# Uncurrying for Innermost Termination and Derivational Complexity*

Harald Zankl,[1] Nao Hirokawa,[2] and Aart Middeldorp[1]

[1] Institute of Computer Science, University of Innsbruck, Austria
{harald.zankl,aart.middeldorp}@uibk.ac.at

[2] School of Information Science, Japan Advanced Institute of Science and Technology, Japan
hirokawa@jaist.ac.jp

First-order applicative term rewriting systems provide a natural framework for modeling higher-order aspects. In earlier work we introduced an uncurrying transformation which is termination preserving and reflecting. In this paper we investigate how this transformation behaves for innermost termination and (innermost) derivational complexity. We prove that it reflects innermost termination and innermost derivational complexity and that it preserves and reflects polynomial derivational complexity. For the preservation of innermost termination and innermost derivational complexity we give counterexamples. Hence uncurrying may be used as a preprocessing transformation for innermost termination proofs and establishing polynomial upper and lower bounds on the derivational complexity. Additionally it may be used to establish upper bounds on the innermost derivational complexity while it neither is sound for proving innermost non-termination nor for obtaining lower bounds on the innermost derivational complexity.

## 1 Introduction

Proving termination of first-order applicative term rewrite systems is challenging since the rules lack sufficient structure. But these systems are important since they provide a natural framework for modeling higher-order aspects found in functional programming languages. Since proving termination is easier for innermost than for full rewriting we lift some of the recent results from [8] from full to innermost termination. For the properties that do not transfer to the innermost setting we provide counterexamples. Furthermore we show that the uncurrying transformation is suitable for proving upper bounds on the (innermost) derivational complexity.

We remark that our approach on proving innermost termination also is beneficial for functional programming languages that adopt a lazy evaluation strategy since applicative term rewrite systems modeling functional programs are left-linear and non-overlapping. It is well known that for this class of systems termination and innermost termination coincide (see [5] for a more general result).

The remainder of this paper is organized as follows. After recalling preliminaries in Section 2, we show that uncurrying preserves innermost non-termination (but not innermost termination) in Section 3. In Section 4 we show that it preserves and reflects derivational complexity of rewrite systems while it only reflects innermost derivational complexity. Section 5 reports on experimental results and we conclude in Section 6.

*This research is supported by FWF (Austrian Science Fund) project P18763 and the Grant-in-Aid for Young Scientists Nos. 20800022 and 22700009 of the Japan Society for the Promotion of Science.

HOR 2010
EPTCS ??, 20??, pp. 1–12, doi:10.4204/EPTCS.??.??

## 2 Preliminaries

In this section we fix preliminaries on rewriting, complexity and uncurrying.

### 2.1 Term Rewriting

We assume familiarity with term rewriting [1, 17]. Let $\mathcal{F}$ be a signature and $\mathcal{V}$ a set of variables disjoint from $\mathcal{F}$. By $\mathcal{T}(\mathcal{F},\mathcal{V})$ we denote the set of terms over $\mathcal{F}$ and $\mathcal{V}$. The size of a term $t$ is denoted $|t|$. A rewrite rule is a pair of terms $(\ell, r)$, written $\ell \to r$, such that $\ell$ is not a variable and all variables in $r$ occur in $\ell$. A term rewrite system (TRS for short) is a set of rewrite rules. A TRS $\mathcal{R}$ is said to be duplicating if there exist a rewrite rule $\ell \to r \in \mathcal{R}$ and a variable $x$ that occurs more often in $r$ than in $\ell$.

Contexts are terms over the signature $\mathcal{F} \cup \{\Box\}$ with exactly one occurrence of the fresh constant $\Box$ (called hole). The expression $C[t]$ denotes the result of replacing the hole in $C$ by the term $t$. A substitution $\sigma$ is a mapping from variables to terms and $t\sigma$ denotes the result of replacing the variables in $t$ according to $\sigma$. Substitutions may change only finitely many variables (and are thus written as $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$). The set of positions of a term $t$ is defined as $\mathcal{P}\mathsf{os}(t) = \{\epsilon\}$ if $t$ is a variable and as $\mathcal{P}\mathsf{os}(t) = \{\epsilon\} \cup \{iq \mid q \in \mathcal{P}\mathsf{os}(t_i)\}$ if $t = f(t_1, \ldots, t_n)$. Positions are used to address occurrences of subterms. The subterm of $t$ at position $p \in \mathcal{P}\mathsf{os}(t)$ is defined as $t|_p = t$ if $p = \epsilon$ and as $t|_p = t_i|_q$ if $p = iq$. We say a position $p$ is to the right of a position $q$ if $p = p_1 i p_2$ and $q = q_1 j q_2$ with $p_1 = q_1$ and $i > j$. For a term $t$ and positions $p, q \in \mathcal{P}\mathsf{os}(t)$ we say $t|_p$ is to the right of $t|_q$ if $p$ is to the right of $q$.

A rewrite relation is a binary relation on terms that is closed under contexts and substitutions. For a TRS $\mathcal{R}$ we define $\to_{\mathcal{R}}$ to be the smallest rewrite relation that contains $\mathcal{R}$. We call $s \to_{\mathcal{R}} t$ a rewrite step if there exist a context $C$, a rewrite rule $\ell \to r \in \mathcal{R}$, and a substitution $\sigma$ such that $s = C[\ell\sigma]$ and $t = C[r\sigma]$. In this case we call $\ell\sigma$ a redex and say that $\ell\sigma$ has been contracted. A root rewrite step, denoted by $s \to_{\mathcal{R}}^{\epsilon} t$, has the shape $s = \ell\sigma \to_{\mathcal{R}} r\sigma = t$ for some $\ell \to r \in \mathcal{R}$. A rewrite sequence is a sequence of rewrite steps. The set of normal forms of a TRS $\mathcal{R}$ is defined as $\mathit{NF}(\mathcal{R}) = \{t \in \mathcal{T}(\mathcal{F},\mathcal{V}) \mid t \text{ contains no redexes}\}$. A redex $\ell\sigma$ in a term $t$ is called innermost if proper subterms of $\ell\sigma$ are normal forms, and rightmost innermost if in addition $\ell\sigma$ is to the right of any other redex in $t$. A rewrite step is called innermost (rightmost innermost) if an innermost (rightmost innermost) redex is contracted, written $\xrightarrow{\mathsf{i}}$ and $\xrightarrow{\mathsf{ri}}$, respectively.

If the TRS $\mathcal{R}$ is not essential or clear from the context the subscript $_{\mathcal{R}}$ is omitted in $\to_{\mathcal{R}}$ and its derivatives. As usual, $\to^+$ ($\to^*$) denotes the transitive (reflexive and transitive) closure of $\to$ and $\to^m$ its $m$-th iterate. A TRS is terminating (innermost terminating) if $\to^+$ ($\xrightarrow{\mathsf{i}}{}^+$) is well-founded.

Let $\mathcal{P}$ be a property of TRSs and let $\Phi$ be a transformation on TRSs with $\Phi(\mathcal{R}) = \mathcal{R}'$. We say $\Phi$ *preserves* $\mathcal{P}$ if $\mathcal{P}(\mathcal{R})$ implies $\mathcal{P}(\mathcal{R}')$ and $\Phi$ *reflects* $\mathcal{P}$ if $\mathcal{P}(\mathcal{R}')$ implies $\mathcal{P}(\mathcal{R})$. Sometimes we call $\Phi$ $\mathcal{P}$ preserving if $\Phi$ preserves $\mathcal{P}$ and $\mathcal{P}$ reflecting if $\Phi$ reflects $\mathcal{P}$, respectively.

### 2.2 Derivational Complexity

For complexity analysis we assume TRSs to be finite and (innermost) terminating.

Hofbauer and Lautemann [10] introduced the concept of derivational complexity for terminating TRSs. The idea is to measure the maximal length of rewrite sequences (derivations) depending on the size of the starting term. Formally, the *derivation height* of a term $t$ (with respect to a finitely branching and well-founded order $\to$) is defined on natural numbers as $\mathrm{dh}(t, \to) = \max\{m \in \mathbb{N} \mid t \to^m u \text{ for some } u\}$. The *derivational complexity* $\mathrm{dc}_{\mathcal{R}}(n)$ of a TRS $\mathcal{R}$ is then defined as $\mathrm{dc}_{\mathcal{R}}(n) = \max\{\mathrm{dh}(t, \to_{\mathcal{R}}) \mid |t| \leqslant n\}$.

Similarly we define the *innermost* derivational complexity as $\mathrm{idc}_{\mathcal{R}}(n) = \max\{\mathrm{dh}(t, \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \mid |t| \leqslant n\}$. Since we regard finite TRSs only, these functions are well-defined if $\mathcal{R}$ is (innermost) terminating. If $\mathrm{dc}_{\mathcal{R}}(n)$ is bounded by a linear, quadratic, cubic, ... function or polynomial, $\mathcal{R}$ is said to have linear, quadratic, cubic, ... or polynomial derivational complexity. A similar convention applies to $\mathrm{idc}_{\mathcal{R}}(n)$.

For functions $f, g\colon \mathbb{N} \to \mathbb{N}$ we write $f(n) \in \mathcal{O}(g(n))$ if there are constants $M, N \in \mathbb{N}$ such that $f(n) \leqslant M \cdot g(n) + N$ for all $n \in \mathbb{N}$.

One popular method to prove polynomial upper bounds on the derivational complexity is via triangular matrix interpretations [13], which are a special instance of monotone algebras. An $\mathcal{F}$-algebra $\mathcal{A}$ consists of a non-empty carrier $A$ and a set of interpretations $f_{\mathcal{A}}$ for every $f \in \mathcal{F}$. By $[\alpha]_{\mathcal{A}}(\cdot)$ we denote the usual evaluation function of $\mathcal{A}$ according to an assignment $\alpha$ which maps variables to values in $A$. An $\mathcal{F}$-algebra $\mathcal{A}$ together with a well-founded order $\succ$ on $A$ is called a *monotone algebra* if every $f_{\mathcal{A}}$ is monotone with respect to $\succ$. Any monotone algebra $(\mathcal{A}, \succ)$ induces a well-founded order on terms: $s \succ_{\mathcal{A}} t$ if for any assignment $\alpha$ the condition $[\alpha]_{\mathcal{A}}(s) \succ [\alpha]_{\mathcal{A}}(t)$ holds. A TRS $\mathcal{R}$ is compatible with a monotone algebra $(\mathcal{A}, \succ_{\mathcal{A}})$ if $l \succ_{\mathcal{A}} r$ for any $l \to r \in \mathcal{R}$.

*Matrix interpretations* $(\mathcal{M}, \succ)$ (often just denoted $\mathcal{M}$) are a special form of monotone algebras. Here the carrier is $\mathbb{N}^d$ for some fixed dimension $d \in \mathbb{N} \setminus \{0\}$. The order $\succ$ is defined on $\mathbb{N}^d$ as $(u_1, \ldots, u_d) \succ (v_1, \ldots, v_d)$ if $u_1 >_{\mathbb{N}} v_1$ and $u_i \geqslant_{\mathbb{N}} v_i$ for all $2 \leqslant i \leqslant d$. If every $f \in \mathcal{F}$ of arity $n$ is interpreted as $f_{\mathcal{M}}(\vec{x_1}, \ldots, \vec{x_n}) = F_1\vec{x_1} + \cdots + F_n\vec{x_n} + \vec{f}$ where $F_i \in \mathbb{N}^{d \times d}$ for all $1 \leqslant i \leqslant n$ and $\vec{f} \in \mathbb{N}^d$ then monotonicity of $\succ$ is achieved by demanding $F_{i(1,1)} \geqslant 1$ for any $1 \leqslant i \leqslant n$. Such interpretations have been introduced in [2].

A matrix interpretation where for every $f \in \mathcal{F}$ all $F_i$ ($1 \leqslant i \leqslant n$ where $n$ is the arity of $f$) are upper triangular is called *triangular* (abbreviated by TMI). A square matrix $A$ of dimension $d$ is of *upper triangular* shape if $A_{(i,i)} \leqslant 1$ and $A_{(i,j)} = 0$ if $i > j$ for all $1 \leqslant i, j \leqslant d$. The next theorem is from [13].

**Theorem 1.** *If a TRS $\mathcal{R}$ is compatible with a TMI $\mathcal{M}$ of dimension $d$ then* $\mathrm{dc}_{\mathcal{R}}(n) \in \mathcal{O}(n^d)$.

Recent generalizations of this theorem are reported in [14, 18].

### 2.3 Uncurrying

This section recalls definitions and results from [8].

An *applicative* term rewrite system (ATRS for short) is a TRS over a signature that consists of constants and a single binary function symbol called application which is denoted by the infix and left-associative symbol $\circ$. In examples we often use juxtaposition instead of $\circ$. Every ordinary TRS can be transformed into an ATRS by currying. Let $\mathcal{F}$ be a signature. The currying system $\mathcal{C}(\mathcal{F})$ consists of the rewrite rules

$$f_{i+1}(x_1, \ldots, x_i, y) \to f_i(x_1, \ldots, x_i) \circ y$$

for every $n$-ary function symbol $f \in \mathcal{F}$ and every $0 \leqslant i < n$. Here $f_n = f$ and, for every $0 \leqslant i < n$, $f_i$ is a fresh function symbol of arity $i$. The currying system $\mathcal{C}(\mathcal{F})$ is confluent and terminating. Hence every term $t$ has a unique normal form $t{\downarrow}_{\mathcal{C}(\mathcal{F})}$. For instance, $\mathsf{f}(\mathsf{a}, \mathsf{b})$ is transformed into $\mathsf{f}\ \mathsf{a}\ \mathsf{b}$. Note that we write $f$ for $f_0$.

Next we recall the uncurrying transformation from [8]. Let $\mathcal{R}$ be an ATRS over a signature $\mathcal{F}$. The *applicative arity* $\mathrm{aa}(f)$ of a constant $f \in \mathcal{F}$ is defined as the maximum $n$ such that $f \circ t_1 \circ \cdots \circ t_n$ is a subterm in the left- or right-hand side of a rule in $\mathcal{R}$. This notion is extended to terms as follows:

| $\mathcal{R}$ | $\mathcal{U}(\mathcal{R})$ | $\mathcal{R}{\downarrow}_{\mathcal{U}(\mathcal{R})}$ | $\mathcal{R}_\eta$ | $\mathcal{R}_\eta{\downarrow}_{\mathcal{U}(\mathcal{R})}$ |
|---|---|---|---|---|
| $\mathsf{id}\ x \to x$ | $\mathsf{id} \circ x \to \mathsf{id}_1(x)$ | $\mathsf{id}_1(x) \to x$ | $\mathsf{id}\ x \to x$ | $\mathsf{id}_1(x) \to x$ |
| $\mathsf{f}\ x \to \mathsf{id}\ \mathsf{f}\ x$ | $\mathsf{id}_1(x) \circ y \to \mathsf{id}_2(x,y)$ | $\mathsf{f}_1(x) \to \mathsf{id}_2(\mathsf{f},x)$ | $\mathsf{f}\ x \to \mathsf{id}\ \mathsf{f}\ x$ | $\mathsf{f}_1(x) \to \mathsf{id}_2(\mathsf{f},x)$ |
| | $\mathsf{f} \circ x \to \mathsf{f}_1(x)$ | | $\mathsf{id}\ x\ y \to x\ y$ | $\mathsf{id}_2(x,y) \to x \circ y$ |

Table 1: Some (transformed) TRSs

$\mathrm{aa}(t) = \mathrm{aa}(f)$ if $t$ is a constant $f$ and $\mathrm{aa}(t_1) - 1$ if $t = t_1 \circ t_2$. Note that $\mathrm{aa}(t)$ is undefined if the head symbol of $t$ is a variable. The uncurrying system $\mathcal{U}(\mathcal{R})$ consists of the rewrite rules

$$f_i(x_1,\ldots,x_i) \circ y \to f_{i+1}(x_1,\ldots,x_i,y)$$

for every constant $f \in \mathcal{F}$ and every $0 \leqslant i < \mathrm{aa}(f)$. Here $f_0 = f$ and, for every $i > 0$, $f_i$ is a fresh function symbol of arity $i$. We say that $\mathcal{R}$ is *left head variable free* if $\mathrm{aa}(t)$ is defined for every non-variable subterm $t$ of a left-hand side of a rule in $\mathcal{R}$. This means that no subterm of a left-hand side in $\mathcal{R}$ is of the form $t_1 \circ t_2$ where $t_1$ is a variable. The uncurrying system $\mathcal{U}(\mathcal{R})$, or simply $\mathcal{U}$, is confluent and terminating. Hence every term $t$ has a unique normal form $t{\downarrow}_{\mathcal{U}}$. The *uncurried* system $\mathcal{R}{\downarrow}_{\mathcal{U}}$ is the TRS consisting of the rules $\ell{\downarrow}_{\mathcal{U}} \to r{\downarrow}_{\mathcal{U}}$ for every $\ell \to r \in \mathcal{R}$. However the rules of $\mathcal{R}{\downarrow}_{\mathcal{U}}$ are not enough to simulate an arbitrary rewrite sequence in $\mathcal{R}$. The natural idea is now to add $\mathcal{U}(\mathcal{R})$, but still $\mathcal{R}{\downarrow}_{\mathcal{U}(\mathcal{R})} \cup \mathcal{U}(\mathcal{R})$ is not enough as shown in the next example from [8].

**Example 2.** Consider the TRS $\mathcal{R}$ in Table 1. Based on $\mathrm{aa}(\mathsf{id}) = 2$ and $\mathrm{aa}(\mathsf{f}) = 1$ we get three rules in $\mathcal{U}(\mathcal{R})$ and can compute $\mathcal{R}{\downarrow}_{\mathcal{U}(\mathcal{R})}$. The TRS $\mathcal{R}$ is non-terminating but $\mathcal{R}{\downarrow}_{\mathcal{U}(\mathcal{R})} \cup \mathcal{U}(\mathcal{R})$ is terminating.

Let $\mathcal{R}$ be a left head variable free ATRS. The *$\eta$-saturated* ATRS $\mathcal{R}_\eta$ is the smallest extension of $\mathcal{R}$ such that $\ell \circ x \to r \circ x \in \mathcal{R}_\eta$ whenever $\ell \to r \in \mathcal{R}_\eta$ and $\mathrm{aa}(\ell) > 0$. Here $x$ is a variable that does not appear in $\ell \to r$. In the following we write $\mathcal{U}_\eta^+(\mathcal{R})$ for $\mathcal{R}_\eta{\downarrow}_{\mathcal{U}(\mathcal{R})} \cup \mathcal{U}(\mathcal{R})$. Note that applicative arities are computed before $\eta$-saturation.

**Example 3.** Consider again Table 1. Since $\mathrm{aa}(\mathsf{id}) = 2$ but $\mathrm{aa}(\mathsf{id}\ x) = 1$ for the rule $\mathsf{id}\ x \to x$ in $\mathcal{R}$ this explains the rule $\mathsf{id}\ x\ y \to x\ y$ in $\mathcal{R}_\eta$. Note that $\mathcal{U}_\eta^+(\mathcal{R})$ is non-terminating.

For a term $t$ over the signature of the TRS $\mathcal{U}_\eta^+(\mathcal{R})$, we denote by $t{\downarrow}_{\mathcal{C}'}$ the result of identifying different function symbols in $t{\downarrow}_{\mathcal{C}}$ that originate from the same function symbol in $\mathcal{F}$. For a substitution $\sigma$, we write $\sigma{\downarrow}_{\mathcal{U}}$ for the substitution $\{x \mapsto \sigma(x){\downarrow}_{\mathcal{U}} \mid x \in \mathcal{V}\}$.

***From now on we assume that every ATRS is left-head variable free.***

We conclude this preliminary section by recalling some results from [8].

**Lemma 4** ([8, Lemma 20])**.** *Let $\sigma$ be a substitution. If $t$ is head variable free then $t{\downarrow}_{\mathcal{U}}\sigma{\downarrow}_{\mathcal{U}} = (t\sigma){\downarrow}_{\mathcal{U}}$.* □

**Lemma 5** ([8, Lemma 15])**.** *If $\mathcal{R}$ is an ATRS then $\to_{\mathcal{R}} = \to_{\mathcal{R}_\eta}$.* □

**Lemma 6** ([8, Lemmata 26 and 27])**.** *Let $\mathcal{R}$ be an ATRS. If $s$ and $t$ are terms over the signature of $\mathcal{U}_\eta^+(\mathcal{R})$ then (1) $s \to_{\mathcal{R}{\downarrow}_{\mathcal{U}}} t$ if and only if $s{\downarrow}_{\mathcal{C}'} \to_{\mathcal{R}} t{\downarrow}_{\mathcal{C}'}$ and (2) $s \to_{\mathcal{U}} t$ implies $s{\downarrow}_{\mathcal{C}'} = t{\downarrow}_{\mathcal{C}'}$.* □

**Lemma 7** ([8, Proof of Theorem 16])**.** *Let $\mathcal{R}$ be an ATRS. If $s \to_{\mathcal{R}} t$ then $s{\downarrow}_{\mathcal{U}} \to^+_{\mathcal{U}_\eta^+(\mathcal{R})} t{\downarrow}_{\mathcal{U}}$.* □

Consequently our transformation is shown to be termination preserving and reflecting.

**Theorem 8** ([8, Theorems 16 and 28])**.** *Let $\mathcal{R}$ be an ATRS. The ATRS $\mathcal{R}$ is terminating if and only if the TRS $\mathcal{U}_\eta^+(\mathcal{R})$ is terminating.* □

# 3 Innermost Uncurrying

Before showing that our transformation reflects innermost termination we show that it does not preserve innermost termination. Hence uncurrying may not be used as a preprocessing transformation for innermost non-termination proofs.

**Example 9.** Consider the ATRS $\mathcal{R}$ consisting of the rules

$$\mathsf{f}\ x \to \mathsf{f}\ x \qquad\qquad \mathsf{f} \to \mathsf{g}$$

In an innermost sequence the first rule is never applied and hence $\mathcal{R}$ is innermost terminating. The TRS $\mathcal{U}_\eta^+(\mathcal{R})$ consists of the rules

$$\mathsf{f}_1(x) \to \mathsf{f}_1(x) \qquad \mathsf{f} \to \mathsf{g} \qquad \mathsf{f}_1(x) \to \mathsf{g} \circ x \qquad \mathsf{f} \circ x \to \mathsf{f}_1(x)$$

and is not innermost terminating due to the rule $\mathsf{f}_1(x) \to \mathsf{f}_1(x)$.

The next example shows that $s \xrightarrow{\mathsf{i}}_{\mathcal{R}} t$ does not imply $s{\downarrow_{\mathcal{U}}} \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}_\eta^+(\mathcal{R})} t{\downarrow_{\mathcal{U}}}$. This is not a counterexample to soundness of uncurrying for innermost termination, but it shows that the proof for the "if-direction" of Theorem 8 (which is based on Lemma 7) cannot be adopted for the innermost case without further ado.

**Example 10.** Consider the ATRS $\mathcal{R}$ consisting of the rules

$$\mathsf{f} \to \mathsf{g} \qquad\qquad \mathsf{a} \to \mathsf{b} \qquad\qquad \mathsf{g}\ x \to \mathsf{h}$$

and the innermost step $s = \mathsf{f}\ \mathsf{a} \xrightarrow{\mathsf{i}}_{\mathcal{R}} \mathsf{g}\ \mathsf{a} = t$. We have $s{\downarrow_{\mathcal{U}}} = \mathsf{f} \circ \mathsf{a}$ and $t{\downarrow_{\mathcal{U}}} = \mathsf{g}_1(\mathsf{a})$. The TRS $\mathcal{U}_\eta^+(\mathcal{R})$ consists of the rules

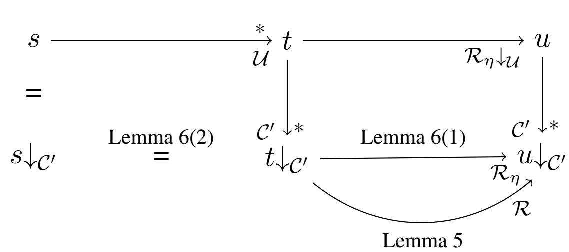
$$\mathsf{f} \to \mathsf{g} \qquad \mathsf{a} \to \mathsf{b} \qquad \mathsf{g}_1(x) \to \mathsf{h} \qquad \mathsf{g} \circ x \to \mathsf{g}_1(x)$$

We have $s{\downarrow_{\mathcal{U}}} \xrightarrow{\mathsf{i}}_{\mathcal{U}_\eta^+(\mathcal{R})} \mathsf{g} \circ \mathsf{a}$ but the step from $\mathsf{g} \circ \mathsf{a}$ to $t{\downarrow_{\mathcal{U}}}$ is not innermost.

The above problems can be solved if we consider terms that are not completely uncurried. The next lemmata prepare for the proof. Below we write $s \rhd t$ if $t$ is a proper subterm of $s$.

**Lemma 11.** *Let $\mathcal{R}$ be an ATRS. If $s$ is a term over the signature of $\mathcal{R}$, $s \in NF(\mathcal{R})$, and $s \to_{\mathcal{U}}^* t$ then $t \in NF(\mathcal{R}_\eta{\downarrow_{\mathcal{U}}})$.*

*Proof.* From Lemma 6(2) we obtain $s{\downarrow_{\mathcal{C}'}} = t{\downarrow_{\mathcal{C}'}}$. Note that $s{\downarrow_{\mathcal{C}'}} = s$ because $s$ is a term over the signature of $\mathcal{R}$. If $t \notin NF(\mathcal{R}_\eta{\downarrow_{\mathcal{U}}})$ then $t \to_{\mathcal{R}_\eta{\downarrow_{\mathcal{U}}}} u$ for some term $u$. Lemma 6(1) yields $t{\downarrow_{\mathcal{C}'}} \to_{\mathcal{R}_\eta} u{\downarrow_{\mathcal{C}'}}$ and Lemma 5 yields $s \to_{\mathcal{R}} u{\downarrow_{\mathcal{C}'}}$. Hence $s \notin NF(\mathcal{R})$, contradicting the assumption. The proof is summarized in the following diagram:




□

**Lemma 12.** $\to_{\mathcal{U}}^* \cdot \rhd \subseteq \rhd \cdot \to_{\mathcal{U}}^*$

*Proof.* Assume $s \to_{\mathcal{U}}^* t \rhd u$. We show that $s \rhd \cdot \to_{\mathcal{U}}^* u$ by induction on $s$. If $s$ is a variable or a constant then there is nothing to show. So let $s = s_1 \circ s_2$. We consider two cases.

- If the outermost $\circ$ has not been uncurried then $t = t_1 \circ t_2$ with $s_1 \to_{\mathcal{U}}^* t_1$ and $s_2 \to_{\mathcal{U}}^* t_2$. Without loss of generality assume that $t_1 \unrhd u$. If $t_1 = u$ then $s \rhd s_1 \to_{\mathcal{U}}^* t_1$. If $t_1 \rhd u$ then the induction hypothesis yields $s_1 \rhd \cdot \to_{\mathcal{U}}^* u$ and hence also $s \rhd \cdot \to_{\mathcal{U}}^* u$.
- If the outermost $\circ$ has been uncurried in the sequence from $s$ to $t$ then the head symbol of $s_1$ cannot be a variable and $\mathrm{aa}(s_1) > 0$. Hence we may write $s_1 = f \circ t_1 \circ \cdots \circ t_i$ and $t = f_{i+1}(t_1', \ldots, t_i', s_2')$ with $t_j \to_{\mathcal{U}}^* t_j'$ for all $1 \leqslant j \leqslant i$ and $s_2 \to_{\mathcal{U}}^* s_2'$. Clearly, $t_j' \unrhd u$ for some $1 \leqslant j \leqslant i$ or $s_2' \unrhd t$. In all cases the result follows with the same reasoning as in the first case. □

The next lemma states (a slightly more general result than) that an innermost root rewrite step in an ATRS $\mathcal{R}$ can be simulated by an innermost rewrite sequence in $\mathcal{U}_\eta^+(\mathcal{R})$.

**Lemma 13.** *For every ATRS $\mathcal{R}$ the inclusion ${}_{\mathcal{U}}^*\!\leftarrow \cdot \xrightarrow{\mathsf{i}}{}_{\mathcal{R}}^{\epsilon} \subseteq \xrightarrow{\mathsf{i}}{}_{\mathcal{U}_\eta^+(\mathcal{R})}^{+} \cdot {}_{\mathcal{U}}^*\!\leftarrow$ holds.*

*Proof.* We prove that $s \xrightarrow{\mathsf{i}}{}_{\mathcal{U}_\eta^+(\mathcal{R})}^{+} r{\downarrow_{\mathcal{U}}}\sigma{\downarrow_{\mathcal{U}}} \; {}_{\mathcal{U}}^*\!\leftarrow r\sigma$ whenever $s \; {}_{\mathcal{U}}^*\!\leftarrow \ell\sigma \xrightarrow{\mathsf{i}}{}_{\mathcal{R}}^{\epsilon} r\sigma$ for some rewrite rule $\ell \to r$ in $\mathcal{R}$. By Lemma 4 and the confluence of $\mathcal{U}$,

$$s \xrightarrow{\mathsf{i}}{}_{\mathcal{U}}^{*} (\ell\sigma){\downarrow_{\mathcal{U}}} = \ell{\downarrow_{\mathcal{U}}}\sigma{\downarrow_{\mathcal{U}}} \to_{\mathcal{U}_\eta^+(\mathcal{R})} r{\downarrow_{\mathcal{U}}}\sigma{\downarrow_{\mathcal{U}}} \; {}_{\mathcal{U}}^*\!\leftarrow r\sigma$$

It remains to show that the sequence $s \xrightarrow{\mathsf{i}}{}_{\mathcal{U}}^{*} (\ell\sigma){\downarrow_{\mathcal{U}}}$ and the step $\ell{\downarrow_{\mathcal{U}}}\sigma{\downarrow_{\mathcal{U}}} \to_{\mathcal{U}_\eta^+(\mathcal{R})} r{\downarrow_{\mathcal{U}}}\sigma{\downarrow_{\mathcal{U}}}$ are innermost with respect to $\mathcal{U}_\eta^+(\mathcal{R})$. For the former, let $s \xrightarrow{\mathsf{i}}{}_{\mathcal{U}}^{*} C[u] \xrightarrow{\mathsf{i}}{}_{\mathcal{U}} C[u'] \xrightarrow{\mathsf{i}}{}_{\mathcal{U}}^{*} (\ell\sigma){\downarrow_{\mathcal{U}}}$ with $u \xrightarrow{\mathsf{i}}{}_{\mathcal{U}}^{\epsilon} u'$ and let $t$ be a proper subterm of $u$. Obviously $\ell\sigma \to_{\mathcal{U}}^* C[u] \rhd t$. According to Lemma 12, $\ell\sigma \rhd v \to_{\mathcal{U}}^* t$ for some term $v$. Since $\ell\sigma \xrightarrow{\mathsf{i}}{}_{\mathcal{R}}^{\epsilon} r\sigma$, the term $v$ is a normal form of $\mathcal{R}$. Hence $t \in \mathit{NF}(\mathcal{R}_\eta{\downarrow_{\mathcal{U}}})$ by Lemma 11. Since $u \xrightarrow{\mathsf{i}}{}_{\mathcal{U}}^{\epsilon} u'$, $t$ is also a normal form of $\mathcal{U}$. Hence $t \in \mathit{NF}(\mathcal{U}_\eta^+(\mathcal{R}))$ as desired. For the latter, let $t$ be a proper subterm of $(\ell\sigma){\downarrow_{\mathcal{U}}}$. According to Lemma 12, $\ell\sigma \rhd u \to_{\mathcal{U}}^* t$. The term $u$ is a normal form of $\mathcal{R}$. Hence $t \in \mathit{NF}(\mathcal{R}_\eta{\downarrow_{\mathcal{U}}})$ by Lemma 11. Obviously, $t \in \mathit{NF}(\mathcal{U})$ and thus also $t \in \mathit{NF}(\mathcal{U}_\eta^+(\mathcal{R}))$. □

The next example shows that it is not sound to replace $\xrightarrow{\mathsf{i}}{}_{\mathcal{R}}^{\epsilon}$ by $\xrightarrow{\mathsf{i}}{}_{\mathcal{R}}$ in Lemma 13.

**Example 14.** Consider the ATRS $\mathcal{R}$ consisting of the rules

$$\mathsf{f} \to \mathsf{g} \qquad \mathsf{f}\ x \to \mathsf{g}\ x \qquad \mathsf{a} \to \mathsf{b}$$

Consequently the TRS $\mathcal{U}_\eta^+(\mathcal{R})$ consists of the rules

$$\mathsf{f} \to \mathsf{g} \qquad \mathsf{f}_1(x) \to \mathsf{g}_1(x) \qquad \mathsf{a} \to \mathsf{b} \qquad \mathsf{f} \circ x \to \mathsf{f}_1(x) \qquad \mathsf{g} \circ x \to \mathsf{g}_1(x)$$

We have $\mathsf{f}_1(\mathsf{a}) \; {}_{\mathcal{U}}^*\!\leftarrow \mathsf{f} \circ \mathsf{a} \xrightarrow{\mathsf{i}}{}_{\mathcal{R}} \mathsf{g} \circ \mathsf{a}$ but $\mathsf{f}_1(\mathsf{a}) \xrightarrow{\mathsf{i}}{}_{\mathcal{U}_\eta^+(\mathcal{R})}^{+} \cdot {}_{\mathcal{U}}^*\!\leftarrow \mathsf{g} \circ \mathsf{a}$ does not hold. To see that the latter does not hold, consider the two reducts of $\mathsf{g} \circ \mathsf{a}$ with respect to $\to_{\mathcal{U}}^*$: $\mathsf{g}_1(\mathsf{a})$ and $\mathsf{g} \circ \mathsf{a}$. We have neither $\mathsf{f}_1(\mathsf{a}) \xrightarrow{\mathsf{i}}{}_{\mathcal{U}_\eta^+(\mathcal{R})}^{+} \mathsf{g}_1(\mathsf{a})$ nor $\mathsf{f}_1(\mathsf{a}) \xrightarrow{\mathsf{i}}{}_{\mathcal{U}_\eta^+(\mathcal{R})}^{+} \mathsf{g} \circ \mathsf{a}$.

In order to extend Lemma 13 to non-root positions, we have to use rightmost innermost evaluation. This avoids the situation in the above example where parallel redexes become nested by uncurrying.

**Lemma 15.** *For every ATRS $\mathcal{R}$ the inclusion ${}_{\mathcal{U}}^*\!\leftarrow \cdot \xrightarrow{\mathsf{ri}}{}_{\mathcal{R}} \subseteq \xrightarrow{\mathsf{i}}{}_{\mathcal{U}_\eta^+(\mathcal{R})}^{+} \cdot {}_{\mathcal{U}}^*\!\leftarrow$ holds.*

*Proof.* Let $s \; {}_{\mathcal{U}}^{*}\!\!\leftarrow t = C[\ell\sigma] \xrightarrow{\mathsf{ri}}_{\mathcal{R}} C[r\sigma] = u$ with $\ell\sigma \xrightarrow{\mathsf{i}}{}_{\mathcal{R}}^{\epsilon} r\sigma$. We use induction on $C$. If $C = \Box$ then $s \; {}_{\mathcal{U}}^{*}\!\!\leftarrow t \xrightarrow{\mathsf{i}}{}_{\mathcal{R}}^{\epsilon} u$. Lemma 13 yields $s \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}_\eta^+(\mathcal{R})} \cdot \; {}_{\mathcal{U}}^{*}\!\!\leftarrow u$. For the induction step we consider two cases.

- Suppose $C = \Box \circ s_1 \circ \cdots \circ s_n$ and $n > 0$. Since $\mathcal{R}$ is left head variable free, $\mathrm{aa}(\ell)$ is defined. If $\mathrm{aa}(\ell) = 0$ then $s = t' \circ s_1' \circ \cdots \circ s_n' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow \ell\sigma \circ s_1 \circ \cdots \circ s_n \xrightarrow{\mathsf{i}}_{\mathcal{R}} r\sigma \circ s_1 \circ \cdots \circ s_n$ with $t' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow \ell\sigma$ and $s_j' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow s_j$ for $1 \leqslant j \leqslant n$. The claim follows using Lemma 13 and the fact that innermost rewriting is closed under contexts. If $\mathrm{aa}(\ell) > 0$ we have to consider two cases. In the case where the leftmost $\circ$ symbol in $C$ has not been uncurried we proceed as when $\mathrm{aa}(\ell) = 0$. If the leftmost $\circ$ symbol of $C$ has been uncurried, we reason as follows. We may write $\ell\sigma = f \circ u_1 \circ \cdots \circ u_k$ where $k < \mathrm{aa}(f)$. We have $t = f \circ u_1 \circ \cdots \circ u_k \circ s_1 \circ \cdots \circ s_n$ and $u = r\sigma \circ s_1 \circ \cdots \circ s_n$. There exists an $i$ with $1 \leqslant i \leqslant \min\{\mathrm{aa}(f), k+n\}$ such that $s = f_i(u_1', \ldots, u_k', s_1', \ldots, s_{i-k}') \circ s_{i-k+1}' \circ \cdots \circ s_n'$ with $u_j' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow u_j$ for $1 \leqslant j \leqslant k$ and $s_j' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow s_j$ for $1 \leqslant j \leqslant n$. Because of rightmost innermost rewriting, the terms $u_1, \ldots, u_k, s_1, \ldots, s_n$ are normal forms of $\mathcal{R}$. According to Lemma 11 the terms $u_1', \ldots, u_k', s_1', \ldots, s_n'$ are normal forms of $\mathcal{R}_\eta{\downarrow_{\mathcal{U}}}$. Since $i - k \leqslant \mathrm{aa}(\ell)$, $\mathcal{R}_\eta$ contains the rule $\ell \circ x_1 \circ \cdots \circ x_{i-k} \to r \circ x_1 \circ \cdots \circ x_{i-k}$ where $x_1, \ldots, x_{i-k}$ are pairwise distinct variables not occurring in $\ell$. Therefore $\tau = \sigma \cup \{x_1 \mapsto s_1, \ldots, x_{i-k} \mapsto s_{i-k}\}$ is a well-defined substitution. We obtain

$$\begin{aligned} s \; &\xrightarrow{\mathsf{i}}{}^{*}_{\mathcal{U}_\eta^+(\mathcal{R})} \; f_i(u_1{\downarrow_{\mathcal{U}}}, \ldots, u_k{\downarrow_{\mathcal{U}}}, s_1{\downarrow_{\mathcal{U}}}, \ldots, s_{i-k}{\downarrow_{\mathcal{U}}}) \circ s_{i-k+1}' \circ \cdots \circ s_n' \\ &\xrightarrow{\mathsf{i}}_{\mathcal{U}_\eta^+(\mathcal{R})} \; (r \circ x_1 \circ \cdots \circ x_{i-k}){\downarrow_{\mathcal{U}}} \tau{\downarrow_{\mathcal{U}}} \circ s_{i-k+1}' \circ \cdots \circ s_n' \\ &\; {}_{\mathcal{U}}^{*}\!\!\leftarrow \; (r \circ x_1 \circ \cdots \circ x_{i-k})\tau \circ s_{i-k+1} \circ \cdots \circ s_n \; = \; r\sigma \circ s_1 \circ \cdots \circ s_n \; = \; t \end{aligned}$$

  where we use the confluence of $\mathcal{U}$ in the first sequence.

- In the second case we have $C = s_1 \circ C'$. Clearly $C'[\ell\sigma] \xrightarrow{\mathsf{ri}}_{\mathcal{R}} C'[r\sigma]$. If $\mathrm{aa}(s_1) \leqslant 0$ or if $\mathrm{aa}(s_1)$ is undefined or if $\mathrm{aa}(s_1) > 0$ and the outermost $\circ$ has not been uncurried in the sequence from $t$ to $s$ then $s = s_1' \circ s' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow s_1 \circ C'[\ell\sigma] \xrightarrow{\mathsf{ri}}_{\mathcal{R}} s_1 \circ C'[r\sigma] = u$ with $s_1' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow s_1$ and $s' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow C'[\ell\sigma]$. If $\mathrm{aa}(s_1) > 0$ and the outermost $\circ$ has been uncurried in the sequence from $t$ to $s$ then we may write $s_1 = f \circ u_1 \circ \cdots \circ u_k$ where $k < \mathrm{aa}(f)$. We have $s = f_{k+1}(u_1', \ldots, u_k', s')$ for some term $s'$ with $s' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow C'[\ell\sigma]$ and $u_i' \; {}_{\mathcal{U}}^{*}\!\!\leftarrow u_i$ for $1 \leqslant i \leqslant k$. In both cases we obtain $s' \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}_\eta^+(\mathcal{R})} \cdot \; {}_{\mathcal{U}}^{*}\!\!\leftarrow C'[r\sigma]$ from the induction hypothesis. Since innermost rewriting is closed under contexts, the desired $s \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}_\eta^+(\mathcal{R})} \cdot \; {}_{\mathcal{U}}^{*}\!\!\leftarrow u$ follows. □

By Lemma 15 and the equivalence of rightmost innermost and innermost termination [16] we obtain the main result of this section.

**Theorem 16.** *An ATRS $\mathcal{R}$ is innermost terminating if $\mathcal{U}_\eta^+(\mathcal{R})$ is innermost terminating.* □

# 4 Derivational Complexity

In this section we investigate how the uncurrying transformation affects derivational complexity for full and innermost rewriting.

## 4.1 Full Rewriting

It is sound to use uncurrying as a preprocessor for proofs of upper bounds on the derivational complexity:

**Theorem 17.** *If $\mathcal{R}$ is a terminating ATRS then* $\mathrm{dc}_{\mathcal{R}}(n) \in \mathcal{O}(\mathrm{dc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n))$.

*Proof.* Consider an arbitrary maximal rewrite sequence $t_0 \to_{\mathcal{R}} t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \cdots \to_{\mathcal{R}} t_m$ which we can transform into the sequence

$$t_0{\downarrow_{\mathcal{U}}} \to^+_{\mathcal{U}^+_\eta(\mathcal{R})} t_1{\downarrow_{\mathcal{U}}} \to^+_{\mathcal{U}^+_\eta(\mathcal{R})} t_2{\downarrow_{\mathcal{U}}} \to^+_{\mathcal{U}^+_\eta(\mathcal{R})} \cdots \to^+_{\mathcal{U}^+_\eta(\mathcal{R})} t_m{\downarrow_{\mathcal{U}}}$$

using Lemma 7. Moreover, $t_0 \to^*_{\mathcal{U}^+_\eta(\mathcal{R})} t_0{\downarrow_{\mathcal{U}}}$ holds. Therefore, $\mathrm{dh}(t_0, \to_{\mathcal{R}}) \leqslant \mathrm{dh}(t_0, \to_{\mathcal{U}^+_\eta(\mathcal{R})})$. Hence $\mathrm{dc}_{\mathcal{R}}(n) \leqslant \mathrm{dc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n)$ holds for all $n \in \mathbb{N}$. □

Next we show that uncurrying preserves polynomial complexity. Hence we disregard duplicating (exponential complexity, cf. [9]) and empty (constant complexity) ATRSs. A TRS $\mathcal{R}$ is called *length-reducing* if $\mathcal{R}$ is non-duplicating and $|\ell| > |r|$ for all rules $\ell \to r \in \mathcal{R}$. The following lemma is an easy consequence of [9, Theorem 23]. Here for a relative TRS $\mathcal{R}/\mathcal{S}$ the derivational complexity $\mathrm{dc}_{\mathcal{R}/\mathcal{S}}(n)$ is based on the rewrite relation $\to_{\mathcal{R}/\mathcal{S}}$ which is defined as $\to^*_{\mathcal{S}} \cdot \to_{\mathcal{R}} \cdot \to^*_{\mathcal{S}}$.

**Lemma 18.** *Let $\mathcal{R}$ be a non-empty non-duplicating TRS over a signature containing at least one symbol of arity at least two and let $\mathcal{S}$ be a length-reducing TRS. If $\mathcal{R} \cup \mathcal{S}$ is terminating then* $\mathrm{dc}_{\mathcal{R}\cup\mathcal{S}}(n) \in \mathcal{O}(\mathrm{dc}_{\mathcal{R}/\mathcal{S}}(n))$. □

Note that the above lemma does not hold if the TRS $\mathcal{R}$ is empty.

**Theorem 19.** *Let $\mathcal{R}$ be a non-empty ATRS. If* $\mathrm{dc}_{\mathcal{R}}(n)$ *is in* $\mathcal{O}(n^k)$ *then* $\mathrm{dc}_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}}(n)$ *and* $\mathrm{dc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n)$ *are in* $\mathcal{O}(n^k)$.

*Proof.* Let $\mathrm{dc}_{\mathcal{R}}(n)$ be in $\mathcal{O}(n^k)$ and consider a maximal rewrite sequence of $\to_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}}$ starting from an arbitrary term $t_0$:

$$t_0 \to_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}} t_1 \to_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}} \cdots \to_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}} t_m$$

By Lemma 6 we obtain the sequence $t_0{\downarrow_{\mathcal{C}'}} \to_{\mathcal{R}} t_1{\downarrow_{\mathcal{C}'}} \to_{\mathcal{R}} \cdots \to_{\mathcal{R}} t_m{\downarrow_{\mathcal{C}'}}$. Thus, $\mathrm{dh}(t_0, \to_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}}) \leqslant \mathrm{dh}(t_0{\downarrow_{\mathcal{C}'}}, \to_{\mathcal{R}})$. Because $|t_0{\downarrow_{\mathcal{C}'}}| \leqslant 2|t_0|$, we obtain $\mathrm{dc}_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}}(n) \leqslant \mathrm{dc}_{\mathcal{R}}(2n)$. From the assumption the right-hand side is in $\mathcal{O}(n^k)$, hence $\mathrm{dc}_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}}(n)$ is in $\mathcal{O}(n^k)$. Since $\mathrm{dc}_{\mathcal{R}}(n)$ is in $\mathcal{O}(n^k)$, $\mathcal{R}$ must be non-duplicating and terminating. Because $\mathcal{U}$ is length-reducing, Lemma 18 yields that $\mathrm{dc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n)$ also is in $\mathcal{O}(n^k)$. □

In practice it is recommendable to investigate $\mathrm{dc}_{\mathcal{R}_\eta\downarrow_{\mathcal{U}}/\mathcal{U}}(n)$ instead of $\mathrm{dc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n)$, see [19]. The next example shows that uncurrying might be useful to enable criteria for polynomial complexity.

**Example 20.** Consider the ATRS $\mathcal{R}$ consisting of the two rules

$$\mathsf{add}\ x\ \mathsf{0} \to x \qquad\qquad \mathsf{add}\ x\ (\mathsf{s}\ y) \to \mathsf{s}\ (\mathsf{add}\ x\ y)$$

The system $\mathcal{U}^+_\eta(\mathcal{R})$ consists of the rules

$$\mathsf{add}_2(x,\mathsf{0}) \to x \qquad\qquad \mathsf{add}_2(x,\mathsf{s}_1(y)) \to \mathsf{s}_1(\mathsf{add}_2(x,y))$$
$$\mathsf{add}_1(x) \circ y \to \mathsf{add}_2(x,y) \qquad \mathsf{add} \circ x \to \mathsf{add}_1(x) \qquad \mathsf{s} \circ x \to \mathsf{s}_1(x)$$

The 2-dimensional TMI $\mathcal{M}$

$$\mathsf{add_2}_{\mathcal{M}}(\vec{x},\vec{y}) = \circ_{\mathcal{M}}(\vec{x},\vec{y}) = \begin{pmatrix}1\,1\\0\,1\end{pmatrix}\vec{x} + \begin{pmatrix}1\,1\\0\,1\end{pmatrix}\vec{y} \qquad \mathsf{add_1}_{\mathcal{M}}(\vec{x}) = \mathsf{s_1}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix}1\,0\\0\,1\end{pmatrix}\vec{x} + \begin{pmatrix}0\\1\end{pmatrix}$$

$$\mathsf{add}_{\mathcal{M}} = \mathsf{s}_{\mathcal{M}} = \mathsf{0}_{\mathcal{M}} = \begin{pmatrix}0\\1\end{pmatrix}$$

orients all rules in $\mathcal{U}^+_\eta(\mathcal{R})$ strictly, inducing a quadratic upper bound on the derivational complexity of $\mathcal{U}^+_\eta(\mathcal{R})$ according to Theorem 1 and by Theorem 17 also of $\mathcal{R}$. In contrast, the TRS $\mathcal{R}$ itself does not admit such an interpretation of dimension 2. To see this, we encoded the required condition as a satisfaction problem in non-linear arithmetic over the integers. MiniSmt [20][1] can prove this problem unsatisfiable by simplifying it into a trivially unsatisfiable constraint. Details can be inferred from the website mentioned in Footnote 4.

## 4.2 Innermost Rewriting

Next we consider innermost derivational complexity. Let $\mathcal{R}$ be an innermost terminating TRS. From a result by Krishna Rao [16, Section 5.1] which has been generalized by van Oostrom [15, Theorems 2 and 3] we infer that $\mathrm{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}}) = \mathrm{dh}(t, \xrightarrow{\mathsf{ri}}_{\mathcal{R}})$ holds for all terms $t$.

**Theorem 21.** *If $\mathcal{R}$ is an innermost terminating ATRS then* $\mathrm{idc}_{\mathcal{R}}(n) \in \mathcal{O}(\mathrm{idc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n))$.

*Proof.* Consider a maximal rightmost innermost rewrite sequence $t_0 \xrightarrow{\mathsf{ri}}_{\mathcal{R}} t_1 \xrightarrow{\mathsf{ri}}_{\mathcal{R}} t_2 \xrightarrow{\mathsf{ri}}_{\mathcal{R}} \cdots \xrightarrow{\mathsf{ri}}_{\mathcal{R}} t_m$. Using Lemma 15 we obtain a sequence

$$t_0 \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}^+_\eta(\mathcal{R})} t'_1 \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}^+_\eta(\mathcal{R})} t'_2 \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}^+_\eta(\mathcal{R})} \cdots \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{U}^+_\eta(\mathcal{R})} t'_m$$

for terms $t'_1, t'_2, \ldots, t'_m$ such that $t_i \to^*_{\mathcal{U}} t'_i$ for all $1 \leqslant i \leqslant m$. It follows that $\mathrm{dh}(t_0, \xrightarrow{\mathsf{i}}_{\mathcal{R}}) = \mathrm{dh}(t_0, \xrightarrow{\mathsf{ri}}_{\mathcal{R}}) \leqslant \mathrm{dh}(t_0, \xrightarrow{\mathsf{i}}_{\mathcal{U}^+_\eta(\mathcal{R})})$ and we conclude $\mathrm{idc}_{\mathcal{R}}(n) \in \mathcal{O}(\mathrm{idc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n))$. ☐

As Example 9 showed, uncurrying does not preserve innermost termination. Similarly, it does not preserve innermost polynomial complexity even if the original ATRS has linear innermost derivational complexity.

**Example 22.** Consider the non-duplicating ATRS $\mathcal{R}$ consisting of the two rules

$$\mathsf{f} \to \mathsf{s} \qquad\qquad \mathsf{f}\ (\mathsf{s}\ x) \to \mathsf{s}\ (\mathsf{s}\ (\mathsf{f}\ x))$$

Since the second rule is never used in innermost rewriting, $\mathrm{idc}_{\mathcal{R}}(n) \in \mathcal{O}(n)$ is easily shown by induction on $n$. We show that the innermost derivational complexity of $\mathcal{U}^+_\eta(\mathcal{R})$ is at least exponential. The TRS $\mathcal{U}^+_\eta(\mathcal{R})$ consists of the rules

$$\mathsf{f} \to \mathsf{s} \qquad \mathsf{f_1}(x) \to \mathsf{s_1}(x) \qquad \mathsf{f_1}(\mathsf{s_1}(x)) \to \mathsf{s_1}(\mathsf{s_1}(\mathsf{f_1}(x))) \qquad \mathsf{f} \circ x \to \mathsf{f_1}(x) \qquad \mathsf{s} \circ x \to \mathsf{s_1}(x)$$

and one can verify that $\mathrm{dh}(\mathsf{f}_1^n(\mathsf{s_1}(x)), \xrightarrow{\mathsf{i}}_{\mathcal{U}^+_\eta(\mathcal{R})}) \geqslant 2^n$ for all $n \geqslant 1$. Hence, $\mathrm{idc}_{\mathcal{U}^+_\eta(\mathcal{R})}(n+3) \geqslant 2^n$ for all $n \geqslant 0$.

[1] `http://cl-informatik.uibk.ac.at/software/minismt/`

Table 2: Innermost termination for 213 ATRSs.

| subterm | matrix (1) | matrix (2) | matrix (3) | matrix (4) |
|---|---|---|---|---|
| $-/+$ | $-/+$ | $-/+$ | $-/+$ | $-/+$ |
| 42 / 55 | 67 / 102 | 111 / 142 | 113 / 144 | 114 / 145 |

Table 3: (Innermost) derivational complexity for 195 (213) ATRSs.

| | TMI (1) | TMI (2) | TMI (3) | TMI (4) |
|---|---|---|---|---|
| | $-/+$ | $-/+$ | $-/+$ | $-/+$ |
| dc | 3 / 4 | 10 / 14 | 12 / 26 | 12 / 28 |
| idc | 3 / 4 | 10 / 14 | 12 / 26 | 12 / 28 |

## 5 Experimental Results

The results from this paper are implemented in the termination prover $\mathsf{T_TT_2}$ [12].[2] Version 7.0.2 of the termination problem data base (TPDB)[3] contains 195 ATRSs for full rewriting and 18 ATRSs for innermost rewriting. All tests have been performed on a single core of a server equipped with eight dual-core AMD Opteron® processors 885 running at a clock rate of 2.6 GHz and 64 GB of main memory.

Experiments[4] give evidence that uncurrying allows to handle significantly more systems. For proving innermost termination we considered two popular termination methods, namely the subterm criterion [7] and matrix interpretations [2] of dimensions one to four. The implementation of the latter is based on SAT solving (cf. [2]). For a matrix interpretation of dimension $d$ we used $5-d$ bits to represent natural numbers in matrix coefficients. An additional bit was used for intermediate results. Both methods are integrated within the dependency pair framework using dependency graph reasoning and usable rules as proposed in [3, 4, 6].

Table 2 shows the number of systems that could be proved innermost terminating. In the table $+$ $(-)$ indicates that uncurrying has (not) been used as preprocessing step, e.g., for the subterm criterion the number of successful proofs increases from 42 to 55 if uncurrying is used as a preprocessing transformation. For the setting based on matrix interpretations the gains are even larger. In the table, the numbers in parentheses denote the dimensions of the matrices.

Table 3 shows how uncurrying improves the performance of $\mathsf{T_TT_2}$ for derivational complexity. In this table we used TMIs as presented in Theorem 1. Coefficients of TMIs are represented with $\max\{2, 5-d\}$ bits; again an additional bit is allowed for intermediate results. If uncurrying is used as preprocessing transformation, TMIs can, e.g., show 14 systems to have at most quadratic derivational complexity while without uncurrying the method only applies to 10 systems. Since $\mathsf{T_TT_2}$ has no special methods for proving *innermost* derivational complexity, the numbers in rows dc and idc coincide.

[2] `http://cl-informatik.uibk.ac.at/software/ttt2/`
[3] `http://termination-portal.org/wiki/TPDB`
[4] `http://cl-informatik.uibk.ac.at/software/ttt2/10hor/`

## 6 Conclusion

In this paper we studied properties of the uncurrying transformation from [8] for innermost rewriting and (innermost) derivational complexity. The significance of these results has been confirmed empirically.

For proving (innermost) termination of applicative systems we mention transformation $\mathcal{A}$ [3] as related work. The main benefit of the approach in [3] is that in contrast to our setting no auxiliary uncurrying rules are necessary. However, transformation $\mathcal{A}$ only works for *proper* ATRSs without head variables in the (left- and) right-hand sides of rewrite rules. Here proper means that any constant always appears with the same applicative arity.

We are not aware of other investigations dedicated to (derivational) complexity analysis of ATRSs. However, we remark that transformation $\mathcal{A}$ preserves derivational complexity.This is straightforward from [11, Lemma 2.1(3)].

As future work we plan to incorporate the results for innermost termination into the dependency pair processors presented in [8].

## References


[1] F. Baader & T. Nipkow (1998): *Term Rewriting and All That*. Cambridge University Press.

[2] J. Endrullis, J. Waldmann & H. Zantema (2008): *Matrix Interpretations for Proving Termination of Term Rewriting*. Journal of Automated Reasoning 40(2-3), pp. 195–220. Available at `http://dx.doi.org/10.1007/s10817-007-9087-9`.

[3] J. Giesl, R. Thiemann & P. Schneider-Kamp (2005): *Proving and Disproving Termination of Higher-Order Functions*. In: Proc. 5th International Workshop on Frontiers of Combining Systems. LNCS 3717, pp. 216–231. Available at `http://dx.doi.org/10.1007/11559306_12`.

[4] J. Giesl, R. Thiemann, P. Schneider-Kamp & S. Falke (2006): *Mechanizing and Improving Dependency Pairs*. Journal of Automated Reasoning 37(3), pp. 155–203. Available at `http://dx.doi.org/10.1007/s10817-006-9057-7`.

[5] B. Gramlich (1995): *Abstract Relations between Restricted Termination and Confluence Properties of Rewrite Systems*. Fundamenta Informaticae 24(1/2), pp. 3–23.

[6] N. Hirokawa & A. Middeldorp (2005): *Automating the Dependency Pair Method*. Information and Computation 199(1-2), pp. 172–199. Available at `http://dx.doi.org/10.1016/j.ic.2004.10.004`.

[7] N. Hirokawa & A. Middeldorp (2007): *Tyrolean Termination Tool: Techniques and Features*. Information and Computation 205(4), pp. 474–511. Available at `http://dx.doi.org/10.1016/j.ic.2006.08.010`.

[8] N. Hirokawa, A. Middeldorp & H. Zankl (2008): *Uncurrying for Termination*. In: Proc. 15th International Conference on Logic for Programming, Artificial Intelligence and Reasoning. LNCS (LNAI) 5330, pp. 667–681. Available at `http://dx.doi.org/10.1007/978-3-540-89439-1_46`.

[9] N. Hirokawa & G. Moser (2008): *Automated Complexity Analysis Based on the Dependency Pair Method*. In: Proc. 4th International Joint Conference on Automated Reasoning. LNCS (LNAI) 5195, pp. 364–380. Available at `http://dx.doi.org/10.1007/978-3-540-71070-7_32`.

[10] D. Hofbauer & C. Lautemann (1989): *Termination Proofs and the Length of Derivations*. In: Proc. 3rd International Conference on Rewriting Techniques and Applications. LNCS 355, pp. 167–177. Available at `http://dx.doi.org/10.1007/3-540-51081-8_107`.

[11] R. Kennaway, J.W. Klop, M.R. Sleep & F.-J. de Vries (1996): *Comparing Curried and Uncurried Rewriting*. Journal of Symbolic Computation 21(1), pp. 15–39.

[12] M. Korp, C. Sternagel, H. Zankl & A. Middeldorp (2009): *Tyrolean Termination Tool 2*. In: *Proc. 20th International Conference on Rewriting Techniques and Applications. LNCS* 5595, pp. 295–304. Available at `http://dx.doi.org/10.1007/978-3-642-02348-4_21`.

[13] G. Moser, A. Schnabl & J. Waldmann (2008): *Complexity Analysis of Term Rewriting Based on Matrix and Context Dependent Interpretations*. In: *Proc. 28th International Conference on Foundations of Software Technology and Theoretical Computer Science. LIPIcs* 2, pp. 304–315. Available at `http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2008.1762`.

[14] F. Neurauter, H. Zankl & A. Middeldorp (2010): *Revisiting Matrix Interpretations for Polynomial Derivational Complexity of Term Rewriting*. In: *Proc. 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning. LNCS (ARCoSS)* 6397, pp. 550–564. Available at `http://dx.doi.org/10.1007/978-3-642-16242-8_39`.

[15] V. van Oostrom (2007): *Random Descent*. In: *Proc. 18th International Conference on Rewriting Techniques and Applications. LNCS* 4533, pp. 314–328. Available at `http://dx.doi.org/10.1007/978-3-540-73449-9_24`.

[16] M.R.K. Krishna Rao (2000): *Some Characteristics of Strong Innermost Normalization*. *Theoretical Computer Science* 239, pp. 141–164. Available at `http://dx.doi.org/10.1016/S0304-3975(99)00215-7`.

[17] TeReSe (2003): *Term Rewriting Systems*. *Cambridge Tracts in Theoretical Computer Science* 55, Cambridge University Press.

[18] J. Waldmann (2010): *Polynomially Bounded Matrix Interpretations*. In: *Proc. 21st International Conference on Rewriting Techniques and Applications. LIPIcs* 6, pp. 357–372. Available at `http://dx.doi.org/10.4230/LIPIcs.RTA.2010.357`.

[19] H. Zankl & M. Korp (2010): *Modular Complexity Analysis via Relative Complexity*. In: *Proc. 21st International Conference on Rewriting Techniques and Applications. LIPIcs* 6, pp. 385–400. Available at `http://dx.doi.org/10.4230/LIPIcs.RTA.2010.385`.

[20] H. Zankl & A. Middeldorp (2010): *Satisfiability of Non-Linear (Ir)rational Arithmetic*. In: *Proc. 16th International Conference on Logic for Programming, Artificial Intelligence and Reasoning. LNCS (LNAI)* 6355, pp. 481–500.