Efficient Certification of Complexity Proofs

Formalizing the Perron–Frobenius Theorem

(Invited Talk Paper)

Jose Divasón
Universidad de La Rioja
Spain
jose.divasonm@unirioja.es

Sebastiaan Joosten
University of Twente
the Netherlands
s.j.c.joosten@utwente.nl

Ondřej Kunčar
Technical University of Munich
Germany
kuncar@in.tum.de

René Thiemann
University of Innsbruck
Austria
rene.thiemann@uibk.ac.at

Akihisa Yamada
National Institute of Informatics
Japan
akihisa.yamada@uibk.ac.at

Abstract

Matrix interpretations are widely used in automated complexity analysis. Certifying such analyses boils down to determining the growth rate of A^n for a fixed non-negative rational matrix A. A direct solution for this task involves the computation of all eigenvalues of A, which often leads to expensive algebraic number computations.

In this work we formalize the Perron–Frobenius theorem. We utilize the theorem to avoid most of the algebraic numbers needed for certifying complexity analysis, so that our new algorithm only needs the rational arithmetic when certifying complexity proofs that existing tools can find. To cover the theorem in its full extent, we establish a connection between two different Isabelle/HOL libraries on matrices, enabling an easy exchange of theorems between both libraries. This connection crucially relies on the transfer mechanism in combination with local type definitions, being a non-trivial case study for these Isabelle tools.

CCS Concepts • **Theory of computation → Algebraic complexity theory**; **Logic and verification**;

Keywords Complexity, Isabelle/HOL, Multivariate Analysis, Spectral Radius

ACM Reference Format:
Jose Divasón, Sebastiaan Joosten, Ondřej Kunčar, René Thiemann, and Akihisa Yamada. 2018. Efficient Certification of Complexity Proofs: Formalizing the Perron–Frobenius Theorem (Invited Talk Paper). In *Proceedings of 7th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP'18).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3167103

CPP'18, January 8–9, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5586-5/18/01.
https://doi.org/10.1145/3167103

1 Introduction

CeTA [17] is an Isabelle-formalized *certifier* which validates various kinds of proofs generated by untrusted program analyzers. One of the supported proofs is complexity proofs for term rewrite systems as generated by analyzers like AProVE, CaT, or TcT [1, 5, 21].

Although the most crucial information is contained in the proof (e.g., a measure function), a certain amount of computation is always left for the certifier, e.g., to verify that the measure decreases in every rewrite step.

This work aims at reducing the amount of computation in validating complexity proofs that use *matrix interpretations* [3]—a special kind of measure function where the domain consists of vectors. Matrix interpretations are an important technique for complexity analysis, for instance in the years 2015–2017 of *the Termination Competition* [6], at least 30 % of the machine readable complexity proofs contain matrix interpretations.

Example 1.1. Consider the following implementation of insertion sort.

$$
\begin{aligned}
\mathsf{sort}(\mathsf{Cons}(x, xs)) &\to \mathsf{insort}(x, \mathsf{sort}(xs)) \\
\mathsf{sort}(\mathsf{Nil}) &\to \mathsf{Nil} \\
\mathsf{insort}(x, \mathsf{Cons}(y, ys)) &\to \mathsf{Cons}(x, \mathsf{Cons}(y, ys)) \quad | \; x \le y \\
\mathsf{insort}(x, \mathsf{Cons}(y, ys)) &\to \mathsf{Cons}(y, \mathsf{insort}(x, ys)) \quad | \; x \not\le y \\
\mathsf{insort}(x, \mathsf{Nil}) &\to \mathsf{Cons}(x, \mathsf{Nil})
\end{aligned}
$$

The complexity analyzer TcT claims the runtime complexity of this example to be $O(n^2)$, using the following matrix

interpretation as a proof.

$$[\![\mathsf{sort}]\!](xs) = \begin{bmatrix} 3 & 3 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot xs$$

$$[\![\mathsf{insort}]\!](x, xs) = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot xs + \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$$

$$[\![\mathsf{Cons}]\!](x, xs) = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{A} \cdot xs + \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

$$[\![\mathsf{Nil}]\!] = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

It is easy to validate that this interpretation proves termination, i.e., that in every rewrite step from s to t the measure decreases: a vector $[\![s]\!]$ is larger than $[\![t]\!]$ if there is a strict decrease of the first entry and a weak decrease elsewhere. For instance, to validate the strict decrease of the first rule for sort, the following computation is performed.

$$[\![\mathsf{sort}(\mathsf{Cons}(x, xs))]\!] = \begin{bmatrix} 3 & 3 & 3 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot [\![xs]\!] + \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} > \begin{bmatrix} 3 & 3 & 3 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot [\![xs]\!] + \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} = [\![\mathsf{insort}(x, \mathsf{sort}(xs))]\!]$$

It remains to validate that this matrix interpretation ensures the $O(n^2)$ runtime of sort, i.e., that the entries of the vector $[\![\mathsf{sort}(\mathsf{Cons}(x_1, \ldots \mathsf{Cons}(x_n, \mathsf{Nil})))]\!]$ are in $O(n^2)$. We have

$$[\![\mathsf{sort}(\mathsf{Cons}(x_1, \ldots \mathsf{Cons}(x_n, \mathsf{Nil})))]\!] = \begin{bmatrix} 3 & 3 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \left(A^n \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} + \sum_{i<n} A^i \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \right) \in O(n \cdot A^n)$$

where A is the coefficient matrix of Cons. Thus, it remains to validate $A^n \in O(n)$.

As illustrated above, the certification of complexity proofs with matrix interpretations boils down to determining the growth rate of matrix powers A^n. There is a conceptually simple algorithm for this task.

We illustrate Algorithm 1 with the help of Figure 1a. In order to guarantee polynomial growth rate, in step 3 we ensure that there is no eigenvalue outside the unit circle (gray). In order to determine the degree of the polynomial we consider eigenvalues on the unit circle (black) and check their Jordan blocks in step 4. Eigenvalues strictly within the unit circle can be ignored (white).

Algorithm 1 works well on Example 1.1: it determines the two eigenvalues 0 and 1 and computes the set of Jordan blocks for eigenvalue 1, which contains only the size-two Jordan block $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ in this case. Hence, one can deduce $A^n \in O(n)$ and thus, the complexity analysis by TcT is validated.

Algorithm 1: Certifying $A^n \in O(n^d)$.

Input: Matrix A and degree d.
Output: Accept or assertion failure.

1 Compute all eigenvalues $\lambda_1, \ldots, \lambda_n$ of A, i.e., all complex roots of the characteristic polynomial of A
2 Let ρ_A be the spectral radius of A, i.e., $\rho_A = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$
3 Assert $\rho_A \leq 1$.
4 For each eigenvalue of λ_i with $|\lambda_i| = 1$, and for each Jordan block of A and λ_i with size s, assert $s \leq d + 1$.
5 Accept

Unfortunately, it is not always the case that Algorithm 1 works well as it requires expensive irrational arithmetic as we will see in the following example.

Example 1.2. Consider the matrix A defined as

$$A = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

The characteristic polynomial is

$$\chi_A = \frac{(x-1)\,(8x^3 - 4x^2 - 2x - 1)}{8}$$

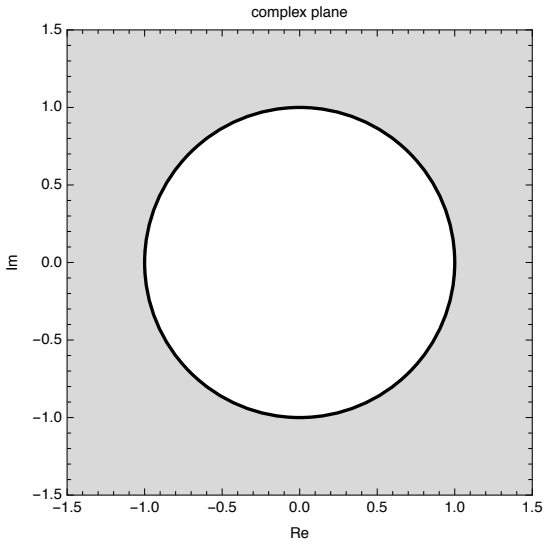
so the eigenvalues $\lambda_1, \ldots, \lambda_4$ of A, indicated in Figure 1b, are precisely the roots of χ_A.

Using our formalized algebraic number library [18], for step 1 we can compute them precisely as follows:

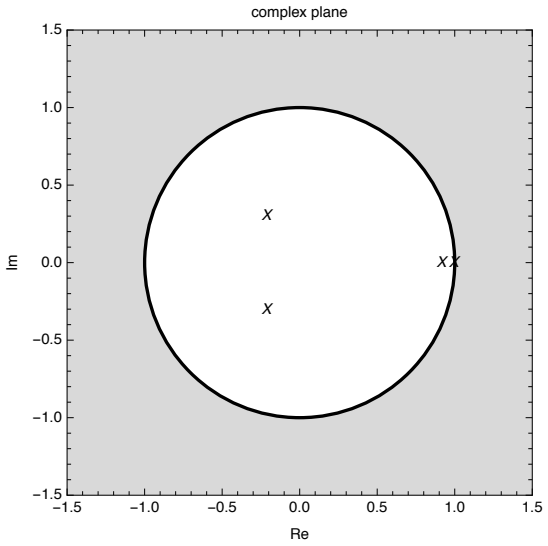
$$\lambda_1 = 1$$
$$\lambda_2 = (\text{root \#1 of } f_1)$$
$$\lambda_3 = (\text{root \#1 of } f_2) + (\text{root \#1 of } f_3)\mathrm{i}$$
$$\lambda_4 = (\text{root \#1 of } f_2) + (\text{root \#2 of } f_3)\mathrm{i}$$

Here, (root #k of f) denotes the k-th greatest real root of polynomial f, and the polynomials $f_1, \ldots, f_4$ are defined as follows:

$$f_1(x) = 8x^3 - 4x^2 - 2x - 1$$
$$f_2(x) = 32x^3 - 16x^2 + 1$$
$$f_3(x) = 1024x^6 + 512x^4 + 64x^2 - 11$$
$$f_4(x) = 64x^6 + 16x^4 + 4x^2 - 1$$



(a) relevant areas



(b) eigenvalues of Example 1.2

Figure 1. Illustration of Algorithm 1

For step 2 we further compute

$$\begin{aligned}
|\lambda_1| &= 1\\
|\lambda_2| &= |(\text{root \#1 of } f_1)| = (\text{root \#1 of } f_1)\\
|\lambda_3| &= \sqrt{(\text{root \#1 of } f_2)^2 + (\text{root \#1 of } f_3)^2}\\
&= (\text{root \#2 of } f_4)\\
|\lambda_4| &= \sqrt{(\text{root \#1 of } f_2)^2 + (\text{root \#2 of } f_3)^2}\\
&= (\text{root \#2 of } f_4)
\end{aligned}$$

and since $|\lambda_2| < 1$ and $|\lambda_3| = |\lambda_4| < 1$, we get $\rho_A = 1$.

We continue to step 4. Omitting details, it turns out that the (only) Jordan block for λ_1 has size $s = 1$, so the algorithm accepts for any d since $1 \leq d + 1$.

In this work, we formalize and utilize *the Perron–Frobenius theorem* [4, 15] to modify Algorithm 1 to avoid the *explicit* computation of all eigenvalues, and moreover reduce the number of considered eigenvalues, so that the gray and black area in Figure 1a are significantly reduced.

The basic version of the Perron–Frobenius theorem is stated as follows:

Theorem 1.3 (Perron–Frobenius, basic version)**.** *For a non-negative real matrix A, the spectral radius ρ_A is an eigenvalue of A.*

A simple consequence is that step 3 of Algorithm 1 can be replaced by only checking that there are no real eigenvalues above 1. Graphically this means that we can switch from Figure 1a to Figure 2a, significantly reducing the gray area.

Based on further results of Perron–Frobenius for irreducible matrices, we arrive at the following key theorem for certifying complexity proofs.

Theorem 1.4. *For a non-negative real matrix A, the characteristic polynomial χ_A can be factored into*

$$\chi_A = f \cdot \prod_{k \in K} (x^k - \rho_A^k)$$

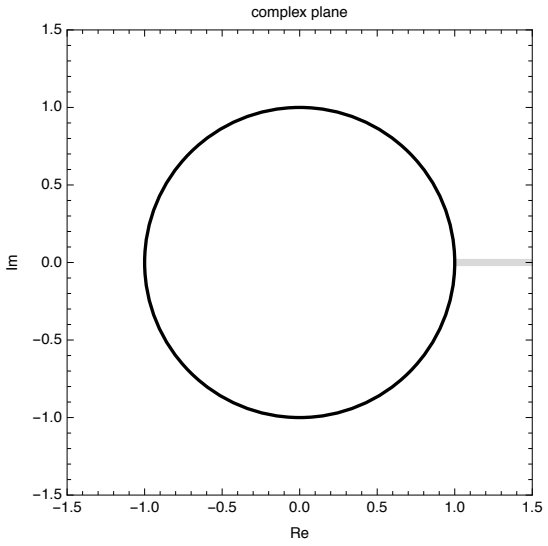
for some polynomial f and non-empty multiset K where all complex roots of f have a norm strictly less than ρ_A.

Theorem 1.4 permits us to reduce the black circle in Figure 1a to a finite number of points, namely to the roots of unity up to a certain degree, determined by the dimension k of the input matrix. Figure 2b shows the roots of unity up to degree 5, labeled by the smallest k at which our algorithm has to consider the point; it suffices to consider only the (potential) eigenvalue 1 for matrices of dimension up to 4, $\{1, -1\}$ for dimension 5, $\{1, -1, \frac{1+i\sqrt{3}}{2}, \frac{1-i\sqrt{3}}{2}\}$ for dimension 6 and 7, and so on. So, in Example 1.2 our improved certification algorithm only requires rational number arithmetic instead of algebraic number computations.

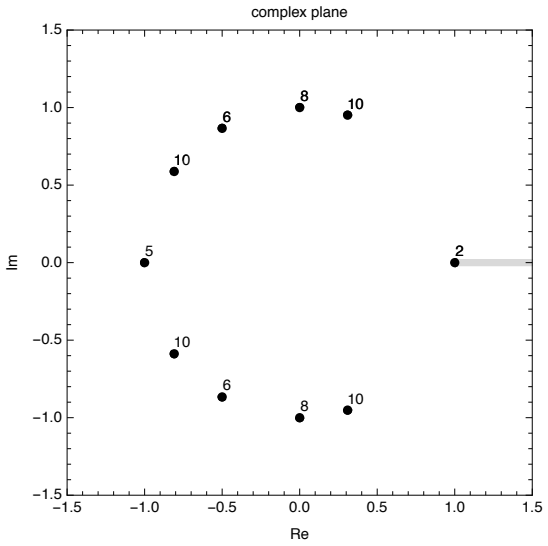
The paper is structured as follows.

We present some preliminaries on linear algebra in Section 2. This section also introduces two different representations of matrices and shows their differences. HMA is Isabelle/HOL's [13] library on multivariate analysis and JNF is our matrix library in the archive of formal proofs (AFP) which allows essential flexibility for formalizing Jordan normal forms [19].

Next, we provide details on our formalization of Theorem 1.3 in Section 3. We closely follow a paper proof [16] using Brouwer's fixpoint theorem and use HMA as matrix representation.



(a) using Theorem 1.3



(b) using Theorem 1.4

Figure 2. Improvements over Algorithm 1

Afterwards, in Section 4 we create a bridge between the HMA world and JNF world which permits to easily transfer theorems from one world to the other.

This bridge is essential for a more elaborate proof of the Perron–Frobenius theorem for irreducible matrices. It is illustrated in Section 5 and contains many more properties than just the one that the spectral radius is an eigenvalue.

In Section 6 we explain the formalization of our key Theorem 1.4. It is not restricted to irreducible matrices and its formalization requires JNF matrices.

We further prove in Section 7 that for matrices of dimension up to 4, the spectral radius is not only an eigenvalue, but it is also the eigenvalue that has the largest Jordan block among all eigenvalues with maximal norm.

Finally, we integrate our findings to an efficient complexity checker and verify its soundness in Section 8. This complexity checker is also integrated in CeTA. It is five times faster than Algorithm 1 on standard benchmarks and easily solves larger examples which were not feasible before.

We conclude in Section 9 and shortly explain why our improved complexity checker also has the potential to increase the power of automated tools.

The whole formalization is available in the AFP:

https://www.isa-afp.org/entries/Perron_Frobenius.html

2 Preliminaries

We assume basic knowledge of linear algebra and analysis. We use letters u, v, x, y, z for vectors, A, B, C, D for matrices, λ for eigenvalues, and i, j for matrix indices. Sometimes x is also used as the variable of a uni-variate polynomial. To denote the i-th row j-th column element of a matrix A, we often write A_{ij} or $A\ i\ j$ in Isabelle sources. We usually write the function application by parenthesis and use juxtaposition for multiplication in mathematical text, whereas in formal Isabelle sources juxtaposition is function application and multiplication is written explicitly. So, the expression $Af(x)$ is written as $A \cdot f\ x$ when presenting it as Isabelle source.

We write $\|v\|_1$ for the linear norm of a vector v and $\|v\|$ for the Euclidean norm, i.e., $\|v\|_1 = \sum_i |v_i|$ and $\|v\| = \sqrt{\sum_i |v_i|^2}$.

We write I for the identity matrix, $det(A)$ for the determinant of a matrix A, and χ_A for the characteristic polynomial of A, i.e., $\chi_A(x) = det\,(xI - A)$. A vector $v \neq 0$ is an *eigenvector* of A with *eigenvalue* λ, if $Av = \lambda v$. It is well known that λ is an eigenvalue of A iff $\chi_A(\lambda) = 0$, and that χ_A is a monic polynomial where the degree of χ_A is the same as the dimension of A. Two matrices A and B are similar if $A = P^{-1}BP$ for some invertible matrix P. Similar matrices have the same characteristic polynomial and the same eigenvalues.

The *spectral radius* ρ_A of A is defined as $\max\{\,|\lambda| \mid \lambda \in \mathbb{C}, \chi_A(\lambda) = 0\,\}$, i.e., ρ_A is the largest norm of the eigenvalues of A. An eigenvalue λ is *maximal* if $|\lambda| = \rho_A$.

For each matrix A we associate a directed graph where there is an edge $i \to j$ iff $A_{ij} \neq 0$. A matrix A is *irreducible* if the graph of A is connected, i.e., for every index i and j there is a non-empty path from i to j.

We compare vectors and matrices pointwise, e.g., $A \geq B$ is defined as $A_{ij} \geq B_{ij}$ for all indices i and j.

The *roots of unity of degree k* are precisely the complex values x satisfying $x^k = 1$. The *primitive* roots of unity of degree k are those roots of unity of degree k which do not satisfy $x^\ell = 1$ for any $0 < \ell < k$. For instance, the roots of unity of degree 4 are 1, −1, i, and −i, whereas only i and −i are primitive roots of unity of degree 4.

In earlier work [19], we formalized the theory of *Jordan normal forms (JNFs)* in Isabelle/HOL. For this paper, it is only important to know that one can prove the soundness of Algorithm 1 with the help of JNFs; that the sum of the sizes

of the Jordan blocks for A and λ is precisely the algebraic multiplicity of λ, i.e., the order of λ as root of χ_A; and that there is a verified algorithm for step 4 in the Algorithm 1: it computes the set of Jordan blocks for A and λ via the Gaussian elimination.

2.1 HMA Matrix Representation

Matrices in HMA are represented using ideas by Harrison [8]: matrices with elements of type α are essentially[1] represented as functions of type $'n \to 'm \to \alpha$ where $'n$ and $'m$ are type variables which are restricted to have finitely many elements. Vectors are represented correspondingly as functions of type $'n \to \alpha$.

The HMA representation has the advantage that the type system can enforce compatible dimensions. For instance, matrix addition will have type $('n \to 'm \to \alpha) \to ('n \to 'm \to \alpha) \to ('n \to 'm \to \alpha)$ and is defined as $A + B = (\%i\, j.\, A\, i\, j + B\, i\, j)$.[2] Consequently, the library contains (unconditional) lemmas like the associativity of matrix addition: $A + (B + C) = (A + B) + C$.

Several results and algorithms on HMA matrices are provided in the Isabelle distribution, e.g., that real vectors form a Euclidean space.

There is however also a disadvantage of this representation: it is cumbersome, if possible, to change the dimension of the matrix, or to decompose matrices into blocks. To wit, consider formulating Strassen's matrix multiplication algorithm using the HMA representation; in the recursion you will have to find new types which represent the upper/lower-left/right blocks of a matrix.

2.2 JNF Matrix Representation

The JNF matrix representation is based on the characteristic function of a matrix, too, but the type of indices is always natural numbers and the dimension is made explicit. To be more precise, matrices have essentially the type $\alpha\ mat = \mathbb{N} \times \mathbb{N} \times (\mathbb{N} \to \mathbb{N} \to \alpha)$.[3]

A disadvantage of this approach compared to HMA is that the type system of Isabelle/HOL cannot express the constraint of compatible dimensions. For instance, matrix addition will have type $\alpha\ mat \to \alpha\ mat \to \alpha\ mat$ and is essentially defined as $(n, m, A) + (n', m', B) = (n, m, (\%i\, j.\, A\, i\, j + B\, i\, j))$. Clearly, there is a problem if $(n, m) \neq (n', m')$. Therefore, the library for JNF matrices works with explicit carriers: *carrier-mat n m* is the set of all matrices with dimension $n \times m$. In the sequel, we often identify a matrix (n, m, A) with its characteristic function A. The dimensions will mostly be visible in constraints on the carrier like $A \in$ *carrier-mat n m*.

Most lemmas in the JNF library require explicit conditions on dimensions; e.g., the associativity of matrix addition is stated as

$$A \in \textit{carrier-mat}\ n\ m \Longrightarrow B \in \textit{carrier-mat}\ n\ m \Longrightarrow C \in \textit{carrier-mat}\ n\ m \Longrightarrow A + (B + C) = (A + B) + C$$

Moreover, there are also auxiliary lemmas which are not needed in the HMA representation at all, such as closure under addition:

$$A \in \textit{carrier-mat}\ n\ m \Longrightarrow B \in \textit{carrier-mat}\ n\ m \Longrightarrow A + B \in \textit{carrier-mat}\ n\ m$$

Hence, working with JNF matrices is more tedious, but it also has some advantages. Changing the dimension of a matrix, or decomposing it, is straightforward using JNF matrices. For instance, for the upcoming formalization of the Perron–Frobenius theorem, the derivation rule for characteristic polynomials is essential. Here, *mat-delete A i j* deletes the i-th row and j-th column of a matrix A in JNF-representation.

Lemma 2.1. $A \in \textit{carrier-mat}\ n\ n \Longrightarrow$
$\textit{pderiv}\ (\textit{char-poly}\ A) = \sum_{i<n} \textit{char-poly}\ (\textit{mat-delete}\ A\ i\ i)$

3 Perron–Frobenius, Basic Version

We start this section with a formalized version of the basic Perron–Frobenius theorem, Theorem 1.3. It additionally contains the property that the spectral radius has an associated *non-negative real* eigenvector.

Theorem 3.1 (Isabelle/HOL version of Theorem 1.3)**.**
$\textit{real-non-neg-mat}\ A \Longrightarrow$
$\exists v.\, \textit{eigen-vector}\ A\ v\ (\textit{of-real}\ (\textit{spectral-radius}\ A)) \wedge$
$\quad \textit{real-non-neg-vec}\ v$

We only present an informal short proof of Theorem 3.1 following a textbook by Serre [16, Theorem 5.2.1]. The proof is based on Brouwer's fixpoint theorem.

Theorem 3.2 (Brouwer for $\mathbb{R}^n$)**.** *Let $S \subseteq \mathbb{R}^n$ be a non-empty, compact and convex set. Let f be a continuous function from S to S. Then f has a fixpoint x, i.e., $x \in S$ and $f(x) = x$.*

Proof of Theorem 3.1. Define $S := \{v \mid \|v\|_1 = 1 \wedge v \geq 0 \wedge Av \geq \rho_A v\}$. Consider two cases.

If there is some $x \in S$ such that $Ax = 0$, then $Ax = 0x$, so x is a non-negative real eigenvector with eigenvalue 0. Since $x \in S$ we conclude

$$0 = Ax \geq \rho_A x$$

and as $x \neq 0$ and $x \geq 0$, this implies $0 \geq \rho_A$. Hence, $\rho_A = 0$ since $\rho_A \geq 0$ by the definition of the spectral radius.

[1] The actual Isabelle/HOL definition uses an isomorphic copy of $'n \to 'm \to \alpha$. In this paper we will neglect this aspect and just identify an HMA matrix with its characteristic function.

[2] In this paper, we use $\%i.\ f\ i$ as syntax for lambda-expressions, since λ is already used to denote eigenvalues.

[3] The actual Isabelle/HOL type definition additionally enforces that the characteristic function returns a fixed value—*undefined*—for indices beyond the dimension. In this way, only the intended part of the characteristic function becomes relevant when specifying matrix equality.

In the other case, we know that $Av \neq 0$ for all $v \in S$. Hence, we can define $f(v) := \frac{1}{\|Av\|_1} Av$ and by Brouwer's fixpoint theorem obtain some $x \in S$ such that $f(x) = x$.[4]

As in the previous case we easily conclude that x is a non-negative eigenvector for eigenvalue $\|Ax\|_1$:

$$Ax = \|Ax\|_1 f(x) = \|Ax\|_1 x$$

Moreover, since $x \in S$ we derive

$$\|Ax\|_1 x = Ax \geq \rho_A x$$

and hence $\|Ax\|_1 \geq \rho_A$. But since by definition $\rho_A \geq \|Ax\|_1$ we conclude $\|Ax\|_1 = \rho_A$. □

The Isabelle/HOL formalization of the above proof requires only 400 lines. It closely follows the informal proof using the HMA library. For the actual formalization we refer to the sources and here only mention some important aspects.

- The paper proof hides conversions between complex and real numbers, which however are frequently occurring in the formalization where there are different types for real and complex numbers.
- In order to apply Brouwer's fixpoint theorem, we need to prove the continuity of the function f. Unfortunately, the Isabelle distribution lacks continuity results for several operations on matrices like matrix multiplication. Here, we are grateful to Fabian Immler who gave us a short tutorial on how these proofs are best conducted within Isabelle/HOL: do not use *continuous-on*, but *tendsto*, *tendsto-intros* and friends.
- In the above proof it is essential to use the linear norm, since otherwise S is not necessarily convex. However, in HMA the vector norm is fixed to the Euclidean norm. Hence, we had to reprove certain lemmas about norms.

Let us now illustrate how to exploit Theorem 1.3 in Example 1.2 from the introduction.

Example 3.3. Instead of computing all eigenvalues, we directly apply step 3 of Algorithm 1. Here, we decide $\rho_A \leq 1$ by checking that there is no real root of χ_A in the interval $(1, \infty)$. The latter property can be easily verified using Sturm's method, whose formalization was already provided by Eberl [2]. Moreover, for step 4 we apply a cheap square-free factorization on χ_A to see that χ_A contains no duplicate factors and hence, no duplicate roots. Thus, the largest Jordan block is of size 1 and we can deduce that $A^n \in O(1)$.

Unfortunately, Theorem 1.3 combined with square-free factorization does not always suffice to precisely determine the asymptotic growth rate of A^n, without explicit computation of the complex eigenvalues.

[4] In order to prove that S is non-empty, pick some (complex) eigenvector to eigenvalue ρ_A, apply the norm-function on all components, and finally divide the whole vector by its linear vector norm.

Example 3.4. Consider the matrix

$$A = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

with characteristic polynomial:

$$\chi_A = \left(\frac{4x^2 + 2x + 1}{4}\right)^2 (x - 1)$$

One can easily check $\rho_A \leq 1$ as in Example 3.3. However, there are two complex roots with multiplicity 2, so we must know whether their norm is precisely 1, and if so, we must compute the sizes of their Jordan blocks. Theorem 1.3 does not provide any help in these tasks, so we would fall back to applying algebraic number computations to determine the complex roots λ_1 and λ_2 of $4x^2 + 2x + 1$ and to further decide whether $|\lambda_1| = 1$ and $|\lambda_2| = 1$ are satisfied—the answer is no.

4 Connecting HMA- and JNF-Matrices

In order to formally prove the stronger Theorem 1.4, we need to combine theorems of the HMA library and the JNF library. To this end, we establish a connection between both representations in the form of *transfer rules* [10]. The connection consists of two parts.

The first part is the definition of correspondence relations between matrices (or vectors, or indices) of JNF and HMA. We define functions to convert between indices, matrices and vectors of the two representations. For instance, for indices *from-nat* :: $\mathbb{N} \to$ *'n* is defined as an arbitrary bijection between $\{0, \ldots, CARD('n) - 1\}$ and the universe of *'n* which has *CARD('n)* many elements. The inverse function is *to-nat* :: *'n* $\to \mathbb{N}$. For matrices we define

definition *from-hma* :: *('n* $\to$ *'m* $\to$ α*)* $\to$ α *mat* ***where***
from-hma A =
(CARD('n), CARD('m), (% i j. A (from-nat i) (from-nat j)))

and a similar definition is available for conversion of vectors. Then it is easy to define when two indices, matrices, etc. are related. All of the following relations take two arguments and return a Boolean, where the first argument is an object of the JNF world, and the second argument is the corresponding object in the HMA world.[5]

definition *rel-I* :: $\mathbb{N} \to$ *'n* $\to$ *bool* ***where***
rel-I i j = *(i = to-nat j)*
definition *rel-M* :: α *mat* $\to$ *('n* $\to$ *'m* $\to$ α*)* $\to$ *bool* ***where***
rel-M A B = *(A = from-hma B)*

The second step of the connection is proving several transfer rules which we will explain by an example.

[5] In the sources, the relations *rel-I* and *rel-M* have the names *HMA-I* and *HMA-M* respectively.

lemma $(\textit{rel-M} \longrightarrow \textit{rel-M} \longrightarrow \textit{rel-M})\ +_{\textit{JNF}}\ +_{\textit{HMA}}$
lemma $(\textit{rel-M} \longrightarrow =)\ \textit{det}_{\textit{JNF}}\ \textit{det}_{\textit{HMA}}$

The first transfer rule states that if we invoke matrix addition on A_1 and A_2 in the JNF world (via $+_{\textit{JNF}}$), and we invoke matrix addition on B_1 and B_2 in the HMA world (via $+_{\textit{HMA}}$), then the resulting matrices will be related by *rel-M*, provided A_1 is related to B_1 and A_2 is related to B_2. Similarly, the second transfer rule states that if A_1 and B_1 are related by *rel-M* then the computed determinants in both worlds are related by the equality relation, i.e, they are equal.

Whereas it was quite easy to prove the transfer rules for matrix addition, multiplication, etc., the most difficult part was actually the transfer rule for determinants. Recall that the determinant of a matrix is defined as a sum ranging over all permutations of the indices, where each summand depends on the sign of the permutation. For the transfer rule for determinants we essentially have to prove the following property.

$$\sum_{p.\ p\ \textit{permutes}\ \{0..<\textit{CARD}('n)\}} \textit{signof}\ p \cdot \prod_{i<\textit{CARD}('n)} A\ (\textit{from-nat}\ i)\ (\textit{from-nat}\ (p\ i)) = \sum_{p.\ p\ \textit{permutes}\ \textit{UNIV}} \textit{signof}\ p \cdot \prod_{i\in\textit{UNIV}} A\ i\ (p\ i))$$

To this end, we convert the set of permutations via the bijections *from-nat* and *to-nat* and at the same time show that the signs do not change by this conversion.

After having installed the transfer rules we can easily transfer lemmas from the JNF world to the HMA world. For instance, we transfer properties on the characteristic polynomial in the HMA world which so far have only been available in the JNF world.

Since the transfer package is bidirectional, we can also transfer statements from HMA into JNF. For instance, Theorem 3.1 is transferred into the following theorem:

$$\begin{aligned}&A \in \textit{carrier-mat}\ \textit{CARD}('n)\ \textit{CARD}('n) \Longrightarrow\\ &\textit{real-nonneg-mat}\ A \Longrightarrow n \neq 0 \Longrightarrow\\ &\exists v.\ v \in \textit{carrier-vec}\ \textit{CARD}('n) \wedge\\ &\quad \textit{eigenvector}\ A\ v\ (\textit{of-real}\ (\textit{spectral-radius}\ A)) \wedge\\ &\quad \textit{real-nonneg-vec}\ v\end{aligned}$$

Here, we would like to replace the expression $\textit{CARD}('n)$ by a variable n to make the theorem applicable to arbitrary dimensions. To this end, we implement a small routine which automatically proves the following theorem from the aforementioned theorem by using the local type definitions [11].

$$\begin{aligned}&A \in \textit{carrier-mat}\ n\ n \Longrightarrow\\ &\textit{real-nonneg-mat}\ A \Longrightarrow n \neq 0 \Longrightarrow\\ &\exists v.\ v \in \textit{carrier-vec}\ n \wedge\\ &\quad \textit{eigenvector}\ A\ v\ (\textit{of-real}\ (\textit{spectral-radius}\ A)) \wedge\\ &\quad \textit{real-nonneg-vec}\ v\end{aligned}$$

The new theorem is more generic and only constraints the new variable n to be non-zero. This constraint is a consequence of the fact that types have to be non-empty; the routine internally creates a local type τ with n elements and then instantiates the previous statement where $'n$ will be τ.

It is worth noting that there is slightly different spelling of constants between Theorem 3.1 and the above statements, e.g. *eigen-vector* and *eigenvector*. This is caused by slightly different names in the HMA and the JNF worlds and this difference has an advantage: without it one would always have to prefix non-overloaded constants for disambiguation, which decreases readability.

5 Perron–Frobenius, Irreducible Matrices

In order to circumvent the limitation of Theorem 1.3 we continue to formalize the Perron–Frobenius theorem for irreducible matrices.

Theorem 5.1 (Perron–Frobenius, irreducible version)**.** *Let A be a non-negative real and irreducible matrix. Then*

1. *ρ_A is an eigenvalue with eigenvector $z > 0$.*
2. *The algebraic multiplicity of ρ_A is 1.*
3. *Every non-negative real eigenvector corresponds to eigenvalue ρ_A.*
4. *There is some $k > 0$ and polynomial f such that $\chi_A = f(x^k - \rho_A^k)$ and the norm of all complex roots of f is strictly less than ρ_A.*

In order to formalize Theorem 5.1, we closely follow a paper proof by Wielandt [20], though we will also see one deviation. As in the proof of Theorem 3.1, we mostly use HMA matrices, but at a certain point also JNF matrices.

Proof. We assume that A is a square matrix of dimension n. Since A is irreducible, $(A + I)^n > 0$. Similar to the proof of Theorem 3.1, we define a compact set: X_1.

$$X := \{x \in \mathbb{R}^n \mid x \geq 0, x \neq 0\} \quad X_1 := \{x \in X \mid \|x\| = 1\}$$

Next, we define a function r from X to real numbers

$$r(x) := \min_{j, x_j \neq 0} \frac{(Ax)_j}{x_j} = \max\{c \mid cx \leq Ax\}$$

with the property that $r(x)x \leq Ax$.

Note that r is neither continuous on X nor on X_1, since the selection of the indices j in the minimum-operation is not continuous. Therefore, we define

$$Y := \{(A + I)^n x \mid x \in X_1\}$$

and prove that r is continuous on Y, the reason being that $r(y) = \min_j \frac{(Ay)_j}{y_j}$ for all $y \in Y$. Hence, we apply the extreme value theorem on r and Y to obtain a maximum z such that $r(z) = \max\{r(y) \mid y \in Y\}$. At this point the formalization slightly differs from the paper proof, since the standard notion of maximum and Isabelle/HOL's notion of the maximum of a set are not the same: the latter only works for finite sets. Therefore, the formalization instead uses Hilbert's choice operator (*SOME* in Isabelle) and contains an explicit statement of membership and maximality.

definition $z = (SOME\ z.\ z \in Y \wedge (\forall y \in Y.\ r\ y \leq r\ z))$
lemma $z \in Y \wedge (y \in Y \longrightarrow r\ y \leq r\ z)$

We further prove that z is also maximal for X, and that z is a positive eigenvector with eigenvalue $r(z)$ by directly translating the paper proof. To be more precise, we show that for any $u \in X$ with $r(u) = r(z)$ it follows that u is an eigenvector with eigenvalue $r(z)$ and $u > 0$. Afterwards, we derive that $r(z)$ is actually ρ_A by proving that any complex eigenvalue λ satisfies $|\lambda| \leq r(z)$. So, at this point we completed part (1) of Theorem 5.1 which is a slightly stronger property than Theorem 3.1: for irreducible matrices we get a positive real eigenvector whereas we only had a non-negative real eigenvector before.

For proving that ρ_A has multiplicity 1, the formalization becomes more interesting. The paper proof works along the following line, where it is shown that the derivative of χ_A at point ρ_A is positive. Here, B_i is the matrix where row i and column i have been deleted from A.

$$\chi'_A(\rho_A) \overset{(*)}{=} \sum_i \chi_{B_i}(\rho_A) \overset{(**)}{>} 0$$

Equality $(*)$ is essentially the derivation rule for characteristic polynomials which says $\chi'_A = \sum_i \chi_{B_i}$ and which is hard to state in the HMA world because of the change of dimensions. Although this lemma has been formalized for JNF-matrices (Lemma 2.1), it is still not possible to convert it back to the HMA world, since there is no operation on HMA matrices which corresponds to *mat-delete*. Therefore, we provide another operation, which erases a specific row and column by overwriting the values by zero. This operation is easy to define in both the JNF- and the HMA-representation and also the proof of the transfer-rule between the constants *mat-erase* (JNF) and *erase-mat* (HMA) as in Section 4 is straightforward. In JNF we then show the following property of the derivative of the characteristic polynomial where *monom* 1 *n* is just one possible way to write the monomial x^n in Isabelle/HOL.

lemma $A \in carrier\text{-}mat\ n\ n \Longrightarrow monom\ 1\ 1 \cdot$
$pderiv\ (char\text{-}poly\ A) = \sum_{i<n} char\text{-}poly\ (mat\text{-}erase\ A\ i\ i)$

The advantage of this characterization of the derivative is to be convertible to HMA via transfer.

lemma $monom\ 1\ 1 \cdot pderiv\ (charpoly\ A) =$
$\sum_i charpoly\ (erase\text{-}mat\ A\ i\ i)$

We clearly see that the latter lemma lives in HMA; for instance, the range of the index i of the summation is implicit by the type and not explicitly bounded by n as before. Using the lemma it is no longer difficult to formalize step $(*)$ where we replace B_i by *erase-mat A i i*.

For proving $(**)$ the essential step is to show for all B that $A \geq B \geq 0$ together with $A \neq B$ implies $\rho_B < \rho_A$. Hence, ρ_A is larger than any root of χ_B and thus, $\chi_B(\rho_A) > 0$. For proving $\rho_B < \rho_A$ we do not follow the paper proof which considers an arbitrary complex eigenvector of B, but instead we apply the Perron–Frobenius Theorem 3.1 to directly restrict to a non-negative real eigenvector u of B for eigenvalue ρ_B, i.e., $u \in X$. Note that it is not possible to use the already proven part (1) of Theorem 5.1 at this point, since B is not necessarily irreducible. By the restriction $u \in X$ it becomes easy to derive $(**)$: $\rho_B \leq \rho_A$ follows from monotonicity via $\rho_B u = Bu \leq Au$ and $\rho_A = \max\{c \mid \exists x \in X.\ cx \leq Ax\}$. Moreover, $\rho_B = \rho_A$ would imply $(A - B)u = 0$ and $u > 0$, a contradiction to $A \neq B$.

At this point we have proved the first two parts of Theorem 5.1, and we skip the explanation of the remaining part as it is again quite close to the paper proof, e.g. by showing that A is similar to $\frac{\lambda}{|\lambda|}A$ for every maximal eigenvalue λ. □

After its proof, let us have a look at Theorem 5.1 from the complexity viewpoint. Here, in particular the last part is interesting. It implies that all maximal eigenvalues have algebraic multiplicity 1, and hence the Jordan blocks of these eigenvalues all have size 1. This permits us to easily determine the matrix growth in Example 3.4 without explicitly computing any eigenvalue.

Example 5.2. The matrix in Example 3.4 is irreducible and $\rho_A = 1$. By Theorem 5.1 the largest Jordan block of a maximal eigenvalue has size 1. Thus, $A^n \in O(1)$ by the soundness of Algorithm 1.

6 Perron–Frobenius, General Case

The Perron–Frobenius Theorem 5.1 implies that for irreducible matrices we can always (and only) derive either constant or exponential growth. Therefore, irreducible matrices are quite limited for complexity analysis.[6] For instance, Theorem 5.1 is not applicable on Example 1.2 since that matrix is not irreducible. So, we would like to generalize Theorem 5.1 to non-irreducible matrices. Since we are mainly interested in the last property of Theorem 5.1, we exactly obtain Theorem 1.4 of the introduction, whose informal proof is as follows.

Proof of Theorem 1.4. The proof is by induction on the dimension n and considers three cases. The irreducible case is handled by Theorem 5.1, and the property is trivial in case the dimension of A is 1. So we remain with the only interesting case that A is not irreducible and $n \geq 2$. Then there exists a permutation π of row and columns such that

$$\pi(A) = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}$$

where the dimensions of B and D are both smaller than n. Since π is a permutation we know that also B and D are non-negative real matrices. Hence, by the induction hypothesis

[6] This knowledge should be exploited when searching for suitable matrix interpretations in automatic complexity tools.

we conclude

$$\chi_B = f_B \prod_{k \in K_B} (x^k - \rho_B^k) \text{ and } \chi_D = f_D \prod_{k \in K_D} (x^k - \rho_D^k).$$

Since A and $\pi(A)$ are similar, they have the same characteristic polynomial. We conclude

$$\chi_A = \chi_{\pi(A)} = \chi_B \chi_D$$

and moreover $\rho_A = \max\{\rho_B, \rho_D\}$. Hence, for $\rho_B = \rho_D$ it suffices to choose $f = f_B f_D$ and $K = K_B \cup K_D$. If $\rho_B < \rho_D$ we just choose $f = \chi_B f_D$ and $K = K_D$ to finish the proof. Finally, the case $\rho_D < \rho_B$ is symmetric to the case $\rho_B < \rho_D$. □

In order to formalize the above informal proof, clearly we need JNF matrices to perform the decomposition of $\pi(A)$ into the four blocks B, C, 0, and D. Here, it turns out that we also have to formalize several results on permutations of matrix indices, e.g., that applying a permutation is a similarity transformation, and that a non-irreducible matrix can always be permuted to the form above, i.e., a block matrix where the lower-left block is 0. Especially the latter fact is quite tedious.

To be more precise, let G be the graph of A. Since A is not irreducible and $n \geq 2$, we get two indices i and j such that there is no path from i to j. Now let I be the set of indices (i.e., nodes of G) that are reachable from i. Next define π as a permutation which moves I to the front—in Isabelle, we define π as a permutation obtained by sorting the list of all indices w.r.t. a suitable order. Finally we prove that $\pi(A)$ has the desired property, since any non-zero value in the lower-left block of $\pi(A)$ would connect a node which is reachable from i to a node which is not reachable by i.

In total, we arrive at the formalized statement of Theorem 1.4 which is available for both HMA and JNF. We do not display a formal version of the theorem explicitly at this point, but instead present a corollary which is tailored for complexity analysis. Here, the matrix A has elements of type *real*, so the second assumption demands that there are no *real* eigenvalues above 1, whereas in the conclusion we know that all *complex* roots of f have a norm below 1.

Corollary 6.1. *non-neg-mat A* $\Longrightarrow$
$\forall x.$ *poly* (*charpoly A*) $x = 0 \longrightarrow x \leq 1 \Longrightarrow$
$\exists K\ f.$ *charpoly* $A = f \cdot \prod_{k \leftarrow K}$ (*monom* 1 $k - 1$) $\wedge$
$\forall x.$*poly* (*map-poly complex-of-real* f) $x = 0 \longrightarrow |x| < 1$

Based on this corollary, we can now prove why it suffices to consider the potential eigenvalues in Figure 2b. The figure states that for matrices of dimension n it suffices to compute the Jordan blocks of the roots of unity of degree at most $\lfloor \frac{n}{2} \rfloor$. This is just a simple counting argument: If there is any maximal eigenvalue λ with norm 1, then by the corollary it must be a root of unity of degree k where $k \in K$. Since Jordan blocks of size 1 can always be ignored in Algorithm 1, we may assume that k has a Jordan block of size 2 or above. But then also the multiplicity of λ must be at least 2, so a multiple of k must occur at least twice in K. However, then the degree of χ_A is at least $2k$, so $k \leq \lfloor \frac{n}{2} \rfloor$.

With this reasoning we can prove the validity of all numbers in Figure 2b except for the potential eigenvalue -1 which is labeled by 5. According to the above counting argument we would have to consider the Jordan blocks of -1 already for matrices of dimension 4. The answer to this difference is provided in the next section.

7 Largest Jordan Blocks

Note that Theorem 1.4 and Corollary 6.1 only provide us with information on the characteristic polynomial of A, but they do not provide insights on the structure of the Jordan blocks of A.

In contrast, the following lemma states that the Jordan blocks of the spectral radius are always the largest ones among all maximal eigenvalues. Hence, it suffices to just compute the Jordan blocks for eigenvalue 1 in Algorithm 1, and thus, Jordan blocks for -1 do not have to be computed for matrices of dimension 4.

Lemma 7.1. *Let A be a non-negative real matrix of dimension $n \leq 4$, and λ a maximal eigenvalue of A. If a Jordan block of A and λ is of size s, then there exists a Jordan block of A and ρ_A with size $t \geq s$.*

Proof. W.l.o.g. we assume that $\rho_A = 1$, as otherwise one can just multiply the matrix by the constant $1/\rho_A$. In the following we just provide a short informal argument and refer to the sources for the details of the formalization via JNF matrices.

By using the counting argument of Corollary 6.1, we see that the only possible violation of the claim is that -1 has a Jordan block of size 2, so in particular K must be the multiset $\{2, 2\}$ and hence $\chi_A = (x^2 - 1)^2$. By Theorem 5.1 we then know that A is not irreducible. Consequently we can obtain a permutation π such that $\pi(A) = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}$. Moreover, the theorem tells us that both B and D must have the characteristic polynomial $x^2 - 1$. Since both B and D are non-negative this is only possible if

$$\pi(A) = \begin{bmatrix} 0 & a & c & d \\ \frac{1}{a} & 0 & e & f \\ 0 & 0 & 0 & b \\ 0 & 0 & \frac{1}{b} & 0 \end{bmatrix}$$

for some $a > 0$ and $b > 0$.

We derive that $\pi(A)$ is similar to E via the invertible matrix P where $g = \frac{-abe+af+bc-d}{2b}$ and $h = \frac{abe+af+bc+d}{2a}$.

$$E = \begin{bmatrix} -1 & g & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} \frac{1}{2} & \frac{-a}{2} & \frac{abe+af-bc-d}{8b} & \frac{abe+af-bc-d}{8} \\ 0 & 0 & \frac{1}{2} & \frac{-b}{2} \\ \frac{1}{a} & 1 & \frac{abe-af+bc-d}{2ab} & 0 \\ 0 & 0 & \frac{1}{b} & 1 \end{bmatrix}$$

Actually, we used Mathematica to obtain g, h, E, and P and then manually copied these definitions into our formalization. Thus A is similar to E, too, and so their Jordan blocks must be identical. So, since A has a Jordan block for -1 of size 2, g must be non-zero. But then also h must be non-zero by the definition of g and h. Thus, there also is a Jordan block for eigenvalue 1 with size 2. □

Currently Lemma 7.1 states the maximality result only for matrices up to dimension 4. We conjecture that it is also true for arbitrary n: among billions of generated matrices we did not find any violation. However, we do not see how to generalize the proof of Lemma 7.1.

8 Improved Certification Algorithm

In order to actually certify the growth rate of the power of non-negative real matrices via Corollary 6.1 and Lemma 7.1, there is still one minor problem, namely the existentially quantified K and f in the corollary have to be computed. To this end, we first prove the soundness of Algorithm 2. It computes K and f and thereby proves that these values are uniquely determined.

Algorithm 2: Computing K and f of Corollary 6.1.

Input: A polynomial $g = f \prod_{k \in K}(x^k - 1)$ where f has no complex roots with norm 1
Output: K and f

```
f := g, K := ∅
k := degree f
while k ≥ 1 do
    if x^k − 1 divides f then
        K := {k} ∪ K, f := f/(x^k − 1)
    else
        k := k − 1
return K and f
```

It is important that the loop in Algorithm 2 goes down from the degree of f to 1. If one would reverse the iteration order, then the algorithm would deliver wrong results: for instance consider $g = x^2 - 1$ with the correct answer $f = 1$ and $K = \{2\}$, but where an iteration with ascending k would result in $f = x + 1$ and $K = \{1\}$.

We are now ready to present the improved certification algorithm for matrix growth.

Algorithm 3: Efficient Certification of $A^n \in O(n^d)$.

Input: A non-negative real matrix A and degree d.
Output: Accept or assertion failure.

```
Assert {x ∈ ℝ. χ_A(x) = 0, x > 1} = ∅ via Sturm's method
Compute K by decomposing χ_A via Algorithm 2
if |K| ≤ d + 1 then accept
Check the Jordan blocks for eigenvalue 1, i.e., assert that
  each Jordan block of A and 1 has size s ≤ d + 1
if dimension of A ≤ 4 then accept
for k ∈ {2, ..., max K} do
    m_k := |{k′ ∈ K. k divides k′}|
    if m_k > d + 1 then
        Check the Jordan blocks for all primitive roots
          of unity of degree k
Accept
```

Algorithm 3 is even more fine-grained than considering all points in Figure 2b, since it precisely determines the set of maximal eigenvalues whose multiplicities may violate the given complexity bound, without explicitly computing them. The value m_k in the algorithm is precisely the algebraic multiplicity of the primitive roots of unity of degree k, and in particular $m_1 = |K|$ is the algebraic multiplicity of 1.

In order to produce the primitive roots of unity of degree k, we apply explicit formulas for $k \leq 4$: $\{1\}$, $\{-1\}$, $\{\frac{-1 \pm \sqrt{3}\mathrm{i}}{2}\}$, and $\{\pm\mathrm{i}\}$ for $k = 1, 2, 3$, and 4, respectively, and otherwise we just invoke a generic complex-root algorithm on $x^k - 1$ which will generate all roots of unity of degree k, even non-primitive ones.

We formalize the soundness of Algorithm 3 by combining the soundness of Algorithm 1 with Corollary 6.1, Lemma 7.1, and the soundness of Algorithm 2.

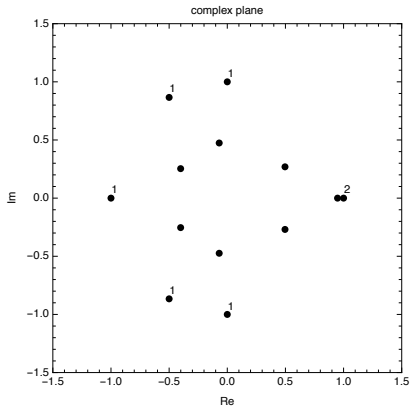
Let us illustrate the improvement of Algorithm 3 over Algorithm 1 and also over Corollary 6.1 in an example.

Example 8.1. Consider some non-negative real matrix A with

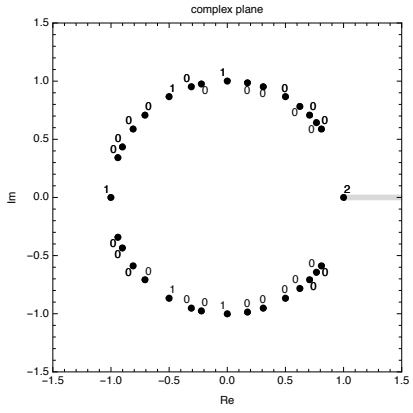
$$\chi_A = \frac{1}{4096}\big(4096x^{21} - 8192x^{20} + 4096x^{19} - 4096x^{18} + 4608x^{17} + 3584x^{16} - 4096x^{15} + 3456x^{14} - 8048x^{13} + 4608x^{12} + 128x^{11} + 488x^{10} - 656x^9 - 119x^7 + 152x^6 - x^4 - 9x^3 + 1\big)$$

where we are interested in checking whether A^n has constant growth, i.e., $d = 0$.

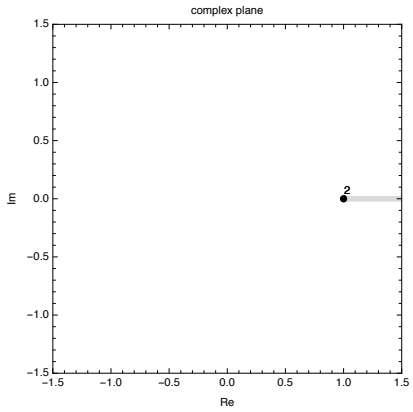
We tested three different algorithms to conduct the following task: check that $\rho_A \leq 1$ and compute all critical



(a) Algorithm 1



(b) Corollary 6.1



(c) Algorithm 3

Figure 3. Different ways to compute critical eigenvalues

eigenvalues λ, i.e., eigenvalues λ with norm 1 which have an algebraic multiplicity of 2 or more, so that a Jordan block computation for λ is required. The execution of the algorithms is illustrated in Figure 3 where each point indicates an explicitly computed potential eigenvalue, and each number indicates a calculated algebraic multiplicity.

(a) Algorithm 1 first explicitly computes all eigenvalues, i.e., the complex roots of χ_A, as shown in Figure 3a. Afterwards it determines their norms, and finally computes the algebraic multiplicity of each maximal eigenvalue. This approach requires expensive algebraic number computations, e.g., the imaginary part of one of the eigenvalues is root #5 of a degree 42 polynomial whose leading coefficient is 75557863725914323419136. We had to abort this computation after one hour.[7] Note that preprocessing the characteristic polynomial by a square-free factorization does not help in this example: the factorization splits χ_A into $\frac{f^2 g}{4096}$ where the roots of $f = 64x^8 - 128x^7 + 64x^6 + 4x^4 - 4x^3 - x + 1$ are precisely the eigenvalues with multiplicity 2 and the roots of g are the eigenvalues with multiplicity 1. Still determining the norms of the complex roots of f (instead of χ_A) took more than one hour.

(b) The next approach first applies Sturm's method to detect $\rho_A = 1$, indicated by the gray line in Figure 3b. Then using Corollary 6.1 we know that the critical eigenvalues can only be roots of unity up to degree 10. For all of these numbers the algebraic multiplicities are calculated and it is then determined that 1 is the only critical eigenvalue. The overall execution took 10.33 seconds.

(c) Finally we invoke Algorithm 3. It first applies Sturm's method and then computes $K = \{3, 4\}$. Next, it figures out that only for $k = 1$ there are critical eigenvalues: $m_k \leq 1 = d + 1$ for $k = 2, 3, 4$. Finally, it returns as critical eigenvalues all roots of unity of degree $k = 1$, i.e., 1. Hence, only one eigenvalue is explicitly computed, cf. Figure 3c. The overall computation took 0.05 seconds.

Example 8.1 uses an artificial large matrix where tremendous improvement in speed is observed. To measure improvements in practice, we extracted all matrix interpretations from complexity proofs of the international termination and complexity competition [6] in the last three years, which amounts to the validation of the growth rate of 6,690 matrices, whose largest dimension was only 5. This low dimension keeps the overhead of algebraic number computations at a reasonable level. Still, processing all 6,690 matrices became five times faster after replacing Algorithm 1 by Algorithm 3.

Finally, we remark that the integration of Algorithm 3 into IsaFoR—the formalization underlying CeTA—was unfortunately not straightforward. The reason is that initially we based our definition of the graph of a matrix on the AFP entry on graphs by Noschinski [14]. However, IsaFoR already depends on an AFP entry on computing strongly-connected components of a graph by Lammich [12]. Since both of these AFP entries define their own versions of graphs in different

[7] All experiments have been conducted on a computer running at 3.5 GHz using compiled Haskell code. This code was generated from the Isabelle sources using Isabelle's code generator [7].

theory files using the *same theory name*, we could not include both AFP entries into IsaFoR in Isabelle 2017. Our solution was to completely rewrite the graph part of our formalization so that it no longer depends on the the AFP entry by Noschinski. Clearly, it would have been more convenient if there were support on resolving theory-name clashes, e.g., by some kind of package or module system.

9 Conclusion

We developed an efficient algorithm which decides $A^n \in O(n^d)$ for non-negative real matrices. Its soundness has been formalized in Isabelle/HOL, and it is heavily based on our formalization of the Perron–Frobenius theorem. A key technical part of the formalization is our connection between matrices in JNF and HMA representations: it permits to arbitrarily switch between both representations.

Since for matrices of dimensions up to 5 no algebraic number computations are required, it also seems to be possible to use our algorithm for synthesis of matrix interpretation: one can write a polynomial-sized SAT or SMT encoding whether a symbolic matrix of dimension up to 5 has an a-priori fixed growth rate by just encoding the computations that are performed in Algorithm 3.

Although our formalization was motivated by the certification of complexity proofs, there are also other applications where it may become useful. For instance, Theorem 5.1 implies that there is a unique eigenspace that contains a non-negative real vector, and moreover this space is 1-dimensional. This property is connected to invariant distributions of stochastic matrices and to convergence of finite irreducible Markov chains. Hence, it will be interesting to connect our work with the recent formalization of Markov chains by Hölzl [9].

Acknowledgments

This research was supported by the Austrian Science Fund (FWF) project Y757. Jose Divasón is partially funded by the Spanish project MTM2014-54151-P. Most of the research was conducted while Sebastiaan Joosten and Akihisa Yamada were working in the University of Innsbruck. The authors are listed in alphabetical order regardless of individual contributions or seniority.

We thank Fabian Immler for his explanations on how to perform continuity proofs in the HMA library.

References

[1] Martin Avanzini, Georg Moser, and Michael Schaper. 2016. TcT: Tyrolean Complexity Tool. In *TACAS 2016 (LNCS)*, Vol. 9636. 407–423.

[2] Manuel Eberl. 2015. A Decision Procedure for Univariate Real Polynomials in Isabelle/HOL. In *CPP 2015*. ACM, 75–83.

[3] Jörg Endrullis, Johannes Waldmann, and Hans Zantema. 2008. Matrix Interpretations for Proving Termination of Term Rewriting. *Journal of Automated Reasoning* 40, 2-3 (2008), 195–220.

[4] Ferdinand Georg Frobenius. 1912. Über Matrizen aus nicht negativen Elementen. In *Sitzungsberichte Preuß. Akad. Wiss.* 456–477.

[5] Jürgen Giesl, Cornelius Aschermann, Marc Brockschmidt, Fabian Emmes, Florian Frohn, Carsten Fuhs, Jera Hensel, Carsten Otto, Martin Plücker, Peter Schneider-Kamp, Thomas Ströder, Stephanie Swiderski, and René Thiemann. 2017. Analyzing Program Termination and Complexity Automatically with AProVE. *Journal of Automated Reasoning* 58, 1 (2017), 3–31. https://doi.org/10.1007/s10817-016-9388-y

[6] Jürgen Giesl, Frédéric Mesnard, Albert Rubio, René Thiemann, and Johannes Waldmann. 2015. Termination Competition (termCOMP 2015). In *CADE-25 (LNCS)*, Vol. 9195. 105–108.

[7] Florian Haftmann and Tobias Nipkow. 2010. Code Generation via Higher-Order Rewrite Systems. In *FLOPS 2010 (LNCS)*, Vol. 6009. 103–117.

[8] John Harrison. 2013. The HOL Light Theory of Euclidean Space. *J. Autom. Reasoning* 50, 2 (2013), 173–190.

[9] Johannes Hölzl. 2017. Markov chains and Markov decision processes in Isabelle/HOL. *Journal of Automated Reasoning* (2017). To appear.

[10] Brian Huffman and Ondřej Kunčar. 2013. Lifting and Transfer: A Modular Design for Quotients in Isabelle/HOL. In *CPP 2013 (LNCS)*, Vol. 8307. 131–146.

[11] Ondřej Kunčar and Andrei Popescu. 2016. From Types to Sets by Local Type Definitions in Higher-Order Logic. In *ITP 2016 (LNCS)*, Vol. 9807. 200–218.

[12] Peter Lammich. 2014. Verified Efficient Implementation of Gabow's Strongly Connected Components Algorithm. *Archive of Formal Proofs* (May 2014). http://isa-afp.org/entries/Gabow_SCC.html, Formal proof development.

[13] Tobias Nipkow, Lawrence C. Paulson, and Makarius Wenzel. 2002. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic*. LNCS, Vol. 2283. Springer.

[14] Lars Noschinski. 2013. Graph Theory. *Archive of Formal Proofs* (April 2013). http://isa-afp.org/entries/Graph_Theory.html, Formal proof development.

[15] Oskar Perron. 1907. Zur Theorie der Matrices. *Math. Ann.* 64 (1907), 248–263.

[16] Denis Serre. 2002. *Matrices: Theory and Applications*. Springer.

[17] René Thiemann and Christian Sternagel. 2009. Certification of Termination Proofs using CeTA. In *TPHOLs'09 (LNCS)*, Vol. 5674. 452–468.

[18] René Thiemann and Akihisa Yamada. 2016. Algebraic Numbers in Isabelle/HOL. In *ITP 2016 (LNCS)*, Vol. 9807. 391–408.

[19] René Thiemann and Akihisa Yamada. 2016. Formalizing Jordan normal forms in Isabelle/HOL. In *CPP 2016*. ACM, 88–99.

[20] Helmut Wielandt. 1950. Unzerlegbare, nicht negative Matrizen. *Mathematische Zeitschrift* 52, 1 (1950), 642–648.

[21] Harald Zankl and Martin Korp. 2014. Modular Complexity Analysis for Term Rewriting. *Logical Methods in Computer Science* 10, 1:19 (2014), 1–34.