

# Towards the Certification of Complexity Proofs<sup>\*</sup>

René Thiemann

Institute of Computer Science, University of Innsbruck, Austria

**Abstract.** We report on our formalization of matrix-interpretation in Isabelle/HOL. Matrices are required to certify termination proofs and we wish to utilize them for complexity proofs, too. For the latter aim, only basic methods have already been integrated, and we discuss some upcoming problems which arise when formalizing more complicated results on matrix-interpretations, which are based on Cayley-Hamilton's theorem or joint-spectral radius theory.

## 1 Introduction

IsaFoR is an Isabelle/HOL [14] Formalization of Rewriting [17]. The initial aim in the development of IsaFoR was the certification of termination proofs of term rewrite systems (TRSs). Here, several important techniques like recursive path orders, polynomial orders, matrix interpretations, and dependency pairs have been formalized in a deep embedding. All these termination techniques are accompanied with executable algorithms which guarantee the correct application of these techniques in some termination proof that should be certified. The corresponding certifier CēTA (*Certified Termination Analysis*) is obtained by invoking Isabelle's code-generator [6] on these executable algorithms.

In the mean time, most termination techniques that are applied in current termination tools for TRSs can indeed be certified, and IsaFoR was extended towards other interesting rewriting related topics like confluence, completion, and complexity analysis.

In the sequel, we will report on our formalization of complexity analysis, where we will concentrate on one specific method: matrix interpretations for inferring polynomial complexity bounds. To this end, we will shortly recapitulate some theory on term rewriting and matrix interpretations in Sec. 2. Our formalization of matrix interpretations for termination proofs is presented in Sec. 3. We discuss the extension to complexity proofs in Sec. 4 where we also discuss some open problems.

All formalizations described in this paper are available from the AFP-entry [16] or from the IsaFoR-library (<http://cl-informatik.uibk.ac.at/software/ceta>).

## 2 Preliminaries

We assume familiarity with term rewriting [1]. Still, we recall the most important notions that are used later on. A *term*  $t$  over a set of *variables*  $\mathcal{V}$  and a set of

<sup>\*</sup> This research is supported by the Austrian Science Fund (FWF): P22767-N13.

function symbols  $\mathcal{F}$  is either a variable  $x \in \mathcal{V}$  or an  $n$ -ary function symbol  $f \in \mathcal{F}$  applied to  $n$  argument terms  $f(t_1, \dots, t_n)$ . We write  $|t|$  for the size of a term.

A *rewrite rule* is a pair of terms  $\ell \rightarrow r$  and a TRS  $\mathcal{R}$  is a set of rewrite rules. The *rewrite relation (induced by  $\mathcal{R}$ )*  $\rightarrow_{\mathcal{R}}$  is the closure under substitutions and under contexts of  $\mathcal{R}$ , i.e.,  $s \rightarrow_{\mathcal{R}} t$  iff there is a context  $C$ , a rewrite rule  $\ell \rightarrow r \in \mathcal{R}$ , and a substitution  $\sigma$  such that  $s = C[\ell\sigma]$  and  $t = C[r\sigma]$ . A TRS  $\mathcal{R}$  is *terminating*, written  $\text{SN}(\mathcal{R})$ , if there is no infinite derivation  $t_1 \rightarrow_{\mathcal{R}} t_2 \rightarrow_{\mathcal{R}} t_3 \rightarrow_{\mathcal{R}} \dots$ .

For a finite and terminating TRS  $\mathcal{R}$ , we its *derivational complexity*  $\text{dc}_{\mathcal{R}} : \mathbb{N} \rightarrow \mathbb{N}$  is defined as  $\text{dc}_{\mathcal{R}}(n) = \max\{k \mid \exists t_1 \dots t_k. |t_1| \leq n \wedge t_1 \rightarrow_{\mathcal{R}} t_2 \rightarrow_{\mathcal{R}} \dots \rightarrow_{\mathcal{R}} t_k\}$ .

One important termination technique is the usage of well-founded monotone algebras. In this approach, it is assumed that there is some algebra  $(\mathcal{A}, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$ , where  $\mathcal{A}$  is the universe and for each function symbol  $f$  of arity  $n$  we have an interpretation function  $f_{\mathcal{A}} : \mathcal{A}^n \rightarrow \mathcal{A}$ . Moreover, there is some well-founded order  $>$  on  $\mathcal{A}$  and all  $f_{\mathcal{A}}$  have to be monotone w.r.t.  $>$  in all their arguments.

Proving termination using well-founded monotone algebras can now be done by demanding for all assignments  $\alpha : \mathcal{V} \rightarrow \mathcal{A}$  and all rules  $\ell \rightarrow r \in \mathcal{R}$  that  $\llbracket \ell \rrbracket_{\alpha} > \llbracket r \rrbracket_{\alpha}$ . The reason is that then every rewrite step  $s = C[\ell\sigma] \rightarrow_{\mathcal{R}} C[r\sigma] = t$  leads to a strict decrease  $\llbracket s \rrbracket_{\alpha} > \llbracket t \rrbracket_{\alpha}$  w.r.t. the well-founded order. Polynomial orders [2, 10] are a well-known instance of well-founded monotone algebras where every  $f_{\mathcal{A}}$  is a polynomial,  $\mathcal{A} = \mathbb{N}$ , and  $>$  is the standard order on the naturals.

Despite proving termination, well-founded monotone algebras can also be used for complexity analysis. Assume  $>$  is the standard order on the naturals. Then for any ground term  $t$ , its interpretation  $\llbracket t \rrbracket$  is a bound on the length of each derivation starting in  $t$ . Hence, if we can find a bound  $b : \mathbb{N} \rightarrow \mathbb{N}$  such that  $\llbracket t \rrbracket \leq b(|t|)$  for all terms  $t$ , then  $\text{dc}_{\mathcal{R}}(n) \leq b(n)$  and thus,  $b$  is also a bound for the derivational complexity.

Unfortunately, if one considers polynomial orders, then the bound  $b$  can be double-exponential and this bound is tight [8]. Even for linear polynomial orders in general one can only infer an exponential bound. Only for a very restricted class of polynomial interpretations (strongly linear interpretations [3]), one achieves a linear complexity bound. So, when using polynomial interpretations we either have to impose severe restrictions to obtain a linear bound, or without this restriction we can only guarantee non-polynomial bounds.

Luckily, it turned out that other well-founded monotone algebras are useful for proving termination [5] and complexity [12]: matrix-interpretations. Matrix interpretations are similar to linear polynomial interpretations except that  $\mathcal{A}$  is the set of  $n$ -dimensional square matrices over some carrier.<sup>1</sup> To be more precise, every  $f_{\mathcal{A}}$  is of the form  $f_{\mathcal{A}}(x_1, \dots, x_n) = M_{f,0} + M_{f,1}x_1 + \dots + M_{f,n}x_n$  and matrices are compared by demanding a strict decrease in the upper-left entry, and a weak decrease in all remaining entries. To ease presentation we here assume that we have  $n$ -dimensional matrices of natural numbers, i.e.,  $\mathcal{A} = \mathbb{N}^{n \times n}$ .

<sup>1</sup> In [5], the universe consists of vectors, and the linear interpretations take matrices as coefficients. However, in the formalization of [4] and also in our formalization of matrix interpretations it was easier to always use matrices.

Using matrix interpretations, there are at least three approaches to estimate the value of  $\llbracket t \rrbracket$  depending on  $|t|$ . In all these techniques one collects the set of all matrix-coefficients  $\mathcal{M} = \{M_{f,i} \mid f \in \mathcal{F}, 1 \leq i \leq \text{arity of } f\}$  and it is easy to see that  $\llbracket t \rrbracket \leq |t| \cdot c \cdot \max\{N_1 \cdots N_{|t|} \mid N_i \in \mathcal{M}\}$  where  $c$  is some constant depending on  $\{M_{f,0} \mid f \in \mathcal{F}\}$ .

- (a) In [12] one approximates  $\{N_1 \cdots N_{|t|} \mid N_i \in \mathcal{M}\}$  by  $M_{\max}^{|t|}$  where  $M_{\max}$  is the pointwise maximum matrix of  $\mathcal{M}$ . Afterwards, a sufficient criterion to bound the value of  $M_{\max}^{|t|}$  is provided: if  $M_{\max}$  is upper triangular where all entries on the diagonal are at most 1, then the overall complexity is within  $\mathcal{O}(|t|^n)$  where  $n$  is the dimension of the matrix. This result is proven using a standard inductive proof.
- (b) In [13] the previous result is extended as follows: If all eigenvalues of the characteristic polynomial of  $M_{\max}$  are at most 1, then the overall complexity is within  $\mathcal{O}(|t|^m)$  where  $m$  is the multiplicity of eigenvalue 1. This result is proven via the theorem of Cayley-Hamilton [15].
- (c) Even more precise estimations can be gained by using theorems from joint spectral radius theory [9, 11] as these do not perform the rough approximation of  $\mathcal{M}$  via  $M_{\max}$ . Unfortunately, the corresponding mathematics is even more complicated than the Cayley-Hamilton theorem.

### 3 Formalizing Matrix Interpretations for Termination

Recall that the aim of our formalization is to obtain an executable program, `CeTA`, that is able to certify proofs with matrix interpretations of arbitrary dimensions.

For the formalization of matrices itself, there are several options:

- If there are dependent types, then the obvious choice is to model matrices in  $\mathbb{N}^{n \times n}$  as lists of lists of length  $n$ . However, we are working in Isabelle/HOL, so this is not an option in our case.
- Alternatively, one can use the idea to model matrices as functions of type  $I \rightarrow I \rightarrow A$  where  $I$  is some finite index type and where the cardinality of  $I$  corresponds to the dimension [7].

Unfortunately, this trick is not possible in our case, since as far as we see, for code-generation it is necessary to instantiate the index type  $I$  above for every dimension that we would require. However, the dimensions of matrices that will be used in the certificates can be arbitrary without any bound on the dimension.

- The representation of Steven Obua uses a type  $\mathbb{N} \rightarrow \mathbb{N} \rightarrow A$  with the restriction that only finitely entries are non-zero.<sup>2</sup> Here, we see two problems, namely executability of matrix comparisons and moreover, this representation does not allow to define a 1-matrix, which would be inconvenient for our purposes.

<sup>2</sup> See `HOL/Matrix-LP/Matrix.thy` for details.

- Define matrixes just as lists of lists and use predicates as *guards* to ensure that the dimensions fit.

We formalized several matrix-operations using the last approach with guards. Essentially, all our theorems look as follows where  $\text{mat}^{m,n}(M)$  is a predicate that ensures that  $M$  represents an  $m \times n$ -matrix, i.e., the outer list is of length  $m$  and all inner lists have length  $n$ .

$$\text{mat}^{m,n}(M) \implies \text{mat}^{m,n}(N) \implies M + N = N + M \quad (1)$$

$$\text{mat}^{m,n}(M) \implies \text{mat}^{m,n}(N) \implies \text{mat}^{m,n}(M + N) \quad (2)$$

$$\text{mat}^{m,n}(M) \implies \text{mat}^{m,n}(N) \implies (M = N) = (\forall ij. M_{ij} = N_{ij}) \quad (3)$$

$$\text{mat}^{m,n}(M) \implies \text{mat}^{m,n}(N) \implies (M + N)_{ij} = M_{ij} + N_{ij} \quad (4)$$

Here, property (1) states that matrix addition is commutative (silently assuming that the addition on the underlying carrier is commutative), but this fact is guarded by the condition that the matrix dimensions of both matrices fit together.

The next kind of property (2) states that matrix addition preserves the dimensions which is required to perform reasoning within contexts, e.g., to prove  $(M + N) + K = K + (M + N)$  where we only know the dimensions of matrices  $M$ ,  $N$ , and  $K$ .

Property (3) was somehow the key to prove most properties of basic matrix operations: Instead of comparing matrices using their representing type, i.e., the inductive type of lists, we do a pointwise comparison. And then a property like (4) just states that the algorithm for addition (which is defined recursively over lists) is correct w.r.t. the pointwise definition of matrix addition. Afterwards, all properties involving addition use the characterization of (4) instead of the concrete implementation on lists. For example, the prove of (1) becomes trivial using (3), (4), and commutativity of the addition on the carrier, and does not require any induction.

Using this representation of matrices, code-generation works without any problems, since all algorithms like matrix-addition, -multiplication, etc. are just algorithms on lists. However, it has one major disadvantage, namely that we cannot use matrices in combination with the standard classes like *group* or *semiring* from the Isabelle-distribution since these require equalities like  $M + N = N + M$  without the additional guards that we impose. As one example consequence, it is not possible to combine the polynomial library of Clemens Ballarin from the Isabelle-distribution<sup>3</sup> with our matrix library.

To this end, we had to develop our own library on linear polynomials which works on guarded operations and requires properties like (1) and (2). It is needless to say that working with these guards is by far more cumbersome than working with the similar unguarded classes from the distribution.

<sup>3</sup> In `HOL/Algebra/abstract/Ring2.thy` one can see the (unguarded) requirements for `HOL/Algebra/poly/Polynomial.thy`

## 4 Formalizing Matrix Interpretations for Complexity

The switch from termination to complexity proofs via matrix interpretations poses one additional challenge, namely that of estimating values or growth-rates of matrices.

So far, we formalized the approach of (a) using triangular matrices. Already in this technique, an unexpected challenge has occurred in formalizing that the linear matrix norm is sub-multiplicative, i.e.,  $\|M \cdot N\| \leq \|M\| \cdot \|N\|$ . In the literature we only found proofs for matrices over real or complex numbers which are based on a suprema- or limit-construction. However, in our setting we would like to have this statement for matrices over arbitrary carriers like the naturals, the integers, or the rationals. Therefore, we developed our own proof which works by induction over the shared dimension of  $M$  and  $N$  and is about 200 lines long (in Isabelle).

For the future, when extending our work towards the more sophisticated methods of (b) and (c) we would like to minimize the effort in finding new proofs, e.g., for (b) we plan to first formalize the theorem of Cayley-Hamilton and then use it in the same way as it is done in [13]. However, here already in the setup there is one major obstacle: the theorem of Cayley-Hamilton requires non-linear polynomials over matrices, and we are not aware of any Isabelle library on non-linear polynomials that is able to deal with guarded semirings like our matrices.

So, the questions to the Isabelle-community would be, whether

- someone has already done work on non-linear polynomials using guarded semirings?
- one should try to generalize the existing classes like semiring and the existing polynomial library from the distribution to work with guards?
- one should develop an independent formalization of non-linear polynomials including guards?
- we overlooked something, and there is a possibility to use matrices of arbitrary dimension in combination with code-extraction and the existing library on polynomials?

## References

1. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
2. A. Ben Cherifa and P. Lescanne. Termination of Rewriting Systems by Polynomial Interpretations and Its Implementation. *Science of Computer Programming*, 9(2):137–159, 1987.
3. G. Bonfante, A. Cichon, J.-Y. Marion, and H. Touzet. Algorithms with polynomial interpretation termination proof. *Journal of Functional Programming*, 11(1):33–53, 2001.
4. P. Courtieu, G. Gbedo, and O. Pons. Improved matrix interpretation. In *Proceedings of the 36th Conference on Current Trends in Theory and Practice of Computer Science*, volume 5901 of *Lecture Notes in Computer Science*, pages 283–295, 2010.

5. J. Endrullis, J. Waldmann, and H. Zantema. Matrix Interpretations for Proving Termination of Term Rewriting. *Journal of Automated Reasoning*, 40(2-3):195–220, 2008.
6. F. Haftmann and T. Nipkow. Code generation via higher-order rewrite systems. In *Proceedings of the 10th International Symposium on Functional and Logic Programming*, volume 6009 of *Lecture Notes in Computer Science*, pages 103–117, 2010.
7. J. Harrison. A HOL theory of euclidean space. In *Proceedings of the 18th International Conference on Theorem Proving in Higher Order Logics*, volume 3603 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2005.
8. D. Hofbauer and C. Lautemann. Termination proofs and the length of derivations. In *Proceedings of the 3rd International Conference on Rewriting Techniques and Applications*, volume 355 of *Lecture Notes in Computer Science*, pages 167–177, 1989.
9. R. Jungers. *The Joint Spectral Radius: Theory and Applications*. Springer, 2009.
10. D. Lankford. On proving term rewriting systems are Noetherian. Technical Report MTP-3, Louisiana Technical University, Ruston, LA, USA, 1979.
11. A. Middeldorp, G. Moser, F. Neurauter, J. Waldmann, and H. Zankl. Joint spectral radius theory for automated complexity analysis of rewrite systems. In *Proceedings of the 4th International Conference on Algebraic Informatics*, volume 6742 of *Lecture Notes in Computer Science*, pages 1–20, 2011.
12. G. Moser, A. Schnabl, and J. Waldmann. Complexity analysis of term rewriting based on matrix and context dependent interpretations. In *Proceedings of the 28th International Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 2 of *Leibniz International Proceedings in Informatics*, pages 304–315, 2008.
13. F. Neurauter, H. Zankl, and A. Middeldorp. Revisiting matrix interpretations for polynomial derivational complexity of term rewriting. In *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, volume 6397 of *Lecture Notes in Computer Science*, pages 550–564, 2010.
14. T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, 2002.
15. H. E. Rose. *Linear Algebra: A Pure Mathematical Approach*. Birkhäuser, 2002.
16. C. Sternagel and R. Thiemann. Executable matrix operations on matrices of arbitrary dimensions. *Archive of Formal Proofs*, 2010.
17. R. Thiemann and C. Sternagel. Certification of termination proofs using CeTA. In *Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics*, volume 5674 of *Lecture Notes in Computer Science*, pages 452–468, 2009.