Certifying Exact Complexity Bounds for Matrix Interpretations*

Jose Divasón,[1] Sebastiaan Joosten,[2] Ondřej Kunčar,[3] René Thiemann,[2] and Akihisa Yamada[2]

[1] Universidad de La Rioja
[2] University of Innsbruck
[3] Technical University of Munich

1 Introduction

CeTA [9] is a *certifier* for complexity proofs of term rewrite systems; i.e., it takes an untrusted, automatically generated proof from complexity analyzers such as AProVE, CaT, or TCT [5,12,1], and tries to validate it. There are three possible outcomes: (1) the proof could be validated, (2) the proof was rejected because it indeed was faulty, and (3) the proof was rejected because at some point in the analysis CeTA applied too coarse estimations or imposed too severe preconditions, so that the desired complexity bound could not be validated.

This work aims at reducing the number of rejected proofs of the kind (3), by improving the support for *matrix interpretations* [4], an important technique for complexity analysis. For instance, in *the Termination Competition 2015* [6], roughly 40 % of the machine readable complexity proofs contain matrix interpretations.

To certify a complexity proof via matrix interpretations, the main task is to estimate the growth rate of the values in the k-th power of a matrix A, for fixed A and increasing k. For instance, consider the matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 1 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Old versions of CeTA did not accept this matrix within complexity proofs as they were restricted to upper triangular matrices. Due to our development, the new version of CeTA admits arbitrary matrices and gives precise complexity bounds, as we will see in the next sections.

2 Spectral radius and Perron-Frobenius theorem

We denote the characteristic polynomial of a matrix A by $\chi_A(\lambda)$. The *spectral radius* $\rho(A)$ of A is defined as $\max\{\, |\lambda| \mid \lambda \in \mathbb{C}, \chi_A(\lambda) = 0 \,\}$.

* This research was partially supported by the Austrian Science Fund (FWF) project Y757. The authors are listed in alphabetical order regardless of individual contributions or seniority.

In an earlier work [11], we formalized the theory of *Jordan normal forms (JNFs)* [7] in Isabelle/HOL. With the help of JNFs we obtained the following theorem about the spectral radius.

Bellow we denote the multiplicity of a root λ of a polynomial f by $\mathsf{m}(f, \lambda)$, and the growth rate of A^k denotes the growth rate of the largest values in A^k.

Theorem 1 (Complexity via spectral radius and multiplicity).

1. *If* $\rho(A) > 1$*, then* A^k *grows exponentially.*
2. *If* $\rho(A) \leq 1$*, then* $A^k \in \mathcal{O}(k^{n-1})$ *for* $n = \max\{\mathsf{m}(\chi_A, \lambda) \mid \chi_A(\lambda) = 0, |\lambda| = 1\}$.

One problem in using the theorem for certifying complexity proofs is that it is expensive to compute all complex roots of χ_A, and also to compute the largest norm among them.

To avoid the expensive computations, in recent work [2] we formalized the Perron-Frobenius theorem in Isabelle/HOL. Here the recent addition of local type definitions [8] was essential.

Theorem 2 (Perron-Frobenius). *If* A *is a non-negative real matrix, then* $\chi_A(\rho(A)) = 0$*; i.e.,* $\rho(A)$ *is a real root of* χ_A*.*

This means that $\rho(A) \leq 1$ iff there is no real root of χ_A in the interval $(1, \infty)$. The latter property can be easily checked using Sturm's method, whose formalization was already provided by Eberl [3].

Another problem is to compute the degree in Theorem 1, step 2. To quickly estimate it we also formalized Yun's square-free factorization algorithm [11].

Theorem 3 (Yun factorization). *Let* $(c, [f_1, \ldots, f_n])$ *be the result of Yun's algorithm applied on polynomial* f*. Then* $f = c \cdot f_1^1 \cdot \ldots \cdot f_n^n$ *and each root* λ *of* f_i *satisfies* $\mathsf{m}(f, \lambda) = i$*.*

Now we can turn Theorem 1 into an executable algorithm as follows.

Algorithm 1: Estimating complexity via spectral radius

Input: A non-negative real matrix A.
Output: A degree d such that $A^k \in \mathcal{O}(k^d)$.
1 Ensure $\rho(A) \leq 1$ with the help of Sturm's method.
2 Return $d = n - 1$ if Yun's algorithm applied on χ_A delivers $(c, [f_1, \ldots, f_n])$.

In the running example, we determine the square-free factorization of χ_A as

$$\chi_A(x) = \frac{1}{2} - \frac{5}{2}x + \frac{9}{2}x^2 - \frac{7}{2}x^3 + x^4 = \frac{1}{2} \cdot (2x - 1)^1 \cdot (x - 1)^3$$

Hence Algorithm 1 concludes $A^k \in \mathcal{O}(k^2)$.

This bound is, however, not tight. Whereas the first line of Algorithm 1 is precise, the second part introduces approximation: it ignores the condition of Theorem 1 that one only has to inspect the multiplicities of roots λ that satisfy $|\lambda| = 1$. Moreover, just taking the multiplicity in Theorem 1 is already imprecise.

3 Jordan normal forms

If one can determine all roots of χ_A, then one can compute the JNF of A, i.e., an invertible matrix P and a JNF J such that $A = PJP^{-1}$. In the running example, the roots are 1 and $\frac{1}{2}$, and the JNF is

$$J = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

which is concisely represented by the list of Jordan blocks: $(2, 1)$, $(1, 1)$, $(1, 0.5)$ indicates a block of size 2 with diagonal entry 1, a block of size 1 with diagonal entry 1, and a block of size 1 with diagonal entry 0.5.

A closed formula is known for k-th powers of JNFs; in this case,

$$J^k = \begin{bmatrix} 1^k & \binom{k}{1}1^{k-1} & 0 & 0 \\ 0 & 1^k & 0 & 0 \\ 0 & 0 & 1^k & 0 \\ 0 & 0 & 0 & 0.5^k \end{bmatrix} = \begin{bmatrix} 1 & k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.5^k \end{bmatrix}$$

whose elements are clearly bounded by k. Since $A^k = PJ^kP^{-1}$, we can derive the tight bound $A^k \in \mathcal{O}(k)$.

Analysing the Jordan blocks of a matrix gives a precise complexity bound.

Theorem 4 (Complexity via JNFs). *Let A have JNF J, represented by blocks $(n_1, \lambda_1), \ldots, (n_l, \lambda_l)$.*

- *If $|\lambda_i| > 1$ for some i, then A^k grows exponentially.*
- *If $|\lambda_i| \leq 1$ for all i, then $A^k \in \Theta(k^{n-1})$ where $n = \max\{\, n_i \mid |\lambda_i| = 1 \,\}$.*

To formalize the theory of JNF in Isabelle/HOL, we developed a new library for matrices that allows us to conveniently work with block matrices. We then formalized the Gram-Schmidt orthogonalization algorithm and the Schur decomposition to turn matrices into upper-triangular form, and formalized an algorithm that turns upper-triangular matrices into JNFs.

4 Algebraic numbers

For a rational (or even integer) matrix A, the roots of χ_A are in general *algebraic* (complex numbers represented as roots of rational polynomials). At the time of development, there was no executable formalization of algebraic numbers in Isabelle; hence we recently developed such a library [10].

Example 5. Consider a matrix A with $\chi_A = \frac{1}{3} + \frac{2}{3}x + x^4$. Using our development [10], we can exactly compute the complex roots of χ_A; the complex roots are expressed via the real roots of $f = -1 - 12x^2 + 144x^6$ and $g = 7 - 216x^2 - 336x^4 - 1248x^6 + 1152x^8 + 6912x^{12}$ as follows:

$\lambda_1 = \textit{root \#1 of } f + (\textit{root \#2 of } g)\mathrm{i} \quad \lambda_2 = \textit{root \#1 of } f + (\textit{root \#3 of } g)\mathrm{i}$
$\lambda_3 = \textit{root \#2 of } f + (\textit{root \#1 of } g)\mathrm{i} \quad \lambda_4 = \textit{root \#2 of } f + (\textit{root \#4 of } g)\mathrm{i}$

Here, real roots are indexed according to the standard order. We can also compute all $|\lambda_i|$, which are precisely root #3 and #4 of the polynomial $h = 1 - 3x^4 - 12x^6 - 9x^8 + 27x^{12}$. As they are strictly less than 1 we can conclude that A^k tends to 0 for increasing k.

5 Conclusion

We developed correct algorithms which can decide whether $A^k \in \mathcal{O}(k^n)$ and mechanized their soundness proofs in Isabelle/HOL. These are integrated in our certifier CeTA: it first tries the efficient Algorithm 1, and if this does not suffice to ensure the desired bound, then it invokes the decision procedure of Theorem 4 in combination with algebraic numbers. In this way, we succeeded in certifying several automatically generated complexity proofs which were not previously certified, and sometimes even improved the bound that complexity analyzers calculated.

References

1. Avanzini, M., Moser, G.: Tyrolean Complexity Tool: Features and usage. In: RTA 2013. LIPIcs, vol. 21, pp. 71–80 (2013)
2. Divasón, J., Kunčar, O., Thiemann, R., Yamada, A.: Perron-Frobenius theorem for spectral radius analysis. Archive of Formal Proofs (May 2016), `http://isa-afp.org/entries/Perron_Frobenius.shtml`, Formal proof development
3. Eberl, M.: A decision procedure for univariate real polynomials in Isabelle/HOL. In: CPP 2015. pp. 75–83. ACM (2015)
4. Endrullis, J., Waldmann, J., Zantema, H.: Matrix interpretations for proving termination of term rewriting. Journal of Automated Reasoning 40(2-3), 195–220 (2008)
5. Giesl, J., Brockschmidt, M., Emmes, F., Frohn, F., Fuhs, C., Otto, C., Plücker, M., Schneider-Kamp, P., Ströder, T., Swiderski, S., Thiemann, R.: Proving termination of programs automatically with AProVE. In: IJCAR 2014. LNCS, vol. 8562, pp. 184–191 (2014)
6. Giesl, J., Mesnard, F., Rubio, A., Thiemann, R., Waldmann, J.: Termination competition (termCOMP 2015). In: CADE-25. LNCS, vol. 9195, pp. 105–108 (2015)
7. Jordan, C.: Trait des substitutions et des quations algbriques. Gauthier-Villars (1870)
8. Kunčar, O., Popescu, A.: From types to sets by local type definitions in higher-order logic. In: ITP 2016. LNCS, (to appear)
9. Thiemann, R., Sternagel, C.: Certification of termination proofs using CeTA. In: TPHOLs'09. LNCS, vol. 5674, pp. 452–468 (2009)
10. Thiemann, R., Yamada, A.: Algebraic numbers in Isabelle/HOL. In: ITP 2016. LNCS, (to appear)
11. Thiemann, R., Yamada, A.: Formalizing Jordan normal forms in Isabelle/HOL. In: CPP 2016. pp. 88–99. ACM (2016)
12. Zankl, H., Korp, M.: Modular complexity analysis for term rewriting. Logical Methods in Computer Science 10(1:19), 1–34 (2014)