# Termination of context-sensitive rewriting by rewriting *

Salvador Lucas

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, E-46071 Valencia, Spain.
e.mail: slucas@dsic.upv.es

**Abstract.** Context-sensitive rewriting allows us to deal with certain context-replacing restrictions in performing reduction steps. This is useful in avoiding undesirable reductions. In this paper, we study termination of context-sensitive rewriting. We give some conditions under which it is possible to analyze the termination of the context-sensitive rewrite relation induced by a term rewriting system by means of any method to prove termination of rewriting. This allows us to apply all known (and forthcoming) results on termination to context-sensitive rewriting.

**Keywords:** term rewriting, functional programming, termination.

## 1 Introduction

Term Rewriting is traditionally considered the operational semantics of functional programming languages [3, 16]. A functional program is a set of equations that are interpreted as left-to-right rewrite rules [5, 15]. Execution of the program for any input data consists of the evaluation of the term using the rewrite rules until it cannot be further reduced (i.e. until a *normal form* is reached). In a terminating system, no infinite computations are possible.

In context-sensitive rewriting [12, 13], for each symbol in the signature $\Sigma$ we fix the set of *replacing* arguments by means of a mapping $\mu : \Sigma \to \wp(I\!N)$ (the *replacement map*) which specifies the set of positions which can eventually be reduced. This is useful to express some requirements on the 'intended meaning' of a function. For example, consider the standard functional definition of the *if-then-else* operation:

$if(true, x, y) \to x$
$if(false, x, y) \to y$

Given an input $if(cond, t, s)$, we are not interested in reductions on the second and third arguments of the operation $if$ until the condition $cond$ has been evaluated. By defining a replacement map such as $\mu(if) = \{1\}$, the undesirable reductions on the second and third immediate subterms of the term $if(cond, t, s)$ can be avoided with no extra control.

* This work has been partially supported by CICYT under grant TIC 95-0433-C03-03.

Context-sensitive rewriting has some strong connections with the lazy strategies of functional programming languages. Actually, the mechanism which allows us to prevent the evaluation of some expressions can also be used to declare that these reductions need not to be done. Although giving support to lazy reduction techniques is not the main motivation of our work, context-sensitive rewriting does allow us to express fixed, meaningful and easily implementable restrictions on rewriting, similarly to the way that lazy reduction techniques approximate 'needed' reductions à la Huet and Lévy [10]. An approach to these questions which is related with our work can be found in [9].

In this paper, we address the problem of the termination of context-sensitive rewriting by using any method to prove that a TRS terminates. Termination of rewriting is normally proved by defining a noetherian ordering $>$ on terms and checking whether $t > s$ whenever $t \rightarrow s$ (Manna and Ness' theorem, [14]). This is equivalent to proving that each instance of every rule $l \rightarrow r$ in the TRS satisfies $l > r$ in a monotonic noetherian ordering $>$ (Kamin and Lévy, [8]). As suggested by Manna and Ness, *'it is often convenient to separate reduction orderings into a homomorphism from terms to an algebra with a noetherian ordering'*. This can be achieved by means of *termination functions* [2, 4]. A termination function is a *homomorphism* from terms to a $\Sigma$-algebra (where $\Sigma$ is the underlying signature). It can also be considered as an *extraction function* from terms to multisets of selected immediate subterms (see [4]). We show that we can often prove termination of context-sensitive rewriting by first transforming the original TRS into a *contracted* TRS on a new signature by means of a *contracting* function. This function can be thought of as a *pseudo*-termination function in that it *projects* the original set of terms onto a new set of *contracted* terms on which we can use the standard methods.

The paper is organized as follows. In Section 2, we briefly recall the technical concepts used in the remainder of the paper. In Section 3, we formulate some basic properties of context-sensitive rewriting. Section 4 provides two basic criteria to ensure termination of context-sensitive rewriting. Section 5 concludes.

## 2 Preliminaries

Let us first introduce the main notations used in the paper from [5, 11]. $V$ denotes a countable infinite set of variables and $\Sigma$ denotes a set of function symbols {f, g, ...}, each with a fixed arity given by a function $ar : \Sigma \rightarrow \mathbb{N}$. By $\mathcal{T}(\Sigma, V)$ we denote the set of terms. $Var(t)$ is the set of variable symbols in the term $t$.

Terms are viewed as labelled trees in the usual way. Occurrences $u, v, \ldots$ are represented by chains of positive natural numbers used to address subterms of $t$. Occurrences are ordered by the standard prefix ordering: $u \leq v$ iff there is a chain $v'$ such that $v = u.v'$. $O(t)$ denotes the set of occurrences of a term $t$. $t|_u$ is the subterm at occurrence $u$ of $t$. $t[s]_u$ is the term $t$ with the subterm at the occurrence $u$ replaced with $s$. We refer to any term $C$ that is the same as $t$ everywhere except below $u$, i.e. there exists a term $s$ such that $C[s]_u = t$, as the *context* within the replacement occurs.

A rewrite rule is an ordered pair $(l, r)$, written $l \to r$, with $l, r \in \mathcal{T}(\Sigma, V)$, $l \notin V$ and $Var(r) \subseteq Var(l)$. $l$ is said to be the left-hand side (*lhs*) of the rule and $r$ is the right-hand side (*rhs*). A TRS is a pair $\mathcal{R} = (\Sigma, R)$ where $R$ is a set of rewrite rules. For a given TRS $\mathcal{R} = (\Sigma, R)$, a term $t$ rewrites to a term $s$ (at the occurrence $u$), written $t \to_{\mathcal{R}} s$, if $t|_u = \sigma(l)$ and $s = t[\sigma(r)]_u$, for some rule $l \to r$ in $R$, occurrence $u$ in $t$ and substitution $\sigma$. $\to_{\mathcal{R}}$ is the one-step rewrite relation for $\mathcal{R}$.

$I\!N_k^+$ is an initial segment $\{1, 2, \ldots k\}$ of the set of positive natural numbers $I\!N^+$, where $I\!N_0^+ = \emptyset$. We let R denote a binary relation on $A$, i.e., $\mathrm{R} \subseteq A \times A$. We will often write $x\mathrm{R}y$ for $(x, y) \in \mathrm{R}$. We let $\mathrm{R}^*$ denote the reflexive-transitive closure of R. A relation R is *noetherian* (or *terminating*) iff there is no infinite sequence $a_1 \mathrm{R}\, a_2 \cdots a_n \mathrm{R}\, a_{n+1} \cdots$.

Given a set $A$, we denote a k-tuple $x_1, \ldots, x_k$ of elements as $\tilde{x}$, i.e. $\tilde{x} \in A^k$. By $\tilde{x}[y]_i$ for any $i, 1 \leq i \leq k$ we denote a k-tuple where $x_i$ is replaced by $y$ in $\tilde{x}$. Given a k-tuple $\tilde{x}$ and a set of positive natural numbers $J = \{j_1, \ldots, j_q\} \subseteq I\!N_k^+$, denote as $\tilde{x}_J$ the $q$-ary tuple of elements in $\tilde{x}$ *extracted* by $J$, i.e. $\tilde{x}_J = x_{j_1}, \ldots, x_{j_q}$. Note that the $i$-th component $x_i$ of $\tilde{x}$ is the $|I\!N_i^+ \cap J|$-th component of $\tilde{x}_J$. Given a set $B$, a function $\psi : A \to B$ and a k-ary tuple $\tilde{x} \in A^k$, we denote as $\psi(\tilde{x})$ the k-ary tuple of transformed-by-$\psi$ elements, i.e. $\psi(x_1), \ldots, \psi(x_k)$.

## 3 Context-sensitive rewriting

### 3.1 Basic definitions of context-sensitive rewriting

The idea behind the concept of context-sensitive rewriting [12, 13] is to impose a syntactic replacement condition which prevents some reductions. This is achieved by using a map on the signature we call replacement map.

**Definition 3.1 (Replacement map)** *Let $\Sigma$ be a signature. A mapping $\mu : \Sigma \to \wp(I\!N)$ is a replacement map (or $\Sigma$-map) for the signature $\Sigma$ iff for all* f $\in \Sigma$. $\mu(\mathrm{f}) \subseteq \mathcal{N}^+_{ar(\mathrm{f})}$.

The ordering $\subseteq$ on $\wp(I\!N)$ extends to an ordering $\sqsubseteq$ on $M_\Sigma$, the set of all $\Sigma$-maps: $\mu \sqsubseteq \mu'$ iff $\mu(\mathrm{f}) \subseteq \mu'(\mathrm{f})$ for all $\mathrm{f} \in \Sigma$. It is immediate to see that $(M_\Sigma, \sqsubseteq, \mu_\perp, \mu_\top)$ is a complete lattice where $\mu_\perp(\mathrm{f}) = \emptyset$ and $\mu_\top(\mathrm{f}) = I\!N^+_{ar(\mathrm{f})}$ for all $\mathrm{f} \in \Sigma$.

The replacement map determines the argument positions which can be reduced for each symbol on the signature. The replacement condition indicates the occurrences which can be rewritten.

**Definition 3.2 (Replacement condition)** *Let $\Sigma$ be a signature and $\mu$ be a $\Sigma$-map. Let $t \in \mathcal{T}(\Sigma, V)$ be a term. The replacement condition is a relation $\gamma_{\mu,t}$ defined on the set of occurrences $O(t)$ as follows:*

$\gamma_{\mu,t}(\varepsilon)$.

$\gamma_{\mu,\mathrm{f}(\tilde{t}[s]_i)}(i.u) \Leftrightarrow (i \in \mu(\mathrm{f})) \wedge \gamma_{\mu,s}(u)$.

*We say that the occurrence $u$ of a term $t$ satisfies the replacement condition $\gamma_{\mu,t}$ iff $\gamma_{\mu,t}(u)$. We also say that $u$ is a replacing occurrence of $t$ or that the subterm $t|_u$ is a replacing subterm.*

We write $\gamma_t(u)$ when the replacement map is clear from the context. Some new notation entailed by the concept of replacement condition: $O^\mu(t)$ is the set of *replacing* occurrences of a term $t$: $u \in O^\mu(t) \Leftrightarrow u \in O(t) \wedge \gamma_t(u)$. $O^\mu_s(t)$ is the (possibly empty) set of replacing occurrences of a subterm $s$ of $t$: $u \in O^\mu_s(t) \Leftrightarrow u \in O^\mu(t) \wedge t|_u = s$. $Var^\mu(t)$ is the set of *replacing variables* of $t$: $x \in Var^\mu(t) \Leftrightarrow O^\mu_x(t) \neq \emptyset$.

**Definition 3.3 (One-step context-sensitive rewrite relation)** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu$ be a $\Sigma$-map. A term $t$ $\mu$-rewrites to a term $s$, written $t \hookrightarrow_{\mathcal{R}(\mu)} s$, if $t \to_{\mathcal{R}} s$ at the* replacing *occurrence $u \in O^\mu(t)$. $\hookrightarrow_{\mathcal{R}(\mu)}$ is the one-step context-sensitive rewrite relation of $\mathcal{R}$ wrt $\mu$. $\hookrightarrow^*_{\mathcal{R}(\mu)}$ is the context-sensitive rewrite relation of $\mathcal{R}$ wrt $\mu$.*

In the sequel, we assume that the TRS $\mathcal{R} = (\Sigma, R)$ and $\Sigma$-map $\mu$ are given, and we drop references to them by just writing $\hookrightarrow$ instead of $\hookrightarrow_{\mathcal{R}(\mu)}$ to denote the one-step context-sensitive rewrite relation.

**Example 3.4** *Let $\rho : \mathrm{f}(x) \to x$ be a rule and $t = \mathrm{g}(\mathrm{f}(\mathrm{a}), \mathrm{f}(\mathrm{b}))$ be a term.*
*a) If $\mu(\mathrm{g}) = \emptyset$, then $\rho$ cannot be applied to reduce $t$, because neither the occurrence $u = 1$ nor $u = 2$ satisfy the replacement condition wrt $\mu$.*
*b) If $\mu(\mathrm{g}) = \{1\}$, then $\rho$ can be applied to reduce* $\mathrm{f}(\mathrm{a})$, *but it cannot be applied to reduce* $\mathrm{f}(\mathrm{b})$.
*c) If $\mu(\mathrm{g}) = \{2\}$, then $\rho$ can be applied to reduce* $\mathrm{f}(\mathrm{b})$, *but it cannot be applied to reduce* $\mathrm{f}(\mathrm{a})$.
*d) If $\mu(\mathrm{g}) = \{1, 2\}$, then $\rho$ can be applied to reduce both* $\mathrm{f}(\mathrm{a})$ *and* $\mathrm{f}(\mathrm{b})$.

By analogy to standard rewriting, context-sensitive is closed under *replacing* context application, i.e., $t \hookrightarrow s \Rightarrow C[t]_u \hookrightarrow C[s]_u$ for every $u \in O^\mu(C)$.

**Remark 3.5** *Note that with the top $\Sigma$-map $\mu_\top$ the replacement condition is satisfied for every term $t$ and occurrence $u \in O(t)$. This is to say that, given a TRS $\mathcal{R}$, the one-step context-sensitive rewrite relation for $\mu_\top$ coincides with the standard one-step rewrite relation, i.e. $\hookrightarrow_{\mathcal{R}(\mu_\top)} = \to_{\mathcal{R}}$.*

Given a TRS, the following proposition relates the one-step context-sensitive rewrite relations induced by comparable replacement maps.

**Proposition 3.6 (Monotonicity of $\hookrightarrow$ with respect to $\sqsubseteq$)** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu, \mu'$ be $\Sigma$-maps. Then $\mu \sqsubseteq \mu' \Rightarrow \hookrightarrow_{\mathcal{R}(\mu)} \subseteq \hookrightarrow_{\mathcal{R}(\mu')}$.*

### 3.2 Examples of application

**Improving the evaluation of expressions.**

The rules which define the *short-cut* boolean operators *and*, *or* of Lisp are:

$$and(true, x) \to x \qquad or(true, x) \to true$$
$$and(false, x) \to false \qquad or(false, x) \to x$$

It is sensible to evaluate up to the normal form the first argument of an input term $and(t,s)$ ($or(t,s)$), since the outcome is necessary to determine which rule has to be applied to the root. According to the rules, if the first argument reduces to *false* (*true*), then any reduction on the second argument is useless, since it does not contribute to computing the value of the resulting expression.

A replacement map $\mu(and) = \mu(or) = \{1\}$, has the effect of delaying the evaluation of redexes which are different from the first argument of the term.

In manipulating lists, another case of study arises. Let us consider the *head* and *tail* functions, whose behavior is as follows:

$head(x :: L) \to x$
$tail(x :: L) \to L$

*head* only requires the evaluation of the head of the list and *tail* only evaluates the rest of the list. This points are only of interest if *cons* (i.e. '::') does not evaluate its arguments systematically. Programming Languages have been conceived where *cons* does not evaluate its arguments (or, at least it does not evaluate its second argument) (see [6, 11]). This behavior can easily be achieved in context-sensitive rewriting, by defining $\mu(::) = \emptyset$ (or $\mu(::) = \{1\}$).

**Mechanizing Inference Systems.**

Context-sensitive rewriting can be formulated as an Inference System by considering each rewrite rule $l \to r$ as a (scheme of) axiom: $\frac{}{l \to r}$, and introducing as many context-passing rules [11] as replacing indices $i \in \mu(\mathrm{f})$ for each $k$-ary function symbol 'f' in the signature. That is, for each *replacing* argument position $i$, we define the (scheme of) rule $\frac{t \to s}{\mathrm{f}(t_1,\ldots,t_{i-1},t,\ldots,t_k) \to \mathrm{f}(t_1,\ldots,t_{i-1},s,\ldots,t_k)}$.

Different computational systems have been formalized by using Inference Systems whose axioms and rules match these patterns, as they are weak $\beta$-reduction, call-by-name reduction strategies of $\lambda$-calculus, and $\pi$-calculus. Therefore, we can use context-sensitive rewriting as a suitable mechanization of the execution of processes of these systems. In this way, we can exploit existing rewriting devices (reduction machines, graph reduction, etc.) for efficient implementations [7, 15].

## 4 $\mu$-termination

We would like to compare the termination behavior of the standard rewriting in a given TRS $\mathcal{R}$ and the context-sensitive rewriting in $\mathcal{R}$ given a replacement map. To distinguish them, we introduce the following preparatory definition.

**Definition 4.1 ($\mu$-termination)** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu$ be a $\Sigma$-map. We say that $\mathcal{R}$ $\mu$-terminates (or $\mathcal{R}$ is $\mu$-terminating) iff $\hookrightarrow_{\mathcal{R}(\mu)}$ is noetherian.*

**Proposition 4.2** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS. Let $\mu, \mu'$ be two $\Sigma$-maps such that $\mu \sqsubseteq \mu'$. If $\mathcal{R}$ $\mu'$-terminates, then $\mathcal{R}$ $\mu$-terminates.*

Proof. By contradiction, using Proposition 3.6. □

Now we consider the Remark 3.5 to show that any criteria establishing the termination of a TRS $\mathcal{R}$ also establishes the $\mu$-termination of $\mathcal{R}$.

**Theorem 4.3** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu$ be a $\Sigma$-map. If $\mathcal{R}$ terminates, then $\mathcal{R}$ $\mu$-terminates.*

Proof. Immediate from Proposition 4.2, since $\rightarrow_{\mathcal{R}} = \hookrightarrow_{\mathcal{R}(\mu_\top)}$. □

From Theorem 4.3, we can directly use any ordering $>$ on $\mathcal{T}(\Sigma, V)$ (for example the recursive path ordering $>^*$ in [1, 2]) to check $\mu$-termination of the context-sensitive rewriting by just considering the rules in the TRS (without taking into account the replacement map). Actually we only need to prove that $\sigma(l) > \sigma(r)$ for all rules $l \rightarrow r$ in the TRS and substitutions $\sigma$. If this comparison succeeds, then Theorem 4.3 ensures that any context-sensitive rewriting will terminate. We refer to this method as the $\mathcal{C}_{\mathcal{R}}$*-criterion* to check $\mu$-termination. Nevertheless, we can consider many $\mu$-terminating TRSs despite the fact that the $\mathcal{C}_{\mathcal{R}}$-test does not succeed.

**Example 4.4** *Let us consider the nonterminating TRS $\mathcal{R} = (\{\mathrm{f}, \mathrm{g}\}, \{\mathrm{f}(\mathrm{g}(x), y) \rightarrow \mathrm{f}(x, \mathrm{f}(\mathrm{g}(x), y))\})$. If we take $\mu$ such that $\mu(\mathrm{f}) = \mu(\mathrm{g}) = \{1\}$, then it is easy to verify that any $\mu$-rewriting on $\mathcal{R}$ will eventually terminate.*

The example shows that the $\mathcal{C}_{\mathcal{R}}$-criterion is not complete. It also illustrates that, when comparing the *lhs* and *rhs* of a rewrite rule $l \rightarrow r$ in order to determine $\mu$-termination of the TRS, we only need to take into account the *replacing* arguments occurring in $l$ and $r$. Roughly speaking, since $\mu(\mathrm{f}) = \{1\}$ we would find $l > r$, i.e. $\mathrm{f}(\mathrm{g}(x), y) > \mathrm{f}(x, \mathrm{f}(\mathrm{g}(x), y))$ if we forget the second immediate argument both in $\mathrm{f}(\mathrm{g}(x), y)$ and $\mathrm{f}(x, \mathrm{f}(\mathrm{g}(x), y))$. This is similar to checking whether $\mathrm{f}(\mathrm{g}(x), \square) > \mathrm{f}(x, \square)$ (where $\square$ expresses the 'unnecessary' arguments, that will not be taken into account in order to prove $l > r$). This idea will be formalized in the following sections.

## 4.1 The contractive transformation

In the following section, we are able to state an interesting result: we can investigate $\mu$-termination of a TRS by studying termination of a contracted TRS built using a new signature we call *$\mu$-contracted signature* which we define as follows.

**Definition 4.5 ($\mu$-contracted signature)** *Let $\Sigma$ be a signature and $\mu$ be a $\Sigma$-map. The $\mu$-contracted signature $\Sigma^\mu$ is as follows: $\mathrm{f}_\mu \in \Sigma^\mu \wedge ar(\mathrm{f}_\mu) = |\mu(\mathrm{f})| \Leftrightarrow \mathrm{f} \in \Sigma$.*

We connect terms from both signatures $\Sigma$ and $\Sigma^\mu$ by means a function.

**Definition 4.6 ($\mu$-contracting function)** *Let $\Sigma$ be a signature and $\mu$ be a $\Sigma$-map. The $\mu$-*contracting function *$\tau_\mu : \mathcal{T}(\Sigma, V) \rightarrow \mathcal{T}(\Sigma^\mu, V)$ is as follows:*

- $\tau_\mu(x) = x$ *if* $x \in V$

– $\tau_\mu(\mathrm{f}(\tilde{t})) = \mathrm{f}_\mu(\tau_\mu(\tilde{t}_{\mu(\mathrm{f})}))$

Note that this function *drops* the non-replacing immediate subterms of a term $t$ and constructs a '$\mu$-contracted' term by joining the (also transformed) replacing subterms below the corresponding operator of the $\mu$-contracted signature. See below for an illustrative example.

We can observe that this is not a termination function (see [4]) since it is neither a *homomorphism* (the operations in the contracted signature could have a lesser arity than the corresponding operation in the original one) nor an *extraction function* (because (i) the extracted-by-$\mu$ subterms are joined by a new function symbol of the contracted signature and (ii) the subterms are also transformed by the contracting function). Nevertheless, it deals with a task similar to termination functions, i.e., translating the original set of terms $\mathcal{T}(\Sigma, V)$ into a set (in this case $\mathcal{T}(\Sigma^\mu, V)$) on which we can demonstrate the terminating behavior of context-sensitive rewriting.

**Example 4.7** *(Continued from Example 4.4) We can transform the TRS under consideration into a 'contracted' TRS $\mathcal{R}^\mu$ by applying $\tau_\mu$. This gives: $\mathcal{R}^\mu = (\{\mathrm{f}_\mu, \mathrm{g}_\mu\}, \{\mathrm{f}_\mu(\mathrm{g}_\mu(x)) \to \mathrm{f}_\mu(x)\})$.*

The next proposition shows how the $\mu$-contracting function changes the location of the immediate subterms.

**Proposition 4.8** *Let $\Sigma$ be a signature and $\mu$ be a $\Sigma$-map. If $i \in \mu(\mathrm{f})$, then we have $\tau_\mu(\mathrm{f}(\tilde{t}[s]_i)) = \mathrm{f}_\mu(\tau_\mu(\tilde{t}_{\mu(\mathrm{f})}[s]_j))$ with $j = |\mathbb{N}_i^+ \cap \mu(\mathrm{f})|$.*

PROOF. Immediate from definition of $\tau_\mu$. □

From this proposition, we know that the $i$-th immediate (replacing) subterm of $t = \mathrm{f}(\tilde{t})$ becomes the $|\mathcal{N}_i^+ \cap \mu(\mathrm{f})|$-th immediate subterm of $\tau_\mu(t)$.

**Proposition 4.9 (Surjectivity of $\tau_\mu$)** *The $\mu$-contracting function $\tau_\mu$ is surjective. That is, for each $s \in \mathcal{T}(\Sigma^\mu, V)$, there exists $t \in \mathcal{T}(\Sigma, V)$ such that $s = \tau_\mu(t)$.*

PROOF. By structural induction on $s \in \mathcal{T}(\Sigma^\mu, V)$.

1. Base case. If $s = x \in V$, define $t = x$. If $s$ is a constant $\mathrm{f}_\mu$, let f be the corresponding symbol of $\Sigma$. Then we consider the following two cases:
   (a) $ar(\mathrm{f}) = 0$. Define $t = \mathrm{f}$.
   (b) $ar(\mathrm{f}) > 0$. By definition of $\tau_\mu$, this means that $\mu(\mathrm{f}) = \emptyset$. Then, we let $t = \mathrm{f}(\tilde{t})$ for any $\tilde{t} \in \mathcal{T}(\Sigma, V)^{ar(\mathrm{f})}$.
2. Let $s = \mathrm{f}_\mu(\tilde{s})$, with $\tilde{s} \in \mathcal{T}(\Sigma^\mu, V)^{ar(\mathrm{f}_\mu)}$. By I.H., there exists $\tilde{t'} \in \mathcal{T}(\Sigma, V)^{ar(\mathrm{f}_\mu)}$ s.t. $\tilde{s} = \tau_\mu(\tilde{t'})$. Define $t = \mathrm{f}(\tilde{t})$, for some $\tilde{t} \in \mathcal{T}(\Sigma, V)^{ar(\mathrm{f})}$ s.t. $\tilde{t}_{\mu(\mathrm{f})} = \tilde{t'}$.

□

This proposition allows us to uniquely consider any term $s \in \mathcal{T}(\Sigma^\mu, V)$ as the image by $\tau_\mu$ of some term $t \in \mathcal{T}(\Sigma, V)$. Therefore, we often write $\tau_\mu(t)$ instead of $s$ when $s = \tau_\mu(t)$.

We need to establish some properties of the contracting function which will be used later. In particular, we are interested in relating the structure of a term in $\mathcal{T}(\Sigma, V)$ and the corresponding term (by $\tau_\mu$) in $\mathcal{T}(\Sigma^\mu, V)$. We introduce the following definition.

**Definition 4.10** *Let $\Sigma$ be a term and $\mu$ be a $\Sigma$-map. Let $t \in \mathcal{T}(\Sigma, V)$. The function $\nu_{\mu,t} : O(t) \to (\mathbb{N} - \{0\})^*$ is as follows:*

- $\nu_{\mu,t}(\epsilon) = \epsilon$
- $\nu_{\mu,\mathrm{f}(\tilde{t}[s]_i)}(i.u) = \epsilon \quad$ *if* $i \notin \mu(\mathrm{f})$
- $\nu_{\mu,\mathrm{f}(\tilde{t}[s]_i)}(i.u) = |\mathcal{N}_i^+ \cap \mu(\mathrm{f})|.\nu_{\mu,s}(u) \quad$ *if* $i \in \mu(\mathrm{f})$

We will write just $\nu_t$ instead of $\nu_{\mu,t}$ when the replacement map $\mu$ is clear from the context. This function allows us to obtain the set of occurrences of a contracted term from the occurrences of the original term.

**Proposition 4.11** *Let $\Sigma$ be a signature, $\mu$ be a $\Sigma$-map and $\tau_\mu$ the $\mu$-contracting function. Let $t \in \mathcal{T}(\Sigma, V)$ and $u \in O(t)$. Then we have $O(\tau_\mu(t)) = \nu_t(O(t))$.*

An interesting fact concerning the contracting function is related to the preservation of the replacing subterms.

**Proposition 4.12 ($\tau_\mu$ preserves the replacing subterms)** *Let $\Sigma$ be a signature, $\mu$ be a $\Sigma$-map and $\tau_\mu$ the $\mu$-contracting function. Let $t \in \mathcal{T}(\Sigma, V)$ and $u \in O^\mu(t)$. Assume that $z = t|_u$. Then $\tau_\mu(z) = \tau_\mu(t)|_v$, where $v \in O(\tau_\mu(t))$ satisfies that $v = \nu_t(u)$.*

Proof. Structural induction. The base case is immediate. For the induction step, assume $t = \mathrm{f}(\tilde{t}[z]_{i.u'})$, where $u = i.u'$. Then, we can apply the induction hypothesis to the $i$-th immediate subterm $s$ of $t$, since $z = s|_{u'}$. Therefore, $\tau_\mu(z) = \tau_\mu(s)|_{v'}$, where $v' = \nu_s(u')$. Since $i.u' \in O^\mu(t)$, $i \in \mu(\mathrm{f})$ and by Proposition 4.8, we get $\tau_\mu(t) = \tau_\mu(\mathrm{f}(\tilde{t}[s]_i)) = \mathrm{f}_\mu(\tau_\mu(\tilde{t}_{\mu(\mathrm{f})}[s]_j))$, with $j = |\mathbb{N}_i^+ \cap \mu(\mathrm{f})|$. Therefore, the subterm $\tau_\mu(z)$ occurs at position $j.v'$ in $\tau_\mu(t)$. By Definition 4.10, we have $\nu_t(i.u') = j.\nu_s(u') = j.v' = v$, and the conclusion follows. $\square$

**Corollary 4.13 ($\tau_\mu$ preserves the replacing variables)** *Let $\Sigma$ be a signature, $\mu$ be a $\Sigma$-map and $\tau_\mu$ the $\mu$-contracting function. Let $t \in \mathcal{T}(\Sigma, V)$. If $x \in Var^\mu(t)$, then $x \in Var(\tau_\mu(t))$.*

**Proposition 4.14** *Let $\Sigma$ be a signature, $\mu$ be a $\Sigma$-map and $\tau_\mu$ the $\mu$-contracting function. Let $t, s \in \mathcal{T}(\Sigma, V)$. If $Var^\mu(t) \subseteq Var^\mu(s)$, then $Var(\tau_\mu(t)) \subseteq Var(\tau_\mu(s))$.*

Proof. Let us take $x \in Var^\mu(t)$. By Corollary 4.13, $x \in Var(\tau_\mu(t))$. Since $Var^\mu(t) \subseteq Var^\mu(s)$, we also have $x \in Var(\tau_\mu(s))$, which proves the claim. $\square$

In the following proposition, we prove that the substitution application commutes with the $\mu$-contracting function.

**Proposition 4.15** *Let $\Sigma$ be a signature, $\mu$ be a $\Sigma$-map and $\tau_\mu$ the $\mu$-contracting function. Let $t \in \mathcal{T}(\Sigma, V)$ and $\sigma : V \to \mathcal{T}(\Sigma, V)$ a substitution. Let $\sigma_\mu : V \to \mathcal{T}(\Sigma^\mu, V)$ such that $\sigma_\mu(x) = \tau_\mu(\sigma(x))$ for all $x \in V$. Then, we have $\tau_\mu(\sigma(t)) = \sigma_\mu(\tau_\mu(t))$.*

Proof. By structural induction. If $t = x \in V$, then the result is immediate, since $\tau_\mu(x) = x$. For the induction step, we take $t = \mathrm{f}(\tilde{t})$ for some $\tilde{t} \in \mathcal{T}(\Sigma, V)^{ar(\mathrm{f})}$. By I.H., $\tau_\mu(\sigma(t)) = \tau_\mu(\sigma(\mathrm{f}(\tilde{t}))) = \tau_\mu(\mathrm{f}(\sigma(\tilde{t}))) = \mathrm{f}_\mu(\tau_\mu(\sigma(\tilde{t}_{\mu(\mathrm{f})}))) = \mathrm{f}_\mu(\sigma_\mu(\tau_\mu(\tilde{t}_{\mu(\mathrm{f})}))) = \sigma_\mu(\mathrm{f}_\mu(\tau_\mu(\tilde{t}_{\mu(\mathrm{f})}))) = \sigma_\mu(\tau_\mu(t))$. □

### 4.2 $\mu$-termination and contracted signature

The $\mu$-contracting function allows us to prove $\mu$-termination in context-sensitive rewriting by proving termination of the 'contracted' TRS using any method. However, there is one complication. The $\mu$-contraction of the rewrite rules $l \to r$ of a TRS could lead to pairs $(\tau_\mu(l), \tau_\mu(r))$ which can not be considered as rewrite rules. For example: the rule $\mathrm{f}(\mathrm{g}(x), y) \to \mathrm{f}(y, \mathrm{f}(\mathrm{g}(x), y))$ with $\mu(\mathrm{f}) = \mu(\mathrm{g}) = \{1\}$ yields (by $\mu$-contracting) the 'rule' $\mathrm{f}_\mu(\mathrm{g}_\mu(x)) \to \mathrm{f}_\mu(y)$, which does not satisfy the standard non-extra variable condition of rewriting rules.

We overcome this problem by imposing a syntactical restriction on the TRSs on which we want to analyse $\mu$-termination.

**Definition 4.16 (TRS with conservative replacing variables)** *Let $\Sigma$ be a signature and $\mu$ be a $\Sigma$-map. A TRS $\mathcal{R} = (\Sigma, R)$ has* conservative replacing variables *if every replacing variable in the rhs of a rule in $R$ is a replacing variable in the lhs of this rule. Formally: $Var^\mu(r) \subseteq Var^\mu(l)$ for every rule $l \to r \in R$.*

**Definition 4.17 ($\mu$-contracted TRS)** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu$ be a $\Sigma$-map. If $\mathcal{R}$ is a TRS with conservative replacing variables, we define the $\mu$-contraction $\mathcal{R}^\mu = (\Sigma^\mu, R^\mu)$ of $\mathcal{R}$ as follows:*

- *$\Sigma^\mu$ is the $\mu$-contracted signature in Definition 4.5.*
- *$R^\mu = \{\tau_\mu(l) \to \tau_\mu(r) \mid l \to r \in R\}$.*

The following proposition establishes the correctness of Definition 4.17.

**Proposition 4.18** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu$ be a $\Sigma$-map. If $\mathcal{R}$ is a TRS with conservative replacing variables, then the $\mu$-contraction $\mathcal{R}^\mu = (\Sigma^\mu, R^\mu)$ of of $\mathcal{R}$ is a TRS.*

Proof. For each rule $\tau_\mu(l) \to \tau_\mu(r) \in R^\mu$, we need to check, whether it verifies $\tau_\mu(l) \notin V$ and $Var(\tau_\mu(r)) \subseteq Var(\tau_\mu(l))$. The first condition is immediate from Definition 4.6 if $\mathcal{R}$ is a TRS. The second condition follows immediately from Definition 4.16 and Proposition 4.14. □

In order to connect $\mu$-rewriting for a TRS and a replacement map and rewriting sequences using the contracted TRS, we introduce the following proposition.

**Proposition 4.19 ($\mu$-rewriting in $\mathcal{R}$ and rewriting in $\mathcal{R}^\mu$)** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with conservative replacing variables and $\mu$ be a $\Sigma$-map. Let $\mathcal{R}^\mu = (\Sigma^\mu, R^\mu)$ be the contraction of $\mathcal{R}$. If $t \hookrightarrow_{\mathcal{R}(\mu)} s$, then $\tau_\mu(t) \to_{\mathcal{R}^\mu} \tau_\mu(s)$.*

PROOF. If $t \hookrightarrow_{\mathcal{R}(\mu)} s$, then there exists $u \in O^\mu(t)$, $l \to r \in R$ and substitution $\sigma : V \to \mathcal{T}(\Sigma, V)$ such that $t|_u = \sigma(l)$ and $s = t[\sigma(r)]_u$. Assume $v = \nu_t(u)$, and $\sigma_\mu : V \to \mathcal{T}(\Sigma^\mu, V)$ a substitution such that $\sigma_\mu(x) = \tau_\mu(\sigma(x))$ for all $x \in V$. By Proposition 4.12 and Proposition 4.15, we have that $\tau_\mu(t)|_v = \tau_\mu(\sigma(l)) = \sigma_\mu(\tau_\mu(l))$. By Definition 4.17, if $l \to r \in R$, then $\tau_\mu(l) \to \tau_\mu(r) \in R^\mu$. Therefore the term $\tau_\mu(t)$ can be rewritten at occurrence $v$ by using rule $\tau_\mu(l) \to \tau_\mu(r) \in R^\mu$ and matching $\sigma_\mu$. The rewritten term is $s' = \tau_\mu(t)[\sigma_\mu(\tau_\mu(r))]_v = \tau_\mu(t)[\tau_\mu(\sigma(r))]_v = \tau_\mu(t[\sigma(r)]_u) = \tau_\mu(s)$. □

The next example shows that the converse statement does not hold, i.e. $\tau_\mu(t) \to_{\mathcal{R}^\mu} \tau_\mu(s)$ does not imply $t \hookrightarrow_{\mathcal{R}(\mu)} s$.

**Example 4.20** *Let $\mathcal{R} = (\{\mathrm{a}, \mathrm{b}, \mathrm{f}\}, \{\mathrm{f}(\mathrm{a}, \mathrm{b}) \to \mathrm{a}\})$ and consider a replacement map $\mu$ such that $\mu(\mathrm{f}) = \{1\}$. Since $\mathcal{R}^\mu = (\{\mathrm{a}_\mu, \mathrm{b}_\mu, \mathrm{f}_\mu\}, \{\mathrm{f}_\mu(\mathrm{a}_\mu) \to \mathrm{a}_\mu\})$ and $\tau_\mu(\mathrm{f}(\mathrm{a}, \mathrm{a})) = \mathrm{f}_\mu(\mathrm{a}_\mu)$, we get $\tau_\mu(\mathrm{f}(\mathrm{a}, \mathrm{a})) \to_{\mathcal{R}^\mu} \tau_\mu(\mathrm{a})$. However, $\mathrm{f}(\mathrm{a}, \mathrm{a}) \not\hookrightarrow_{\mathcal{R}(\mu)} \mathrm{a}$.*

The following theorem establishes the $\mu$-termination of a TRS and a replacement map when the contracted TRS terminates.

**Theorem 4.21 ($\mu$-termination of $\mathcal{R}$ when $\mathcal{R}^\mu$ terminates)** *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu$ be a $\Sigma$-map. Let $\mathcal{R}^\mu = (\Sigma^\mu, R^\mu)$ be the contracted TRS of $\mathcal{R}$. If $\mathcal{R}$ is a TRS with conservative replacing variables, then $\mathcal{R}$ $\mu$-terminates if $\mathcal{R}^\mu$ terminates.*

PROOF. By contradiction. Assume that $\mathcal{R}$ does not $\mu$-terminate. Then there exists an infinite $\mu$-derivation $t_1 \hookrightarrow t_2 \hookrightarrow \cdots \hookrightarrow t_n \hookrightarrow \cdots$ in $\mathcal{R}$. By Proposition 4.19, we get an infinite derivation $\tau_\mu(t_1) \to \tau_\mu(t_2) \to \cdots \to \tau_\mu(t_n) \to \cdots$ in $\mathcal{R}^\mu$, which contradicts the initial hypothesis. □

This theorem gives us a new criterion to detect $\mu$-termination of a TRS. Actually, we can use any standard method to check termination of the $\mu$-contracted TRS. We refer to this new method as the $\mathcal{C}^\mu_\mathcal{R}$*-criterion* to check $\mu$-termination.

**Example 4.22** *We can apply the $\mathcal{C}^\mu_\mathcal{R}$-criterion to demonstrate the $\mu$-termination of the TRS $\mathcal{R}$ in the Example 4.4, using the $\mu$-contracted TRS $\mathcal{R}^\mu$ obtained in Example 4.7. By using a recursive path ordering $>^*$ on $\mathcal{T}(\Sigma^\mu, V)$ based on any precedence on $\Sigma^\mu = \{\mathrm{f}_\mu, \mathrm{g}_\mu\}$, we get $\mathrm{f}_\mu(\mathrm{g}_\mu(x)) >^* \mathrm{f}_\mu(x)$, i.e. $\mathcal{R}^\mu$ terminates, and therefore $\mathcal{R}$ $\mu$-terminates.*

Sometimes we can not use directly the $\mathcal{C}^\mu_\mathcal{R}$-criterion, but it turns still useful in combination with Proposition 4.2.

**Example 4.23** *Let us consider the TRS $\mathcal{R}$ and the replacement map $\mu$ of Example 4.4. Let $\mu'$ be given as follows: $\mu'(f) = \{1\}$ and $\mu'(g) = \emptyset$. Then, $\mathcal{R}$ does* not *satisfies the conservative replacing variables property w.r.t $\mu'$. Hence we can not apply the $\mathcal{C}^{\mu'}_\mathcal{R}$-criterion. However, since $\mu' \sqsubseteq \mu$ and $\mathcal{R}$ is $\mu$-terminating (see Example 4.22), by Proposition 4.2, $\mathcal{R}$ is $\mu'$-terminating.*

### 4.3 Checking $\mu$-termination

We use the above results to establish a suitable criterion to check $\mu$-termination. First, an example reveals the *uncompleteness* of the $\mathcal{C}^\mu_\mathcal{R}$-criterion.

**Example 4.24** *Let us consider the TRS* $\mathcal{R} = (\{\mathrm{f}, \mathrm{g}\}, \{\mathrm{f}(\mathrm{g}(x)) \to \mathrm{f}(x)\})$*, and a replacement map* $\mu$ *given by* $\mu(\mathrm{f}) = \emptyset$*. For this replacement map,* $\mathcal{R}$ *is a TRS with conservative replacing variables. Then, it is easy to demonstrate that* $\mathcal{R}$ *is a* $\mu$*-terminating TRS (it suffices to consider the* $\mathcal{C}_\mathcal{R}$*-criterion and apply a recursive path ordering, for example, to* $\mathcal{R}$*), but the* $\mu$*-contracted TRS* $\mathcal{R}^\mu = (\{\mathrm{f}_\mu, \mathrm{g}_\mu\}, \{\mathrm{f}_\mu \to \mathrm{f}_\mu\})$ *is not terminating.*

Since we could establish the $\mu$-termination of $\mathcal{R}$ by means of the $\mathcal{C}_\mathcal{R}$-criterion, the example points out the fact that the $\mathcal{C}^\mu_\mathcal{R}$-criterion does not include the $\mathcal{C}_\mathcal{R}$-criterion. On the other hand, from Example 4.4 we conclude that the $\mathcal{C}_\mathcal{R}$-criterion does not include the $\mathcal{C}^\mu_\mathcal{R}$-criterion either.

The example 4.24 also shows that Theorem 4.21 has not a symmetric counterpart, that is, we can not ensure termination of a $\mu$-contracted TRS from the $\mu$-termination of the original.

Now we can formulate a mixed criterion involving both $\mathcal{C}_\mathcal{R}$ and $\mathcal{C}^\mu_\mathcal{R}$ criteria to test $\mu$-termination.

**Theorem 4.25 ($\mu$-termination of a TRS)** *Let* $\mathcal{R} = (\Sigma, R)$ *be a TRS with conservative replacing variables given a* $\Sigma$*-map* $\mu$*. Let* $\mathcal{R}^\mu = (\Sigma^\mu, R^\mu)$ *be the contracted TRS of* $\mathcal{R}$*. Then* $\mathcal{R}$ $\mu$*-terminates if* $\mathcal{R}$ *or* $\mathcal{R}^\mu$ *terminates.*

Proof. Immediate from Theorem 4.3 and Theorem 4.21. □

Note that this theorem only applies to TRSs with conservative replacing variables. Fortunately, for general TRSs, we still have the Theorem 4.3 ($\mathcal{C}_\mathcal{R}$-criterion) to prove $\mu$-termination.

## 5 Conclusions

Context-sensitive rewriting allows us to deal with context-replacing restrictions in performing reduction steps, thus avoiding some undesirable reductions. Termination is always preserved or improved, as we have shown in this paper.

The restrictions imposed by context-sensitive rewriting are easily implementable. However, in order to get complete benefits from context-sensitive rewriting, more research is needed. In particular, it is interesting to study conditions to ensure that, by defining a concrete replacement map for a given TRS, computations are restricted but not *too much*, i.e., they will be able to reduce sufficiently a term by context-sensitive rewriting. This would allow us, implicitly, to define the adequate replacement map which is powerful enough to perform computations with context-sensitive rewriting. This task will be addressed in a further work.

## References


1. N. Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science* 17:279-301, 1982.
2. N. Dershowitz. Termination of rewriting. *Journal on Symbolic Computation*, 3:69-115, 1987.
3. N. Dershowitz. A Taste of Rewrite Systems. In P.E. Lauer, editor, *Proc. of Functional Programming, Concurrency, Simulation and Automated Reasoning*, LNCS 693:199-228, Springer-Verlag, Berlin, 1993.
4. N. Dershowitz and C. Hoot. Natural Termination. *Theoretical Computer Science* 142(2):179-207, 1995.
5. N. Dershowitz and J.P. Jouannaud. Rewrite Systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 243-320. Elsevier, Amsterdam and The MIT Press, Cambridge, MA, 1990.
6. D.P. Friedman and D.S. Wise. CONS should not evaluate its arguments. In S. Michaelson and R. Milner, editors, *Automata, Languages and Programming*, pages 257-284, Edinburgh University Press, 1976.
7. J. Goguen, C. Kirchner and J. Meseguer. Concurrent Term Rewriting as a Model of Computation. In J.H. Fasel, R.M. Keller, editors, *Graph Reduction. Proceedings. of a Workshop.* LNCS 279:53-93. Springer-Verlag, Berlin, 1987.
8. S. Kamin and J.J. Lévy. Two generalizations of the recursive path ordering. University of Illinois, 1980.
9. J.F.Th. Kamperman and H.R. Walters. Lazy Rewriting and Eager Machinery. In J. Hsiang, editor, *Proc. of the 6th International Conference on Rewriting Techniques and Applications, RTA'95*, LNCS 914:147-162, Springer-Verlag, Berlin, 1995.
10. G. Huet and J.J. Lévy. Computations in orthogonal term rewriting systems. In J.L. Lassez and G. Plotkin, editors, *Computational logic: essays in honour of J. Alan Robinson.* MIT Press, Cambridge, MA, 1991.
11. R. Lalement. Computation as Logic. Masson-Prentice Hall International, 1993.
12. S. Lucas. Computational properties in context-sensitive rewriting. In M. Alpuente and M.I. Sessa, editors, *Proc. of 1995 Joint Conference on Declarative Programming, GULP-PRODE'95*, pages 435-446, 1995.
13. S. Lucas. Fundamentals of context-sensitive rewriting. In M. Bartŏsek, J. Staudek and J. Wiedermann, editors, *Proc. of XXII Seminar on Current Trends in Theory and Practice of Informatics, SOFSEM'95*, LNCS 1012:405-412, Springer-Verlag, Berlin, 1995.
14. Z. Manna and S Ness. On the termination of Markov algorithms. In *Proc. of the Third Hawaii International Conference on System Science*, pages 789-792, 1970.
15. S. L. Peyton-Jones. The Implementation of Functional Programming Languages. Prentice-Hall International, London, 1987.
16. C. Reade. Elements of Functional Programming. Addison-Wesley Publishing Company, 1987.


This article was processed using the LTEX macro package with LLNCS style