/or commutativity and/or identity axioms. Maude supports rewriting modulo such axioms and also checks automatically property (iv) (it actually checks a somewhat weaker condition for identity axioms that still ensures a least sort for each B -equivalence class).

(Refl)	$\frac{}{u \rightarrow^* v}$ if $u =_B v$
(Tran)	$\frac{u \rightarrow u' \quad u' \rightarrow^* v}{u \rightarrow^* v}$
(Cong)	$\frac{u_i \rightarrow u'_i}{f(u_1, \dots, u_i, \dots, u_k) \rightarrow f(u_1, \dots, u'_i, \dots, u_k)}$ where $f \in \Sigma$ and $1 \leq i \leq k = ar(f)$
(Repl)	$\frac{\sigma(u_1) \rightarrow^* \sigma(v_1) \quad \dots \quad \sigma(u_n) \rightarrow^* \sigma(v_n)}{u \rightarrow v}$ where $\ell \rightarrow r \Leftarrow u_1 \rightarrow v_1 \dots u_n \rightarrow v_n \in \mathcal{R}$, σ is an OS-substitution, $u =_B \sigma(\ell)$ and $v = \sigma(r)$

Fig. 1. Inference rules for order-sorted rewrite theories

Rewrite rules $\ell \rightarrow r \Leftarrow c$ in OSRTs are classified according to the distribution of variables among ℓ , r , and c , as follows: type 1, if $\mathcal{V}ar(r) \cup \mathcal{V}ar(c) \subseteq \mathcal{V}ar(\ell)$; type 2, if $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)$; type 3, if $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell) \cup \mathcal{V}ar(c)$; and type 4, if no restriction is given. An n -OSRT contains only rewrite rules of types $m \leq n$. A 3-OSRT \mathcal{R} is called *deterministic* if for each rule $\ell \rightarrow r \Leftarrow s_1 \rightarrow t_1, \dots, s_n \rightarrow t_n$ in \mathcal{R} and each $1 \leq i \leq n$, we have $\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(\ell) \cup \bigcup_{j=1}^{i-1} \mathcal{V}ar(t_j)$. If for all specializations ν $LS(\nu(\ell)) \geq LS(\nu(r))$ then we say that the OS-rule $\ell \rightarrow r \Leftarrow c$ is *sort-decreasing*. We call an OSRT $\mathcal{R} = (\Sigma, B, R)$ *sort-decreasing* if all rules in R are so.

We write $t \rightarrow_{\mathcal{R}} u$ (resp. $t \rightarrow_{\mathcal{R}}^* u$) iff there is a well-formed proof tree for $t \rightarrow u$ (resp. $t \rightarrow^* u$) for \mathcal{R} using the inference system in Figure 1. As usual, $\rightarrow_{\mathcal{R}}$ is the *one-step* rewrite relation for the OSRT \mathcal{R} and $\rightarrow_{\mathcal{R}}^*$ is the *zero-or-more-steps* rewrite relation for \mathcal{R} . We write $t \rightarrow_{\mathcal{R}}^0 u$ if $t =_B u$; $t \rightarrow_{\mathcal{R}}^1 u$ if $t \rightarrow_{\mathcal{R}} u$, and $t \rightarrow_{\mathcal{R}}^n u$, for some $n > 1$ if there is a term t' such that $t \rightarrow_{\mathcal{R}} t'$ and $t' \rightarrow_{\mathcal{R}}^{n-1} u$.

Operational Termination. Given a logic \mathcal{L} (defined by its inference rules), one has the notion of a *theory* or *specification* \mathcal{S} in such a logic, so that \mathcal{L} 's inference system becomes specialized to each such specification \mathcal{S} to derive its provable theorems φ . Assume that we have an *interpreter* for the logic \mathcal{L} , that is, an *inference machine* that, given a theory \mathcal{S} and a goal formula φ will try to incrementally build a proof tree for φ . Intuitively, we will call \mathcal{S} *terminating* if for any φ the interpreter either finds a proof in finite time, or fails in all possible attempts also in finite time. In the same vein, we can say that a predicate π (for instance, \rightarrow or \rightarrow^* in the inference system of Figure 1) is operationally terminating if for any goal φ such that $\varphi = \pi(t_1, \dots, t_k)$ for terms t_1, \dots, t_k , φ is operationally terminating. The notion of *operational termination* captures this fact, meaning that, given an initial goal, an interpreter will either succeed in

finite time in producing a closed proof tree, or will fail in finite time, not being able to close or extend further any of the possible proof trees, after exhaustively searching all such proof trees [7]. In the following, according to the previous discussion, we speak about operational 1-termination of a OSRT as the operational termination of \rightarrow (with respect to the inference system of Figure 1). By operational termination of an OSRT we then mean the operational termination of \rightarrow^* . Similarly, we say that a term t is operationally (1-)terminating if every goal $t \rightarrow^* u$ (resp. $t \rightarrow u$) is operationally terminating for all terms u . We call \mathcal{R} *ground operationally (1-)terminating* iff all $t \in \mathcal{T}_\Sigma$ are so.

One last issue important for executability purposes is (strong) *B-coherence*. This means that if $t \rightarrow_{\mathcal{R}}^1 u$ and $t =_B t'$, then there exists a u' such that $t' \rightarrow_{\mathcal{R}}^1 u'$ and $u =_B u'$. For axioms B such as combinations of associativity, commutativity and identity, Maude automatically completes the user-specified rules so that they become *B-coherent*. In this paper we will assume that *all OSRTs are B-coherent*.

3 Orderings, Quasi-Decreasingness, and (Strong) Operational Termination

A binary relation R on a set A is *terminating* (or well-founded) if there is no infinite sequence $a_1 R a_2 R a_3 \dots$. Given $f : A^k \rightarrow A$ and $i \in \{1, \dots, k\}$, we say that f is *i-monotonic* on its i -th argument (or that f is *i-monotone* with respect to R) if $f(x_1, \dots, x_{i-1}, x, \dots, x_k) R f(x_1, \dots, x_{i-1}, y, \dots, x_k)$ whenever $x R y$, for all $x, y, x_1, \dots, x_k \in A$. We say that R is monotonic if, for all symbols f , f is monotonic w.r.t. R . In [7] we have shown that operational termination of deterministic 3-CTRSs (which are special deterministic 3-OSRTs where the set of sorts S contains a single sort and the set of equations B is empty) is equivalent to *quasi-decreasingness*, i.e., the existence of a well-founded partial ordering \succ on terms satisfying that: (1) the one-step rewriting relation is contained in \succ : $\rightarrow_{\mathcal{R}} \subseteq \succ$, (2) the strict subterm relation is contained in \succ : $\triangleright \subseteq \succ$, and (3) for every rule $\ell \rightarrow r \Leftarrow s_1 \rightarrow t_1, \dots, s_n \rightarrow t_n$, substitution σ , and index i , $0 \leq i < n$, if $\sigma(s_j) \rightarrow_{\mathcal{R}}^* \sigma(t_j)$ for every $1 \leq j \leq i$, then $\sigma(\ell) \succ \sigma(s_{i+1})$. In the following, we generalize this result to deterministic 3-OSRTs under the assumptions on B stated in Section 2. We use *strong* operational termination and operational termination as synonymous. This is done to distinguish it from a notion of *weak* operational termination presented later. Now we address the problem of defining appropriate orderings for dealing with order-sorted terms and rewrite theories.

3.1 Orderings for Order-Sorted Terms

A strict ordering \succ_s on terms of sort s is an irreflexive and transitive binary relation on $\mathcal{T}_\Sigma(\mathcal{X})_s$. A strict ordering $\succ_{[s]}$ on terms of sort in the connected component $[s]$ (of S/\equiv_{\leq}) is an irreflexive and transitive binary relation on $\mathcal{T}_\Sigma(\mathcal{X})_{[s]}$.

Remark 1. Order-sorted rewriting proceeds by transforming terms of the same connected component $[s] \in S/\equiv_{\leq}$. Therefore, orderings $\succ_{[s]}$ indexed by con-

nected components of sorts, rather than by sorts, are more appropriate for compatibility with the order-sorted rewrite relation. Indeed, note that $\rightarrow_{\mathcal{R}}^+ = (\rightarrow_{\mathcal{R}[s]}^+)$ is a well-founded S -ordering if the one-step rewrite relation is terminating, and that it is monotonic if \mathcal{R} is sort-decreasing. On the other hand, we can always obtain an ordering \succ_s on terms of sort s as follows: $\succ_s = \succ_{[s]} \cap \mathcal{T}_{\Sigma}(\mathcal{X})_s^2$.

A strict S -ordering $\succ_S = \{\succ_{[s]}\}_{[s] \in S/\equiv_{\leq}}$ is an S -sorted strict ordering on $\mathcal{T}_{\Sigma}(\mathcal{X})$, i.e., given terms $u, v \in \mathcal{T}_{\Sigma}(\mathcal{X})$, $u \succ_S v$ if and only if $u, v \in \mathcal{T}(\mathcal{F}, \mathcal{X})_{[s]}$ for some $[s] \in S/\equiv_{\leq}$ and $u \succ_{[s]} v$. An S -ordering \succ_S is: *well-founded* if its components $\succ_{[s]}$ are well-founded for all $s \in S$; *stable* if for all S -sorted substitution σ , $s \in S$, and terms $u, v \in \mathcal{T}_{\Sigma}(\mathcal{X})_{[s]}$ $u \succ_{[s]} v$, then $\sigma(u) \succ_{[s]} \sigma(v)$; *monotonic* if for all $f : s_1 \cdots s_k \rightarrow s \in \Sigma$ and terms $u_i, v_i \in \mathcal{T}_{\Sigma}(\mathcal{X})_{[s_i]}$ for $1 \leq i \leq k$, if $u_i \succ_{[s_i]} v_i$, then $f(u_1, \dots, u_i, \dots, u_k) \succ_{[s]} f(u_1, \dots, v_i, \dots, u_k)$. An S -ordering \succ_S on $\mathcal{T}_{\Sigma}(\mathcal{X})$ is compatible with a set of equations B on $\mathcal{T}_{\Sigma}(\mathcal{X})$ if for all terms u, u', v , whenever $u \succ_S v$ and $u' =_B u$, we have $u' \succ_S v$ (in short: $=_B \circ \succ \subseteq \succ$). The previous definitions generalize to arbitrary relations (quasi-orderings \succsim , equivalences \approx , etc.) on order-sorted terms.

Remark 2. S -sorted orderings cannot compare terms in different connected components. Still, S -sorted orderings are the natural ones when comparing the left- and right-hand sides of the rules of an order-sorted (conditional) rewrite system.

A term ordering \succ is a strict order on $\mathcal{T}_{\Sigma}(\mathcal{X})$. An S -sorted ordering \succ_S on $\mathcal{T}_{\Sigma}(\mathcal{X})$ defines a term ordering on $\mathcal{T}_{\Sigma}(\mathcal{X})$: $u \succ v$ iff $\exists [s] \in S/\equiv_{\leq}$ such that $u \succ_{[s]} v$. A term ordering which is *not* S -sorted is the subterm relation \supseteq : $\forall u, v \in \mathcal{T}_{\Sigma}(\mathcal{X})$, $u \supseteq v$ if either $u = v$ or $u = f(u_1, \dots, u_k)$ for some $f : s_1 \cdots s_k \rightarrow s \in \Sigma$ and $u_i \supseteq v$ for some i , $1 \leq i \leq k$. We write $u \triangleright v$ if $u \supseteq v$ and $u \neq v$.

3.2 Quasi-Decreasingness and (Strong) Operational Termination of deterministic 3-OSRTs

After the previous discussion, we can provide a generalization to deterministic 3-OSRTs of the usual notion of quasi-decreasingness for deterministic 3-CTRSs.

Definition 1 (Quasi-decreasingness). *A deterministic 3-OSRT (Σ, B, R) is quasi-decreasing if there is a well-founded term ordering \succ on $\mathcal{T}_{\Sigma}(\mathcal{X})$ satisfying: (1) $\rightarrow_{\mathcal{R}} \subseteq \succ$, (2) $=_B \circ \succ \subseteq \succ$, (3) $\triangleright \subseteq \succ$, and (4) for every rule $l \rightarrow r \leftarrow u_1 \rightarrow v_1, \dots, u_n \rightarrow v_n$, S -sorted substitution σ , and index i , $0 \leq i < n$, if $\sigma(u_j) \rightarrow_{\mathcal{R}}^* \sigma(v_j)$ for every $1 \leq j \leq i$, then $\sigma(l) \succ \sigma(s_{i+1})$.*

Quasi-decreasingness is a sufficient condition for operational termination of deterministic 3-OSRTs.

Theorem 1. *Let \mathcal{R} be a deterministic 3-OSRT. If \mathcal{R} is quasi-decreasing, then it is operationally terminating.*

Quasi-decreasingness is also necessary for operational termination of order-sorted and *sort-decreasing* rewrite theories. Due to our assumption that the equations B are sort-preserving and the B -coherence assumption, sort-decreasingness is stable under B -equivalence classes.

Remark 3. Our definition of sort-decreasing conditional rule does *not* impose anything to the conditional part of the rules. In this paper, we need sort-decreasingness to ensure *monotonicity* of conditional rewriting (see Proposition 1). This holds without any further restriction on the conditions of the rules.

Thanks to the stability of sort-decreasing rules under B -equality ensured by the assumptions on B we then have:

Proposition 1. [9] *Let \mathcal{R} be a sort-decreasing OSRT, $t, u, v \in \mathcal{T}_\Sigma(\mathcal{X})$ and $p \in \text{Pos}(t)$. If $t = t[u]_p$ and $u \rightarrow v$, then $t[u]_p \rightarrow t[v]_p$.*

Without sort-decreasingness, this important result does not hold (see [9]). This assumption is essential in our proof of the following result.

Theorem 2. *Let \mathcal{R} be a sort-decreasing deterministic 3-OSRT. If \mathcal{R} is operationally terminating, then it is quasi-decreasing.*

Thus, quasi-decreasingness characterizes operational termination of order-sorted, sort-decreasing rewrite theories.

Corollary 1. *A sort-decreasing deterministic 3-OSRT \mathcal{R} is operationally terminating if and only if it is quasi-decreasing.*

4 Computing with Normal Rewrite Theories

Definition 2 (Irreducible forms and weak termination). *Let \mathcal{R} be an OSRT and s, t be terms. We say that t is irreducible if, for any term u , $t \not\rightarrow_{\mathcal{R}} u$. $\text{Irr}(\mathcal{R})$ (resp. $\text{GIrr}(\mathcal{R})$) is the set of irreducible terms (resp. ground terms) of \mathcal{R} .*

If s rewrites to an irreducible term t , we say that s has a (not necessarily unique) irreducible form t , denoted $s \rightarrow t$. If every term s has an irreducible form, i.e., $s \rightarrow t$ for some irreducible term t , then \mathcal{R} is called weakly terminating.

Terminating OSRTs are weakly terminating (in general, the opposite is not true).

Definition 3 (Normal form, weak normalization). *A term t is called a normal form if it is irreducible and operationally 1-terminating. Let $\text{NF}(\mathcal{R})$ (resp. $\text{GNF}(\mathcal{R})$) be the set of normal forms (resp. ground normal forms) of \mathcal{R} .*

If $s \rightarrow t$ and t is a normal form, we then write $s \rightarrow^! t$ and call t a normal form of s . If every (ground) term s has a normal form, i.e., $s \rightarrow^! t$ for some normal form t , then \mathcal{R} is called weakly (ground) operationally terminating (or weakly (ground) normalizing).

Remark 4 (Notation). If \mathcal{R} is confluent and weakly operationally terminating, then we write $t \rightarrow_{\mathcal{R}}^! u$ for $t \rightarrow_{\mathcal{R}} u$, denote such a u by $u = t!_{\mathcal{R}}$ or $u = \text{can}_{\mathcal{R}}(t)$, and call it the \mathcal{R} -canonical form of t which is unique up to B -equality.

Note that $\rightarrow_{\mathcal{R}/B} \supseteq \rightarrow_{\mathcal{R}/B}^!$ and $\text{NF}(\mathcal{R}) \subseteq \text{lrr}(\mathcal{R})$ (this inclusion can be strict!).

Example 1. The one-step rewrite relation for $a \rightarrow b \Leftarrow a \rightarrow c$ (a single rule OSRT) is empty. Hence, a is irreducible. However, a is *not* a normal form: every attempt to prove a reduction step on a starts an infinite proof tree.

There can also be *reducible* terms that are *not* operationally 1-terminating.

Example 2. Term $f(a)$ is not operationally 1-terminating in the 2-CTRS \mathcal{R} :

$$g(a) \rightarrow c(b) \tag{1}$$

$$b \rightarrow f(a) \tag{2}$$

$$f(x) \rightarrow x \Leftarrow g(x) \rightarrow c(y) \tag{3}$$

Since $g(a) \rightarrow c(b)$, we have $f(a) \rightarrow a$ by means of a finite proof tree. However, since the evaluation of the condition could *continue* beyond $c(b)$

$$g(a) \rightarrow c(\underline{b}) \rightarrow c(f(a))$$

and the term $f(a)$ can start a new (deep) proof tree, we also have an infinite (well-formed) proof tree for the goal $f(a) \rightarrow u$ with u arbitrary.

Remark 5. Note that \mathcal{R} in Example 2 is *terminating*, i.e., there is no infinite rewrite sequence $t_1 \rightarrow_{\mathcal{R}} t_2 \rightarrow_{\mathcal{R}} \dots$. This is easy to see, because the *underlying* TRS $\mathcal{R}_u = \{\ell \rightarrow r \mid \ell \rightarrow r \Leftarrow c \in \mathcal{R}\}$ is clearly terminating.

Definition 4 (Normal and strongly deterministic rewrite theory). A deterministic OSRT \mathcal{R} is called normal (resp. ground normal) if the set $\text{lrr}(\mathcal{R})$ (resp. the set $\text{Glrr}(\mathcal{R})$) is operationally terminating, i.e., every irreducible (ground) term is a (ground) normal form: $\text{lrr}(\mathcal{R}) = \text{NF}(\mathcal{R})$ (resp. $\text{Glrr}(\mathcal{R}) = \text{GNF}(\mathcal{R})$).

A normal OSRT $\mathcal{R} = (\Sigma, B, R)$ is called strongly deterministic if for each $\ell \rightarrow r \Leftarrow s_1 \rightarrow t_1, \dots, s_n \rightarrow t_n$ in R , and each substitution θ such that $\theta(x) \in \text{NF}(\mathcal{R})$ for each $x \in \mathcal{X}$, we have: $\theta(t_1), \dots, \theta(t_n) \in \text{NF}(\mathcal{R})$.

The B -coherence assumption then gives us:

Proposition 2. *If a strongly deterministic 3-OSRT \mathcal{R} is (ground) confluent and weakly normalizing, then \mathcal{R} is (ground) normal.*

Remark 6. Ground normality is the minimum prerequisite for executability. For ground normal and ground confluent deterministic 3-OSRT \mathcal{R} , each ground term t has *at most* one normal form up to B -equality and the process $t \mapsto [t!_{\mathcal{R}}]_B$ defines a recursive partial function, since \mathcal{R} need not even be weakly terminating.

In order to prove that a strongly deterministic OSRT $\mathcal{R} = (\Sigma, B, R)$ is ground normal, we can proceed as follows:

1. Identify a subsignature of constructors Ω with nonempty sorts such that the rules in R decompose as a disjoint union $R_{(\Sigma-\Omega)} \cup R_\Omega$, where the R_Ω have only Ω terms in their rules and conditions, and each $\ell \rightarrow r \Leftarrow s_1 \rightarrow t_1, \dots, s_n \rightarrow t_n$ in $R_{(\Sigma-\Omega)}$ has $\ell = f(t_1, \dots, t_n)$ for some $f \in \Sigma - \Omega$. We also assume that the axioms B decompose as a disjoint union $B_{(\Sigma-\Omega)} \cup B_\Omega$ with the B_Ω involving only Ω terms, and the $B_{(\Sigma-\Omega)}$ *not* Ω -equations. This yields an ORST inclusion $\mathcal{R}_\Omega \subseteq \mathcal{R}$, with $\mathcal{R}_\Omega = (\Omega, B_\Omega, R_\Omega)$.
2. Prove (by inductive theorem proving) that for all defined symbols $f \in \Sigma - \Omega$, say with rank $f : s_1 \cdots s_n \rightarrow s$, the following inductive property holds:

$$\forall x_1 \in \mathcal{T}_{\Omega_{s_1}}, \dots, x_n \in \mathcal{T}_{\Omega_{s_n}}, \exists y f(x_1, \dots, x_n) \rightarrow_{\mathcal{R}}^1 y$$

Then if \mathcal{R}_Ω is *operationally terminating*, \mathcal{R} is *ground normal* and, furthermore, $\text{GNF}(\mathcal{R}) \subseteq \mathcal{T}_\Omega$. That is, an inductive proof of ground reducibility w.r.t. the constructors shows that $t \in \mathcal{T}_\Sigma$ is a ground normal form iff:

1. $t \in \mathcal{T}_\Omega$; and
2. $t \in \text{GNF}(\mathcal{R}_\Omega)$.

The assumptions on B give us:

Proposition 3. *Let $\mathcal{R} = (\Sigma, B, R)$ be a normal, sort-decreasing, confluent, strongly deterministic 3-OSRT such that R is finite. If \mathcal{R} is weakly operationally terminating, then the function $t \mapsto [t]_{\mathcal{R}}!$ is total recursive and preserves sorts.*

Note that, otherwise, if \mathcal{R} is confluent but *not* weakly operationally terminating, then the function $t \mapsto [u]_B$ with $t \rightarrow u$ may not be recursive, even if each t has an *irreducible* form. Implicit in Proposition 3 is the fact that, under such conditions plus the assumptions on B , when we interpret each $\ell \rightarrow r \Leftarrow s_1 \rightarrow t_1, \dots, s_n \rightarrow t_n$ in R as a *conditional equation* $\ell = r \Leftarrow s_1 = t_1, \dots, s_n = t_n$, normal forms define an *algebra* $\mathcal{C}_{\Sigma/R,B}$, called the *canonical term algebra* of \mathcal{R} . Specifically, for each sort s we define $\mathcal{C}_{\Sigma/R,B,s} = \text{GNF}(\mathcal{R})/_B \cap \mathcal{T}_{\Sigma/B}$, that is, the set of B -equivalence classes of ground normal forms of sort s , and, for each $f : s_1 \cdots s_n \rightarrow s$ in Σ its interpretation in $\mathcal{C}_{\Sigma/R,B}$ maps each tuple $([t_1]_B, \dots, [t_n]_B)$ with $[t_i]_B \in \mathcal{C}_{\Sigma/R,B,s_i}$ to the B -equivalence class $[f(t_1, \dots, t_n)]_{\mathcal{R}}!$, which is well-defined and unique because of confluence, sort-decreasingness and B -coherence. The agreement between the operational semantics of \mathcal{R} when terms are normalized by rewriting, and the mathematical semantics of \mathcal{R} when its rules are interpreted as conditional equations can then be expressed for such general OSRTs as follows:

Corollary 2. *For $\mathcal{R} = (\Sigma, B, R)$ a sort-decreasing, ground confluent and weakly ground operationally terminating strongly deterministic 3-OSRT, the canonical term algebra $\mathcal{C}_{\Sigma/\mathcal{R}}$ is a computable algebra. Furthermore, $\mathcal{T}_{\Sigma/R \cup B} \simeq \mathcal{C}_{\Sigma/R,B}$.*

Computing the normal form $t!_{\mathcal{R}}$ of a term t under the assumptions of Corollary 2 is somewhat complex, and can be computationally expensive. It is therefore useful to seek conditions under which we can more efficiently compute normal forms. We consider two such conditions, which can be executed in Maude in a straightforward manner.

The first case is that of a strongly deterministic 3-OSRT that is sort-decreasing, ground confluent, 1-terminating, and ground weakly terminating and has a finite number of rewrite rules. Under such conditions, the `search` command in Maude asking for the fully-reduced first result for the given input ground term will compute such a normal form. This assumes that the rules in the theory are expressed as *rules* in a Maude system module and *not as equations* in a functional module, even though the module does indeed have a *functional semantics*. A simple theory transformation, easily definable in Maude's `META-LEVEL` module, can transform the given functional module into its associated system module. Let us illustrate this general method with an example. Note that in this example the set B of axioms is empty. The functional module `fmod WEAK-NORM endfm` expresses the rewrite rules R as conditional equations, whereas the system module `mod WEAK-NORM endm` expresses them explicitly as rewrite rules.

```
fmod WEAK-NORM is
  protecting BOOL .
  sorts Nat Nat? .
  subsort Nat < Nat? .
  op 0 : -> Nat .           op s : Nat -> Nat .
  op _+_ : Nat Nat -> Nat . op even : Nat -> Bool .
  ops f g : Nat? -> Nat? .
  vars N M : Nat .
  eq N + 0 = N .           eq N + s(M) = s(N + M) .
  eq even(0) = true .     eq even(s(0)) = false .
  eq even(s(s(N))) = even(N) . eq g(N) = N .
  eq f(N) = N + N .
  ceq f(N) = g(f(N)) if true := even(N) .
endfm
```

This module is sort-decreasing, weakly terminating and ground confluent. By the technique presented in Section 5, we can prove it normal. Giving to Maude the term `f(0)` for evaluation leads to non-terminating behavior. That is, the usual operational semantics for evaluating operationally terminating confluent theories cannot be relied upon to compute normal forms. This problem can be solved by transforming the above functional module into a system module, that is, by transforming equations into rules, and using Maude's `search` command:

```
mod WEAK-NORM is
  protecting BOOL .
  sorts Nat Nat? .
  subsort Nat < Nat? .
  op 0 : -> Nat .           op s : Nat -> Nat .
  op _+_ : Nat Nat -> Nat . op even : Nat -> Bool .
  ops f g : Nat? -> Nat? .
  vars N M : Nat .
  rl N + 0 => N .           rl N + s(M) => s(N + M) .
  rl even(0) => true .     rl even(s(0)) => false .
  rl even(s(s(N))) => even(N) . rl g(N) => N .
  rl f(N) => N + N .
```

```

    crl f(N) => g(f(N)) if even(N) => true .
endm

```

The normal form of a term can then be obtained by searching for the *first* result of a *terminating* computation from the given term. By confluence such a result is unique up to B -equality, exists by weak operational termination, and can be found by search without risk of looping thanks to 1-termination:

```

Maude> search [1] f(0) =>! N:Nat .
search in WEAK-NORM : f(0) =>! N .
Solution 1 (state 5)
states: 9 rewrites: 12 in 0ms cpu (0ms real) (44943 rewrites/second)
N --> 0
Maude> search [1] f(s(s(0))) =>! N:Nat .
search in WEAK-NORM : f(s(s(0))) =>! N .
Solution 1 (state 14)
states: 20 rewrites: 35 in 0ms cpu (0ms real) (55118 rewrites/second)
N --> s(s(s(0)))
Maude> search [1] f(s(s(s(s(0)))) =>! N:Nat .
search in WEAK-NORM : f(s(s(s(s(0)))) =>! N .
Solution 1 (state 27)
states: 35 rewrites: 70 in 1ms cpu (1ms real) (57189 rewrites/second)
N --> s(s(s(s(s(s(0))))))

```

The second case where execution of a weakly operationally terminating deterministic OSRT can be achieved using execution mechanisms already available in Maude and yields again a full agreement between operational and mathematical semantics is the one of context-sensitive OSRTs under some reasonable assumptions. A *context-sensitive* [6] OSRT is a four-tuple $\mathcal{R} = (\Sigma, B, R, \mu)$, where (Σ, B, R) is an OSRT, and μ maps each $f : s_1 \cdots s_n \rightarrow s$ in Σ to a subset $\mu(f) \subseteq \{1, \dots, n\}$ of the argument positions of f under which rewriting is allowed. The operational semantics of context-sensitive OSRTs is defined by restricting the inference system of Figure 1 with the single restriction that, in the (Cong) Rule, i with $1 \leq i \leq k$ must furthermore satisfy $i \in \mu(f)$.

The Lemma below states the required conditions on $\mathcal{R} = (\Sigma, B, R, \mu)$ yielding the desired agreement between operational and mathematical semantics. This result relies on the notion of μ -sufficient completeness and of the algebra $\mathcal{C}_{\mathcal{R}}^{\mu}$ of term in μ -normal form (see [5]).

Lemma 1. *If \mathcal{R} is a confluent, sort decreasing and strongly deterministic context-sensitive 3-OSRT $\mathcal{R} = (\Sigma, B, R, \mu)$, which is μ -operationally terminating and μ -sufficiently complete for $\Omega \subseteq \Sigma$ a subsignature of free constructors modulo B , then:*

1. \mathcal{R} is ground weakly operationally terminating.
2. $\mathcal{C}_{\mathcal{R}}^{\mu} |_{\Omega} = \mathcal{T}_{\Omega/B}$.
3. For each $t \in \mathcal{T}_{\Sigma}$, $t!_{\mathcal{R},B} = t!_{\mathcal{R},B}^{\mu}$, that is, the normal form and the μ -normal form of t (which can be computed by Maude's reduce command) coincide.
4. $\mathcal{T}_{\Sigma/E \cup B} \simeq \mathcal{C}_{E/B}^{\mu}$ (agreement between operational and denotational semantics).

Under the assumptions of Lemma 1, we compute normal forms as follows: since Maude supports the execution of confluent context-sensitive 3-OSRTs $\mathcal{R} = (\Sigma, B, R, \mu)$ specified as functional modules, we just use the `reduce` command to compute normal μ -forms, which under the assumptions in Lemma 1 are also ordinary normal forms in the underlying OSRT (Σ, B, R) . We can illustrate these ideas with the following example of a context-sensitive 3-OSRT in Maude:

```
fmod FACTORIAL is
  protecting NAT .
  op monus : Nat Nat -> Nat .
  op _~_ : Nat Nat -> Bool [comm] .
  op [_,-,-] : Bool Nat Nat -> Nat [strat (1 0)] .
  op fact : Nat -> Nat .
  vars N M : Nat .
  eq monus(s(N),s(M)) = monus(N,M) .
  ceq monus(N,M) = N if M := 0 .
  ceq monus(N,M) = 0 if N := 0 .
  eq N ~ N = true .
  eq s(N) ~ s(M) = N ~ M .
  eq 0 ~ s(N) = false .
  eq [true,N,M] = N .
  eq [false,N,M] = M .
  eq fact(N) = [(N ~ 0),s(0),N * fact(monus(N,s(0)))] .
endfm
```

This theory, though ground confluent, is clearly non-terminating because of the last equation. Here, μ does not restrict any argument positions, except for the if-then-else operator $[-,-,-]$, where $\mu([-,-,-]) = \{1\}$, as specified by the `strat` attribute. It is, however, operationally μ -terminating and has 0 and `s`, and `true`, `false` as free constructors. Here are some evaluations:

```
Maude> red fact(2) .
reduce in FACTORIAL : fact(2) .
rewrites: 15 in 0ms cpu (0ms real) (192307 rewrites/second)
result NzNat: 2
Maude> red fact(3) .
reduce in FACTORIAL : fact(3) .
rewrites: 21 in 0ms cpu (0ms real) (10500000 rewrites/second)
result NzNat: 6
Maude> red fact(4) .
reduce in FACTORIAL : fact(4) .
rewrites: 27 in 0ms cpu (0ms real) (692307 rewrites/second)
result NzNat: 24
Maude> red fact(5) .
reduce in FACTORIAL : fact(5) .
rewrites: 33 in 0ms cpu (0ms real) (358695 rewrites/second)
result NzNat: 120
```

We end this section with the following result that, though well-known (see, e.g., [13]), has an easier proof with a rewrite theory with axioms B of associativity

and identity for strings. In some sense this result shows how wild the beasts in the general menagerie of OSRTs can be, and illustrates the need for notions such as that of normal theory to obtain reasonable computational behaviors.

Theorem 3. *There is a 2-OSRTs \mathcal{R} and a sort s such that the set $\text{lrr}(\mathcal{R})_s \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})_s$ of \mathcal{R} -irreducible terms is not recursively enumerable, so it is not even semi-decidable if a term is \mathcal{R} -irreducible.*

5 Proving Order-Sorted Rewrite Theories Normal

1-operationally terminating rewrite theories are *normal*. The opposite is not true.

Example 3. The CTRS \mathcal{R} in Example 2 is *not* 1-operationally terminating. However, \mathcal{R} is normal: assume that there is a minimal irreducible term s having an infinite well-formed proof tree whose strict subterms are normal forms. Since f is the only symbol defined by a conditional rule, $s = f(t)$ for some normal form t . Since $f(t)$ is irreducible, the evaluation of the condition in the rule cannot succeed, i.e., $g(t)$ must be irreducible. Since t is a normal form, $g(t)$ cannot start any infinite well-formed tree. Contradiction.

Remark 7. As noticed in Remark 5, \mathcal{R} in Example 2 is terminating. Since \mathcal{R} is *not* 1-operationally terminating and a fortiori not operationally terminating, it follows from Example 3 that neither 1-operational termination nor operational termination of \mathcal{R} follow from the termination and normality of \mathcal{R} .

An interesting feature in the treatment of innermost termination problems using the dependency pair approach [1] is that, since the variables in the right-hand side of the dependency pairs are in normal form, the rules which can be used to connect contiguous dependency pairs are usually a proper subset of the rules in the TRS. This leads to the notion of *usable rules* [1, Definition 32] which simplifies the proofs of innermost termination of rewriting.

In our analysis of normal rewrite theories we have a similar situation: when an irreducible term $t = f(t_1, \dots, t_k)$ is tested to see whether it is a normal form, we know that all possible reductions derived from a proof $t \rightarrow x$ (for a fresh variable x) cause the evaluation of the conditional part c of some conditional rule $f(\ell_1, \dots, \ell_k) \rightarrow r \leftarrow c$. Therefore, if we single out those rules that *can be* involved in any attempt to evaluate $\sigma(c)$ for some σ such that $t = \sigma(f(\ell_1, \dots, \ell_k))$, we can obtain a more precise analysis. The notion of usable rule provides an upper, purely syntactic, approximation to the set of rules that eventually apply to a term t during any possible rewriting on t . We keep the original flavor of the original, unsorted notion in the following definition.

Definition 5 (Usable rules for a rewrite theory). *Let $\mathcal{R} = (\Sigma, B, R)$ be an OSRT. Let $\text{RULES}(\mathcal{R}, t)$ be the set of rules defining symbols occurring in t :*

$$\text{RULES}(\mathcal{R}, t) = \{\ell \rightarrow r \leftarrow c \in R \mid \exists p \in \text{Pos}(t), \text{root}(\ell) = \text{root}(t|_p)\}$$

Then, the set of usable rules of \mathcal{R} for t is:

$$\mathcal{U}(\mathcal{R}, t) = \text{RULES}(\mathcal{R}, t) \cup \bigcup_{l \rightarrow r \leftarrow c \in \text{RULES}(\mathcal{R}, t)} \mathcal{U}(\mathcal{R}', r) \cup \bigcup_{s_i \rightarrow t_i \in c} \mathcal{U}(\mathcal{R}', s_i)$$

where $\mathcal{R}' = \mathcal{R} - \text{RULES}(\mathcal{R}, t)$.

That is: we consider both unconditional and conditional rules and add the rules that could be *used* to evaluate the conditions in the rules. Since we are dealing with OSRTs $\mathcal{R} = (\Sigma, B, R)$, rewriting happens *modulo* B . This raises the issue of whether the above definition of usable rules is overly syntactic, that is, not stable under B -equality. The key issue is whether in the (Repl) rule in the inference system of Figure 1 the top symbol of the redex u coincides with that of the lefthand side l . This is the case by requiring the axioms B to be as follows:

$$B = \bigcup_{f: [s_1] \cdots [s_n] \rightarrow [s] \in \Sigma} B_f$$

where B_f is a set of equations $u = v$ with $u, v \in \mathcal{T}_{\{f\}}(\mathcal{X}) - \mathcal{X}$, i.e., only symbol f is allowed (and must) to occur in the equations belonging to B_f . Associativity and commutativity axioms satisfy this requirement, which can even be made to work for identity axioms by performing the semantics-preserving transformation described in [3]. Now we can give the main result of this section. For an OSRT $\mathcal{R} = (\Sigma, B, R)$, we say that B *preserves* the \mathcal{R} -normal forms if for all \mathcal{R} -normal forms t , if $t =_B u$, then u is an \mathcal{R} -normal form. *B-coherence*, which is a usual requirement for working OSRTs, implies this property. By \mathcal{R}_C we denote the OSRT obtained as the union of $\mathcal{U}(\mathcal{R}, s)$ for all *lhs*'s conditions in the rules of \mathcal{R} :

$$\mathcal{R}_C = \bigcup_{l \rightarrow r \leftarrow c} \bigcup_{s \rightarrow t \in c} \mathcal{U}(\mathcal{R}, s)$$

Theorem 4. *A deterministic 3-OSRT $\mathcal{R} = (\Sigma, B, R)$ is normal if B preserves the \mathcal{R} -normal forms and \mathcal{R}_C is operationally terminating.*

Example 4. Consider the functional module WEAK-NORM in Section 4. Here, \mathcal{R}_C is the unconditional subOSRT consisting of the rules defining **even**. Note that \mathcal{R}_C has no conditional rule and is clearly terminating, hence operationally terminating. We conclude that, as claimed, WEAK-NORM is a normal OSRT.

Now we show that Theorem 4 does *not* characterize normality of OSRTs:

Example 5. Consider the following deterministic 1-CTRS:

$$\begin{array}{l} a \rightarrow b \quad f(x) \rightarrow x \leftarrow c \rightarrow d, a \rightarrow c \\ b \rightarrow a \end{array}$$

Every term $f(t)$ is *irreducible* and also a normal form because the unsatisfiable condition $c \rightarrow d$ prevents the looping evaluation of the condition $a \rightarrow c$. However, $\mathcal{R}_C = \{a \rightarrow b, b \rightarrow a\}$ is not (operationally) terminating.

6 Conclusions and Future Work

The results presented in this paper can be viewed from two complementary perspectives: one more theoretical, and another more practical. At the theoretical level, we have investigated parts of the *terra incognita* of conditional term rewriting by asking and providing precise answers to innocent-sounding questions such as: what is a normal form? How can normal forms be effectively computed? How should the notion of weakly normalizing system be understood in the conditional case? How can good executability properties be ensured for a theory? There is, however, a more practical aspect to all these results. It consists in taking to heart the idea that rewrite theories are an excellent framework for declarative programming and formal specification and verification. From this second perspective, the questions asked and answered include: what is the most general notion possible of a conditional rule-based program for which normal forms can be computed? What is the appropriate term normalization operational semantics? How can it be made more efficient? What are the most general possible requirements under which conditional functional programs can be given an initial algebra semantics which *fully agrees* with their operational semantics?

Future work should further investigate proof methods and supporting tools for all the properties discussed here. For example, although the characterization of the operation termination of an OSRT in terms of quasi-decreasingness offers in principle a complete proof method, we are currently investigating a far-reaching generalization to the conditional case of the dependency pair method that seems considerably more effective for mechanizing actual proofs. In general, the development of *intrinsic methods* for proving both strong and weak operational termination of OSRTs seems both quite attractive and sorely needed.

With regard to checking normality of OSRTs, Example 5 shows that the notion of operational termination of OSRTs is perhaps too strong to capture normality of some OSRTs. In [8] we have introduced the *weaker* notion of V -termination for CTRSs, which captures the absence of infinite computations involving an infinite number of failed attempts to issue a *single* rewriting step (which we associate to a *vertical* dimension of nontermination in [8]). The definition of V -termination of CTRSs is based on the Dependency Pairs for CTRSs (2D DPs) described in [8]. The interesting feature is that V -termination can be *independently* proved in the 2D DP Framework. Unfortunately, V -termination does not yield a valid criterion to prove CTRSs normal.

Example 6. The following variant \mathcal{R}' of \mathcal{R} in Example 5

$$\begin{array}{l} a \rightarrow b \qquad f(x) \rightarrow x \Leftarrow a \rightarrow c \\ b \rightarrow a \end{array}$$

is *not* normal because terms $f(t)$ are *irreducible* (since $a \rightarrow^* c$ cannot be satisfied), but they are *not* normal forms because $f(t) \rightarrow u$ starts an infinite well-formed tree (i.e., \mathcal{R}' is *not* 1-terminating). However, by using the methods in [8] it is easy to prove that \mathcal{R}' is V -terminating.

Since 1-termination implies V -termination, the following hierarchy of properties

$$\begin{array}{c} \text{Operational termination} \Rightarrow 1\text{-termination} \Rightarrow V\text{-termination} \\ \Downarrow \\ \text{Normal theory} \end{array}$$

suggests now the development of techniques for proving 1-termination as an important topic for further research.

References

1. T. Arts and J. Giesl. Termination of Term Rewriting Using Dependency Pairs. *Theoretical Computer Science*, 236(1–2):133–178, 2000.
2. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. All About Maude – A High-Performance Logical Framework. LNCS 4350, 2007.
3. F. Durán, S. Lucas, J. Meseguer, Termination Modulo Combinations of Equational Theories, In *Proc of FroCoS'09*, LNAI 5749:246–262, Springer-Verlag, 2009.
4. J. Goguen and J. Meseguer. Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. *Theoretical Computer Science*, 105:217–273, 1992.
5. J. Hendrix and J. Meseguer. On the Completeness of Context-Sensitive Order-Sorted Specifications In *Proc. of RTA'07*, LNCS 4533:229-245, Springer-Verlag, 2007.
6. S. Lucas. Context-Sensitive Computations in Functional and Functional Logic Programs. *Journal of Functional and Logic Programming*, 1998(1):1–61, 1998.
7. S. Lucas, C. Marché, and J. Meseguer. Operational termination of conditional term rewriting systems. *Information Processing Letters*, 95:446–453, 2005.
8. S. Lucas and J. Meseguer. 2D Dependency Pairs for Proving Operational Termination of CTRSs. In S. Escobar, editor, *Proc. of the 10th International Workshop on Rewriting Logic and its Applications, WRLA'14, this volume* of LNCS, 2014.
9. S. Lucas and J. Meseguer. Order-Sorted Dependency Pairs. In *Proc. of PPDP'08*, pp. 108-119, ACM Press, 2008.
10. J. Meseguer. Membership algebra as a logical framework for equational specification. In *Proc. of WADT'97*, LNCS 1376:18–61. Springer-Verlag, 1998.
11. J. Meseguer. On Coherence. *To appear* 2014.
12. E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer-Verlag, Apr. 2002.
13. TeReSe, Term Rewriting Systems, Cambridge University Press, 2003.