The Derivational Complexity of the Bits Function and the Derivation Gap Principle*

Harald Zankl and Martin Korp
Institute of Computer Science
University of Innsbruck
6020 Innsbruck, Austria
{harald.zankl,martin.korp}@uibk.ac.at

Abstract

In this note we present two proofs that the derivational complexity of the bits function is linear. The first proof is intuitive but not very suitable for implementation while the second one has been found automatically. Using the second proof idea allows the complexity tool CaT to show linear derivational complexity of the bits function for which no other current contemporary analyzer can infer a polynomial upper bound. In the second part of this note we generalize the weight gap principle of (Hirokawa and Moser, 2008).

1 Introduction

Hofbauer and Lautemann [5] consider the length of derivations as a measurement for the complexity of terminating rewrite systems. The resulting notion of *derivational complexity* relates the length of a rewrite sequence to the size of its starting term. Thereby it is, e.g., a suitable metric for the complexity of deciding the word problem for a given confluent and terminating rewrite system (since the decision procedure rewrites terms to normal form).

To show (feasible) upper bounds on the derivational complexity currently few techniques are known. Typically termination criteria are restricted such that polynomial upper bounds can be inferred. The early work by Hofbauer and Lautemann [5] considers polynomial interpretations, suitably restricted, to admit quadratic derivational complexity. Match-bounds [2] and arctic matrix interpretations (AMIs) [7, 8] induce linear derivational complexity and triangular matrix interpretations (TMIs) [9] admit polynomially long derivations (the dimension of the matrices yields the degree of the polynomial). All these methods share the property that until now they have been used directly only, meaning that a single termination technique has to orient all rules in one go. However, using direct criteria exclusively is problematic due to their restricted power.

In the sequel we consider the TRS $\mathcal{R}_{\mathsf{bits}}$ (nontermin/AG01/#4.28 from [10])

$$\begin{aligned} \mathsf{half}(0) &\to 0 & \mathsf{bits}(0) &\to 0 \\ \mathsf{half}(\mathsf{s}(0)) &\to 0 & \mathsf{bits}(\mathsf{s}(x)) &\to \mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x)))) \\ \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) &\to \mathsf{s}(\mathsf{half}(x)) \end{aligned}$$

and present two proofs that the derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$ is linear, i.e., a term of size n admits at most derivations of length $\mathcal{O}(n)$. This result is somehow surprising due to the last rule. (Note that already a single rule $\mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{f}(x))$ causes the derivational complexity to be quadratic.) The first proof is presented in Section 3 and is based on a low level reasoning exploiting the structure of the TRS $\mathcal{R}_{\mathsf{bits}}$. Although the reasoning is intuitive it is hard to automate. The second proof given in Section 4 builds on a recent result by the authors that allows to combine different complexity criteria for a single TRS [11]. This approach is very suitable for implementation and linear derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$ can be inferred completely automatically by our complexity analyzer CaT.[1] Note that currently no other tool can show linear (not even polynomial) derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$.

* This research is supported by FWF (Austrian Science Fund) project P18763.

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

After the presentation of the bits example we focus on the weight gap principle introduced in [4]. We show in Section 5 how it can be generalized such that in principle arbitrary techniques can be plugged in.

2 Preliminaries

We assume familiarity with rewriting [1]. A *relative* TRS $\mathcal{R}/\mathcal{S}$ consists of two TRSs $\mathcal{R}$ and $\mathcal{S}$ with the rewrite relation $\to_{\mathcal{R}/\mathcal{S}} = \to_{\mathcal{S}}^* \cdot \to_{\mathcal{R}} \cdot \to_{\mathcal{S}}^*$. The *derivation length* of a term t with respect to a relation $\to$ and the set of terms is defined as follows: $\mathsf{dl}(t, \to) = \max\{m \mid \exists u\ t \to^m u\}$. The *derivational complexity* computes the maximal derivation length of all terms up to a given size, i.e., $\mathsf{dc}(n, \to) = \max\{\mathsf{dl}(t, \to) \mid |t| \leqslant n\}$. Sometimes we say that $\mathcal{R}$ ($\mathcal{R}/\mathcal{S}$) has linear, quadratic, etc. derivational complexity if $\mathsf{dc}(n, \to_{\mathcal{R}})$ ($\mathsf{dc}(n, \to_{\mathcal{R}/\mathcal{S}})$) can be bounded by a linear, quadratic, etc. polynomial in n. TMIs of dimension one are called SLIs. For the remainder of this note we assume that TRSs are finite.

3 Hand-Made Proof

To simplify the presentation of the first proof of the linear derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$ we abbreviate $s^n(0)$ by n and drop parentheses if no confusion can arise. At first we show that each derivation induced by a term of the form $\mathsf{bits}\ n$ has at most length $3n$. To this end we need the following technical lemma.

Lemma 1. *Let* $t = \mathsf{bits}\ n$. *If* $n = 0$ *then* $t \to 0$. *If* $n > 0$ *then* $t \to^{2+\frac{n}{2}} \mathsf{s}\ \mathsf{bits}\ \frac{n}{2}$.

Proof. We have $t \to \mathsf{s}\ \mathsf{bits}\ \mathsf{half}\ n \to^{n/2} \mathsf{s}\ \mathsf{bits}\ \frac{n}{2} \mathsf{half}\ m \to \mathsf{s}\ \mathsf{bits}\ \frac{n}{2}$ where $m = 0$ or $m = 1$. □

Lemma 2. *Let* $t = \mathsf{bits}\ n$. *Then* $\mathsf{dl}(t, \to_{\mathcal{R}}) \leqslant 3n$.

Proof. We restrict our attention to a special reduction. Since $\mathcal{R}$ is terminating (see [3, Example 5]) and has the diamond property this is fine. Using Lemma 1 we obtain the finite rewrite sequence

$$t \to^{2+\frac{n}{2}} \mathsf{s}\ \mathsf{bits}\ \frac{n}{2} \to^{2+\frac{n}{4}} \mathsf{s}\,\mathsf{s}\ \mathsf{bits}\ \frac{n}{4} \to^{2+\frac{n}{8}} \cdots \to^{2+\frac{n}{2^k}} \mathsf{s}^{\,k}\mathsf{bits}\ \frac{n}{2^k}$$

where $k = \lceil \log(n) \rceil$. Since $k \leqslant n$ we have

$$\sum_{1 \leqslant i \leqslant k} (2 + \frac{n}{2^i}) \leqslant 2n + n \cdot \sum_{1 \leqslant i \leqslant n} \frac{1}{2^i} \leqslant 3n$$

and hence $\mathsf{dl}(t, \to_{\mathcal{R}}) \leqslant 3n$ as desired □

At next we prove that each term of the form $\mathsf{bits}^{\,m} n$ admits at most linear derivation length. To prove this claim we need the property that $\log^i(n) \leqslant \frac{n}{2^i}$ if $\log^i(n) > 0$ for all $n > 3$ and $i \in \mathbb{N}$. This is show below.

Lemma 3. *For any* $n > 3$ *and* $i \in \mathbb{N}$ *with* $\log^i(n) > 0$ *we have* $\log^i(n) \leqslant \frac{n}{2^i}$.

Proof. We perform induction on i. If $i = 0$ then $n \leqslant \frac{n}{2^0} = n$. For the step case we claim that $\log(n) \leqslant \frac{n}{2}$ for all $n > 3$. Using the claim as well as monotonicity of log (on $n > 0$) we obtain:

$$\log^{i+1}(n) = \log(\log^i(n)) \leqslant \log(\frac{n}{2^i}) \leqslant \frac{n}{2^{i+1}}$$

It remains to show that the claim is correct. We have $\log(n) \leqslant \frac{n}{2} \Leftrightarrow 2\log(n) \leqslant n \Leftrightarrow 2^{2\log(n)} \leqslant 2^n \Leftrightarrow 2^{\log(n)} 2^{\log(n)} \leqslant 2^n \Leftrightarrow n \cdot n \leqslant 2^n \Leftrightarrow n^2 \leqslant 2^n$. The latter can be shown by an easy induction on n. □

Lemma 4. *Let $n > 3$ and $t = \mathsf{bits}^m\, n$. Then $\mathsf{dl}(t, \to_{\mathcal{R}}) \leqslant 6n + m$.*

Proof. First we assume that n is large enough such that all $\log^i(n)$ are positive. By Lemma 2 we can then estimate a sequence as follows:

$$t \to^{\leqslant 3n} \mathsf{s}\,\mathsf{bits}^{m-1}\,\log(n) \to^{\leqslant 3\log(n)} \mathsf{s}\,\mathsf{s}\,\mathsf{bits}^{m-2}\,\log^2(n) \to^{\leqslant 3\log^2(n)} \cdots \to^{\leqslant 3\log^{m-1}(n)} \mathsf{s}^{\,m}\,\log^m(n)$$

Using Lemma 3 we conclude that this sequence admits at most

$$3n + \sum_{1 \leqslant i \leqslant m} 3\log^i(n) \leqslant 3n + \sum_{1 \leqslant i \leqslant m} 3\frac{n}{2^i} \leqslant 3n + 3n \sum_{1 \leqslant i \leqslant m} \frac{1}{2^i} \leqslant 6n$$

steps. In the other case there is a $k \leqslant m$ such that $t \to^{\leqslant 6n} \mathsf{s}^k\,\mathsf{bits}^{m-k}\,0 \to^{m-k} \mathsf{s}^k\,0$. Putting things together finishes the proof. □

Since bits 4 admits longer derivations as bits 3, by Lemma 4 we obtain the following corollary.

Corollary 5. *Let $t = \mathsf{bits}^m\, n$. Then $\mathsf{dl}(t, \to_{\mathcal{R}}) \leqslant 6n + m + 24$.* □

Since the terms considered in Corollary 5 have longest derivations we obtain the next result.

Corollary 6. *The derivational complexity of the TRS $\mathcal{R}$ is linear.* □

4 Automated Proof

Contemporary methods for proving the derivational complexity of TRSs (automatically) comprise match-bounds [2], AMIs [7, 8] and TMIs [9]. These criteria can be preceded by (complexity-preserving) transformations. Usually one method must orient all rewrite rules strictly to deduce an upper bound on the derivational complexity of a TRS. Our experiments revealed that none of the above methods can conclude polynomial derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$ on its own. Recently [11] introduced a modular approach that allows to combine different criteria for complexity analysis by relative rewriting. Suddenly a polynomial (even linear!) upper bound on the derivational complexity can be inferred automatically. We recall the main results that allow to handle the TRS $\mathcal{R}_{\mathsf{bits}}$. The first theorem presented is one of the main results from [11].

Theorem 7. *Let $(\mathcal{R}_1 \cup \mathcal{R}_2)/\mathcal{S}$ be a relative TRS and let t be a term such that t terminates with respect to $(\mathcal{R}_1 \cup \mathcal{R}_2)/\mathcal{S}$. Then $\mathcal{O}(\mathsf{dl}(t, \to_{(\mathcal{R}_1 \cup \mathcal{R}_2)/\mathcal{S}})) = \max\{\mathcal{O}(\mathsf{dl}(t, \to_{\mathcal{R}_1/\mathcal{S}_1})), \mathcal{O}(\mathsf{dl}(t, \to_{\mathcal{R}_2/\mathcal{S}_2}))\}$.* □

The next theorem generalizes the *weight gap principle* introduced in [4] to relative rewriting.

Theorem 8. *Let $(\mathcal{R}_1 \cup \mathcal{R}_2)/\mathcal{S}$ be a relative TRS, $\mathcal{R}_1$ be non-duplicating, and let $\mathcal{M}$ be an SLI such that $\mathcal{R}_2 \subseteq \succ_{\mathcal{M}}$ and $\mathcal{S} \subseteq \succeq_{\mathcal{M}}$. Then for any $\mathcal{R}_1$ and $\mathcal{M}$ there exist constants K and L such that*

$$K \cdot \mathsf{dl}(t, \to_{\mathcal{R}_1/(\mathcal{R}_2 \cup \mathcal{S})}) + L \cdot |t| \geqslant \mathsf{dl}(t, \to_{(\mathcal{R}_1 \cup \mathcal{R}_2)/\mathcal{S}})$$

whenever t is terminating on $(\mathcal{R}_1 \cup \mathcal{R}_2)/\mathcal{S}$. □

Using the above theorems it is easy to show that the TRS $\mathcal{R}_{\mathsf{bits}}$ admits linear derivational complexity.

Theorem 9. *The derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$ is at most linear.*

Proof. Obviously $\mathsf{dl}(t,\to_{\mathcal{R}})$ is equal to $\mathsf{dl}(t,\to_{\mathcal{R}/\varnothing})$. By Theorem 8 with an SLI that counts the symbols $\mathsf{bits},\mathsf{half},\mathsf{s}$ further progress is achieved and the derivational complexity of $\mathcal{R}_{\mathsf{bits}}/\varnothing$ can be estimated by analyzing the complexity of the rule $\mathsf{bits}(\mathsf{s}(x)) \to \mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x))))$ relative to the other rules. For the last step there is an arctic interpretation of dimension three, i.e.,

$$\mathsf{bits}_{\mathcal{A}}(\vec{x}) = \begin{pmatrix} 0\,0\,0 \\ 0\,0\,3 \\ \$\,0\,0 \end{pmatrix}\vec{x} \qquad \mathsf{half}_{\mathcal{A}}(\vec{x}) = \begin{pmatrix} 0\,\$\,\$ \\ 0\,\$\,\$ \\ 0\,\$\,\$ \end{pmatrix}\vec{x} \qquad \mathsf{s}_{\mathcal{A}}(x) = \begin{pmatrix} 0\,\$\,0 \\ 0\,1\,2 \\ 2\,\$\,0 \end{pmatrix}x \qquad \mathsf{0}_{\mathcal{A}} = \begin{pmatrix} 0 \\ \$ \\ \$ \end{pmatrix}$$

that orients the problematic rule strictly and the remaining rules weakly. Hence the derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$ is at most linear. □

Note that the proof of Theorem 9 has been constructed automatically by CaT within a few seconds. Furthermore we stress that the involved reasoning is on a higher level compared to the handmade proof. This also eases the task of (automated) certification.

5 Derivation Gap Principle

Based on the success of the modular complexity approach using relative rewriting, in this section we aim to generalize the weight gap principle from [4] into the so-called *derivation gap principle*. Provided that $\mathcal{R}$ satisfies some property, the new result allows us to establish an upper bound on the derivational complexity of $\mathcal{R}\cup\mathcal{S}$ based on the derivational complexities of $\mathcal{R}/\mathcal{S}$ and $\mathcal{S}$.

Theorem 10. *Let $\mathcal{R}\cup\mathcal{S}$ be a TRS and t be terminating on $\mathcal{R}\cup\mathcal{S}$. If there exists a constant Δ such that for any $s\to_{\mathcal{R}} t$ we have $\mathsf{dl}(s,\to_{\mathcal{S}})+\Delta \geqslant \mathsf{dl}(t,\to_{\mathcal{S}})$ then $\mathsf{dc}(n,\to_{\mathcal{R}\cup\mathcal{S}}) \in \mathcal{O}(\mathsf{dc}(n,\to_{\mathcal{R}/\mathcal{S}})+\mathsf{dc}(n,\to_{\mathcal{S}}))$.*

Proof. We show that under the above assumptions there exists a constant K such that $\mathsf{dl}(t,\to_{\mathcal{R}\cup\mathcal{S}}) \leqslant K\cdot\mathsf{dl}(t,\to_{\mathcal{R}/\mathcal{S}})+\mathsf{dl}(t,\to_{\mathcal{S}})$. Consider a maximal derivation in $\mathcal{R}\cup\mathcal{S}$, written as follows:

$$s_1 \to_{\mathcal{S}}^{k_1} t_1 \to_{\mathcal{R}} s_2 \to_{\mathcal{S}}^{k_2} t_2 \to_{\mathcal{R}} \cdots \to_{\mathcal{R}} s_m \to_{\mathcal{S}}^{k_m} t_m$$

Since the derivation is maximal we have $\mathsf{dl}(s_1,\to_{\mathcal{R}\cup\mathcal{S}}) \leqslant \mathsf{dl}(s_1,\to_{\mathcal{R}/\mathcal{S}})+\sum_{1\leqslant i\leqslant m} k_i$. Obviously we have $\mathsf{dl}(s_1,\to_{\mathcal{S}}) \geqslant \mathsf{dl}(t_1,\to_{\mathcal{S}})+k_1$. Because $\mathsf{dl}(t_1,\to_{\mathcal{S}}) \geqslant \mathsf{dl}(s_2,\to_{\mathcal{S}})-\Delta$ according to underlying assumption we obtain $\mathsf{dl}(s_1,\to_{\mathcal{S}}) \geqslant \mathsf{dl}(s_2,\to_{\mathcal{S}})-\Delta+k_1$. An easy induction proof shows $\mathsf{dl}(s_1,\to_{\mathcal{S}})+m\cdot\Delta \geqslant \sum_{1\leqslant i\leqslant m} k_i$. Hence $\mathsf{dl}(s_1,\to_{\mathcal{R}\cup\mathcal{S}}) \leqslant \mathsf{dl}(s_1,\to_{\mathcal{R}/\mathcal{S}})+\mathsf{dl}(s_1,\to_{\mathcal{S}})+\Delta\cdot\mathsf{dl}(s_1,\to_{\mathcal{R}/\mathcal{S}})$ which can be simplified to $\mathsf{dl}(s_1,\to_{\mathcal{R}\cup\mathcal{S}}) \leqslant (\Delta+1)\cdot\mathsf{dl}(s_1,\to_{\mathcal{R}/\mathcal{S}})+\mathsf{dl}(s_1,\to_{\mathcal{S}})$. Taking $K=\Delta+1$ concludes the proof. □

To implement the above theorem the question arises how to check that $s\to_{\mathcal{R}} t$ implies the desired $\mathsf{dl}(s,\to_{\mathcal{S}})+\Delta \geqslant \mathsf{dl}(t,\to_{\mathcal{S}})$ for some constant Δ. Here the idea is to test $\mathsf{dl}(l,\to_{\mathcal{S}})+\Delta \geqslant \mathsf{dl}(r,\to_{\mathcal{S}})$ for any $l\to r\in\mathcal{R}$ and demand that additionally $C[l]+\Delta \geqslant C[r]$ and $l\sigma+\Delta \geqslant r\sigma$ holds for all contexts C and substitutions σ. Note that $\mathsf{dl}(l,\to_{\mathcal{S}})$ can always be under-approximated by 0 while $\mathsf{dl}(r,\to_{\mathcal{S}})$ can be over-approximated, e.g., by some interpretation of r.

As we know from [4] SLIs can be used to get a concrete instance of Theorem 10. Below we give counterexamples that TMIs, AMIs, and match-bounds do not adhere to the derivation gap principle without further ado. As future work we plan to restrict these criteria accordingly such that they become applicable for Theorem 10. The TRS $\mathcal{R}\cup\mathcal{S}$ in the next example was proposed by Hofbauer [6].

Example 11. Consider the following two TRSs

$$\mathcal{R} = \{\mathsf{c}(\mathsf{L}(x)) \to \mathsf{R}(x)\} \qquad \mathcal{S} = \{\mathsf{R}(\mathsf{a}(x)) \to \mathsf{b}(\mathsf{b}(\mathsf{R}(x))), \mathsf{R}(x) \to \mathsf{L}(x), \mathsf{b}(\mathsf{L}(x)) \to \mathsf{L}(\mathsf{a}(x))\}$$

We observe that the derivational complexity of $\mathcal{R} \cup \mathcal{S}$ is exponential because $\mathsf{c}^n(\mathsf{L}(\mathsf{a}(x))) \to^* \mathsf{L}(\mathsf{a}^{2^n}(x))$. Furthermore the derivational complexity of $\mathcal{R}/\mathcal{S}$ is linear (count c's). Since the TMI $\mathcal{M}_T$ with

$$\mathsf{a}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\vec{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \mathsf{b}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\vec{x} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \mathsf{R}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 3 \\ 0 & 0 \end{pmatrix}\vec{x} + \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad \mathsf{L}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\vec{x}$$

orients all rules in $\mathcal{S}$ strictly—and hence gives a quadratic upper bound on $\mathsf{dc}(n, \to_{\mathcal{R}\cup\mathcal{S}})$—(in general) TMIs cannot adhere to Theorem 10. The problem is that although there exists a Δ with $\mathsf{dl}(l, \to_{\mathcal{S}}) + \Delta \geqslant \mathsf{dl}(r, \to_{\mathcal{S}})$ for all $l \to r \in \mathcal{R}$ this property is not closed under contexts; if arbitrary TMIs are considered. Similarly the AMI $\mathcal{M}_A$ (inducing at most linear derivational complexity of $\mathcal{S}$) with

$$\mathsf{a}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 0 & \$ \\ 3 & 3 \end{pmatrix}\vec{x} \qquad \mathsf{b}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 2 \\ \$ & 0 \end{pmatrix}\vec{x} \qquad \mathsf{R}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 1 & 3 \\ 0 & 2 \end{pmatrix}\vec{x} \qquad \mathsf{L}_{\mathcal{M}}(\vec{x}) = \begin{pmatrix} 0 & \$ \\ \$ & \$ \end{pmatrix}\vec{x}$$

violates the same requirement in Theorem 10 as $\mathcal{M}_T$ above. A similar reasoning holds for match-bounds; one easily verifies that the TRS $\mathcal{S}$ is match-bounded by 2.

Summary and Conclusion

In this note we gave two proofs that the derivational complexity of the TRS $\mathcal{R}_{\mathsf{bits}}$ is linear. The first proof gives much insight into the reductions this system admits but the argument is hard for automation. The second proof has been generated automatically, is formal but does not give much insight into the system. The second part of the note generalized the weight gap principle of Hirokawa and Moser [4]. As future work we plan to investigate restrictions of TMIs, AMIs and match-bounds such that they adhere to Theorem 10.

References

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

[2] Geser, A., Hofbauer, D., Waldmann, J., Zantema, H.: On tree automata that certify termination of left-linear term rewriting systems. I&C 205(4), 512–534 (2007)

[3] Hirokawa, N., Middeldorp, A.: Polynomial interpretations with negative coefficients. In: AISC 2004. LNCS (LNAI), vol. 3249, pp. 185–198 (2004)

[4] Hirokawa, N., Moser, G.: Automated complexity analysis based on the dependency pair method. In: IJCAR 2008. LNCS, vol. 5195, pp. 364–379 (2008)

[5] Hofbauer, D., Lautemann, C.: Termination proofs and the length of derivations (preliminary version). In: RTA 1989. LNCS, vol. 355, pp. 167–177 (1989)

[6] Hofbauer, D., Waldmann, J.: Complexity bounds from relative termination proofs. Talk at the Workshop on Proof Theory and Rewriting, Obergurgl (2006). Available from `http://www.imn.htwk-leipzig.de/~waldmann/talk/06/rpt/rel/main.pdf`

[7] Koprowski, A., Waldmann, J.: Arctic termination ... below zero. In: RTA 2008. LNCS, vol. 5117, pp. 202–216 (2008)

[8] Koprowski, A., Waldmann, J.: Max/plus tree automata for termination of term rewriting. AC 19(2), 357–392 (2009)

[9] Moser, G., Schnabl, A., Waldmann, J.: Complexity analysis of term rewriting based on matrix and context dependent interpretations. In: FSTTCS 2008. LIPIcs, vol. 2, pp. 304–315 (2008)

[10] People, V.: Termination problem data base (2010). See `http://termination-portal.org/`

[11] Zankl, H., Korp, M.: Modular complexity analysis via relative complexity. In: RTA 2010. LIPIcs. (2010). To appear.