A complete characterization of termination of $\mathsf{0}^p\,\mathsf{1}^q \quad\to\quad \mathsf{1}^r\,\mathsf{0}^s$

Hans ZANTEMA
Universiteit Utrecht, P.O. Box 80089, NL-3508 Utrecht, The Netherlands
E-mail: `hansz@cs.ruu.nl`
and
Alfons GESER
Lehrstuhl für Programmiersysteme, Universität Passau, D-94030 Passau, Germany
E-mail: `geser@fmi.uni-passau.de`

Abstract

We completely characterize termination of one-rule string rewriting systems of the form $\mathsf{0}^p\,\mathsf{1}^q \to \mathsf{1}^r\,\mathsf{0}^s$ for every choice of positive integers p, q, r, and s. For the simply terminating cases, we give a sharp estimate of the complexity of derivation lengths.

1 Introduction

A term rewriting system R terminates, if every R-derivation $t_1 \to_R t_2 \to_R \cdots$ is finite. Much of the success of term rewriting is due to the availability of powerful termination criteria. String rewriting is a special case of term rewriting where function symbols are of arity 1, and may be taken as characters.

Termination of term rewriting systems is known to be undecidable, even for the special case of string rewriting systems [12], and even for left-linear, one-rule term rewriting systems [5]. The question whether termination is decidable for one-rule string rewriting systems is still open. In this paper we give a decision procedure for a non-trivial subclass of one-rule string rewriting systems, namely

$$(Z) \qquad \mathsf{0}^p\,\mathsf{1}^q \to \mathsf{1}^r\,\mathsf{0}^s$$

for positive integer numbers p, q, r, s. More precisely, we prove the following theorem.

Theorem 1.1 *Let p, q, r, s denote positive integer numbers. Then the system $\mathsf{0}^p\,\mathsf{1}^q \to \mathsf{1}^r\,\mathsf{0}^s$ terminates if, and only if*

1. *$p \geq s$, or*
2. *$q \geq r$, or*
3. *$p < s < 2p$ and $q \not| r$, or*
4. *$q < r < 2q$ and $p \not| s$.*

Here $x|y$ means that x is a divisor of y. We show that this class contains

1. simply terminating systems (cases 1 and 2) of
   (a) linear ($p > s$ or $q > r$),
   (b) quadratic ($p = s$, $q = r$), and
   (c) exponential derivation lengths ($p = s$, $q < r$, or $p < s$, $q = r$),
2. non-terminating systems ($2p \leq s$, $2q \leq r$, or $p < s < 2p$, $q < r$, $q|r$, or $q < r < 2q$, $p < s$, $p|s$), and
3. non-simply terminating systems (cases 3 and 4).

The main goal of the paper is not only proving this theorem, but also showing how a very difficult termination proof can be given in a purely transformational style. We give such a proof for the non-simply terminating case. Instead of giving a complicated recursively defined ordering as is quite usual in termination proofs, we transform the system a number of times. For every transformation the termination of the original system follows from termination of the transformed system, which is equivalent to saying that the transformation preserves non-termination. This preservation follows from theorems that are generally applicable. These theorems follow the underlying ideas of *transformation ordering* [2, 3] and *dummy elimination* [9].

The paper is organized as follows. First we treat the case of simple termination in section 3. Here the termination proof is routine and we extend our attention to derivation lengths, on which we obtain sharp bounds. In section 4 we deal with the non-terminating cases.

The remainder and the main part of the paper is devoted to the difficult, non-simply terminating case $p < s < 2p$, $q < r$, $q \not| r$; the other non-simply terminating case is obtained by symmetry. In section 5, we describe how the system is transformed a number of times. In one step a fresh symbol $\Box$ is introduced in a right hand side of a rule, whose purpose is to stand there as a proof for the absence of information flow. This step is called *dummy introduction*, and is treated in detail in section 6. We employ an impoverished form of transformation order to prove preservation of non-termination. The next step is *dummy elimination*, the symbol $\Box$ is removed again by splitting the rule $l \to r_1 \Box r_2$ into two rules $l \to r_1$, $l \to r_2$. In section 8 the representation of strings over $\mathtt{0}$ and $\mathtt{1}$ is changed by describing such a string by $\mathtt{0}^m\,\mathtt{1}^n$ packages. In this representation termination of the final system is proved by a lexicographical argument. We conclude by comparing related work.

2 Basic notions

We assume that the reader is familiar with term rewriting, and in particular with termination proofs. A comprehensive survey on termination of rewriting is [7].

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

For $\to$ a binary relation, $\to^{-1}$ and $\leftarrow$ denote the inverse relation: $s \leftarrow t$ holds if $t \to s$. Likewise $\to^+$ and $\to^*$ denote the transitive, transitive-reflexive closure of $\to$, respectively. R/S abbreviates for S^*RS^*, and $\leftrightarrow$ for the symmetric closure $\to \cup \leftarrow$ of $\to$.

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, or wellfounded, 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.

As a well-known fact, a *string rewriting system* can be considered a term rewriting system. For this purpose every character is taken as a unary function, concatenation of strings becomes composition of functions, and a fixed variable (say, z) is appended at the right end of the string. The left context of the string is the context of the respective term. The right context is the substitution for z. So for instance the rewrite system $\mathtt{0\,0\,1\,1} \to \mathtt{1\,1\,1\,0\,0\,0}$ translates to

$$\mathtt{0}(\mathtt{0}(\mathtt{1}(\mathtt{1}(z)))) \to \mathtt{1}(\mathtt{1}(\mathtt{1}(\mathtt{0}(\mathtt{0}(\mathtt{0}(z))))))$$

We will however, for sake of simplicity, stick to the string representation, and will occasionally use some of the vocabulary of string rewriting. Concatenation will be denoted by juxtaposition, and the empty string (i.e. the term z) will be denoted by ε. For surveys on string rewriting see [13] and [4].

3 Simple termination

If we try to apply well-known simplification orders like the recursive path order (rpo) [6] to a one-rule rewrite system of the form $\mathtt{0}^p\,\mathtt{1}^q \to \mathtt{1}^r\,\mathtt{0}^s$, we find that rpo can handle the case $p \geq s$ for arbitrary q, r, using the precedence $\mathtt{0} > \mathtt{1}$. The same is done by polynomial interpretation $[\mathtt{0}](x) = (r+1)x, [\mathtt{1}](x) = x + 1$ [14]. So in this case obviously Z is simply terminating. Moreover, since the interpretation is linear, the derivation length $D(n)$ is at most exponential in n [15]. Here $D(n)$ is defined to be the maximal number of steps in a reduction starting with a string of length n. Below we show that there are systems having linear, quadratic, and exponential derivation lengths.

Now it is easy to see that exchanging $\mathtt{0}$ by $\mathtt{1}$ and reversing strings gives only a renamed copy of the problem. By this symmetry argument the case $q \geq r$, with arbitrary p, s is simply terminating as well.

Proposition 3.1 *Z is simply terminating if $p \geq s$ or $q \geq r$.*

It is a surprising fact that in spite of the symmetry neither rpo nor one-level polynomial interpretations are able to handle the case $p < s, q = r$. Both techniques imply ω-termination as introduced in [18]. In [18] it is proved that the system $\mathtt{0\,1} \to \mathtt{1\,0\,0}$ is not ω-terminating; the same holds for the more general case $p < s, q = r$.

3.1 Linear and quadratic derivation lengths

If the number of $\mathtt{0}$ symbols strictly decreases, then obviously the length of a derivation is bounded by the number of $\mathtt{0}$ symbols in the initial term. The complexity of derivation lengths is thus linear in this case.

Proposition 3.2 *If $p > s$ or $q > r$ then $D(n) = O(n)$.*

If the number of $\mathtt{0}$ symbols remains constant, i.e. if $p = s$ holds, then substrings $\mathtt{0}^p$ behave exactly like single characters. So it suffices to treat the case $p = s = 1$.

Now consider the case $p = s = 1 = q = r$. The rewrite system $\mathtt{0}\,\mathtt{1} \to \mathtt{1}\,\mathtt{0}$ amounts to swap adjacent symbols if the former symbol is smaller (w.r.t. $\mathtt{0} < \mathtt{1}$) than the latter. This is nothing but the well-known *bubblesort* algorithm. Bubblesort has quadratic worst-case complexity.

Proposition 3.3 *If $p = s$ and $q = r$ then $D(n) = O(n^2)$.*

Starting with the string $\mathtt{0}^{np}\,\mathtt{1}^{nq}$ of which the size is linear in n indeed yields a reduction of which the length is quadratic in n.

3.2 Exponential derivation lengths

The considerations of the previous subsection leave the case $p = s = 1$, $q < r$. We claim that here for every choice of $q < r$, the derivation lengths are indeed exponential in the worst case. For instance, consider the case $q = 1$, $r = 2$. The rewrite system is $\mathtt{0}\,\mathtt{1} \to \mathtt{1}\,\mathtt{1}\,\mathtt{0}$. In a nutshell, every $\mathtt{0}$ symbol doubles the number of $\mathtt{1}$ symbols right to it. A worst-case initial term is $\mathtt{0}^n\,\mathtt{1}$, for it has the normal form $\mathtt{1}^{2^n}\,\mathtt{0}^n$, and every rewrite step contributes only 1 to the length of a term. So the derivation has length $2^n - 1$.

Things are however much less easy in the general case; many terms initiate only derivations of polynomial length. The following example is typical. Let $q = 2$, $r = 3$, so Z is $\mathtt{0}\,\mathtt{1}\,\mathtt{1} \to \mathtt{1}\,\mathtt{1}\,\mathtt{1}\,\mathtt{0}$. The term $\mathtt{0}^n\,\mathtt{1}\,\mathtt{1}$ undergoes the following derivation to normal form.

$$\mathtt{0}^n\,\mathtt{1}\,\mathtt{1} \to_Z \mathtt{0}^{n-1}\,\mathtt{1}\,\mathtt{1}\,\mathtt{1}\,\mathtt{0} \to_Z \mathtt{0}^{n-2}\,\mathtt{1}\,\mathtt{1}\,\mathtt{1}\,\mathtt{0}\,\mathtt{1}\,\mathtt{0} \to_Z^{n-2} \mathtt{1}\,\mathtt{1}(\mathtt{1}\,\mathtt{0})^n$$

This derivation has length n, so is linear. The shortness is caused by the fact that $\mathtt{1}$ symbols are not used up completely since 3 is not divisible by 2. We better provide a pattern which is free from such losses. An easy induction on k shows that the following holds.

Proposition 3.4 *For Z the system $\mathtt{0}\,\mathtt{1}^q \to \mathtt{1}^r\,\mathtt{0}$, the following derivation holds.*

$$\mathtt{0}^k\,\mathtt{1}^{q^k} \to_Z^* \mathtt{1}^{r^k}\,\mathtt{0}^k$$

This is exactly the pattern which works without any losses. Its derivation length is $\frac{r^k - q^k}{r-q}$. Note however that this length may be not exponential in the size of the initial term. For, the initial term has length $k + q^k$, which is itself exponential in k, if $q > 1$. And by an easy calculation, the asymptotic derivation length is described by

$$d_Z(\mathtt{0}^k\,\mathtt{1}^{q^k}) = \begin{cases} O(r^{n-2}), & \text{if } q = 1, \\ n^{O(\log_q r)}, & \text{else} \end{cases}$$

where n is the size $k + q^k$ of the initial term. This is an exponential function for $q = 1$ but a polynomial else.

To finally achieve a worst-case pattern we choose a fixed number k which is large enough to lead to a (at least) a duplication of the exponent of $\mathtt{1}$. Thus we can simulate the behaviour of case $q = 1$, $r = 2$ by a macro step.

Proposition 3.5 *Let Z denote the system $\mathtt{0}\,\mathtt{1}^q \to \mathtt{1}^r\,\mathtt{0}$, where $1 < q < r$, and let k be the least integer $\geq \frac{1}{\log_2 \frac{r}{q}}$. Then the term $\mathtt{0}^{nk}\,\mathtt{1}^{q^k}$ initiates a Z-derivation of a length exponential in n, for every $n > 0$.*

Proof *By definition, k is the smallest number such that $r^k \geq 2q^k$ holds. Let c denote the (constant) length of the derivation $\mathtt{0}^k\,\mathtt{1}^{q^k} \to_Z^c \mathtt{1}^{r^k}\,\mathtt{0}^k$. By an easy induction on m one shows that $\mathtt{0}^k\,\mathtt{1}^{mq^k} \to_Z^{cm} \mathtt{1}^{mr^k}\,\mathtt{0}^k$ holds. We prove by induction on n that every term of the form $\mathtt{0}^{nk}\,\mathtt{1}^{mq^k}$ has a derivation length $\geq c(2^n - 1)m$. The base case $n = 1$ is given. For $n > 1$, there is a derivation starting with*

$$\mathtt{0}^{nk}\,\mathtt{1}^{mq^k} = \mathtt{0}^{(n-1)k}\,\mathtt{0}^n\,\mathtt{1}^{mq^k} \to_Z^{cm} \mathtt{0}^{(n-1)k}\,\mathtt{1}^{mr^k}\,\mathtt{0}^k \underset{r^k \geq 2q^k}{=} \mathtt{0}^{(n-1)k}\,\mathtt{1}^{2mq^k}\,\mathtt{1}^{mr^k - 2mq^k}\,\mathtt{0}^k$$

whose substring $\mathtt{0}^{(n-1)k}\,\mathtt{1}^{2mq^k}$ by inductive hypothesis still makes for at least $c(2^{(n-1)} - 1)2m = c(2^n - 2)m$ steps. Altogether there are $cm + c(2^n - 2)m = c(2^n - 1)m$ steps. This finishes the proof of the intermediate lemma. The claim is immediate by $m = 1$.

Note that since k is fixed, the initial term indeed has size linear in n. Thus we have exponential derivation lengths, in the size of the initial term.

4 Non-termination

As we claim "if and only if" in our main theorem, we should be able to prove non-termination for the following cases:

1. $2p \leq s$, $2q \leq r$,
2. $p < s < 2p$, $q < r$, $q|r$,
3. $q < r < 2q$, $p < s$, $p|s$.

To do it we employ the well-known fact that every looping derivation extends to an infinite derivation. For a string rewriting relation $\to$, a proper derivation $t \to^+ u$ is called *looping*, if $u = vtw$ holds for some v, w. Indeed we have:

Lemma 4.1 *For $p \geq s$, $q \geq r$, the system Z has a looping derivation.*

Proof *The following derivation is looping. (The re-occurrence of the initial string, $\mathtt{0}^p\,\mathtt{1}^{2q}$, is marked by a frame box.)*

$$\mathtt{0}^p\,\mathtt{1}^{2q} \to \mathtt{1}^r\,\mathtt{0}^s\,\mathtt{1}^q \to \mathtt{1}^r\,\mathtt{0}^{s-p}\,\mathtt{1}^r\,\mathtt{0}^s = \mathtt{0}^{s-2p}\,\boxed{\mathtt{0}^p\,\mathtt{1}^{2q}}\,\mathtt{1}^{r-2q}$$

To prove that the other case is looping as well, is more difficult. First we establish a lemma saying that a certain suffix can be reached. We use the notation $t \to ...u$ to express the fact that t admits a derivation to a string which has suffix u.

Lemma 4.2 *If $p < s < 2p$, $1 = q < r$, then $\mathtt{0}^{pk}\,\mathtt{1}^{m+1} \to_Z^* ...\mathtt{0}^{sk}$ holds for all nonnegative k, m.*

Proof *By induction on k and m lexicographically. If $k = 0$ then the claim is trivial. So let $k > 0$. Here we get:*

$$\mathtt{0}^{pk}\,\mathtt{1}^{m+1} \to_Z \mathtt{0}^{p(k-1)}\,\mathtt{1}^{r}\,\mathtt{0}^{s}\,\mathtt{1}^{m} \underset{IH,(k-1,r)}{\to_Z^*} \ldots \mathtt{0}^{s(k-1)}\,\mathtt{0}^{s}\,\mathtt{1}^{m} = \ldots \mathtt{0}^{sk}\,\mathtt{1}^{m}$$

If $m = 0$ then we are finished. Else, the derivation continues by

$$\ldots \mathtt{0}^{sk}\,\mathtt{1}^{m} \underset{s>p}{=} \ldots \mathtt{0}^{pk}\,\mathtt{1}^{m} \underset{IH,(k,m-1)}{\to_Z^*} \ldots \mathtt{0}^{sk}$$

This finishes the proof.

This lemma is used to prove our claim:

Lemma 4.3 *If $p < s < 2p$, $q < r$, $q|r$, or symmetrically, $q < r < 2q$, $p < s$, $p|s$, then Z has a looping derivation.*

Proof *Again, we may assume $p < s < 2p$, $1 = q < r$. Then the following derivation is looping. (The re-occurrence of the initial string, $\mathtt{0}^{p^2}\,\mathtt{1}^2$, is marked by a frame box.)*

$$\mathtt{0}^{p^2}\,\mathtt{1}^{2} \to_Z \mathtt{0}^{p^2-p}\,\mathtt{1}^{r}\,\mathtt{0}^{s}\,\mathtt{1}^{2} \to_Z \mathtt{0}^{p^2-p}\,\mathtt{1}^{r}\,\mathtt{0}^{s-p}\,\mathtt{1}^{r}\,\mathtt{0}^{s} \underset{lemma\ 4.2}{\to_Z^*} \ldots \mathtt{0}^{s(p-1)}\,\mathtt{0}^{s-p}\,\mathtt{1}^{r}\,\mathtt{0}^{s} =$$

$$\ldots \mathtt{0}^{(s-1)p}\,\mathtt{1}^{r}\,\mathtt{0}^{s} \underset{\substack{s-1\geq p\\ r\geq 2}}{=} \ldots \boxed{\mathtt{0}^{p^2}\,\mathtt{1}^{2}}\,\mathtt{1}^{r-2}\,\mathtt{0}^{s}$$

5 The proof architecture for the complex case

Finally, we are left with the case $p < s < 2p$, $q < r$, $q \not| r$, and its symmetric counterpart. Here, as we are going to show below, Z is terminating again, but no longer simply terminating. The termination proof is very involved. Nevertheless, we found that standard methods applied and could do much of the clerical work for the termination proof.

Lemma 5.1 *If $p < s < 2p$, $q < r$, $q \not| r$, then Z is self-embedding.*

Proof *Let x be some positive integer number. Consider the following derivation. (Redexes are underlined.)*

$$\mathtt{0}^{p}\,\mathtt{1}^{qx} = \underline{\mathtt{0}^{p}\,\mathtt{1}^{q}}\,\mathtt{1}^{q(x-1)} \to_Z \mathtt{1}^{r}\,\mathtt{0}^{s-p}\,\underline{\mathtt{0}^{p}\,\mathtt{1}^{q}} \to_Z^* (\mathtt{0}^{s-p}\,\mathtt{1}^{r})^{x}\,\mathtt{0}^{s}$$

Now we have an embedding if we can choose $x = x_1 + x_2$ so that $\mathtt{0}^p$ is embedded in $(\mathtt{0}^{s-p}\,\mathtt{1}^{r})^{x_1}$ and $\mathtt{1}^{qx}$ is embedded in $(\mathtt{0}^{s-p}\,\mathtt{1}^{r})^{x_2}$. An easy calculation shows that this is the case whenever $x_1 \geq \lceil\frac{p}{s-p}\rceil$ and $x \geq \lceil\frac{rx_1}{r-q}\rceil$ hold; such x_1 and x are easily found.

As we will show in the remainder of the paper, Z is terminating.

The rewrite system Z, whose termination we want to prove, is transformed to a rewrite system C in such a way that termination of C entails termination of Z. In the same way, C is transformed to another system, B, next B to S, and, finally, S to R.

$$Z \mapsto C \mapsto B \mapsto S \mapsto R$$

Let us briefly explain how the rewrite systems look like, and by which intuition we justify the steps.

5.1 Step 1: Encompassment by left and right contexts

The first transformation step is easy: We consider the rule in every pair of left and right context each of length 1. As the alphabet is $\{\mathtt{0},\mathtt{1}\}$, we get the four-rule rewrite system $C = \{(1),(2),(3),(4)\}$, as follows.

(1) $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{0} \to \mathtt{0}\,\mathtt{1}^r\,\mathtt{0}^s\,\mathtt{0}$$

(2) $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0}\,\mathtt{1}^r\,\mathtt{0}^s\,\mathtt{1}$$

(3) $$\mathtt{1}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{0} \to \mathtt{1}\,\mathtt{1}^r\,\mathtt{0}^s\,\mathtt{0}$$

(4) $$\mathtt{1}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{1}\,\mathtt{1}^r\,\mathtt{0}^s\,\mathtt{1}$$

It is obvious that for every Z-derivation, starting with term t, there is a corresponding C-derivation of the same length, starting e.g. with $\mathtt{0}\,t\,\mathtt{0}$. For this reason, as soon as we have proved termination of $\to_C$, termination of $\to_Z$ follows. The purpose of this transformation is simply to split up Z into four cases which may be treated each in a different way.

5.2 Step 2: Dummy introduction

The main idea for this step is that there is no "information flow" between the left half, $\mathtt{0}\,\mathtt{1}^r$, and the right half, $\mathtt{0}^s\,\mathtt{1}$, of the right hand side, $\mathtt{0}\,\mathtt{1}^r\,\mathtt{0}^s\,\mathtt{1}$, 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", $\square$, between the two, without changing the applicability of rewrite steps. Thus the termination proof of rewrite system C reduces to that of the rewrite system $B = \{(1),(2'),(3),(4)\}$, where rule (2) has been replaced by the following rule.

(2') $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0}\,\mathtt{1}^{r'}\,\square\,\mathtt{0}^{s'}\,\mathtt{1}$$

In section 6 we prove that this step indeed preserves non-termination, provided that $p \not| s$, $q \not| r$, $s' > s - s \bmod p$, and $r' > r - r \bmod q$ hold.

5.3 Step 3: Dummy elimination

The introduction of the $\square$ symbol enables a further step which splits rule (2') into two rules

(21) $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0}\,\mathtt{1}^{r'}$$

(22) $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0}^{s'}\,\mathtt{1}$$

This transforms B towards a system $S = \{(1),(21),(22),(3),(4)\}$. In section 7 we prove that this step is non-termination preserving.

5.4 Step 4: Relative termination

In an arbitrary string consider the number of nonempty *packages* of zeroes (separated by ones). In system S each rule decreases the number of packages, rule (3) even strictly. Let R denote $S \setminus \{(3)\}$. Obviously any S-derivation contains only finitely many (3)-steps. Hence termination of S follows from termination of R. This can als be stated as $\to_{(3)}$ terminates *relative to* $\to_R$; relative termination is studied in [10]. Termination of R for $s' = p + 1$ is proven in section 8.

6 Dummy introduction

The idea that leads to this step can be expressed informally as follows. Call a string d *dead* in context (v, w) if every derivation starting from a string of the form $lvdwr$ will only take steps in the part strictly left or strictly right of d in the string. In particular never a nonempty part of d appears in a redex. (But the definition makes sense even when d is empty.) When one replaces the dead part by a new symbol, no derivation (and, particularly, no *infinite* derivation) disappears. The introduction of this new symbol so may be used as a non-termination preserving transformation step. By construction, the new symbol only appears at right hand sides of the new rewrite system; such a symbol is called a *dummy symbol* in [9].

Technically, dummies are introduced by rewrite steps using rules of the form

$$vdw \to v \Box w$$

where d is dead for C in context (v, w). These rules are collected in an additional rewrite system T. In the next subsection we present abstract criteria for which termination of C can be concluded from termination of B.

6.1 A new abstract commutation criterion

The commutation property is first expressed by a local criterion on abstract reduction systems.

Lemma 6.1 *Let $\to_B$, $\to_T$, and $\to_C$ be arbitrary binary relations on a given set. If*

1. *$\to_B$ terminates,*
2. *$\to_C \subseteq \to_B^+ \leftarrow_T^*$,*
3. *$\leftarrow_T \to_C \subseteq \to_C^+ \leftarrow_T^*$*

then $\to_C$ terminates.

Proof *We have*

$$\to_{T^{-1}} \to_C \;=\; \leftarrow_T \to_C \;\subseteq\; \to_C^+ \leftarrow_T^* \;=\; \to_C^+ \to_{T^{-1}}^* \;\subseteq\; \to_C^+ \to_{C \cup T^{-1}}^* \;=\; \to_C \to_{C \cup T^{-1}}^* .$$

Since $\to_C \to_C \subseteq \to_C \to_{C \cup T^{-1}}^*$ we obtain

$$\to_{C \cup T^{-1}} \to_C \;\subseteq\; \to_C \to_{C \cup T^{-1}}^* ;$$

straightforward induction yields

$$\to_{C \cup T^{-1}}^* \to_C \;\subseteq\; \to_C \to_{C \cup T^{-1}}^* .$$

Using this property and premise 2, we obtain by the following diagram that for every element t having an infinite $\to_C$-derivation there exists an element t' again having an infinite $\to_C$-derivation satisfying $t \to_B^+ t'$.

$$
\begin{tikzcd}
t \arrow[r, "C"] \arrow[dr, "B^+"'] & {} \arrow[r, "C"] \arrow[d, "T^*"] & {} \arrow[r, "C"] & {} & \cdots \\
 & t' \arrow[r, "C"] & {} \arrow[r, "C"] \arrow[u, "(C \cup T^{-1})^*"'] & {} \arrow[u, "(C \cup T^{-1})^*"'] & \cdots
\end{tikzcd}
$$

Repeating the argument shows that the existence of t having an infinite $\to_C$-derivation leads to an infinite $\to_B$-derivation of t, contradicting the termination of $\to_B$.

6.2 Application to Term Rewriting Systems

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

By a straightforward critical pair analysis, lemma 6.1 yields the following result for term rewriting systems.

Theorem 6.1 *Let B, T, and C be term rewriting systems. If*

1. *$\to_B$ terminates,*
2. *$C \subseteq \to_B^+ \leftarrow_T^*$,*
3. *$CP(T,C) \subseteq \to_C^+ \leftarrow_T^*$,*
4. *T left-linear and non-erasing,*
5. *C left-linear,*

then $\to_C$ terminates.

We stipulate that our theorem is applicable not only in string rewriting, but in proper term rewriting as well. The following is a witness.

Example 6.1 *Let C be given by the rule*

$$f(h(x)) \to h(f(g(h(x),x))).$$

Since C is self-embedding all methods for simple termination fail. Let k be a new binary function symbol, and let T be the system $g(h(x),y) \to k(x,y)$. Choose B to be $f(h(x)) \to h(f(k(x,x)))$. Now all conditions are satisfied: there are no critical pairs and $\to_B$ terminates by recursive path order with precedence $f > h > k$. So theorem 6.1 applies, by which $\to_C$ terminates.

6.3 Termination by completion

Let us now apply theorem 6.1 to do the dummy introduction step, $C \mapsto B$. Here B was obtained from C by replacing the rule $\mathtt{0\,0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0\,1}^r\,\mathtt{0}^s\,\mathtt{1}$ by $\mathtt{0\,0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0\,1}^{r'}\,\Box\,\mathtt{0}^{s'}\,\mathtt{1}$. For the dead part d we have $d = \mathtt{0}^{r-r'}\,\mathtt{1}^{s-s'}$ which may be empty. The notion of dead part preassumes $r \geq r'$, $s \geq s'$, but the construction below works as well without this restriction. The precise definition of r' and s' is postponed.

Recall that for string rewriting systems, conditions 4 and 5 are always satisfied. Let T consist at least of the rule

$$\mathtt{0\,1}^r\,\mathtt{0}^s\,\mathtt{1} \to \mathtt{0\,1}^{r'}\,\Box\,\mathtt{0}^{s'}\,\mathtt{1},$$

then condition 2 is satisfied. In order to check condition 3 we consider a typical critical pair. Other critical pairs are either trivial — overlapping in a pair of contexts — or similar. The overlapping region of the peak string is marked by a frame box, else the redex is underlined.

$$\begin{array}{ccc}
\mathtt{0\,0}^{p-1}\,\boxed{\mathtt{0\,1}^q\,\mathtt{1}}\,\mathtt{1}^{r-q-1}\,\mathtt{0}^s\,\mathtt{1} & \xrightarrow[C]{} & \mathtt{0\,1}^r\,\mathtt{0}^{s-1}\,\underline{\mathtt{0\,1}^{r-q}\,\mathtt{0}^s\,\mathtt{1}} \\
\big\downarrow{\scriptstyle T} & & \big\downarrow{\scriptstyle T} \\
\underline{\mathtt{0\,0}^p\,\mathtt{1}^q\,\mathtt{1}}\,\mathtt{1}^{r'-q-1}\,\Box\,\mathtt{0}^{s'}\,\mathtt{1} & \xrightarrow[C]{} & \mathtt{0\,1}^r\,\mathtt{0}^s\,\mathtt{1}^{r'-q}\,\Box\,\mathtt{0}^{s'}\,\mathtt{1}
\end{array}$$

The $\to_T$ arrow at the right column is required by the critical pair condition, but not satisfied. Hence condition 3 is not yet satisfied. Like in the Knuth/Bendix completion procedure, we simply add a new rule

$$\mathtt{0\,1}^{r-q}\,\mathtt{0}^s\,\mathtt{1} \to \mathtt{0\,1}^{r'-q}\,\Box\,\mathtt{0}^{s'}\,\mathtt{1}$$

to T, such that the condition *becomes* satisfied. The same idea underlies the "termination by completion" method [3].

With the new rule we again check for critical pairs, add corresponding rules, and so forth, until we get no more critical pairs. We can easily read off that the exponents of $\mathtt{1}$ are of the form $iq+r \bmod q$ at the left hand side, and $iq+r''$ at the right hand side, where $r'' = r' - q\lfloor\frac{r}{q}\rfloor = r' - r + r \bmod q$. The number r'' should be positive, otherwise some critical pairs would not close. In other words, our construction works only if $r' > r - r \bmod q$. Of course, we might choose $r' = r$ but we will see that $r' = r - r \bmod q + 1$ suits our purposes better.

In the same way, critical pairs with C rules from the right cause new rules in T where $\mathtt{0}$ gets new exponents. Finally we arrive at

$$T \underset{\text{def}}{=} \{ \;\; \mathtt{0\,1}^{iq+r \bmod q}\,\mathtt{0}^{jp+s \bmod p}\,\mathtt{1} \;\;\to\;\; \mathtt{0\,1}^{iq+r''}\,\Box\,\mathtt{0}^{jp+s''}\,\mathtt{1} \;\;\mid\;\; i \in \{0,\ldots,\left\lfloor\frac{r}{q}\right\rfloor\},\quad j \in \{0,\ldots,\left\lfloor\frac{s}{p}\right\rfloor\} \;\;\}$$

Observe that the requirement $q \not| r$ indeed turns out necessary for the diagrams to work. For, if $r \bmod q = 0$, then the T-rule for $i = 0 = j$, which is $\mathtt{0\,0}^{s \bmod p}\,\mathtt{1} \to \mathtt{0\,1}^{r''}\,\Box\,\mathtt{0}^{s''}\,\mathtt{1}$

chases the following diagram

$$\begin{tikzcd}
\mathtt{1}\,\mathtt{0}^{p-s\bmod p-1}\boxed{\mathtt{0}\,\mathtt{0}^{s\bmod p}\,\mathtt{1}}\,\mathtt{1}^{q-1}\,\mathtt{0} \arrow[r, "C"'] \arrow[d, "T"] & \mathtt{1}\,\mathtt{1}^{r}\,\mathtt{0}^{s}\,\mathtt{0} \arrow[d, "\not{T}"] \\
\mathtt{1}\,\mathtt{0}^{p-s\bmod p-1}\,\mathtt{0}\,\mathtt{1}^{r''}\,\square\,\mathtt{0}^{s''}\,\mathtt{1}\,\mathtt{1}^{q-1}\,\mathtt{0} & {}
\end{tikzcd}$$

The diagram does not close as the string $\mathtt{1}\,\mathtt{1}^{r}\,\mathtt{0}^{s}\,\mathtt{0}$ is not able to develop a $\square$ symbol, and C steps cannot get rid of $\square$.

We conclude:

> If $p \not| s$, $q \not| r$, and $r' > r - r \bmod q$, $s' > s - s \bmod p$, then T satisfies the critical pair criterion:
> $$CP(T, C) \subseteq \to_C \leftarrow_T$$

Hence theorem 6.1 can be applied and termination of $\to_B$ implies termination of $\to_C$.

7 Dummy elimination

Let $\square$ be a symbol which only occurs at right hand sides of a string rewriting system. This symbol $\square$ can never be removed by any rewrite rule and will act as a separator between parts of the string. Intuitively an infinite derivation can be localized between these separators, hence a rule

$$l \quad \to \quad r_1 \,\square\, r_2 \,\square \cdots \square\, r_n$$

may be split into n rules

$$l \to r_1, \qquad \ldots, \qquad l \to r_n$$

whose termination can be easier to prove. In this section we formalize this idea.

Definition 7.1 *For each string of the form $s = r_1 \,\square\, r_2 \cdots \square\, r_n$ where $r_i \in (\mathcal{A} \setminus \{\square\})^*$ for all $i \in \{1, \ldots, n\}$, let $\mathcal{E}(s) =_{\text{def}} \{r_1, \ldots, r_n\}$.*

Lemma 7.1 *Let B be a string rewriting system on the alphabet $\mathcal{A}$ where the symbol $\square \in \mathcal{A}$ does not occur on left hand sides of B. Let $S = \{l \to u \mid (l \to r) \in R \wedge u \in \mathcal{E}(r)\}$. Then $\to_B$ terminates if $\to_S$ terminates.*

Proof *In the definition of $\mathcal{E}(s)$ it does not make any difference whether $\mathcal{E}(s)$ is considered as a set or as a multiset. Here we consider $\mathcal{E}(s)$ as a multiset in order to apply well-foundedness of the multiset order.*

Let $\to_S$ be terminating. Define order $>$ on strings on $\mathcal{A} \setminus \{\square\}$ by $v > w$ if there exist q, q' such that $v \to_S^+ qwq'$. Clearly $>$ is an order. Assume $v_1 > v_2 > v_3 > \cdots$ with $v_i \to_S^+ q_i v_{i+1} q_i'$, then

$$v_1 \to_S^+ q_1 v_2 q_1' \to_S^+ q_1 q_2 v_3 q_2' q_1' \to_S^+ q_1 q_2 q_3 v_4 q_3' q_2' q_1' \to_S^+ \cdots$$

contradicting termination of $\to_S$, hence $>$ is well-founded.

We claim that $s \to_B t$ implies $\mathcal{E}(s) >^{mult} \mathcal{E}(t)$. Suppose $s \to_B t$ using rule $l \to r$ in B, which means that s is of the form $s = s_1 l s_2$. Let us first assume that s_1, s_2 do not contain $\Box$, whence $\mathcal{E}(s) = \{s_1 l s_2\}$. If r does not contain $\Box$, then $\mathcal{E}(t) = \{s_1 r s_2\}$, and the claim follows by $s_1 l s_2 \to_S s_1 r s_2$, as rule $l \to r$ is also in S. Else, suppose that $r = r_1 \Box r_2 \Box \cdots \Box r_n$ with $n > 1$. Then $\mathcal{E}(t) = \{s_1 r_1, r_2, r_3, \ldots, r_{n-1}, r_n s_2\}$. Now by defintion $s_1 l s_2$ is greater than every element of $\mathcal{E}(t)$, hence again the claim follows. By closure under multiset union, this reasoning carries over to the case where s_1 or s_2 contain dummy symbols and the claim has been proved. Since $>^{mult}$ is wellfounded, termination of $\to_B$ follows.

Remark. This multiset comparison is not closed under left and right contexts. For example, let $>$ be a simplification order that satisfies $\mathtt{1} > \mathtt{0\,0}$, on strings over the alphabet $\{\mathtt{0}, \mathtt{1}\}$, for instance recursive path order with precedence $\mathtt{1} > \mathtt{0}$. Let $s = \mathtt{0\,0} \Box \mathtt{1}$, $t = \mathtt{1} \Box \mathtt{0}$. Then $\mathcal{E}(s) >^{mult} \mathcal{E}(t)$, but *not* $\mathcal{E}(vs) >^{mult} \mathcal{E}(vt)$. An order that works in the same spirit, and is moreover closed under left and right contetxs, may be defined using a recursive path order construct instead of multisets [11].

In [9] a general dummy elimination theorem is proved for term rewriting instead of string rewriting. Our lemma can also be proved using that theorem.

8 Finish of the proof

It remains to prove termination of R consisting of the rules

(1) $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{0} \to \mathtt{0}\,\mathtt{1}^r\,\mathtt{0}^s\,\mathtt{0}$$

(4) $$\mathtt{1}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{1}\,\mathtt{1}^r\,\mathtt{0}^s\,\mathtt{1}$$

(21) $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0}\,\mathtt{1}^{r'}$$

(22) $$\mathtt{0}\,\mathtt{0}^p\,\mathtt{1}^q\,\mathtt{1} \to \mathtt{0}^{s'}\,\mathtt{1}$$

In this system we still have some freedom in choosing r' and s'; the validity of dummy introduction only required $r' > r - r \bmod q$ and $s' > s - s \bmod p$. Here we require $p < s < 2p$, hence we may choose $s' = p + 1$ and replace s' in rule (22) by $p + 1$.

Now we switch the representation of a string

$$\mathtt{0}^{m_1}\,\mathtt{1}^{n_1} \ldots \mathtt{0}^{m_k}\,\mathtt{1}^{n_k}$$

to a sequence of pairs of non-negative integers,

$$(m_1, n_1) \ldots (m_k, n_k)$$

where for uniqueness we require that except possibly m_1, n_k, all numbers are positive. Now R can be presented in the form

(1) $$(m + p, q)(m', z) \to (m, r)(m' + s, z)$$

(4) $$(z, n)(p, n' + q) \to (z, n + r)(s, n')$$

(21) $$(m + p, n + q) \to (m, n + r' - 1)$$

(22) $$(m + p, n + q) \to (m + p, n)$$

where $z \geq 0$ and $m, m', n, n' > 0$.

Choose any well-founded order $\sqsupset$ on non-negative integers for which $p \sqsupset s$ and $n+p \sqsupset n$ for all n, for example $n \sqsupseteq n' \iff f(n) \geq f(n')$ for

$$f(n) \underset{\text{def}}{=} \begin{cases} n+p, & \text{if } p|n, \\ n, & \text{else} \end{cases}$$

Now we see that by an R-reduction step of type (1), (4) and (21) of the sequence $(m_1, n_1) \ldots (m_k, n_k)$ the string $(m_1, \ldots, m_k)$ lexicographically decreases according to $\sqsupset$, while it remains the same by a step of type (22). Hence any R-reduction contains only finitely many steps of type (1), (4) and (21). Since the rule (22) is clearly terminating, we conclude that $\rightarrow_R$ terminates.

It can be shown that this proof works only with the choice $s' = p + 1$, whence the requirement $s < 2p$ is essential.

9 Related Work

Our work on this subject began with proving termination of the one-rule string rewriting system, sometimes called "Zantema's problem",

$$\texttt{0011} \rightarrow \texttt{111000}$$

which corresponds to the case $p = 2 = q$, $r = 3 = s$, of this paper. To our knowledge, there is a proof sketch by Nachum Dershowitz and Charles Hoot [8], and a detailed proof including a treatment of derivation lengths by Elias Tahhan-Bittar [16]. Dershowitz/Hoot's line of argument is by minimal counterexample, and by forward closures. Tahhan-Bittar uses the notion of "inner redex" and shows termination by the fact that all inner redexes terminate. Our notion of dead part corresponds to his "strongly irreducible" strings. He could extend his termination result to prove a sharp upper bound for the lengths of derivation.

Theorem 9.1 ([16]) *If $p = 2 = q$, $r = 3 = s$ then $D(n) = 2n - 6$.*

A completely different approach is currently investigated by Jan-Willem Klop (personal communication). He uses a reasoning by cases, visualized at rectangular figures where `0` characters are represented by upwards arrows, and `1` characters by rightbound arrows. Rewrite steps are understood as commuting diagrams.

The notion of "transformation ordering" and "termination by completion" have been coined by Françoise Bellegarde and Pierre Lescanne [2, 3].

Theorem 9.2 (Transformation order, [2, 3]) *Let B, C, and T be term rewriting systems. If*

1. *$\rightarrow_B \cup \rightarrow_T$ terminates,*
2. *$\rightarrow_T$ is confluent,*
3. *T is non-erasing and left-linear, and*
4. *$CP(T, B)$ is* cooperative*, i.e. it satisfies $CP(T, B) \subseteq (\rightarrow_B / \rightarrow_T)^+ \leftarrow_T^*$,*

then even

$$>\underset{\text{def}}{=} (\to_B / \leftrightarrow_T)^+ \cup \to_T^+$$

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

In fact, an earlier attack to our problem has shown that the transformation ordering is quite useful, too, to describe the dummy introduction step [11]. On closer observation however we found that termination of the transformer system is an unnatural requirement: T steps did not occur as C rules, and the critical pair criteria worked as well without assuming any normal forms. We felt that we could do without it.

The result is our theorem 6.1. It is a criterion similar to the quasi-commutation criterion of Leo Bachmair and Nachum Dershowitz. Quasi-commutation is the property $\to_T \to_B \subseteq \to_B \to_{B\cup T}^*$, which, provided that $\to_B$ terminates, is equivalent to $\to_T \to_B \subseteq \to_B^+ \to_T^*$.

Theorem 9.3 ([1]) *Let B, T be term rewriting systems. If*

1. *$\to_B$ terminates,*
2. *$CP(T, B) \subseteq \to_B \to_{B\cup T}^*$,*
3. *B is left-linear and non-erasing,*
4. *T is right-linear*

then $\to_B \to_T^*$ terminates.

But it is not quite the same. Comparing the abstract versions, we find we can simulate their version, by $\to_C \; = \; \to_B \leftarrow_T^*$ with T inverted, but not (naively) vice versa. As a counterexample choose $a \to_B b, b \to_B c, a \to_C c, b \to_T c$. Here $\to_B$ terminates, and so $\to_C$ by our criteria, but $\to_B \leftarrow_T^*$ does not terminate. Though we do not claim that we have essentially improved over Bachmair/Dershowitz' criterion: We have the same technical restrictions (left-linearity, non-erasingness) as they have, and a slightly different critical pair condition which is put on C rather than on B. By accident, this suits our needs better in the case of dummy introduction, as critical pairs with C are more comfortable to handle.

10 Conclusions

We gave a complete and precise characterization when a one-rule string rewriting system Z of the form $\mathtt{0}^p\,\mathtt{1}^q \to \mathtt{1}^r\,\mathtt{0}^s$ terminates, where p, q, r, s are positive integers. For the simply terminating cases we gave sharp upper bounds for the complexity of derivation lengths.

We attacked the difficult, non-simply terminating case, $p < s < 2p$, $q < r$, $q \nmid r$, by a series of transformation steps, each preserving non-termination. We demonstrated how to design a termination proof and how to split it into small steps each of which can be supported by standard methods. For the dummy introduction, we used an impoverished form of transformation order. Dummy elimination is about to become a standard method. Another standard method, *semantic labelling* [17] turned out not to support the dummy introduction step, but a twin-labelling (first label as usual, then label the reversed strings) looks promising.

Of course, we would like to have an estimate of the derivation length in the non-simple termination case, too. We expect that derivation lengths are linear, as in the case $p = 2 = q$, $r = 3 = s$. On close observation of the termination proof, we get that each of the transformation steps, except the dummy elimination step, preserves the length of derivations. Dummy elimination, however, gives only an exponential upper bound.

Acknowledgments

We want to thank John Tromp and Elias Tahhan-Bittar for fruitful discussions on this topic.

References

[1] Leo Bachmair and Nachum Dershowitz. Commutation, transformation, and termination. In *8th Conf. Automated Deduction*, pages 5–20. Springer Lecture Notes in Computer Science230, 1985.

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

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

[4] Ronald Book and Friedrich Otto. *String-rewriting systems.* Texts and Monographs in Computer Science. Springer, New York, 1993.

[5] 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.

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

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

[8] 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.

[9] Maria C. F. Ferreira and Hans Zantema. Termination by dummy elimination. submitted, 1994.

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

[11] Alfons Geser. A solution to Zantema's problem. Technical Report MIP-9314, Universität Passau, Passau, Germany, December 1993. Also availabe in: R. Berghammer, G. Schmidt (eds.): Proc. coll. “Programmiersprachen und Grundlagen der Programmierung”, Bericht Nr. 9309, Univ. der Bundeswehr, München, Germany, pp. 88-96, Dec. 1993.

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

[13] M. Jantzen. *Confluent string rewriting*, volume 14 of *EATCS Monographs on Theoretical Computer Science*. Springer, Berlin, 1988.

[14] Dallas S. Lankford. On proving term rewriting systems are noetherian. Technical Report MTP-3, Louisiana Technical University, Math. Dept., Ruston, LA, 1979.

[15] V. C. S. Meeussen and H. Zantema. Derivation lengths in term rewriting from interpretations in the naturals. In H. A. Wijshoff, editor, *Computing Science in the Netherlands*, pages 249 – 260, November 1993. Also appeared as report RUU-CS-92-43, Utrecht University.

[16] Elias Tahhan-Bittar. Non-erasing, right-linear orthogonal term rewrite systems, application to Zantema's problem. Technical Report RR2202, INRIA, France, 1994.

[17] H. Zantema. Termination of term rewriting by semantic labelling. Technical Report RUU-CS-92-38, Utrecht University, December 1992. Extended and revised version appeared as RUU-CS-93-24, July 1993, accepted for special issue on term rewriting of Fundamenta Informaticae.

[18] H. Zantema. Termination of term rewriting: interpretation and type elimination. *Journal of Symbolic Computation*, 17:23–50, 1994.