A solution to Zantema's problem

Alfons GESER
Universität Passau, Fakultät für Mathematik und Informatik
Postfach 2540, D-94030 Passau
Phone: +49 851 509 353, Email: geser@fmi.uni-passau.de

December 20, 1993

Abstract

Hans Zantema raised the question: Does the one-rule string rewriting system $\mathtt{00II} \to \mathtt{III000}$ terminate? Two successful termination proofs have been given by Nachum Dershowitz: One by minimal counterexample, and the other, by a forward closure argument. We give a different, more constructive, termination proof, following Zantema's original proof attempt, and using Bellegarde/Lescanne's transformation order method to verify a subtle transformation step.

Keywords: combinatorial problems, programming calculi, string rewriting, term rewriting, termination, transformation ordering

1 Introduction

A term rewriting system R terminates, if every R-derivation $t_1 \underset{R}{\to} t_2 \underset{R}{\to} \cdots$ is finite. Much of the success of term rewriting is due to the availability of powerful termination criteria. Although termination of rewrite systems is known to be undecidable, even for string rewriting systems [8], and even for left-linear, one-rule term rewriting systems [3], only few known rewrite systems have a *difficult* termination problem. (See [11] for one famous "exception of the rule".) Recently, Hans Zantema communicated a surprisingly small string rewriting system,

$$\mathtt{00II} \to \mathtt{III000} \tag{Z}$$

which is not at all easy to see terminating, and even more difficult to prove terminating. To my knowledge, there are two successful termination proofs of Nachum Dershowitz [6], and an unfinished proof attempt of Hans Zantema. Dershowitz gives one proof by minimal counterexample, and another one by forward closures. We will follow Zantema's way since it appears more intuitive. The detailed layout of his idea is sketched in section 3. Basically, Zantema's rewrite system (Z) is transformed into a rewrite system C (for "context"), which thereafter is transformed into another rewrite system B (for "block"). The transformations are chosen such that termination of rewrite system B entails termination of C, and termination of C entails termination of Z.

$$\begin{array}{cl}
Z & \text{terminating rewrite system} \\
\Big\downarrow & \qquad\quad \Uparrow \\
C & \text{terminating rewrite system} \\
\Big\downarrow & \qquad\quad \Uparrow \\
B & \text{terminating rewrite system}
\end{array}$$

Two parts of the proof which are new, and need special attention, are treated in the subsequent two sections: The proof that B is a terminating rewrite system, in section 4, and the justification of the subtle transformation step $C \mapsto B$, in section 5. The latter step is performed mechanically, using the "transformation order" approach of Françoise Bellegarde and Pierre Lescanne [1], [2]. The two steps can be understood independently of each other, but only together the two parts form a proof of termination of Z.

2 Basic notions

We assume that the reader is familiar with term rewriting, in particular with termination proofs. A comprehensive survey on termination of rewriting is [5].

A binary relation $\to \subseteq S \times S$ on a set S is said to terminate, if there is no infinite $\to$-derivation $t_1 \to t_2 \to \cdots$.

A *term rewriting system* R (or *rewrite system* for short) is any (usually finite) binary relation on terms. A binary relation $\to$ on terms is called *stable*, if $s \to t \Longrightarrow s\sigma \to t\sigma$ for every substitution σ, and terms s and t. A binary relation $\to$ on terms is called *monotonic*, if $s \to t \Longrightarrow c(s) \to c(t)$ for every context c, and terms s and t. The *R-rewrite relation* $\underset{R}{\to}$, a binary relation on terms, is defined as the stable and monotonic closure of R, i.e. the closure under application of a substitution and closure under context. We call R a *terminating rewrite system*, if its rewrite relation $\underset{R}{\to}$ terminates.

A binary relation on terms is called an *order* if it is irreflexive and transitive, and a *quasiorder* if it is reflexive and transitive. A quasiorder $\gtrsim$ defines an order $>$ by $s > t$, if $s \gtrsim t$ and not $t \gtrsim s$, and an equivalence relation $\sim$ by $s \sim t$, if $s \gtrsim t$ and $t \gtrsim s$. We say that $\gtrsim$ *strictly* satisfies a property P if both $\gtrsim$ and $>$ satisfy P. By abuse, we call a quasiorder $\gtrsim$ terminating if $>$ is terminating.

Given two quasiorders $\gtrsim_1$, $\gtrsim_2$, their lexicographic combination, $(\gtrsim_1, \gtrsim_2)$, is a quasiorder $\gtrsim$ defined by $\gtrsim =_{\text{def}} >_1 \cup (\sim_1 \cap \gtrsim_2)$. Strict stability, strict monotonicity, and termination are preserved by lexicographic combination.

The rewrite rule $l \to r$ *overlaps* with a rule $l' \to r'$ if there is a position p in l such that the subterm of l at p (call this term t) is not a variable, and $t\sigma = l'\sigma'$ holds for a pair of substitutions σ, σ'. If there is an overlap, then the most general pair $(r\sigma, t')$, where t' is $l\sigma$ at p replaced by $r'\sigma'$, is called the *critical pair* of $l \to r$ with $l' \to r'$ at p.

If R, S are rewrite systems, the set of (R, S)-critical pairs is the set of all critical pairs (s, t) of a rule from R with a rule from S, together with all pairs (t, s) where (s, t) is a critical pair of a rule from S with one from R.

R is called a *confluent* rewrite system, if its rewrite relation $\underset{R}{\to}$ is confluent, i.e. $\underset{R}{\overset{*}{\leftarrow}}\underset{R}{\overset{*}{\to}} \subseteq \underset{R}{\overset{*}{\to}}\underset{R}{\overset{*}{\leftarrow}}$ holds. Knuth and Bendix [10] showed that for terminating term rewriting systems R, confluence is decidable: There $\underset{R}{\to}$ is confluent if and only if, every (R, R)-critical pair (s, t) satisfies $s \underset{R}{\overset{*}{\to}}\underset{R}{\overset{*}{\leftarrow}} t$.

The relations R^+ and R^* denote the transitive, transitive-reflexive closure of R, respectively. R/S abbreviates for S^*RS^*, and $\leftrightarrow$ for the symmetric closure $\leftarrow \cup \to$ of $\to$. A rewrite rule $l \to r$ is called *non-erasing* if each variable in l also appears in r, and *left-linear* if each variable occurs at most once in l. A rewrite system is non-erasing, left-linear, respectively, if each of its rules is so.

3 The proof architecture

As a well-known fact, a string rewriting system can be considered a term rewriting system. To do this, every character is taken as a unary function, concatenation of strings becomes composition of functions, and the right context of a string is replaced by a variable (say, x).

The left context of the string is the context of the respective term. So we have functions $\mathtt{O}$, $\mathtt{I}$, of arity 1, and the rewrite system Z translates to

$$\mathtt{O}(\mathtt{O}(\mathtt{I}(\mathtt{I}(x)))) \to \mathtt{I}(\mathtt{I}(\mathtt{I}(\mathtt{O}(\mathtt{O}(\mathtt{O}(x))))))$$

We will however, for sake of simplicity, stick to the string representation, and instead of context and substitution, we will speak of *left and right contexts*. Accordingly, stability and monotonicity will be named *closure under left and right context*, respectively. Concatenation will be denoted by juxtaposition, and the empty string will be denoted by ε. For a survey on string rewriting see [9].

The termination proof follows Zantema's proof attempt closely in spirit. The original rewrite system Z is transformed in two steps to a rewrite system for which we can provide a termination proof.

3.1 Step 1

The first transformation step is easy: We consider the rule in every pair of left and right context each of length 1. We thus get the four-rule rewrite system C.

(C1) $$\mathtt{OOOIIO} \to \mathtt{OIIIOOOO}$$

(C2) $$\mathtt{OOOIII} \to \mathtt{OIIIOOOI}$$

(C3) $$\mathtt{IOOIIO} \to \mathtt{IIIIOOOO}$$

(C4) $$\mathtt{IOOIII} \to \mathtt{IIIIOOOI}$$

It is obvious that an infinite Z-derivation, starting with term t, issues an infinite C-derivation, starting e.g. with $\mathtt{O}t\mathtt{O}$. For this reason, we can prove termination of $\underset{Z}{\to}$ by termination of $\underset{C}{\to}$. The purpose of this transformation is simply to split up Z into four cases which we may treat specifically.

3.2 Step 2

Each rule increases the total number of zeroes, as well as the total number of ones. However if we count *packages* of zeroes (separated by ones), every rule, except rule (C2) decreases the number of packages. The task of step 2 is to handle rule (C2) on this account.

The main idea in Zantema's termination proof attempt is now that there is no "information flow" between the left half, $\mathtt{OIII}$, and the right half, $\mathtt{OOOI}$, of the right hand side, $\mathtt{OIIIOOOI}$, of this rule. More precisely, there is no redex that needs a proper part of both the left and right half. This fact allows one to introduce a "barrier", or "block", $\Box$, between the two, without changing the applicability of rewrite steps. One may thus reduce the termination proof of rewrite system C to that of the following rewrite system B.

(B1) $$\mathtt{OOOIIO} \to \mathtt{OIIIOOOO}$$

(B2) $$\mathtt{OOOIII} \to \mathtt{OIII}\,\Box\,\mathtt{OOOI}$$

(B3) $$\mathtt{IOOIIO} \to \mathtt{IIIIOOOO}$$

(B4) $$\mathtt{IOOIII} \to \mathtt{IIIIOOOI}$$

This is the second transformation. For its correctness, we will provide a formal, mechanizable, proof in section 5. Zantema designed for a further transformation to split rule (B2) into two rules

(1) $$\mathtt{OOOIII} \to \mathtt{OIII}$$

(2) $$\mathtt{OOOIII} \to \mathtt{OOOI}$$

but I wonder whether this step is still termination preserving. Here we leave Zantema's proof line, and rather prove by a recursive-path-order like construction that $\underset{B}{\to}$ terminates. The proof is done in the next section.

4 B is a terminating rewrite system

The proof proceeds in three steps. First we try to get rid of block symbols, and cut down rule (B2). Next we compare the number of "0" packages. Third the lengths of packages is compared lexicographically, and the order is finished by a comparison of the total number of "I" symbols. We try to give striking reasons for our line of argument.

4.1 Step 1

Above we introduced the block symbol, $\Box$, to separate the two parts of the right hand side in rule (B2). Thus, as we will show, it is enough to compare the left hand side with each part of the right hand side separately. For this purpose, we define a quasiorder $\gtrsim_{seq} \subseteq \{\mathtt{0}, \mathtt{I}, \Box\}^*$, based on an appropriate given quasiorder $\gtrsim_0 \subseteq \{\mathtt{0}, \mathtt{I}\}^*$ on strings not containing $\Box$.

Definition 4.1 ($\gtrsim_{seq}$) *Let $s_1, \ldots, s_m, t_1, \ldots, t_n \in \{\mathtt{0}, \mathtt{I}\}^*$. Then*

$s_1 \Box \cdots \Box s_m \gtrsim_{seq} t_1 \Box \cdots \Box t_n$, *if*

1. $m > n = 0$, *or*
2. $m, n > 0$, *and*
   (a) $s_1 >_0 t_1$ *and* $s_1 \Box \cdots \Box s_m >_{seq} t_2 \Box \cdots \Box t_n$, *or*
   (b) $s_1 \sim_0 t_1$ *and* $s_2 \Box \cdots \Box s_m \gtrsim_{seq} t_2 \Box \cdots \Box t_n$, *or*
   (c) $s_2 \Box \cdots \Box s_m \gtrsim_{seq} t_1 \Box \cdots \Box t_n$

The reader familiar with term rewriting may recognize that $\gtrsim_{seq}$ is defined on $\gtrsim_0$ by the same scheme as the recursive path order [4] is defined on a precedence order. As a corollary, one gets from there: If $\gtrsim_0$ is a terminating quasiorder, then so is $\gtrsim_{seq}$.

We will, unusually, not employ closure under left and right contexts so as to infer a decrease of each rewrite step from a decrease of each rule. Instead, we will directly prove that an arbitrary rewrite step decreases.

To order an arbitrary B-rewrite step

$$l_1 \Box \cdots \Box l_m s r_1 \Box \cdots \Box r_n >_{seq} l_1 \Box \cdots \Box l_m t r_1 \Box \cdots \Box r_n$$

where $l_1, \ldots, l_m, r_1, \ldots, r_n \in \{\mathtt{0}, \mathtt{I}\}^*$, and $s \to t$ is a rewrite rule in B, it is sufficient to show that $l_m s r_1 >_0 l_m t r_1$ in the case where t contains no block symbol, and $l_m s r_1 >_0 l_m t_1$, $l_m s r_1 >_0 t_2 r_1$ in the case $t = t_1 \Box t_2$. In other words, we have used $\Box$ only to split the problematic right hand side of rule (B2). Here is a good point to collect together what we still have to prove.

Summary 1 *B is a terminating rewrite system, if $\gtrsim_0$ is a terminating quasiorder that satisfies the following inequations for each pair $l, r \in \{\mathtt{0}, \mathtt{I}\}^*$.*

(I1)	$l\mathtt{000II0}r >_0 l\mathtt{0III0000}r$
(I2)	$l\mathtt{000III}r >_0 l\mathtt{0III}$
(I3)	$l\mathtt{000III}r >_0 \mathtt{000I}r$
(I4)	$l\mathtt{I00II0}r >_0 l\mathtt{IIII0000}r$
(I5)	$l\mathtt{I00III}r >_0 l\mathtt{IIII000I}r$

Informally, we have got rid of the $\Box$ symbol again, but moreover, each inequation does no longer increase the number of packages.

4.2 Step 2

Next we are going to develop the quasiorder $\gtrsim_0 \subseteq \{\mathtt{0},\mathtt{I}\}^*$ on strings not containing the block symbol, as a lexicographic combination of quasiorders $\gtrsim_1$, $\gtrsim_2$, $\gtrsim_3$,

$$\gtrsim_0 =_{\text{def}} (\gtrsim_1, \gtrsim_2, \gtrsim_3)$$

so as to satisfy these inequations. For this purpose we interpret a string $s = \mathtt{0}^{m_1}\mathtt{I}^{n_1}\cdots\mathtt{0}^{m_k}\mathtt{I}^{n_k}$ as two sequences $(m_1,\ldots,m_k)$ and $(n_1,\ldots,n_k)$, which give the lengths of $\mathtt{0}$ and $\mathtt{I}$ packages in s, respectively, each from left to right. For uniqueness, let us require that k is as small as possible. We will write $|s|$ for k, and s_i for m_i.

The first order, $\gtrsim_1$, compares the number of packages

$$s \gtrsim_1 t, \quad \text{if} \quad |s| \geq |t|$$

This way, we can solve inequation (I4), and keep the other inequations, for the case $l = r = \varepsilon$. To carry over this for the case of nonempty l or r, we use the following properties.

Proposition 4.1 *Let $s,t,l,r \in \{\mathtt{0},\mathtt{I}\}^*$. Then the following holds.*

1. *If s and t begin with the same character, then $|s| - |t| = |ls| - |lt|$.*
2. *If s and t end with the same character, then $|s| - |t| = |sr| - |tr|$.*
3. *If s begins with $\mathtt{0}$, and l contains at least one $\mathtt{I}$, then $|ls| > |s|$.*
4. *If s ends with $\mathtt{I}$, and r contains at least one $\mathtt{0}$, then $|sr| > |s|$.*

It is straightforward to solve thus

- inequation (I4) completely,
- inequation (I2) for the case where r contains at least one $\mathtt{0}$,
- inequation (I3) for the case where l contains at least one $\mathtt{I}$

This leaves us:

Summary 2 *B is a terminating rewrite system, if $\gtrsim_2$ and $\gtrsim_3$ are terminating quasiorders such that $\gtrsim_0$ satisfies (I1), (I5), and, for all $x \in \mathbb{N}$,*

(I2')	$l\mathtt{000IIII}^x >_0 l\mathtt{0III}$
(I3')	$\mathtt{0}^x\mathtt{000III}r >_0 \mathtt{000I}r$

We remark that for these remaining comparisons $s >_0 t$, we may rely on $|s| = |t|$ and on the fact that s and t have the same first character and the same last character or are both empty.

4.3 Step 3

Since now the length of the sequences is bounded, the sequence components $(m_1,\ldots,m_k)$ of s and $(m'_1,\ldots,m'_k)$ of t may be compared lexicographically. It is well known that lexicographic extension on sequences of bounded length preserves termination. It is advisable to compare the lengths of "$\mathtt{0}$" packages from left to right, ignoring the lengths of intermediate "$\mathtt{I}$" packages. A good reason for this decision is that each rule in question takes two zeroes from some package $\mathtt{0}^{s_i}$ and puts three zeroes into the next package $\mathtt{0}^{s_{i+1}}$ at the right. Rule (I5) is an exception on this account since it removes the *last* pair of zeroes from a package such that the three new zeroes appear one position to much to the left. To try to solve this by a comparison of "I" packages seems to be a bad idea. For, a lexicographic comparison of consecutive packages of ones from left to right fails at the set of pairs $m+1 > m+4$, and a comparison from right to left cannot be suitably interleaved with the corresponding comparison of "0" from left to right. We find that we better compare $m+3 > m+1$ (for (I1) and for (I2')) as well as $2 > 3$ (for (I5)). To achieve this, even lengths are added a "*malus*" of 2 before comparison. This way, 2 is lifted to 4 and so becomes greater than 3, which is not lifted. This *malus* is formalized by the mapping raise : $\mathbb{N} \to \mathbb{N}$ defined by

$$\text{raise}(x) =_{\text{def}} \begin{cases} x+2 & \text{if } x \text{ is even} \\ x & \text{else} \end{cases}$$

Note that $\text{raise}(x) = \text{raise}(y)$, if and only if $x = y$ holds. Moreover, $x - y > 1$ implies $\text{raise}(x) > \text{raise}(y)$. There is, however, a small problem with closure under left and right contexts. For instance, $\mathtt{00} >_2 \mathtt{000}$ due to $\text{raise}(2) = 4 > 3 = \text{raise}(3)$, but $\mathtt{000} <_2 \mathtt{0000}$ due to $\text{raise}(3) = 3 < 6 = \text{raise}(4)$. Informally, pairs of zero packages whose lengths differ by 1, like 00 and 000 should compare only when protected by I contexts to the left and to the right: $\mathtt{I00I} >_2 \mathtt{I000I}$ is safe. Fortunately, rule (I5) is of this protected form. We thus arrive at the following definition.

Definition 4.2 ($\gtrsim_2$) *$s \gtrsim_2 t$, if*

1. *$|s| = |t|$, and*
2. *s and t begin with the same character and end with the same character, or $s = t = \varepsilon$, and*
3. *for all i, $s_i = t_i$ holds, or there is an i such that both*
   (a) *$\text{raise}(s_i) > \text{raise}(t_i)$, and*
   (b) *$1 < i$ and $n_i \neq 0$, or $s_i - t_i > 1$, and*
   (c) *$s_j = t_j$ for all $j < i$.*

We are going to prove that $\gtrsim_2$ is transitive (which is not obvious). Let $s \gtrsim_2 t \gtrsim_2 u$. We claim that $s \gtrsim_2 u$ holds. The case where $s = t$ or $t = u$ is trivial. Otherwise, let i be the smallest index where $s_i \neq t_i$, and let i' be the smallest index where $t_{i'} \neq u_{i'}$ holds. If $i \neq i'$, then the smaller of i, i' is the wanted index i'' where $\text{raise}(s_{i''}) > \text{raise}(u_{i''})$. So let $i = i'$ hold. In this case, either $1 < i = i' < k$, then i is the wanted index, or both $s_i - t_i > 1$ and $t_i - u_i > 1$ hold. In the latter case, $s_i - u_i > 1$ follows.

Next we prove that $\gtrsim_2$ is strictly closed under left and right contexts. Let $s \gtrsim_2 t$. If $s = t = \varepsilon$, then $\mathtt{0}s = s\mathtt{0} \gtrsim_2 \mathtt{0}t = t\mathtt{0}$, as well as $\mathtt{I}s = s\mathtt{I} \gtrsim_2 \mathtt{I}t = \mathtt{I}$ hold, due to $s_1 = t_1$. Otherwise, if both s and t begin with 0, or both s and t begin with I, then $(\mathtt{0}s)_1 - (\mathtt{0}t)_1 = (s_1+1) - (t_1+1) = s_1 - t_1$. According to the definition of $\gtrsim_2$, this difference is either 0 or greater than 1. Anyway then, the comparison of $s \gtrsim_2 t$ carries over to $\mathtt{0}s \gtrsim_2 \mathtt{0}t$ in a straightforward way. If they begin with 0, then $(\mathtt{I}s)_1 = 1 = (\mathtt{I}t)$, and again, the comparison carries over, with indices i and j incremented. If s and t both begin with I, then the comparison trivially carries over to $\mathtt{I}s \gtrsim_2 \mathtt{I}t$.

Using $\gtrsim_2$, we can now handle the inequations (I1), (I2'), and (I5) completely, and (I3') for the case $x > 0$. Now the last order $\gtrsim_3$ should be designed so as to satisfy the missing inequation

$$\mathtt{000III}r >_3 \mathtt{000I}r$$

This is no longer difficult. For instance, we may count the total number of I symbols.

$s = \mathtt{0}^{m_1}\mathtt{I}^{n_1}\cdots\mathtt{0}^{m_k}\mathtt{I}^{n_k} \gtrsim_3 t = \mathtt{0}^{m_1}\mathtt{I}^{n'_1}\cdots\mathtt{0}^{m_k}\mathtt{I}^{n'_k}$, if
$n_1 + \cdots + n_k \geq n'_1 + \cdots + n'_k$

Obviously, $\gtrsim_3$ is a quasiorder, closed under left and right contexts, and terminating.

We claim that $\gtrsim_0$ terminates. The basic argument is that a terminating order, like $>$ on $\mathbb{N}$, extended to a lexicographic order on sequences, like $\gtrsim_2$, terminates provided there is a length bound. Such a length bound is ascertained by the $\gtrsim_1$ comparison. As $\gtrsim_1$, $\gtrsim_2$, $\gtrsim_3$ terminate, so does their lexicographic combination, $\gtrsim_0$.

Summary 3 *Every application of a rule in B strictly decreases by $\gtrsim_{seq}$, and $\gtrsim_{seq}$ is a terminating quasiorder.*

This finishes the termination proof for rewrite system B.

Theorem 1 *The rewrite relation $\underset{B}{\longrightarrow}$ terminates.*

5 The transformation

Now let us consider the transition from rewrite system C to B more precisely. The transition can be described by a transformer rewrite system T. As it will turn out, T needs the four rules

(T1)	$\mathtt{000I000I} \to \mathtt{000I} \,\Box\, \mathtt{000I}$
(T2)	$\mathtt{000I0III} \to \mathtt{000I} \,\Box\, \mathtt{0III}$
(T3)	$\mathtt{0III000I} \to \mathtt{0III} \,\Box\, \mathtt{000I}$
(T4)	$\mathtt{0III0III} \to \mathtt{0III} \,\Box\, \mathtt{0III}$

The purpose of T is to put a block symbol, $\Box$, at where we claim information cannot pass through. Rewrite system B is so nothing but the T-normal form of C. (The right hand side of rule (C2) has been rewritten by rule (T3).) To show that termination of the rewrite system B entails termination of the rewrite system C, we use the *transformation order* method.

Theorem 2 (Transformation order, [1], [2])
If

1. *$B \cup T$ is a terminating rewrite system,*
2. *T is a confluent rewrite system,*
3. *T is non-erasing and left-linear, and*
4. *every (T, B)-critical pair (s, t) is* cooperative, *i.e. it satisfies $s\ (\underset{B}{\longrightarrow} \,/\, \underset{T}{\longrightarrow})^{+} \overset{*}{\underset{T}{\longleftarrow}} t$,*

then even

$$> =_{\text{def}} \ (\underset{B}{\longrightarrow} \,/\, \underset{T}{\longleftrightarrow})^{+} \cup \overset{+}{\underset{T}{\longrightarrow}}$$

terminates. Moreover, every rewrite system C that satisfies $C \subseteq >$, is a terminating rewrite system.

For the proof of termination of $>$, we refer to [1] or [2]. See also [7] for a generalized version. It is easy to show that $>$ is transitive, so an order. By definition through rewrite relations, $>$ is trivially stable and monotonic. Together with termination this cares for the proof that C is a terminating rewrite system.

For our example, it is obvious that $C \subseteq >$ holds. Rule (C2) for instance satisfies

$$\mathtt{000III} \underset{(B2)}{\longrightarrow} \mathtt{0III} \,\Box\, \mathtt{000I} \underset{(T3)}{\longleftarrow} \mathtt{0III000I}$$

So let us check against the conditions of the theorem. First, we have to extend the termination proof of rewrite system B towards one of $B \cup T$. This is trivial since we can order the T rules with the termination order above. String rewriting systems, considered as term rewriting systems are always left-linear and non-erasing. Next we have to check whether the critical pair conditions hold.

Below we will develop the derivation which is, by the way, a beautiful illustration for the elegance of the transformation order method. To get a short presentation, we will schematize critical pair diagrams, and for this purpose we will employ schema variables x, y, z, etc. each to be instantiated by $\mathtt{0}$ or $\mathtt{I}$. At the peak string we mark the overlapping substring by a frame box, everywhere else the redex is underlined.

First we prove that every (T,T)-critical pair is locally confluent. In fact we see that T is even *strongly* confluent. (At most one step is needed to join the critical pair.)

(3)
$$\begin{tikzcd}
\mathtt{0}xx\mathtt{I00}\boxed{\mathtt{0I}}\mathtt{II0}zz\mathtt{I} \arrow[r, "T"'] \arrow[d, "T"] & \underline{\mathtt{0}xx\mathtt{I000I}}\mathtt{II}\,\square\,\mathtt{0}zz\mathtt{I} \arrow[d, "T"] \\
\mathtt{0}xx\mathtt{I}\,\square\,\mathtt{00}\underline{\mathtt{0III0}zz\mathtt{I}} \arrow[r, "T"'] & \mathtt{0}xx\mathtt{I}\,\square\,\mathtt{000III}\,\square\,\mathtt{0}zz\mathtt{I}
\end{tikzcd}$$
(T1), (T3) with (T3), (T4)

(4)
$$\begin{tikzcd}
\mathtt{0}xx\mathtt{I}\boxed{\mathtt{0}yy\mathtt{I}}\mathtt{0}zz\mathtt{I} \arrow[r, "T"'] \arrow[d, "T"] & \underline{\mathtt{0}xx\mathtt{I0}yy\mathtt{I}}\,\square\,\mathtt{0}zz\mathtt{I} \arrow[d, "T"] \\
\mathtt{0}xx\mathtt{I}\,\square\,\underline{\mathtt{0}yy\mathtt{I0}zz\mathtt{I}} \arrow[r, "T"'] & \mathtt{0}xx\mathtt{I}\,\square\,\mathtt{0}yy\mathtt{I}\,\square\,\mathtt{0}zz\mathtt{I}
\end{tikzcd}$$
(T1), (T3) with (T1), (T2), and

This makes a total of 12 (T,T)-critical pairs.

Next we prove that every (T,B)-critical pair is cooperative. First for the critical pairs of rule (B2). Observe that in addition to rule (T3) the rules (T1), (T2), and (T4) are needed to close the diagrams. This explains our choice of T above.

(5)
$$\begin{tikzcd}
\mathtt{00}\boxed{\mathtt{0III}}\mathtt{0}xx\mathtt{I} \arrow[r, "B"'] \arrow[d, "T"] & \mathtt{0III}\,\square\,\underline{\mathtt{000I0}xx\mathtt{I}} \arrow[d, "T"] \\
\underline{\mathtt{000III}}\,\square\,\mathtt{0}xx\mathtt{I} \arrow[r, "B"'] & \mathtt{0III}\,\square\,\mathtt{000I}\,\square\,\mathtt{0}xx\mathtt{I}
\end{tikzcd}$$
(B2) with (T3), (T4)

(6)
$$\begin{tikzcd}
\mathtt{0}xx\mathtt{I}\boxed{\mathtt{000I}}\mathtt{II} \arrow[r, "B"'] \arrow[d, "T"] & \underline{\mathtt{0}xx\mathtt{I0III}}\,\square\,\mathtt{000I} \arrow[d, "T"] \\
\mathtt{0}xx\mathtt{I}\,\square\,\underline{\mathtt{000III}} \arrow[r, "B"'] & \mathtt{0}xx\mathtt{I}\,\square\,\mathtt{0III}\,\square\,\mathtt{000I}
\end{tikzcd}$$
(T1), (T3) with (B2)

Notice that the diagrams work the same way for rule (C2) instead of (B2). And here are the other critical pairs.

(7)
$$\begin{tikzcd}
x\mathtt{00II}\boxed{\mathtt{0}}yy\mathtt{I0}zz\mathtt{I} \arrow[r, "B"'] \arrow[d, "T"] & x\mathtt{III000}\underline{\mathtt{0}yy\mathtt{I0}zz\mathtt{I}} \arrow[d, "T"] \\
\underline{x\mathtt{00II0}}yy\mathtt{I}\,\square\,\mathtt{0}zz\mathtt{I} \arrow[r, "B"'] & x\mathtt{III0000}yy\mathtt{I}\,\square\,\mathtt{0}zz\mathtt{I}
\end{tikzcd}$$
(B1), (B3) with (T1), ..., (T4)

(8)
$$\begin{tikzcd}
\mathtt{0}xx\mathtt{I}\boxed{\mathtt{000I}}\mathtt{I0} \arrow[r, "B"'] \arrow[d, "T"] & \underline{\mathtt{0}xx\mathtt{I0III}}\mathtt{0000} \arrow[d, "T"] \\
\mathtt{0}xx\mathtt{I}\,\square\,\underline{\mathtt{000II0}} \arrow[r, "B"'] & \mathtt{0}xx\mathtt{I}\,\square\,\mathtt{0III0000}
\end{tikzcd}$$
(T1), (T3) with (B1)

(9)
$$\begin{tikzcd}
\mathtt{I0}\boxed{\mathtt{0III}}\mathtt{0}xx\mathtt{I} \arrow[r, "B"'] \arrow[d, "T"] & \mathtt{IIII}\underline{\mathtt{000I0}xx\mathtt{I}} \arrow[d, "T"] \\
\underline{\mathtt{I00III}}\,\square\,\mathtt{0}xx\mathtt{I} \arrow[r, "B"'] & \mathtt{IIII000I}\,\square\,\mathtt{0}xx\mathtt{I}
\end{tikzcd}$$
(B4) with (T3), (T4)

Altogether, we count 24 (T, B)-critical pairs.

Summary 4 *The critical pair diagrams prove the following:*

1. *Every (T, T)-critical pair (s, t) satisfies $s \underset{T}{\rightarrow}\underset{T}{\leftarrow} t$.*
2. *Every (T, B)-critical pair (s, t) satisfies $s \underset{B}{\rightarrow}\underset{T}{\leftarrow} t$.*
3. *Every (T, C)-critical pair (s, t) satisfies $s \underset{C}{\rightarrow}\underset{T}{\leftarrow} t$.*

Now that all conditions for the transformation order are satisfied we may conclude:

Theorem 3 *C is a terminating rewrite system.*

Conclusions and future work

We gave a proof that Hans Zantema's string rewrite system `OOII` $\rightarrow$ `IIIOOO` terminates. The proof proceeds by transforming the original system Z in two steps (via a rewrite system C) to a system B. The latter system can be proven terminating by a recursive-path-order like construction on top of a lexicographic combination of hand-made interpretations. The termination proof of B is interesting as it shows how to work with wellfounded orders, having the subterm property, and with monotonic, wellfounded orders, without going into the trap of simplification orders.

The main goal of this article, however, is to demonstrate how the transformation order method of Françoise Bellegarde and Pierre Lescanne provides an elegant, formal, and mechanizable correctness proof for the intuitive, uncertain transformation step from C to B: The introduction of a "barrier" symbol, $\Box$, defined by a rewrite system T, whose purpose is to separate parts of a term that cannot communicate with each other. The method that we have exercised here can also be applied for other termination problems where the basic termination argument is that parts can be separated.

In addition to the termination proof, as the critical pair diagrams easily carry over to rule (C2) instead of (B2), we have:

Corollary 1 *T-rewrite steps (in either direction) do not change the length of C-derivations.*

This property might be useful to decompose, using T-rewrite steps, the following conjecture of Zantema's (personal communication) on the length of rewrite derivations.

Zantema's conjecture *There is some constant $c \in \mathbb{N}$ such that any C-rewrite derivation starting from a string of n characters has length at most $2n + c$.*

Acknowledgements

Thanks to all participants of the *First Termination Workshop* at St. Andrews, Scotland, most of all to Pierre Lescanne and Hans Zantema, for many hot discussions that finally led to this article. In particular I would like to thank Ursula Martin and the staff for providing such an excellent scientific atmosphere.

References

[1] Françoise Bellegarde and Pierre Lescanne. Transformation ordering. In *2nd TAPSOFT*, pages 69–80. Springer Lecture Notes in Computer Science 249, 1987.

[2] Françoise Bellegarde and Pierre Lescanne. Termination by completion. *Applicable Algebra in Engineering, Communication, and Computation*, 1:79–96, 1990.

[3] Max Dauchet. Simulation of turing machines by a left-linear rewrite rule. In *Proc. 3rd Int. Conf. Rewriting Techniques and Applications*, pages 109–120. LNCS 355, 1989.

[4] Nachum Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17(3):279–301, March 1982.

[5] Nachum Dershowitz. Termination of rewriting. *J. Symbolic Computation*, 3(1&2):69–115, Feb./April 1987. Corrigendum: 4, 3, Dec. 1987, 409-410.

[6] Nachum Dershowitz and Charles Hoot. Topics in termination. In Claude Kirchner, editor, *5th Conf. Rewriting Techniques and Applications*, pages 198–212. Springer Lecture Notes in Computer Science 690, 1993.

[7] Alfons Geser. *Relative termination.* PhD thesis, Fakultät für Mathematik und Informatik, Universität Passau, Passau, Germany, 1990. Also available as: Report 91-03, Ulmer Informatik-Berichte, Universität Ulm, 1991.

[8] Gérard Huet and Dallas Lankford. On the uniform halting problem for term rewriting systems. Technical Report 283, INRIA, 1978.

[9] M. Jantzen. Confluent string rewriting and congruences. *Bulletin of the EATCS*, 28:52–72, February 1986.

[10] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[11] Pierre Lescanne. Rewrite orderings and termination of rewrite systems. In A. Tarlecki, editor, *Proc. Mathematical Foundations of Computer Science 16*, volume 520 of *Springer Lecture Notes in Computer Science*, pages 17–27. Springer Verlag, 1991.