DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN
UNIVERSIDAD POLITÉCNICA DE VALENCIA

P.O. Box: 22012 E-46071 Valencia (SPAIN)

Informe Técnico / Technical Report

Ref. No.:	DSIC-II/01/05	**Pages:** 10
Title:	Operational Termination of Conditional Term Rewriting Systems	
Author(s):	Salvador Lucas, Claude Marché, and José Meseguer	
Date:	February 11, 2005	
Keywords:	Program analysis, term rewriting, termination.	

V^o B^o
Leader of research Group

Author(s)

Operational Termination of Conditional Term Rewriting Systems *

Salvador Lucas† Claude Marché‡ José Meseguer§

1 Introduction

Conditional Term Rewriting Systems (CTRSs [5, 11]) play a fundamental role in algebraic specification of abstract data types and in rule-based programming languages such as Elan [1], Maude [4], OBJ [10], CafeOBJ [7], Haskell [9], etc. The operational meaning of specifications in such sophisticated languages, which support advanced features such as evaluation strategies, rewriting modulo, use of extra variables in conditions, partiality, higher-order, and expressive type systems is often formalized in a proof-theoretic style by means of an inference system (see, e.g., [2, 3, 14]) instead than just by a rewriting relation. Thus, the corresponding language interpreters should be better understood as *inference* machines instead than just as *rewriting* engines.

Current notions of CTRS termination do *not* fit this framework: consider the single conditional rule [15, Section 7.2.1],

$$a \to b \text{ if } f(a) \to b$$

Since we have no unconditional rule, the relation $\to$ is trivially empty, hence well-founded, i.e., no infinite rewriting sequence with $\to$ is possible. Moreover, as remarked by Ohlebusch, it is also *effectively terminating* in the sense of Marchiori & Ohlebusch, discussed in Section 4 below. Although effective termination is intended to provide a notion of termination which (in Marchiori's words) implies *effective computability for CTRSs* [13, Section 5.1], an implementation will typically loop[1] when trying to reduce a (!). So what is the *right* notion of termination for CTRSs?

Our proposal in this paper (explained in Section 2) is to describe conditional rewriting by means of an inference system and capture termination as the *absence of infinite inference*: that is, all proof attempts must either successfully terminate, or they must fail in finite time. We call this notion *operational termination*. Our notion of operational termination is *parametric* on the inference system. In our view this new notion has two key advantages:

*This research was partly supported by bilateral CNRS-DSTIC/UIUC research project "Rewriting calculi, logic and behavior", and by ONR Grant N00014-02-1-0715 and NSF Grant CCR-0234524; Salvador Lucas was partially supported by Spanish MEC grant SELF TIN 2004-07943-C04-02.

†DSIC, Universidad Politécnica de Valencia, Spain, slucas@dsic.upv.es

‡PCRI, LRI (CNRS UMR 8623), INRIA Futurs, Université Paris-Sud, France, Claude.Marche@lri.fr

§CS Dept., University of Illinois at Urbana-Champaign, USA, meseguer@cs.uiuc.edu

[1] We have tried versions of this CTRS both in Maude and Haskell implementations.

1. It corresponds to what interpreters actually do. Thus, it is a sound abstract notion of termination of CTRSs which captures what the users of such systems get. As shown by the simple example above, the notions of well-foundedness or of 'effective termination' fail to capture actual termination.

2. On the other hand, by being parametric in the inference system, it extends naturally to settings in which new features are added, or even to settings such as Horn logic without equality where the rewriting relation may be absent [2, 3, 14]. By contrast, existing CTRS termination notions do not seem to be easily extensible to such general settings.

In Section 3, we specialize this new notion to CTRSs thus leading to the notion of operational termination of CTRSs. We show that operational termination of TRSs (where no conditional rules are present) coincide with the usual well-foundedness notion of termination of TRSs. In Section 4, we compare our new notion with already existing notions of termination of CTRSs, which were intended to provide suitable approximations to effective termination (which is undecidable). We prove that operational termination of CTRSs is, in fact, equivalent to a very general notion proposed for 3-CTRSs, namely the notion of *quasi-decreasingness* [15], which is currently the most general one which is intended to be checked by comparing parts of the CTRS by means of term orderings. Therefore, existing methods for proving quasi-decreasingness of CTRSs (see [15, Section 7.2]) immediately apply to prove operational termination of CTRSs. Another interesting aspect of this result is that it shows the agreement between two in principle quite different computational definitions of termination, one proof-theoretic, and the other based on an ordering and very general, so that both capture the same essential intuition in different ways. Moreover, since it is known that quasi-decreasingness implies effective termination [15, Theorem 7.2.42], operational termination also does. The converse does not hold, as shown by the CTRS above.

2 Operational termination

Our notion of operational termination is *parametric* on the inference system. We assume a logic $\mathcal{L}$ defined by inference rules. Given a logic $\mathcal{L}$, 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. Such provable theorems are exactly the formulas φ in the syntax of $\mathcal{S}$ for which we can derive a *closed* proof tree of the form

$$\frac{T_1 \ \ldots \ T_n}{\varphi}$$

using $\mathcal{L}$'s inference system. We make this more precise in the following:

Definition 1 *Let $\mathcal{S}$ be a theory in a logic $\mathcal{L}$ endowed with an inference system. Then, the set of (finite) proof trees for $\mathcal{S}$ in $\mathcal{L}$ and the head of a proof tree are defined inductively as follows. A proof tree is either*

- *an* open goal*, simply denoted as G, where G is a formula in $\mathcal{S}$. Then, we denote head$(G) = G$.*

- *a* derivation tree *with G as its head, denoted as*

$$\frac{T_1 \quad \cdots \quad T_n}{G}(\Delta)$$

where G is a formula in $\mathcal{S}$, Δ is a derivation rule in $\mathcal{L}$, and T_1,...,T_n are proofs trees such that

$$\frac{head(T_1) \quad \cdots \quad head(T_n)}{G}$$

is an instance of the rule Δ.

We say that a proof tree is closed *whenever it is finite and contains no open goals. Notice the difference between G, an open goal, and $\overline{G}$, a goal closed by a rule without premises.*

Definition 2 *A proof tree T is a* proper prefix *of a proof tree T' if there are one or more open goals $G_1, \ldots, G_n$ in T such T' is obtained from T by replacing each G_i by a derivation tree T_i having G_i as its head. We denote this as $T \subset T'$.*

An infinite proof tree *is an infinite increasing chain of finite trees, that is, a sequence $\{T_i\}_{i \in \mathbb{N}}$ such that for all i, $T_i \subset T_{i+1}$.*

We now 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. The interpreter however should have some reasonable *strategy*, since otherwise it could easily fail to find a proof because it postpones some proofs forever (see a concrete example in Section 3 below). We now characterize the proof trees with computational meaning, by means of the notion of well-formed proof tree.

Definition 3 (Well-formed proof tree) *We say that a proof tree T is* well-formed *if it is either an open goal, or a closed proof tree, or a derivation tree of the form*

$$\frac{T_1 \quad \cdots \quad T_n}{G}(\Delta)$$

where for each j T_j is itself well-formed, and there is $i \leq n$ such that T_i is not closed, for any $j < i$ T_j is closed, and each of the T_{i+1} ,...,T_n is an open goal. An infinite proof tree is well-formed *if it is an ascending chain of well-formed finite proof trees.*

The above definition of well-formed proof tree tries to capture in a formal way the operational behavior of an interpreter: this is expressed by the left-to-right way of chosing goals in the proof tree, so that, if the interpreter has not finished its computation, it has always a *leftmost goal* that it is working on.

Definition 4 (Operational termination) *Let $\mathcal{S}$ be a theory in a logic $\mathcal{L}$ endowed with an inference system. $\mathcal{S}$ is called* operationally terminating *if no infinite well-formed tree for $\mathcal{S}$ exists.*

So operational termination means 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.

3 CTRS logic and operational termination of CTRSs

In this section, we show how the notions introduced in Section 2 specialize to CTRSs and conditional rewriting, thus leading to a notion of operational termination of CTRSs. More expressive inference systems dealing with sorts, order-sorted specifications, evaluation strategies, memberships, etc. (see [2, 3, 6, 14]), can be managed similarly.

3.1 Conditional Term Rewriting Systems

A conditional rule is as follows: $l \to r$ if $s_1 = t_1, \cdots, s_n = t_n$, where $l, r, s_1, t_1, \cdots, s_n, t_n$ are terms [15]. As usual, l and r are called the left- and right-hand sides of the rule, and the sequence $s_1 = t_1, \cdots, s_n = t_n$ (often denoted c) is the *conditional part* of the rule. Rewrite rules $l \to r$ if c are classified according to the distribution of variables among l, r, and c, as follows: type 1, if $\mathrm{Var}(r) \cup \mathrm{Var}(c) \subseteq \mathrm{Var}(l)$; type 2, if $\mathrm{Var}(r) \subseteq \mathrm{Var}(l)$; type 3, if $\mathrm{Var}(r) \subseteq \mathrm{Var}(l) \cup \mathrm{Var}(c)$; and type 4, if no restriction is given. An n-CTRS contains only rewrite rules of type $m \leq n$.

It is well-known that the conditions $s_i = t_i$ for $1 \leq i \leq n$ can be interpreted in a number of different ways [5]. As in [15], we are mainly concerned with *oriented* CTRSs, i.e., those whose (conditional) rules are written as follows:

$$l \to r \ \text{ if } \ s_1 \to t_1, \cdots, s_n \to t_n$$

indicating that the conditions $s_i \to t_i$ for $1 \leq i \leq n$ are intended to express the reachability of (instances of) t_i from (instances of) s_i. An oriented 3-CTRS R is called *deterministic* if for each rule $l \to r$ if $s_1 \to t_1, \ldots, s_n \to t_n$ in R and each $1 \leq i \leq n$, we have $\mathrm{Var}(s_i) \subseteq \mathrm{Var}(l) \cup \bigcup_{j=1}^{i-1} \mathrm{Var}(t_j)$.

Let R be a CTRS. We inductively define unconditional TRSs R_n for $n \in \mathbb{N}$ by $R_0 = \emptyset$ and $R_{n+1} = \{l\sigma \to r\sigma \mid l \to r \text{ if } s_1 \to t_1, \ldots, s_n \to t_n \in R \text{ and } s_i\sigma \to_{R_n}^* t_i\sigma \text{ for all } j \in \{1, \ldots, n\}\}$. The rewrite relation $\to_R$ associated with a CTRS R is $\bigcup_{n \in \mathbb{N}} \to_{R_n}$.

A CTRS R is *terminating* if $\to_R$ is a well-founded relation. A deterministic 3-CTRS is *quasi-decreasing* if there is a well-founded partial ordering $\succ$ on $\mathcal{T}(\mathcal{F}, \mathcal{X})$ satisfying $\to_R \subseteq \succ$, $\rhd \subseteq \succ$ (where $\rhd$ is the strict subterm relation), and for every rule $l \to r$ if $s_1 \to t_1, \ldots, s_n \to t_n$, substitution σ, and index i, $0 \leq i < n$, if $s_j\sigma \to_R^* t_j\sigma$ for every $1 \leq j \leq i$, then $l\sigma \succ s_{i+1}\sigma$.

3.2 Operational termination of CTRSs

We can define the rewriting relation associated to a CTRS R by means of the inference rules of Figure 1. Note the refinement of the rewrite relation $t \to t'$ into two relations, namely $t \to^1 t'$ and $t \to^* t'$. For each atom $A = u \to v$ appearing in a condition of a conditional rule in R we use the meta-notation $A^\bullet$ to denote $u \to^* v$.

Given a CTRS R, we write $R \vdash t \to^1 u$, resp. $R \vdash t \to^* u$, whenever $t \to^1 u$, resp. $t \to^* u$ are derivable using the inference rules in Figure 1. The following proposition shows that our inference rules capture conditional rewriting in the usual sense of Section 3.1.

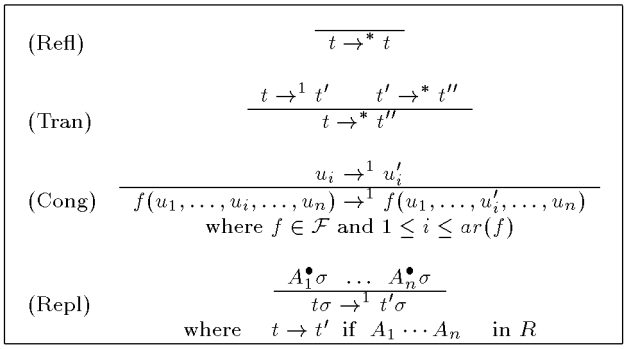

Figure 1: Inference rules for conditional rewriting

Proposition 1 *Let R be a CTRS, and t, s be terms. Then $s \to_R t$ if and only if $R \vdash s \to^1 t$, and $s \to_R^* t$ if and only if $R \vdash s \to^* t$.*

Proof. The if part can be proved by induction of proof trees; the only if part by induction on the n such that $s \to_{R_n} t$. □

When $\mathcal{L}$ is the CTRS logic in Figure 1 and $\mathcal{S}$ is a CTRS R, the notion of operational termination in Definition 4 specializes to the concept of *operational termination of CTRSs*. Now we can motivate our choice of well-formed proof tree (Definition 3) on the basis of a concrete (although representative) case. Consider the CTRS inference system. Notice that it is always possible to build trivial infinite proof trees using only Transitivity:

$$\dfrac{t_1 \to^1 t_2 \qquad \dfrac{t_2 \to^1 t_3 \qquad \dfrac{t_3 \to^1 t_4 \quad \vdots}{t_3 \to^* t}}{t_2 \to^* t}}{t_1 \to^* t}$$

This infinite tree is *not* well-formed and would never be built by a reasonable interpreter, because the interpreter will resolve one goal at a time, before attempting the next, whereas in the above tree, goals $t_i \to^1 t_{i+1}$ are left open.

Now, we can see that the CTRS in Section 1 is not operationally terminating, because we have an infinite proof tree

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{a \to^1 b}}{f(a) \to^1 f(b)} \qquad f(b) \to^* b}{f(a) \to^* b}}{a \to^1 b}$$

Note also that, since TRSs are a particular case of 1-CTRSs where no conditional rule is allowed, term rewriting with TRSs is also captured by the inference system of Figure 1 and Proposition 1. Moreover, if R is a TRS, then it is not difficult to see that R is operationally terminating if and only if R is terminating, that is, if and only if $\to_R$ is well-founded.

In the following section, we explore the relationship between the notion of operational termination and other well-known CTRS termination notions.

4 Effective termination, quasi-decreasingness, and operational termination of CTRSs

Kaplan noticed that, in sharp contrast with the unconditional case, termination of CTRSs, i.e., well-foundedness of the (conditional) rewrite relation $\rightarrow_R$, does not have the desired or expected computational features. For instance, in general for a terminating relation $\rightarrow_R$ it is not decidable whether a term is a normal form or not [11]. In fact, Kaplan established the following result.

Theorem 1 [12, Theorem 1.4] *There exists a conditional TRS R such that $\rightarrow_R$ is confluent and terminating, but such that the relation $\rightarrow_R$ is not decidable and whose set of normal forms is not computable.*

After Kaplan's work, different attempts to formulate appropriate notions of termination for CTRSs have been made [12, 5], see [15] for a survey. Marchiori devised a notion of termination for CTRSs which (in Marchiori's words) implies *effective computability for CTRSs* [13, Section 5.1]. Such a notion was described as follows:

Definition 5 (Effective Termination [13, 15]**)** *A CTRS R is effectively terminating if $\rightarrow_R$ is terminating and the set $\Delta^*(s) = \{t \in \mathcal{T}(\mathcal{F},\mathcal{X}) \mid s \rightarrow_R^* t\}$ of all R-reducts of a term $s \in \mathcal{T}(\mathcal{F},\mathcal{X})$ is finite and computable.*

The crucial difference between termination and effective termination of a CTRS is that, for a terminating CTRS R, membership of a pair $t \rightarrow t'$ in the rewriting relation may in general be undecidable. Thus, Theorem 1 tells us that termination does not imply effective termination.

Giesl and Arts introduced the notion of *quasi-decreasingness*[2] [8, Definition 1] as defined in Section 3.1. As shown in [15], the notion of *quasi-decreasingness* is currently the most general one which is intended to be checked by comparing parts of the CTRS.

We prove two theorems showing that operational termination of (deterministic 3-)CTRSs R (viewed as theories) w.r.t. the CTRS logic is equivalent to quasi-decreasingness of R.

Theorem 2 *Let R be a deterministic 3-CTRS. If R is quasi-decreasing, then it is operationally terminating.*

Proof. Since R is quasi-decreasing, there is a well-founded partial ordering $\succ$ which includes both $\rightarrow_R$ and $\rhd$ and such that for every rule $l \rightarrow r$ if $s_1 \rightarrow t_1, \ldots, s_n \rightarrow t_n$, substitution σ, and index i, $0 \leq i \leq n$, if $s_j\sigma \rightarrow_R^* t_j\sigma$ for every $1 \leq j \leq i$, then $l\sigma \succ s_{i+1}\sigma$.

Assume that R is not is operationally terminating. Then, there is an infinite well-formed proof tree T for some judgement G (i.e., $G = head(T)$). We prove that there is a judgement G' at level 1 or 2 in T which is the head of an infinite well-formed proof tree T' and such that $left(G) = s \succ s' = left(G')$, where $left(s \rightarrow^1 t) = s$ and $left(s \rightarrow^* t) = s$.

1. In case $G = s \rightarrow^1 t$, G must be the head of an instance of an inference rule Δ, where Δ is either the Congruence or Replacement rule.

[2] Actually, the original notion by Giesl and Arts was called *left-right decreasingness*, but we use Ohlebusch's terminology in [15].

(a) If Δ is the Congruence rule, then $s = f(u_1, \ldots, u_i, \ldots, u_n)$, $t = f(u_1, \ldots, u'_i, \ldots, u_n)$ and there is an infinite well-formed proof tree T' (at level 1 in T) whose head is $G' = head(T') = u_i \to^1 u'_i$. Since $\rhd \subseteq \succ$, we have $left(G) = s \succ u_i = left(G')$ as required.

(b) If Δ is the Replacement rule, then $s = l\sigma$ for some conditional rule $l \to r$ if $s_1 \to t_1, \cdots, s_n \to t_n$ and substitution σ and there is $i \in \{1, \ldots, n\}$ such that $T_1, \ldots, T_{i-1}$ are well-formed finite proof trees with $head(T_j) = s_j\sigma \to^* t_j\sigma$ for $1 \leq j < i$ (and hence, $s_j\sigma \to^*_R t_j\sigma$ for all $1 \leq j < i$) and T_i is infinite. Let $G_i = head(T_i) = s_i\sigma \to^* t_i\sigma$. Now, we have $s = l\sigma \succ s_i\sigma$. Thus, we let $G' = G_i$ and conclude $left(G) = s \succ s_i\sigma = left(G')$.

2. Now consider the case $G = s \to^* t$. Since T is infinite, the only possibility is to use Transitivity. Thus, we have two proof trees T_1 and T_2 such that either T_1 or T_2 are infinite. Here, $G_1 = head(T_1) = s \to^1 s'$ for some term s' and $G_2 = head(T_2) = s' \to^* t$.

   (a) If T_1 is infinite, then $left(G) = s = left(G_1)$. We can apply item 1 above to G_1 and T_1 to conclude that there is a judgement G' at level 1 in T_1 (i.e., at level 2 in T) such that $left(G) = s \succ u' = left(G')$.

   (b) If T_1 is finite, then $s \to_R s'$, i.e., $s \succ s'$ and T_2 is infinite. Then, we let $G' = G_2$ and conclude $left(G) = s \succ s' = left(G')$.

Thus, given the well-formed infinite proof tree T, we can define an infinite sequence $s = u_1 \succ u_2 \succ \cdots$ which contradicts the well-foundedness of $\succ$. □

Theorem 3 *Let R be a deterministic 3-CTRS. If R is operationally terminating, then it is quasi-decreasing.*

Proof. If R is operationally terminating, then $\to_R$ is a well-founded ordering which is closed under contexts (although it is not necessarily closed under substitutions). By [15, Lemma 7.2.4], the relation $(\to_R \cup \rhd)^+$ is a well-founded partial ordering on terms. We write $s \rightsquigarrow t$ if there is a conditional rule $l \to r$ if $s_1 \to t_1, \cdots, s_n \to t_n$, a position p of s, a substitution σ and $i \in \{1, \ldots, n\}$ such that $s|_p = l\sigma$, $t = s_i\sigma$ and $s_j\sigma \to^*_R t_j\sigma$ for all $1 \leq j < i$. We now prove that $\succ = (\to_R \cup \rhd \cup \rightsquigarrow)^+$ is a well-founded (strict) ordering. Transitivity is obvious. Irreflexivity is a consequence of well-foundedness, which we prove as follows: assume that $\succ$ is not well-founded. Then, there is an infinite $\succ$-decreasing sequence A where, since $(\to_R \cup \rhd)^+$ is a well-founded partial ordering, we must have an infinite number of $\rightsquigarrow$-steps which isolate finite $(\to_R \cup \rhd)$-sequences. We can, then, write A as follows:

$$u_1 \, (\to_R \cup \rhd)^* u'_1 \rightsquigarrow u_2 \, (\to_R \cup \rhd)^* u'_2 \rightsquigarrow u_3 \cdots$$

In fact, since $(\to_R \cup \rhd)^+ = (\rhd^* \circ \to_R)^+ \circ \rhd^*$ and the relation $\rightsquigarrow$ already takes into account subterms, we can eventually mix the last subterm steps of each sequence $u_i \, (\to_R \cup \rhd)^+ u'_i$ with the $\rightsquigarrow$-step on u'_i. Then, we can write each

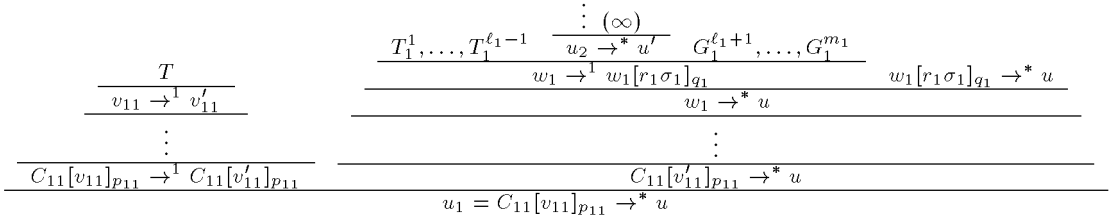

Figure 2: Proof of Theorem 3

sequence from u_i to u_{i+1} as

$$\begin{array}{lcl} u_i & = & v_{i1} \\ & \rhd^* & v'_{i1} \\ & \to_R & v_{i2} \\ & \vdots & \\ & (\rhd^* \circ \to_R) & v_{in_i+1} \\ & \rightsquigarrow & u_{i+1} \end{array}$$

where $v_{ij} = C_{ij}[v'_{ij}]_{q_{ij}}$ for $i \geq 1$ and $1 \leq j \leq n_i$. Thus, we can also write:

$$\begin{array}{lll} u_i & = & C_{i1}[v'_{i1}]_{p_{i1}} \\ & \to_R & C_{i1}[v_{i2}]_{p_{i1}} \\ & \to_R & \\ & \vdots & \\ & \to_R & C_{i1}[\cdots C_{in_i}[v_{in_i+1}]_{p_{in_i}} \cdots]_{p_{i1}} \\ & \rightsquigarrow & u_{i+1} \end{array}$$

Now, each step $C_{ij}[v'_{ij}]_{p_{ij}} \to_R C_{ij}[v_{ij+1}]_{p_{ij}}$ has a finite proof tree T_{ij} whose head is $C_{ij}[v'_{ij}]_{p_{ij}} \to^1 C_{ij}[v_{ij+1}]_{p_{ij}}$ and after $|p_{ij}|$ applications of the Congruence rule there is a proof of $v'_{ij} \to^1 v_{ij+1}$.

On the other hand, since $w_i = C_{i1}[\cdots C_{in_i}[v_{in_i+1}]_{p_{in_i}} \cdots]_{p_{i1}} \rightsquigarrow u_{i+1}$, there is a rule $l_i \to r_i$ if $s_i^1 \to t_i^1, \ldots, s_i^{k_i} \to t_i^{k_i}$, a position $q_i \geq p_{i1}.\cdots.p_{in_i}$, a substitution σ_i, and an index $\ell_i \in \{1, \ldots, k_i\}$ such that $w_i|_{q_i} = l_i\sigma_i$, $u_{i+1} = s_i^{\ell_i}\sigma_i$ and $s_i^j\sigma_i \to_R^* t_i^j\sigma_i$ for all $1 \leq j < \ell_i$. Thus, there is a proof tree

$$\frac{T_i^1, \ldots, T_i^{\ell_i - 1}, T_i^{\ell_i}, G_i^{\ell_i+1}, \ldots, G_i^{m_i}}{G_i}$$

with root $G_i = w_i[l_i\sigma_i]_{q_i} \to^1 w_i[r_i\sigma_i]_{q_i}$, well-formed trees T_i^j headed by $s_i^j\sigma_i \to_R^* t_i^j\sigma_i$ for $1 \leq j < \ell_i$, an *infinite* tree $T_i^{\ell_i}$ with $left(head(T_i^{\ell_i})) = u_{i+1} = s_i^{\ell_i}\sigma_i$ which continues the tail of A after u_{i+1} as explained before, and open goals $G_i^{\ell_i+1}, \ldots, G_i^{m_i}$. Figure 2 illustrates these facts.

Therefore, assumming that $\succ$ is not well-founded contradicts operational termination of R. Also, since well-foundedness implies irreflexivity of $\succ$, it turns out that $\succ$ is a well-founded strict ordering on terms. Now, it is not difficult to see that $\succ$ makes R quasi-decreasing. $\square$

5 Concluding Remarks

The work presented here advances the theoretical foundations of a longer-term collaborative research effort to develop both theory and tools to reason about the termination of specifications in advanced equational languages. There is at present a substantial gap between the input languages of current termination tools and the expressive features of advanced equational languages. In fact, the research in this paper started when we tried to apply the existing notions of termination of CTRSs to Maude specifications [4]. When we found that the simple example in the introduction was not terminating when the corresponding versions of it as programs were executed in a number of interpreters, we felt that, in order to reason about specifications written in such languages, both the notions of CTRS and of CTRS termination need to be generalized. Thus, in this paper, we have proposed an inference-system-based approach to generalize both CTRSs and CTRS termination. Specifically, we have proposed a general notion, called *operational termination*, that is parametric on the underlying inference system and closely corresponds to the termination of a language interpreter. Operational termination of TRSs (where no conditional rules are present) coincides with the usual notion of termination of TRSs. On the other hand, we have proven that operational termination of CTRSs is equivalent to quasi-decreasingness (Theorems 2 and 3). Interestingly, this shows an agreement between two in principle quite different computational definitions of termination, one proof-theoretic, and the other based on an ordering and very general, so that both capture the same essential intuition in different ways. As a consequence of our equivalence result, existing methods for proving quasi-decreasingness of CTRSs immediately apply to prove operational termination of CTRSs. Moreover, operational termination implies effective termination, although the converse does not hold.

Future work

One important advantage of our operational termination notion is its *parametricity*. We plan to use our framework and results to explore other instances of this notion: for example to reason about the termination of higher-order rewriting systems with sophisticated type systems, of rewrite theories involving rewriting with order-sorted specifications, memberships, both equations and non-equational rules, evaluation strategies, or even of inference systems not involving a rewriting relation, etc. [2, 3, 6, 14]. In this sense, the transformational approach initiated in [6] is worth to be further investigated in the new framework presented in this paper. However, *intrinsic termination criteria* that can be used at the level of the given inference system, without requiring theory transformations (e.g., term orderings used in proofs of quasi-decreasingness of CTRSs), should also be investigated.

References

[1] P. Borovanský, C. Kirchner, H. Kirchner, and P.-E. Moreau. ELAN from a rewriting logic point of view. *Theoretical Computer Science*, 285:155–185, 2002.

[2] A. Bouhoula, J.-P. Jouannaud, and J. Meseguer. Specification and proof in membership equational logic. *Theoretical Comput. Sci.*, 236:35–132, 2000.

[3] R. Bruni and J. Meseguer. Generalized Rewrite Theories. In J.C.M. Baeten, J.K. Lenstra, J. Parrow, and G.J. Woeginger (editors), *Proc. of 30th International Colloquium on Automata, Languages and Programming, ICALP'03*, LNCS 2719:252-266, Springer-Verlag, Berlin, 2003.

[4] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. Quesada. Maude: specification and programming in rewriting logic. *Theoretical Comput. Sci.*, 285(2):187–243, Aug. 2002.

[5] N. Dershowitz and M. Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75:111–138, 1990.

[6] F. Durán, S. Lucas, J. Meseguer, C. Marché, and X. Urbain. Proving Termination of Membership Equational Programs. In P. Sestoft and N. Heintze, editors, *Proc. of ACM SIGPLAN 2004 Symposium on Partial Evaluation and Program Manipulation, PEPM'04*, pages 147-158, ACM Press, New York, 2004.

[7] K. Futatsugi and R. Diaconescu. *CafeOBJ Report.* World Scientific, AMAST Series, 1998.

[8] J. Giesl and T. Arts. Verification of Erlang Processes by dependency pairs. *Applicable Algebra in Engineering, Communications and Computing*, 12:39–72, 2001.

[9] P. Hudak, S.J. Peyton-Jones, and P. Wadler. Report on the Functional Programming Language Haskell: a non–strict, purely functional language. *Sigplan Notices*, 27(5):1-164, 1992.

[10] J. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ. In *Software Engineering with OBJ: Algebraic Specification in Action.* Kluwer, 2000.

[11] S. Kaplan. Conditional rewrite rules. *Theoretical Computer Science*, 33:175–193, 1984.

[12] S. Kaplan. Simplifying conditional term rewriting systems: Unification, termination and confluence. *Journal of Symb. Comp.*, 4:295–334, 1987.

[13] M. Marchiori. Unravelings and ultra-properties. In M. Hanus and M. Rodríguez-Artalejo, editors, *Proc. of ALP'96*, volume 1039 of *LNCS*, pages 107–121. Springer-Verlag, 1996.

[14] J. Meseguer. Membership algebra as a logical framework for equational specification. In F. Parisi-Presicce, editor, *Proceedings WADT'97*, volume 1376 of *LNCS*, pages 18–61. Springer-Verlag, 1998.

[15] E. Ohlebusch. *Advanced Topics in Term Rewriting.* Springer-Verlag, Apr. 2002.