# Computable analysis, exact real arithmetic and analytic functions in Coq*

Florian Steinberg[1] and Holger Thies[2]

[1]INRIA, Saclay [2]Kyushu University, Fukuoka

Computable analysis is the application of computability and complexity theory to problems on real numbers and other uncountable spaces. Some of the basic aspects of computable analysis have recently been formalized in the Incone library. In this library, a represented space $\mathbf{X}$ is a record consisting of a type $X$ of abstract objects to compute over, countable inhabited types $\mathbf{Q_X}$ and $\mathbf{A_X}$ of *questions* and *answers*, and a partial surjective function $\delta_{\mathbf{X}} \colon (\mathbf{Q_X} \to \mathbf{A_X}) \to X$ called *representation*. Each $\varphi \colon \mathbf{Q_X} \to \mathbf{A_X}$ with $\delta_{\mathbf{X}}(\varphi) = x$ is called a *name* for $x$. Topological and computability theoretical properties of functions $f :\subseteq \mathbf{X} \to \mathbf{Y}$ can be reduced to computing on names by means of *realizers*, functions $F : (\mathbf{Q_X} \to \mathbf{A_X}) \to (\mathbf{Q_Y} \to \mathbf{A_Y})$ mapping names for elements $x \in \mathbf{X}$ to names for $f(x) \in \mathbf{Y}$. Realizers can often be defined as functions in Coq without resorting to any axioms, thus making them executable inside of Coq and accessible to code extraction. Consequently, a definition of such a realizer is considered as evidence that $f$ is computable. While the realizer should not use any non-computational components in its definition, classical reasoning and controlled use of choice axioms can be used for proofs of correctness and other properties. In particular the abstract data types are captured through classical descriptions along the lines of the axiomatization of the reals in Coq's standard library.

Using Incone, a representation for real numbers via rational approximations and realizers for arithmetic operations and a limiting procedure can be defined by using the types for real and rational numbers from Coq's standard library. Here, the questions are rational accuracy requirements and answers are rational approximations. A more detailed description of the library and some applications can be found in our recent work [STT19]. While the former mainly deals with more theoretical aspects of computable analysis, here we consider applications to actual computations with real numbers and functions and (maybe more interestingly) operators such as integration or differentiation mapping real functions to real functions. Incone provides a *function space construction* that takes represented spaces $\mathbf{X}$ and $\mathbf{Y}$ and returns the represented space $\mathbf{X}^{\mathbf{Y}}$ of continuous functions from $\mathbf{Y}$ to $\mathbf{X}$. This construction can be used to make the source and target spaces of such operators, i.e. $\mathbb{R}^{\mathbb{R}}$, accessible to computation. However, it is well known that in this general setting many important operators such as maximization or differentiation become infeasible or even uncomputable. Thus, real number complexity theory is usually concerned with similar operations on smaller spaces of functions whose representations provide additional information.

An example for such a space is $C^\omega(\overline{B_1(0)})$, the space of analytic functions that have no singularities in the closed complex unit ball and return real values on real arguments. Any $f \in C^\omega(\overline{B_1(0)})$ has a power series expansion $a : \mathbb{N} \to \mathbb{R}$ such that $f(x) = \sum_{i=0}^{\infty} a_n x^n$ for all $x \in [-1, 1]$. On one hand, many operators such as differentiation or arithmetic operations correspond to rather simple manipulations on the power series coefficients. On the other hand, the power series alone is not sufficient for evaluation of the infinite sum as no tail-estimate is

*This work was supported by JSPS KAKENHI Grant Number JP18J10407, by the Japan Society for the Promotion of Science (JSPS), Core-to-Core Program (A. Advanced Research Networks), by the ANR project FastRelax (ANR-14-CE25-0018-01) of the French National Agency for Research and by EU-MSCA-RISE project 731143 Computing with Infinite Data (CID).

available. To mend this issue, one adds to the power series integer constants $A, k \in \mathbb{N}$ called a *series bound* that are such that $\forall n \in \mathbb{N}, |a_n| \leq A\left(1+\frac{1}{k}\right)^{-n}$. That such constants always exist follows from basic theorems about analytic functions. The exact form of the bounds is chosen such that the effort for many basic operations scales polynomially in the size of the series bound [KMRZ15]. We represent the space of power series with radius of convergence bigger than one as follows: a name of the power series is a name as series of real numbers (using Incone's infinite product construction) together with a series bound, i.e. integers $A$ and $k$ as above. A realizer, e.g. for differentiation thus has to produce from a power series and a series bound for $f$ not only a power series for $f'$ but also an appropriate series bound. We implemented the evaluation procedure that witnesses that the above space of power series is canonically isomorphic to a space of functions. We also defined realizers for standard operations (e.g. arithmetic, derivatives and anti-derivatives) and elementary functions like the exponential and trigonometric functions. All realizers come with full formal proofs of their correctness. Our proofs rely on results about real numbers from the standard library and make heavy use of the treatment of power series in the Coquelicot library [BLM15] and also use the Coq-Interval library [Mel08] to prove inequalities over real numbers.

In spite of that the definitions of representations and realizers have been taken from results in real number complexity theory where they are used to show polynomial-time computability, our implementations are not very competitive in terms of efficiency. One reason for this is that computing with rationals is not feasible. An equivalent but more practical way to encode real numbers uses sequences of intervals with multi-precision floating point numbers as endpoints, an approach similar to that used in many software packages for exact real arithmetic. Further improvements can be made by using more sophisticated ideas from verified numerics such as affine arithmetic or Taylor models. Similar methods have already been formalized in Coq, for instance in the Coq-Interval library. The notion of correctness used in these libraries is weaker than correctness in the sense of computable analysis as diameters of return intervals are not required to approach zero as the precision goes to infinity. For a limited number of simpler operations, for instance arithmetic operations on the reals, bounds are easy to come by and we started proving them so that interval operations can be used to obtain algorithms in the sense of computable analysis. For more elaborate problems like evaluation of basic analytic functions, we hope to be able to modify our algorithms to produce a fallback source of approximations used only if enclosures gained from the Coq-Interval library are not tight enough. The idea is to use the series bound to obtain a rough upper bound of the expected approximation quality of the fallback algorithm and check this bound against the diameter of the return value in the Coq-Interval library. As the estimations are fairly simple computations and it is not expected that the fallback algorithm is executed often, we hope to achieve an acceptable overhead over the operations from the interval library together with full correctness proofs of the merged algorithms in the sense of computable analysis.

# References


[BLM15] Sylvie Boldo, Catherine Lelay, and Guillaume Melquiond. Coquelicot: A user-friendly library of real analysis for Coq. *Mathematics in Computer Science*, 9(1):41–62, 2015.

[KMRZ15] Akitoshi Kawamura, Norbert Th. Müller, Carsten Rösnick, and Martin Ziegler. Computational Benefit of Smoothness. *Journal of Complexity*, 2015.

[Mel08] Guillaume Melquiond. Proving bounds on real-valued functions with computations. In *International Joint Conference on Automated Reasoning*, pages 2–17. Springer, 2008.

[STT19] Florian Steinberg, Laurent Thery, and Holger Thies. Quantitative continuity and computable analysis in Coq. HAL preprint, short version accepted for presentation at ITP 2019, April 2019.