Computable analytic functions in `iRRAM`

It is well known that it is impossible to computably evaluate power series uniformly, even if the radius of convergence is known (e.g. [Mül95]). This can be mended by enriching a name of a real sequence with suitable discrete advice. More precisely, in [KMRZ12] it is shown that a given power series $(a_n)_{n\in\mathbb{N}}$ can be evaluated on the closed unit disc $\mathbb{D}$ in polynomial time if, in addition to the numbers a_n, two integers k and A fulfilling

$$|a_n| \leq \frac{A}{r^n}, \quad \text{where} \quad r = \sqrt[k]{2} \tag{1}$$

are specified. Here, r has to be strictly smaller than the radius of convergence.

Thus, we may also regard triples $((a_n)_{n\in\mathbb{N}}, k, A)$ satisfying equation (1) as names for the corresponding sum functions. Those are exactly the functions which can be extended to an analytic function on an open superset of $\mathbb{D}$. This representation is very well behaved with respect to many operators which are known to be hard when operating more generally on smooth functions. For example, parametric maximization resp. antiderivation on $\mathcal{C}^\infty([0,1])$ are known to preserve polytime computability if and only if $\mathcal{P} = \mathcal{NP}$ resp. $\mathcal{P} = \#\mathcal{P}$ (cf. [Ko91]). In [KMRZ12] it is shown that these operators are uniformly polytime computable with respect to representation of analytic functions indicated above.

`iRRAM` is a `C++` library for arbitrary precision real arithmetic and analysis allowing to compute limits of sequences of real numbers and functions (cf. [Mü01]). The implementation of real numbers in `iRRAM` is closely connected to a refinement $\rho_{\mathbb{D}}$ of the standard representation ρ of real numbers (e.g. [Wei00]). Functions from the natural numbers to the reals may then be identified with the product representation $\rho_{\mathbb{D}}^\omega$. The above representation for analytic functions on the closed unit disc canonically yields the declaration of a `C++` class interface.

We describe an efficient implementation of such a class and implementations of the standard arithmetic operations thereon as well as derivation, antiderivation and evaluation operators. While the transfer of the algorithms specified for these operators in [KMRZ12] is mostly straight forward, difficulties arise elsewhere: Evaluation should be as fast as possible. This is not the case if the predefined limit operator is used. Also it should be possible to specify alternate algorithms for evaluation; In particular, the fast implementations of some special functions in `iRRAM` should be accessible. Moreover, it might be useful to cache computed values in some situations. We discuss how these problems can be solved appropriately.

One of the main advantages of `iRRAM` over other attempts to implement arbitrary precision real arithmetic is that it avoids trees of pointers which arise when modeling real numbers as algorithms. The approach to analytic functions presented here does not avoid such trees. One can argue that these problems are less severe for functions since arithmetic operations of real numbers are used more excessively than those of functions. However, similar problems are encountered when trying to implement a composition operator.

References

[KMRZ12] Akitoshi Kawamura, Norbert Th. Müller, Carsten Rösnick, and Martin Ziegler, *Parameterized uniform complexity in numerics: from smooth to analytic, from np-hard to polytime*, pre-print (2012).

[Ko91] Ker-I Ko, *Complexity theory of real functions*, Progress in Theoretical Computer Science, Birkhäuser Boston Inc., Boston, MA, 1991. MR 1137517 (93i:03057)

[Mül95] Norbert Th. Müller, *Constructive aspects of analytic functions*, Computability and Complexity in Analysis (Ker-I Ko and Klaus Weihrauch, eds.), Informatik Berichte, vol. 190, FernUniversität Hagen, September 1995, CCA Workshop, Hagen, August 19–20, 1995, pp. 105–114.

[Mü01] NorbertTh. Müller, *The irram: Exact arithmetic in c++*, Computability and Complexity in Analysis (Jens Blanck, Vasco Brattka, and Peter Hertling, eds.), Lecture Notes in Computer Science, vol. 2064, Springer Berlin Heidelberg, 2001, pp. 222–252 (English).

[Wei00] Klaus Weihrauch, *Computable analysis*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2000, An introduction. MR 1795407 (2002b:03129)