Foreword

We are very pleased to welcome you to the *Fourteenth International Conference on Computability and Complexity in Analysis* (CCA 2017), which is organized at the School of Computing of KAIST, Daejeon, Republic of Korea. The CCA series of conferences provide an outstanding forum for reporting and discussing the latest developments about the theory of computability and complexity over real-valued data.

As you can attest from the abstracts which you can find in these proceedings, the topics touched by the CCA conferences are broad, ranging from more theoretical results to software implementations of exact real computation, and encompass fields like theoretical computer science, domain theory, logic, constructive mathematics, computer arithmetic, numerical mathematics. The CCA conferences have an established tradition of providing a welcoming forum where people from diverse areas can meet, exchange ideas, and initiate or strengthen collaborations, and we hope that CCA 2017 will be no exception.

This conference could not have happened without the contributions of many people. In particular we want to express our gratitude to the members of the Programme Committee, Johanna Franklin, Vassilios Gregoriades, Peter Hertling, Gyesik Lee, Alexander Melnikov, Takako Nemoto, Arno Pauly, Cristóbal Rojas, Frank Stephan, and Klaus Weihrauch, for their hard work reviewing and carefully recommending the contributions which will be presented at this conference. We would also like to thank the invited speakers, Hee-Kap Ahn, Verónica Becher, Anders Hansen, Takayuki Kihara, Amaury Pouly, and Linda Brown Westrick, for their availability to share with us their insights and knowledge. We are also very grateful to Sunyoung Kim and Sewon Park from the local organizing committee, for dealing with all the local aspects pertinent to the organization of this event. The conference would also not be possible without the excellent contributions submitted by the authors. We thank all the authors for their willingness to participate at CCA 2017 and present their latest research and ideas.

Several institutions gave us support in many ways. We would like to thank KAIST with its School of Computing, the *National Research Foundation of Korea* (NRF), and the *Association for Symbolic Logic* (ASL) for their highly appreciated sponsorship. Gratitude finally due to the CCA Steering Committee chaired by Vasco Brattka.

We sincerely hope that you enjoy this conference and your time in Daejeon, and that this event will further stimulate research in the field of computability and complexity over real-valued data.

Daniel Graça (Programme Committee chair) and
Martin Ziegler (Organizing Committee chair)

Table of Contents

Geometric Algorithms in the Presence of Obstacles

Hee-Kap Ahn*

Many classical algorithms for computing geometric structures such as Voronoi diagrams and convex hulls and algorithms assume that input objects are given in d-dimensional Euclidean space, for $d \geq 1$ and the distance between two points is simply the length of the straight line segment. However, there are obstacles in the problem domain in many practical cases such as robot motion planning in polygonal environments (industrial robots in manufacturing facilities and service robots at home) and urban planning for cities amidst rivers and mountains. Then the distance between two points is typically measured by the length of the shortest path that connects the two points and avoids the obstacles. It is called the geodesic distance to distinguish it from the Euclidean distance.

Most of the classical geometric problems are naturally extended to the geodesic setting. For instance, the problem of computing the geodesic diameter (and its counterpart, the geodesic center) in the presence of obstacles is a generalization of the problem of computing the diameter (and its counterpart, the center) of a convex polygon or a set of points. These problems received a lot of attention from the computational geometry community since the early 1980s [3, 4, 5, 10, 11].

This abstract provides a few recent algorithmic results on computing the geodesic center and the Voronoi diagrams of points in a simple polygon.

Let P be a closed simple polygon with n vertices. For any two points in P, the geodesic distance between them is the length of the shortest path that connects them among all paths contained in P. The geodesic center of P is the unique point in P that minimizes the largest geodesic distance to all other points of P. In 1989, Pollack et al. [10] gave an $O(n \log n)$-time algorithm that computes the geodesic center of P. Since then, a longstanding question has been whether this running time can be improved (indeed, this problem was explicitly posed by Pollack et al. [10] and later by Mitchell [6, Chapter 27]). In 2015, Ahn et al. showed how to compute the geodesic center of P in $O(n)$ time.

Given a set of sites in a simple polygon, a geodesic Voronoi diagram partitions the polygon into regions based on distances to sites under the geodesic metric. Papadopoulou and Lee [9] and Aronov et al. [2] gave algorithms for the geodesic nearest-point and the farthest-point Voronoi diagrams of a set of point sites lying inside a simple polygon, respectively. However, there still are gaps between these running times and trivial lower bounds. Very recently, Oh, Barba, and Ahn [8] presented an $O((n+m)\log\log n)$-time algorithm to compute the farthest-point geodesic Voronoi diagram for m sites lying on the boundary of a simple n-gon. This was the first improvement on the computation of farthest-point geodesic Voronoi diagrams since 1993 [2]. They also claimed that their algorithm can be extended for arbitrary sites in the polygon with running time $O(n \log\log n + m\log(n+m)))$.

In 2017, Oh and Ahn [7] presented algorithms for computing the nearest-point, higher-order and farthest-point Voronoi diagrams of m point sites in a simple n-gon, which improve the best known ones for $m \leq n/\operatorname{polylog} n$. These algorithms close the gaps of the running times towards the lower bounds. The algorithm for the geodesic nearest-point Voronoi diagram is optimal for

*Department of Computer Science and Engineering, POSTECH, heekap@postech.ac.kr

$m \leq n/\log^3 n$. Since the algorithm by Papadopoulou and Lee [9] is optimal for $m \geq n$. this algorithm together with the one by Papadopoulou and Lee gives the optimal running time for computing the diagram, except for the case that $n/\log^3 n < m < n$.

Similarly, the algorithm for the geodesic farthest-point Voronoi diagram is optimal for $m \leq n/\log^2 n$. Since the algorithm by Aronov et al. [2] is optimal for $m \geq n$, our algorithm together with the one by Aronov et al. gives the optimal running time for computing the diagram, except for the case that $n/\log^2 n < m < n$. This answers the question posed by Mitchell in the Handbook of Computational Geometry [6, Chapter 27] on the geodesic nearest-point and farthest-point Voronoi diagrams, except for the short intervals of $n/\operatorname{polylog} n < m < n$ stated above.

References

[1] Hee-Kap Ahn, Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, Matias Korman, and Eunjin Oh. A linear-time algorithm for the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 56(4):836–859, 2016.

[2] Boris Aronov, Steven Fortune, and Gordon Wilfong. The furthest-site geodesic Voronoi diagram. *Discrete & Computational Geometry*, 9(3):217–255, 1993.

[3] T. Asano and G.T. Toussaint. Computing the geodesic center of a simple polygon. Technical Report SOCS-85.32, McGill University, 1985.

[4] B Chazelle. A theorem on polygon cutting with applications. In *Proceedings 23rd Annual Symposium on Foundations of Computer Science, FOCS*, pages 339–349, 1982.

[5] John Hershberger and Subhash Suri. Matrix searching with the shortest-path metric. *SIAM Journal on Computing*, 26(6):1612–1634, 1997.

[6] Joseph S. B. Mitchell. Geometric shortest paths and network optimization. In *Handbook of Computational Geometry*, pages 633–701. Elsevier, 2000.

[7] Eunjin Oh and Hee-Kap Ahn. Voronoi diagrams for a moderate-sized point-set in a simple polygon. In *Proceedings of the 33rd International Symposium on Computational Geometry (SoCG 2017)*, pages 1–15, 2017.

[8] Eunjin Oh, Luis Barba, and Hee-Kap Ahn. The farthest-point geodesic voronoi diagram of points on the boundary of a simple polygon. In *32nd International Symposium on Computational Geometry (SoCG 2016)*, volume 51 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:15, 2016.

[9] Evanthia Papadopoulou and D.T.Lee. A new approach for the geodesic Voronoi diagram of points in a simple polygon and other restricted polygonal domains. *Algorithmica*, 1998 (4):319–352, 1998.

[10] Richard Pollack, Micha Sharir, and Günter Rote. Computing the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 4(6):611–626, 1989.

[11] Subhash Suri. Computing geodesic furthest neighbors in simple polygons. *Journal of Computer and System Sciences*, 39(2):220–235, 1989.

Computable Examples of Absolutely Normal Numbers

Verónica Becher
Universidad de Buenos Aires & CONICET
vbecher@dc.uba.ar

May 29, 2017

The most basic form of randomness for real numbers was formalized by Emile Borel more than 100 hundred years ago, and he called it *normality*. For an integer b greater than or equal to 2, the base b-expansion of a real number x is a sequence of integers $a_1, a_2 \ldots$, where $0 \leq a_i < b$ for every i, such that

$$x = \lfloor x \rfloor + \sum_{i=1}^{\infty} a_i b^{-i}.$$

We require that $a_i < b - 1$ infinitely many times to ensure that every rational number has a unique representation.

Definition. *Let b be an integer greater than or equal to 2. A real number is simply normal to base b if each digit $0, \ldots, b-1$ occurs in its b-ary expansion with the same frequency $1/b$. It is normal to base b if it is simply normal to every base b^k, for $k = 1, 2, \ldots$ and it is absolutely normal if it is normal to every integer base b.*

As first noted by D. D. Wall, normality can also be stated in the theory of uniform distribution (see the book by Bugeaud, 2012): a real number x is normal to base b if the sequence $(b^n x)_{n \geq 0}$ is uniformly distributed modulo 1 in the unit interval. Then, the speed of convergence to normality to base b of a real number x is measured using the classical notion of discrepancy of the sequence $(b^n x)_{n \geq 0}$.

Borel showed that almost all (with respect to Lebesgue measure) real numbers are absolutely normal. The results of Gál and Gál (1964), Philipp (1975) and Fukuyama (2008) give, for each base b gave the discrepancy of the sequence $(b^n x)_{n \geq 0}$ for almost all real numbers x.

All known computable examples of absolutely normal numbers have been obtained by algorithms specifically made to comply the definition of normality. For a designated base, these algorithms output the expansion of the computed number, one digit after the other. Unfortunately, for all these algorithms, the computational speed is obtained at the expense of the speed of convergence to normality. We measure computational speed as time complexity and, as usual, we say that an algorithm has complexity of order $f(n)$ if the algorithm outputs the first n digits after performing in the order of $f(n)$ elementary operations.

The current best results are a nearly linear-time algorithm with slow convergence to normality (Lutz and Mayordomo, 2016) and a triply exponential-time algorithm with a speed of convergence to normality equal to that realized by almost all real numbers (Becher, Slaman and Scheerer 2017). I will also comment on algorithms that combine of normality with some other mathematical properties.

References

[1] É. Borel. Les probabilités d'enombrables et leurs applications arithmétiques. *Supplemento di Rendiconti del Circolo Matematico di Palermo*, 27:247–271, 1909.

[2] V. Becher, A-M. Scheerer and T. Slaman. On absolutely normal numbers and their discrepancy estimate. *Acta Arithmetica*, to appear, 2017.

[3] Yann Bugeaud. *Distribution Modulo One and Diophantine Approximation.* Number 193 in Cambridge Tracts in Mathematics. Cambridge University Press, Cambridge, UK, 2012.

[4] K. Fukuyama. The law of the iterated logarithm for discrepancies of $\{\theta^n x\}$. *Acta Mathematica Hungarica*, 118(1):155–170, 2008.

[5] S. Gál and L. Gál. The discrepancy of the sequence $\{(2^n x)\}$. *Koninklijke Nederlandse Akademie van Wetenschappen Proceedings. Seres A 67 = Indagationes Mathematicae*, 26:129–143, 1964.

[6] J. Lutz and E. Mayordomo Computing Absolutely Normal Numbers in Nearly Linear Time preprint *arXiv:1611.05911*, 2016.

[7] W. Philipp. Limit theorems for lacunary series and uniform distribution mod 1. *Acta Arithmetica*, 26(3):241–251, 1975.

COMPUTING THE NON-COMPUTABLE - ON THE ROLE OF COMPUTABILITY THEORY IN THE SCIENCES

A. C. HANSEN

Abstract. Computability theory is a mainstay in the foundations of computational mathematics. Moreover, there are vast areas in the sciences where non-computable problems arise, including computational quantum physics and chemistry, condensed matter physics, statistical mechanics, computational biology etc. In this talk I will precent recent results showing how the emerging fields of mathematics of information, statistical estimation in data sciences, machine learning as well as image and signal processing turn out to be filled with non-computable problems. However, the paradox is that many of the former and latter problems are computed with great success on a daily basis in applications. For example, some of the most recent advances in Magnetic Resonance Imaging (MRI) are based on non-computable problems, yet they work incredibly well in practice and are implemented in the next generation of MRI machines.

This paradox demonstrates the challenge of making computability theory relevant to the sciences, scientific computing and computational mathematics. In this talk I will discuss this issue, the new findings and demonstrate how new developments in the Solvability Complexity Index (SCI) hierarchy helps to bridge the gap between computability theory and scientific computing as well as the sciences. In particular, the SCI hierarchy allows for a rich classification theory explaining the above paradox.

DAMTP, University of Cambridge
E-mail address: ach70@cam.ac.uk

Martin-like phenomena in the classification of real-valued functions

Takayuki Kihara

Graduate School of Informatics, Nagoya University, Japan
kihara@i.nagoya-u.ac.jp

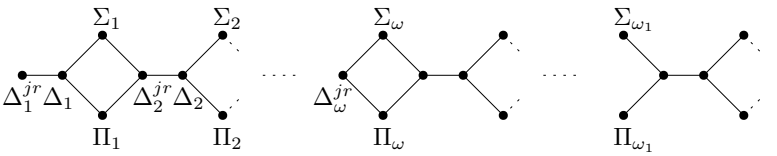
Throughout this talk, we assume AD^+, which is an extension of the axiom of determinacy[1] introduced by Woodin [9]. However, if we restrict our attention to Borel sets and Baire functions, every result presented in this talk can be provable within ZFC.

Recently, Day-Downey-Westrick [2] introduced the notions of m-, tt-, and T-reducibility for real-valued functions. They connected these reducibility notions with the Bourgain hierarhcy, the Kechris-Louveau ranks, etc.

In this talk, we give a full description of DDW's m-degrees, and clarify the relationship between DDW's T-degrees and the uniform Martin conjecture.

To present our results, we need some preliminaries. By Wadge [8] and Martin-Monk, non-self-dual pairs of Wadge pointclasses are well-ordered, say $(\Gamma_\alpha, \check{\Gamma}_\alpha)_{\alpha<\Theta}$, where Θ is the height of the Wadge degrees. Van Wesep [7] and Steel [5] showed that exactly one of Γ_α or $\check{\Gamma}_\alpha$ has the separation property. By Π_α, we denote the one which has the separation property, and by Σ_α, we denote the other one. Then define $\Delta_\alpha = \Sigma_\alpha \cap \Pi_\alpha$.

Theorem 1. *The structure of the DDW-m-degrees of real-valued functions looks like the following figure:*



That is, each successor selfdual Wadge degree splits into two degrees (which are linearly ordered), and nonselfdual Wadge degrees remain the same. (Here, Δ_α^{jr} indicates that the Lipschitz σ-join-reducible Δ_α sets.)

In particular, the DDW-m-degrees of real-valued functions form a semi-well-order of length Θ.

Day-Downey-Westrick [2] defined T-reducibility for real-valued functions as parallelized continuous Weihrauch reducibility, that is, *f is DDW-T-reducible to g* if there are continuous functions H, K such that $f = K \circ \widehat{g} \circ H$, where $\widehat{g}$ is the parallelization of g.

[1] $\mathsf{ZF}+\mathsf{DC}_{\mathbb{R}}+$"Every set of reals is ∞-Borel"+"Ordinal Determinacy".

We show that the DDW-T-degrees (hence the parallel continuous Weihrauch degrees) are exactly the *natural* Turing degrees in the context of the uniform Martin conjecture. Here, we assume that a decision problem P is *natural* if it is relativizable (that is, P^X exists for any oracle X), and if $X \equiv_T Y$ implies $P^X \equiv_T P^Y$ uniformly. For natural decision problems P, Q, we say that P *is Martin-reducible to* Q if there is an oracle C such that $P^X \leq_T Q^X \oplus C$ for any X. See also [6, 4, 1, 3]. (This is slightly different from the original definition, but only on constant functions.) Clearly, one can think of a natural decision problem P as a real-valued function $\iota_P : 2^\omega \to \mathbb{R}$ (via $2^\omega \hookrightarrow \mathbb{R}$).

Theorem 2. *The map $P \mapsto \iota_P$ induces an isomorphism between the Martin degrees of natural decision problems and the DDW-T-degrees of real-valued functions.*

In particular, the DDW-T-degrees of real-valued functions form a well-order of length Θ.

References

1. Howard Becker. A characterization of jump operators. *J. Symbolic Logic*, 53(3):708–728, 1988.
2. Adam Day, Rod Downey, and Linda B. Westrick. Turing-, tt-, and m-reductions for functions in the Baire hierarchy. Talk at Dagstuhl Seminar 17081 "Computability Theory", 2017.
3. Andrew Marks, Theodore Slaman, and John Steel. Martin's conjecture, arithmetic equivalence, and countable Borel equivalence relations. In *Ordinal Definability and Recursion Theory: The Cabal Seminar, Volume III*, pages 493–519. Cambridge University Press, 2016.
4. Theodore A. Slaman and John R. Steel. Definable functions on degrees. In *Cabal Seminar 81–85*, volume 1333 of *Lecture Notes in Math.*, pages 37–55. Springer, Berlin, 1988.
5. John R. Steel. Determinateness and the separation property. *J. Symbolic Logic*, 46(1):41–44, 1981.
6. John R. Steel. A classification of jump operators. *J. Symbolic Logic*, 47(2):347–358, 1982.
7. Robert A. Van Wesep. Separation principles and the axiom of determinateness. *J. Symbolic Logic*, 43(1):77–81, 1978.
8. William Wilfred Wadge. *Reducibility and Determinateness on the Baire Space*. ProQuest LLC, Ann Arbor, MI, 1983. Thesis (Ph.D.)–University of California, Berkeley.
9. W. Hugh Woodin. *The axiom of determinacy, forcing axioms, and the nonstationary ideal*, volume 1 of *de Gruyter Series in Logic and its Applications*. Walter de Gruyter & Co., Berlin, 1999.

A universal polynomial differential equation, and some consequences for computability theory*

Amaury Pouly
MPI-SWS, Germany
pamaury@mpi-sws.org

An astonishing fact was established by Lee A. Rubel (1981): there exists a fixed non-trivial fourth-order polynomial differential algebraic equation (DAE) such that for any positive continuous function φ on the reals, and for any positive continuous function $\epsilon(t)$, it has a $\mathcal{C}^\infty$ solution with $|y(t)-\varphi(t)| < \epsilon(t)$ for all t. Lee A. Rubel provided an explicit example of such a polynomial DAE:

$$3y'^4y''y''''^2 - 4y'^4y'''^2y'''' + 6y'^3y''^2y'''y'''' + 24y'^2y''^4y'''' - 12y'^3y''y'''^3 - 29y'^2y''^3y'''^2 + 12y''^7 = 0.$$

Other examples of universal DAE have later been proposed by other authors.

However, while these results may seem very surprising, their proofs are quite simple and are frustrating for a computability theorist, or for people interested in modeling systems in experimental sciences. First, the involved notions of universality is far from usual notions of universality in computability theory because the proofs heavily rely on the fact that constructed DAE does not have unique solutions for a given initial data. Indeed, in general a DAE may not have a unique solution, given some initials conditions. But Rubel's DAE *never* has a unique solution, even with a countable number of conditions of the form $y^{(k_i)}(a_i) = b_i$. This is very different from usual notions of universality where one would expect that there is clear unambiguous notion of evolution for a given initial data, for example as in computability theory. Second, the proofs usually rely on solutions that are piecewise defined. Hence they cannot be analytic, while analycity is often a key expected property in experimental sciences. Third, the proofs of these results can be interpreted more as the fact that (fourth-order) polynomial algebraic differential equations is a too loose a model compared to classical ordinary differential equations. In particular, one may challenge whether the result is really a universality result.

The question whether one can require the solution that approximates φ to be the unique solution for a given initial data is a well known open problem [Rubel 1981, page 2], [Boshernitzan 1986, Conjecture 6.2]. In this talk, we describe how we solve this open problem and show that Rubel's statement holds for polynomial ordinary differential equations (ODEs). Since polynomial ODEs have a unique solution given an initial data, this positively answers Rubel's open problem. More precisely, we show the following:

Theorem 1. *There exists a* ***fixed*** *polynomial vector p (with rational coefficients) in d variables such that for any functions $f \in C^0(\mathbb{R})$ and $\varepsilon \in C^0(\mathbb{R}, \mathbb{R}_{>0})$, there exists $\alpha \in \mathbb{R}^d$ such that there exists a unique solution $y : \mathbb{R} \to \mathbb{R}^d$ to*

$$y(0) = \alpha, \qquad y' = p(y).$$

Furthermore, this solution satisfies that $|y_1(t) - f(t)| \leqslant \varepsilon(t)$ for all $t \in \mathbb{R}$, and it is analytic.

The proof uses ordinary differential equation programming. We believe it sheds some light on computability theory for continuous-time models of computations. It also demonstrates that ordinary differential equations are indeed universal in the sense of Rubel and hence suffer from the same problem as DAEs for modelization: a single equation is capable of modelling any phenomenon with arbitrary precision, meaning that trying to fit a model based on polynomial DAEs or ODEs is too general (if it has a sufficient dimension).

*Joint work with Olivier Bournez, LIX, Ecole Polytechnique, France, bournez@lix.polytechnique.fr

The story does not stop here however. The existence of such a universal differential equation is not satifying from the point of view of computability theory. Indeed, Theorem 1 does not explain *how to construct α from f*, it could even be non-constructive, making the entire construction barely more useful than Rubel's solution. Fortunately, the proof gives us an algorithm to build α, provided we have an appropriate representation of f and ε:

Theorem 2. *In Theorem 1, the map*

$$(f, \varepsilon) \mapsto \alpha$$

is $([\rho \to \rho]^2, \rho)-$computable, where ρ denotes the Cauchy representation.

This theorem has several consequences. First it is reasurring in the sense that Theorem 1 really gives us a universal differential equation, *one that can be used for computability*. But more importantly, this theorem gives us an alternative representation of computable functions over compact intervals. Indeed, it was shown in [Bournez et al., 2007] that polynomial differential equations compute all real computable functions on computable compact intervals. More precisely, they show the following:

Theorem 3. *A function $f : [a, b] \to \mathbb{R}$ is computable*[1] *if and only if there exists $d \in \mathbb{N}$ and two polynomials $q : \mathbb{R} \to \mathbb{R}^d$, $p : \mathbb{R}^d \to \mathbb{R}^d$ with computable coefficients, such that for any $x \in [a, b]$, there exists*[2] *$y : \mathbb{R}_{\geqslant 0} \to \mathbb{R}^d$ such that for all $t \geqslant 0$,*

$$y(0) = q(x), \qquad y'(t) = p(y(t)),$$

and $|y_1(t) - f(x)| \leqslant y_2(t)$ and $\lim_{t\to\infty} y_2(t) = 0$.

One can see this result as a particular *analog representation* of computable functions over compact intervals, even though at the moment there is no real theory on the subject (contrary to the well-studied theory of representations started by Weihrauch). We claim that Theorem 2 and Theorem 3 can be used together to define a new representation of computable functions:

Theorem 4. *A function $f : [0, 1] \to \mathbb{R}$ is computable if and only if there exists $d \in \mathbb{N}$, a computable $\tau > 1$, a polynomial $p : \mathbb{R}^d \to \mathbb{R}^d$ with computable coefficients and $y : \mathbb{R}_{\geqslant 0} \to \mathbb{R}^d$ such that for all $t \geqslant 0$,*

$$y(0) = q(x), \qquad y'(t) = p(y(t)),$$

and for all $n \in \mathbb{N}$ and $x \in [0, 1]$,

$$|f(x) - y_1(n\tau + x)| \leqslant 2^{-n}.$$

Note that we only gave the definition over $[0, 1]$ for simplicity but it works over $[a, b]$. Intuitively, y gives better and better approximations of f, with a "period" $\tau > 1$. This representation is different from the previous one on several points: the solution y gives us access to all of f at once, whereas the other requires to "re-run" the system if we want the value for a different input. In some sense, Theorem 3 is more like a machine computing f, whereas Theorem 4 is more like an effective and analog version of Weierstrass Approximation Theorem.

[1] In the usual sense of Computable Analysis.
[2] The solution to $y' = p(y)$ is necessarily unique.

Turing, tt- and m-reducibilities for functions in the Baire hierarchy

Linda Brown Westrick

May 29, 2017

This is joint work with Adam Day and Rod Downey.

What would be appropriate definitions for Turing, tt- and m-reducibility for arbitrary (possibly very discontinuous) functions $f : 2^\omega \to \mathbb{R}$?

Following a Weihrauch-like approach, one could imagine a continuous "computation" of such a function f from another function g proceeding as follows. Given an input $X \in 2^\omega$, the algorithm continuously transforms X into a number of inputs $Y_0, Y_1, \ldots$, which are given to g. Then, using names for $g(Y_0), g(Y_1), \ldots$ as an oracle, the algorithm must continuously produce a name for $f(X)$.

Taking this as our notion of Turing computability ($f \leq_{\mathbf{T}} g$), we obtain notions of $\leq_{\mathbf{tt}}$ and $\leq_{\mathbf{m}}$ by appropriate choice of representation for elements of $\mathbb{R}$ and appropriate restrictions on the use of the oracle. For example, informally, $f \leq_{\mathbf{m}} g$ if for each $p \in \mathbb{Q}$ there is a $q \in \mathbb{Q}$ and a continuous h such that $f(X) \lesssim p$ if and only if $g(h(X)) \lesssim q$. Here $\lesssim$ abbreviates a longer expression involving $<$ that permits an error up to some ε.

For f, g in Baire class 1, we have characterized the degrees of these reducibilities. They align precisely with the classification of the Baire 1 functions using the α rank considered in [KL90]. In the following theorem, for $f : 2^\omega \to \mathbb{R}$ a Baire 1 function, let $|f|$ denote its α rank, and let ξ_f denote the least ordinal such that $|f| \leq \omega^{\xi_f}$. It is known that sets of the form $\{f : \xi_f \leq \mu\}$ enjoy good closure properties.

Theorem 1. *Characterization of the $\leq_{\mathbf{m}}, \leq_{\mathbf{tt}}$ and $\leq_{\mathbf{T}}$ degrees for Baire 1 functions f and g:*

1. *If $|f| < |g|$ then $f \leq_{\mathbf{m}} g$. Furthermore, each set $\{f : |f| = \mu + 1\}$ contains exactly four $\mathbf{m}$-equivalence classes, and if μ is a limit, then $\{f : |f| = \mu\}$ is an $\mathbf{m}$-equivalence class.*
2. *$f \leq_{\mathbf{tt}} g$ if and only if $\xi_f \leq \xi_g$.*
3. *All discontinuous Baire 1 functions are $\leq_{\mathbf{T}}$ equivalent.*

The lowest $\mathbf{m}$-degrees recreate some recognizable classes. If $|f| = 1$, then f is continuous, and the set of continuous functions is divided into two $\mathbf{m}$-degrees, the constant functions and the non-constant functions. The smallest two of the four $\leq_{\mathbf{m}}$ degrees corresponding to $|f| = 2$ are incomparable, and consist of exactly the lower semi-continuous functions and upper semi-continuous functions, respectively.

Let the j_n denote the "nth jump function", the function $j_n : 2^\omega \to (0, 1)$ such that the binary expansion of $j_n(X)$ is $X^{(n)}$. The Baire n functions are

exactly those functions f for which $f \leq_{\mathbf{T}} j_n$. But with respect to $\mathbf{m}$-reductions, the nth jump function is the weakest Baire n function:

Theorem 2. *Concerning the structure of the $\leq_{\mathbf{m}}$ degrees:*

1. *For each Baire n function f, we have $f \leq_{\mathbf{m}} j_{n+1}$.*
2. *For each n and f, if f is Baire but not Baire n, then either*

$$j_{n+1} \leq_{\mathbf{m}} f \text{ or } -j_{n+1} \leq_{\mathbf{m}} f.$$

References

[KL90] A. S. Kechris and A. Louveau. A classification of Baire class 1 functions. *Trans. Amer. Math. Soc.*, 318(1):209–236, 1990.

MONTE CARLO COMPUTABILITY AND INVERTING JUMPS

VASCO BRATTKA, RUPERT HÖLZL, AND RUTGER KUYPER

Abstract. We prove a result that allows to invert jumps and that can be used to transfer separation results from one level of the Weihrauch lattice to the next higher level. We apply this result to separate higher probabilistic classes in the Weihrauch lattice and we discuss the notion of Monte Carlo computability and some applications of it in this context.

1. Introduction

We use $\leq_{\mathrm{W}}$ and $\leq_{\mathrm{sW}}$ to denote *Weihrauch reducibility* and *strong Weihrauch reducibility*, respectively, and f' denotes the *jump* of a problem in the Weihrauch lattice as introduced in [5]. It is easy to see that jumps are monotone with respect to strong Weihrauch reducibility. The following is taken from [5, Proposition 5.6].

Lemma 1.1. *$f \leq_{\mathrm{sW}} g \Longrightarrow f' \leq_{\mathrm{sW}} g'$.*

Here we provide a result that allows an implication in the inverse direction.

Theorem 1.2. *$f' \leq_{\mathrm{W}} g' \Longrightarrow (f \leq_{\mathrm{W}} g$ relative to the halting problem). An analogous statement holds with respect to strong Weihrauch reducibility.*

Here "relative to the halting problem" means that the reduction functions are allowed to use the halting problem as an oracle. The proof of this result is based on a theorem of Brattka, Hendtlass and Kreuzer [6, Theorem 14.11]

Weihrauch reductions with respect to the halting problem are, in particular, continuous Weihrauch reductions. Hence this result highlights the importance of continuous separations: they can be transferred to jumps. We apply this result to separate probabilistic problems. In particular, we are introduced in the concept of Monte Carlo computability that can be defined as follows [7].

Definition 1.3 (Monte Carlo computability)**.** Let (X, δ_X) and (Y, δ_Y) be represented spaces. A problem $f\colon\subseteq X \rightrightarrows Y$ is said to be *Monte Carlo computable* if there exists a computable function $F_1\colon\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ and a limit computable function $F_2\colon\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{S}$ such that $\langle \mathrm{dom}(f\delta_X) \times 2^{\mathbb{N}}\rangle \subseteq \mathrm{dom}(F_2)$ and for each $p \in \mathrm{dom}(f\delta_X)$ the following hold:

(1) $S_p := \{r \in 2^{\mathbb{N}} \colon F_2\langle p, r\rangle = 0\}$ is non-empty and $\mu_{2^{\mathbb{N}}}(S_p) > 0$,
(2) $\delta_Y F_1\langle p, r\rangle \in f\delta_X(p)$ for all $r \in S_p$.

This definition is similar to the definition of Las Vegas computability presented in [2, 3], except that the failure recognition mechanism only allows to recognize failures in the limit. It turns out that this concept can be characterized as follows.

Theorem 1.4. *$f \leq_{\mathrm{W}} \mathsf{PC}'_{\mathbb{R}}$ if and only if f is Monte Carlo computable.*

In [4] it was proved that $\mathsf{PC}'_{\mathbb{R}} \equiv_{\mathrm{W}} \mathsf{WWKL}' \times \mathsf{C}'_{\mathbb{N}}$. The above definition of Monte Carlo computability gives us an alternative proof of the following result that was already proved by Bienvenu and Kuyper [1] in a different way.

Date: May 24, 2017.

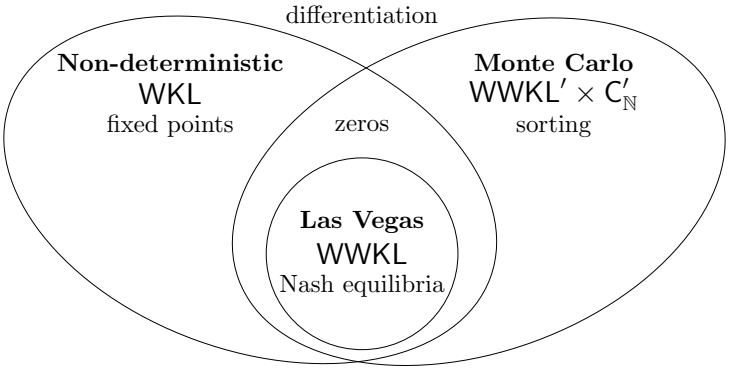


FIGURE 1. Classes of problems

Theorem 1.5. $\mathsf{PC}'_{\mathbb{R}} * \mathsf{PC}'_{\mathbb{R}} \equiv_{\mathrm{W}} \mathsf{PC}'_{\mathbb{R}}$.

This means that Monte Carlo computable problems are closed under composition and hence Monte Carlo computability is a natural notion of computability.

We can separate several notions of computability by natural problems as displayed in Figure 1. Here sorting infinite sequences is a basic problem that was introduced and studied by Neumann and Pauly [8].

REFERENCES

[1] L. Bienvenu and R. Kuyper. Parallel and serial jumps of Weak Weak König's Lemma. In A. Day et al. (eds.), *Computability and Complexity: Essays Ded. to Rod G. Downey on the Occasion of His 60th Birthday*, vol. 10010 of *LNCS*, pp. 201–217. Springer, 2017.

[2] V. Brattka, G. Gherardi, and R. Hölzl. Las Vegas computability and algorithmic randomness. In E. W. Mayr et al. *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, vol. 30 of *LIPIcs*, pp. 130–142, Dagstuhl, Germany, 2015.

[3] V. Brattka, G. Gherardi, and R. Hölzl. Probabilistic computability and choice. *Information and Computation*, 242:249–286, 2015.

[4] V. Brattka, G. Gherardi, R. Hölzl, and A. Pauly. The Vitali covering theorem in the Weihrauch lattice. In A. Day et al. (eds.), *Computability and Complexity: Essays Ded. to Rod G. Downey on the Occasion of His 60th Birthday*, vol. 10010 of *LNCS*, pp. 188–200. Springer, 2017.

[5] V. Brattka, G. Gherardi, and A. Marcone. The Bolzano-Weierstrass theorem is the jump of weak Kőnig's lemma. *Annals of Pure and Applied Logic*, 163:623–655, 2012.

[6] V. Brattka, M. Hendtlass, and A. P. Kreuzer. On the uniform computational content of computability theory. arXiv 1501.00433, 2015.

[7] V. Brattka, R. Hölzl, and R. Kuyper. Monte Carlo computability. In H. Vollmer et al. (eds.), *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*, vol. 66 of *LIPIcs*, pp. 17:1–17:14, Dagstuhl, Germany, 2017.

[8] E. Neumann and A. Pauly. A topological view on algebraic computation models. arXiv 1602.08004, 2016.

FACULTY OF COMPUTER SCIENCE, UNIVERSITÄT DER BUNDESWEHR MÜNCHEN, GERMANY AND DEPARTMENT OF MATHEMATICS & APPLIED MATHEMATICS, UNIVERSITY OF CAPE TOWN, SOUTH AFRICA

E-mail address: Vasco.Brattka@cca-net.de

FACULTY OF COMPUTER SCIENCE, UNIVERSITÄT DER BUNDESWEHR MÜNCHEN, GERMANY

E-mail address: r@hoelzl.fr

SCHOOL OF MATHEMATICS AND STATISTICS, VICTORIA UNIVERSITY OF WELLINGTON, WELLINGTON, NEW ZEALAND

E-mail address: mail@rutgerkuyper.com

A minimal representation for continuous functions

Franz Brauße* and Florian Steinberg†

Kawamura and Cook's framework of second-order representations is accepted as introducing the right notion of complexity for operators in analysis. For this reason the part of real complexity theory that considers complexity of operators and functionals has become very active in the past years [KC12, FGH14, FZ15, Ste17, and many more]. One of the celebrated results that contributed to the popularity and acceptance of the framework of second-order representations is that Kawamura and Cook succeeded to provide a second-order representation of the set of continuous functions on the unit interval which is minimal up to polynomial-time reductions with the property that the evaluation operator is polynomial-time computable. The paper provides a variation of this very result.

While working very well for theoretical considerations, the framework of second-order representations imposes some assumptions on the encodings that lead to extensive padding and seems unnatural in practical applications. Furthermore, some of the theoretical predictions seem to be out of sync with the behavior of popular implementations of real complexity theory: `iRRAM` is a framework for error-free real arithmetic based on the ideas of real complexity theory [Mül]. In `iRRAM` it is possible to implement functions and, as long as the implementation of the function is reasonable, evaluation of the function is fast. Computing the modulus of continuity of a function, in contrast, does not seem to be possible in a reasonable amount of time. This seems to contradict the theory where, in the framework of second-order representations introduced by Kawamura and Cook [KC12], one can prove that polynomial-time computability of evaluation implies polynomial-time computability of a modulus. Thus, it seems reasonable to assume that the behavior of `iRRAM` on functions can not be modeled by second-order representations.

The content of the paper

The paper uses a more relaxed notion of encoding than the one provided by Kawamura and Cook's second-order representations. It proves that in this more general setting it is possible to partially recover Kawamura and Cook's construction of a weakest representation of the continuous functions on the unit interval such that evaluation is fast. Fix an appropriate encoding of the dyadic numbers and let # be an separator symbol for encoding of pairs.

Definition 1 Define **the representation ξ_C of** $C([0,1])$ as follows: A string function φ is a ξ_C-name of a function $f \in C([0,1])$ if and only if

1. For all $r \in \mathbb{D} \cap [0,1]$ and $n \in \omega$ there are $q \in \mathbb{D}$ and $m \in \omega$ such that
$$\varphi(2^n \# r) = 2^m \# q \quad \text{and} \quad f([r \pm 2^{-m}] \cap [0,1]) \subseteq [q \pm 2^{-n}].$$
2. For all $r, q \in \mathbb{D} \cap [0,1]$ it holds that
$$\varphi(2^n \# r) = 2^m \# q \quad \Rightarrow \quad m \leq |\varphi|(n).$$

*Universität Trier, 54286 Trier, room H 420; Email: brausse@informatik.uni-trier.de; supported by the German Research Foundation (DFG), project WERA, grant MU 1801/5-1

†Technische Universität Darmstadt, Schloßgartenstraße 7, 64289 Darmstadt, Building S215, room 203; Email: steinberg@mathematik.tu-darmstadt.de

Note that the length of a name can be increased arbitrarily without interfering with these conditions by changing the values of the string function on strings that do not contain any #. Using this it is quite easy to see that the above indeed defines a representation, i.e. that any continuous function has a name.

Also note that the second condition implies that $|\varphi|$ is a modulus of continuity of $\xi_C(\varphi)$ in the following sense: A function $\mu : \omega \to \omega$ is called **modulus of continuity** of $f \in C([0,1])$ if it fulfills $\forall x, y \in [0,1]\, |x-y| \leq 2^{\mu(n)} \Rightarrow |f(x) - f(y)| \leq 2^{-n}$. The above is automatically fulfilled for $\mu(n) := |\varphi|\,(n+1)$ and $f := \xi_C(\varphi)$.

The paper proves that evaluation with respect to this representation is fast. Here being 'fast' is more restrictive than polynomial-time computability and called 'hyper-linear-time computability'.

Theorem 2 *Evaluation is hyper-linear-time computable with respect to ξ_C.*

The algorithm for evaluation with respect to ξ_C is strikingly similar to how `iRRAM` works internally.

The paper continues to prove that for any other representation such that evaluation is hyper-linear-time computable, there is a fast translation to ξ_C. To make be able to prove minimality of the representation it is necessary to restrict the translations that are considered 'fast' even further than the hyper-linear-time computability that was used for the evaluation procedure.

Theorem 3 *For a representation ξ of $C([0,1])$ the following are equivalent:*

1. *The evaluation operator is hyper-linear-time computable with respect to ξ.*
2. *The representation ξ can be translated to ξ_C in 2-independent hyper-linear time.*

Since ξ_C-names need not be length-monotone and the modulus of continuity is still encoded in the length of a name, it is not clear how to compute a modulus of a function in polynomial time. Indeed, the paper proves that it is impossible to compute a modulus of a function in polynomial-time with respect to ξ_C.

Theorem 4 *With respect to ξ_C it is impossible to find a modulus of continuity of a function in polynomial time.*

This implies that ξ_C is not polynomial-time translatable to the minimal second-order representation constructed by Kawamura and Cook. From the minimality results proven by Kawamura and Cook it follows that ξ_C is not polynomial-time equivalent to any second-order representation.

A full version of the paper can be found on the arXiv [BS17].

References

[BS17] Franz Brauße and Florian Steinberg, *A minimal representation for continuous functions*, https://arxiv.org/abs/1703.10044, 2017, preprint.

[FGH14] Hugo Férée, Walid Gomaa, and Mathieu Hoyrup, *Analytical properties of resource-bounded real functionals*, J. Complexity **30** (2014), no. 5, 647–671. MR 3239272

[FZ15] Hugo Férée and Martin Ziegler, *On the computational complexity of positive linear functionals on c[0;1]*, 2015, MACIS conference.

[KC12] Akitoshi Kawamura and Stephen Cook, *Complexity theory for operators in analysis*, ACM Trans. Comput. Theory **4** (2012), no. 2, 5:1–5:24.

[Mül] Norbert Th. Müller, *iRRAM - exact arithmetic in c++*, http://irram.uni-trier.de/, https://github.com/norbert-mueller/iRRAM, [Online; accessed 26-March-2017].

[Ste17] Florian Steinberg, *Complexity theory for spaces of integrable functions*, to appear in Logical Structures in Computer Science, 2017.

Noetherian Quasi-Polish Spaces (short abstract)*

Matthew de Brecht
Graduate School of Human and Environmental Studies
Kyoto University, Japan
matthew@i.h.kyoto-u.ac.jp

Arno Pauly
Département d'Informatique
Université libre de Bruxelles, Belgium
Arno.M.Pauly@gmail.com

Given a represented space $\mathbf{X}$, the open subsets of $\mathbf{X}$ are in bijective correspondence with the continuous functions from $\mathbf{X}$ to the Sierpiński space $\mathbb{S}$. It is therefore natural to view the open sets $\mathcal{O}(\mathbf{X})$ as a represented space, by identifying $\mathcal{O}(\mathbf{X})$ with the represented function space $\mathcal{C}(\mathbf{X}, \mathbb{S})$ of continuous maps from $\mathbf{X}$ to $\mathbb{S}$.

It is well known that compactness of a space $\mathbf{X}$ can be characterized as the singleton $\{X\}$ being open in the space $\mathcal{O}(\mathbf{X})$. In other words, a represented space $\mathbf{X}$ is (computably) compact if and only if there is a (computable) continuous map from $\mathcal{O}(\mathbf{X})$ to $\mathbb{S}$ which maps X to $\top$ and all other open subsets to $\bot$. Equivalently, this means that universal quantification over a compact space preserves open predicates.

The Δ^0_2-subsets of a represented countably based topological space are in bijection with the realizable maps from the space into $\mathbb{S}^\nabla$, where $\mathbb{S}^\nabla$ is Sierpiński space with a modified representation corresponding to finite mind-change computability [2]. We can extend this observation for topological spaces to a definition for arbitrary represented spaces and identify $\Delta^0_2(\mathbf{X})$ with the function space $\mathcal{C}(\mathbf{X}, \mathbb{S}^\nabla)$. Then $\Delta^0_2(\mathbf{X})$ is again a represented space, and it makes sense to discuss the Δ^0_2-subsets of $\Delta^0_2(\mathbf{X})$. Other classes of Borel sets can be interpreted as represented spaces in this way, which provides a promising approach to descriptive set theory within the category of represented spaces [5].

This suggests higher-order analogues to compactness: For example, we can investigate the spaces $\mathbf{X}$ where $\{X\}$ is a Δ^0_2-subset of the space $\Delta^0_2(\mathbf{X})$. Call this notion ∇-compactness. As Δ^0_2 is self-dual, we find that both universal and existential quantification over ∇-compact spaces preserve Δ^0_2-predicates:

Proposition 1. The following are equivalent for a represented space $\mathbf{X}$:

1. $\mathbf{X}$ is (computably) ∇-compact.
2. For any represented space $\mathbf{Y}$, the function $\forall : \Delta^0_2(\mathbf{X} \times \mathbf{Y}) \to \Delta^0_2(\mathbf{Y})$ mapping R to $\{y \in \mathbf{Y} \mid \forall x \in \mathbf{X}\ (x, y) \in R\}$ is well defined and a (computable) continuous function.
3. For any represented space $\mathbf{Y}$, the function $\exists : \Delta^0_2(\mathbf{X} \times \mathbf{Y}) \to \Delta^0_2(\mathbf{Y})$ mapping R to $\{y \in \mathbf{Y} \mid \exists x \in \mathbf{X}\ (x, y) \in R\}$ is well defined and a (computable) continuous function.

□

Recall that a space is *Noetherian* if and only if every (open) subset is compact [4]. We introduce an effective version of this notion which we call ∇-*computably Noetherian*. Within the setting of quasi-Polish spaces [1], we can fully characterize the ∇-compact spaces:

Theorem 2. A quasi-Polish space is (computably) ∇-compact if and only if it is (∇-computably) Noetherian. □

*This work was supported by JSPS Core-to-Core Program, A. Advanced Research Networks. The first author was supported by JSPS KAKENHI Grant Number 15K15940. The second author was supported by the ERC inVEST (279499) project.

Note that the restriction to quasi-Polish spaces is sufficiently general to include plenty of examples. For example, the countable ordinal $\omega + 1$, ordered by $0 < 1 < \ldots < \omega$ and equipped with the Scott-topology, is a Noetherian quasi-Polish space. Similarly, any countable successor ordinal can be viewed as a Noetherian quasi-Polish space, and additional examples can be found from the theory of well-quasiorders.

Another source of examples comes from algebraic geometry: the prime spectrum of any countable Noetherian ring with the Zariski topology is a Noetherian quasi-Polish space. A simple example is the prime spectrum of the integers, which is homeomorphic to the sobrification of the natural numbers with the cofinite topology (sobrification adds only a single point).

An extended abstract is available as [3].

References

[1] Matthew de Brecht (2013): *Quasi-Polish spaces.* *Annals of Pure and Applied Logic* 164(3), pp. 354–381.

[2] Matthew de Brecht (2014): *Levels of discontinuity, limit-computability, and jump operators.* In Vasco Brattka, Hannes Diener & Dieter Spreen, editors: *Logic, Computation, Hierarchies*, de Gruyter, pp. 79–108.

[3] Matthew de Brecht & Arno Pauly (2016): *Noetherian Quasi-Polish Spaces.* arXiv 1607.07291. Available at `http://arxiv.org/abs/1607.07291`.

[4] Jean Goubault-Larrecq (2007): *On Noetherian Spaces.* In: *22nd Annual IEEE Symposium on Logic in Computer Science*, pp. 453–462, doi:10.1109/LICS.2007.34.

[5] Arno Pauly & Matthew de Brecht (2015): *Descriptive Set Theory in the Category of Represented Spaces.* In: *30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 438–449, doi:10.1109/LICS.2015.48.

Computing Periods...

Junhee Cho and Martin Ziegler, KAIST

Abstract. A *period* is the difference between the volumes of two semi-algebraic sets. Recent research has located these in low levels of the Grzegorczyk Hierarchy, that is, established a structural complexity-theoretic upper bound. The present extended abstract describes work in progress on their refined resource-oriented parameterized computational complexity.

An open question in Algebraic Model Theory asks for a real number which is not a *period*, that is, not 'definable' as the difference between two volumes of semi-algebraic sets; see [KZ01, Problem 3]. Recall that semi-algebraic means (a finite Boolean combination of) sets of solutions polynomial inequalities with integer coefficients

$$S_p := \big\{(x_1,\ldots,x_d) \,\big|\, p(x_1,\ldots,x_d) > 0\big\} \subseteq \mathbb{R}^d, \quad p \in \mathbb{Z}[x_1,\ldots,x_d] \ . \tag{1}$$

The family of periods includes transcendental $\pi = \mathrm{vol}\{(x,y) : x^2+y^2 < 1\}$ yet is countable, hence missing most reals. Towards an explicit example, a recent series of works has gradually narrowed down periods to be computable in the sense of Recursive Analysis, of *elementary* computational complexity [Yos08] in the Grzegorczyk Hierarchy, and in fact even lower:

Fact 1 *Let* Lower Elementary *be the smallest class of total multivariate functions* $f : \mathbb{N}^d \to \mathbb{N} = \{0,1,2,\ldots\}$ *containing the constants, projections, successor, modified difference* $x \dot{-} y = \max\{x-y,0\}$*, and is closed under composition and bounded summation* $f(\vec{x},y) = \sum_{z=0}^{y} g(\vec{x},z)$*.*

Write $\mathcal{M}^2 = \mathcal{E}^2$ *for the smallest class of such* f *containing the constants, projections, successor, modified difference, binary multiplication, and is closed under composition and bounded search* $\mu(f)(\vec{x},y) = \min\{z \le y : f(\bar{z},z) = 0\}$*.*

A real number r *is* lower elementary*/in* $\mathcal{M}^2$ *if there exist lower elementary/*$\mathcal{M}^2$ *functions* $f,g,h : \mathbb{N} \to \mathbb{N}$ *with* $\left|r - \frac{f(N)-g(N)}{h(N)}\right| < 1/N$ *for all* $N > 0$*.*

a) All functions from $\mathcal{M}^2$ *are lower elementary; and the latter functions grow at most polynomially in the value of the arguments. In terms of the binary input length and with respect to bit-cost, lower elementary functions are computable using a linear amount of memory for intermediate calculations and output, that is, they belong to the complexity class* $\mathsf{FSPACE}(n)$*.*

b) $\mathsf{FSPACE}(n)$ *is closed under bounded summation and therefore coincides with the class of lower elementary functions. The* 0/1*-valued functions (that is, decision problems) in* $\mathcal{M}^2$ *exhaust the class* $\mathsf{SPACE}(n)$ [Rit63, §4]*; cmp.* [Kut87]*.*

c) π *and* $e = \sum_n 1/n!$ *and Liouville's transcendental number* $L = \sum_n 10^{-n!}$ *and the Euler-Mascheroni Constant* $\gamma = \lim_n \big(-\ln(n) + \sum_{k=1}^{n} 1/k\big)$ *are all lower elementary* [Sko08, §3]*.*

d) The set of lower elementary real numbers constitutes a real closed field: Binary sum and product and reciprocal of lower elementary real numbers, as well as any real root of a non-zero polynomial with lower elementary coefficients, are again lower elementary [SWG12, Theorem 2]*.*

e) Arctan, natural logarithm and exponential as well as Γ *and* ζ *function map lower elementary reals to lower elementary reals* [TZ10, §9]*.*

f) Natural logarithm maps periods to periods; $\zeta(s)$ *is a period for every integer* $s \ge 2$ [KZ01, §1.1]*.*

g) Periods are lower elementary [TZ10, Corollary 6.4]*.*

h) Given a Boolean expression $\varphi(x_1,\ldots,x_m)$ *as well as the degrees and coefficients of the polynomials* p_j *defining its constituents* S_{p_j}*, deciding whether the semi-algebraic set* $\varphi\big(S_{p_1},\ldots,S_{p_m}\big)$ *is non-empty/of given dimension* [Koi99] *is complete for the complexity class* $\mathsf{NP}^0_{\mathbb{R}} \supseteq \mathsf{NP}$*.*

Item a) follows by structural induction. Together with b) it relates resource-oriented to Grzegorczyk's structural Complexity Theory. Common efficient and practical algorithms tailored for

approximating L, e, γ, or the period π do so up to absolute error $1/N := 2^{-n}$ within time polynomial in the binary precision parameter $n = \log_2 N$ [Kan03]; whereas the best runtime bound known for $\mathsf{SPACE}(n)$ is only exponential [Pap94, Problem 7.4.7]. Note that the hardness Result h) does not seem to entail a lower bound on the problem of approximating the volume.

This raises the question, driving the present work in progress, of whether or not periods in general admit polynomial-time algorithms; and how/what further parameters affect their computational bit-complexity in addition to the binary output precision n [Ko91, Wei03]. Indeed we agree [KZ01, Problem 2] that efficient Reliable High-Precision Numerics and Experimental *Transcendental* Mathematics as computational tools can provide enriched insight into questions including, but not restricted to [Ret12], explicit candidates for non-periods.

We restrict to (volumes of) semi-algebraic sets inside the unit cube $[0;1)^d$. One approach to the 1D case $d = 1$ subdivides the interval $[0;1)$ into sub-intervals $[a \cdot 2^{-n}, (a+1) \cdot 2^{-n})$, $\mathbb{N} \ni a < 2^n$; evaluates the polynomial(s) signs on a random point from each sub-interval; and counts those with positive sign, divided by 2^n: Since a polynomial of degree k can have at most k roots, this will approximate the true volume up to error $k \cdot 2^{-n}$. Moreover with high probability a random point will avoid all roots, hence rendering the sign computable; cmp. [MPPZ16, Definition 2]. The following suggests a way of generalizing this to higher dimensions:

Lemma 2. *Fix a d-variate real or complex power series around zero* $f(\vec{x}) = \sum_{\vec{j}} c_{\vec{j}} \cdot x_1^{j_1} \cdots x_d^{j_d}$ *with* $\vec{j} = (j_1, \ldots, j_d)$ *ranging over* $\mathbb{N}^d$*, converging absolutely and uniformly for all* $\vec{x} = (x_1, \ldots, x_d) \in [-R, +R]^d$*. Abbreviate* $|\vec{x}| := |x_1| + \cdots + |x_d|$ *and for* $0 < r < R$ *consider the condition*

$$|c_{\vec{0}}| \quad > \quad \sum_{\vec{j} \neq \vec{0}} |c_{\vec{j}}| \cdot r^{|\vec{j}|} \quad . \tag{2}$$

a) If $0 < r < R$ *satisfies Condition (2), then* f *has no root in* $[-r, +r]^d$*.*
b) If $f(\vec{0}) \neq 0$*, then there exists* $r > 0$ *such that Condition (2) holds.*
c) Suppose f *is a polynomial of total degree* $k = \max\{j_1 + \cdots + j_d : c_{j_1,\ldots,j_d} \neq 0\}$ *and consider the* N^d *cubes* $\prod_{j=1}^{d} \big[A_j/N, (A_j+1)/N\big) \subseteq [0,1)$ *in* $[0,1)^d$*,* $A_1, \ldots, A_d \in \{0, 1, \ldots, N-1\} =: [N]$*. Then at most* $\mathcal{O}\big(k + d^2 \cdot N\big)^{d-1} \cdot (k + d^2)$ *of them contain a root of* f*.*

References

[Kan03] Yasumasa Kanada. 計算機による円周率計算（特集 円周率 π）. *J. Mathematical Culture*, 1(1):72–83, 2003.

[Ko91] Ker-I Ko. *Complexity Theory of Real Functions.* Progress in Theoretical Computer Science. Birkhäuser, Boston, 1991.

[Koi99] Pascal Koiran. The real dimension problem is NP_R-complete. *Journal of Complexity*, 15(2):227–238, 1999.

[Kut87] Miroslaw Kutylowski. Small Grzegorczyk classes. *Journal of the London Mathematical Society*, 36(2):193–210, 1987.

[KZ01] Maxim Kontsevich and Don Zagier. Periods. In Björn Engquist and Wilfried Schmid, editors, *Mathematics unlimited — 2001 and beyond*, pages 771–808. Springer, 2001.

[MPPZ16] Norbert Müller, Sewon Park, Norbert Preining, and Martin Ziegler. On formal verification in imperative multivalued programming over continuous data types. *arXiv*, 1608.05787, 2016.

[Pap94] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Ret12] Robert Rettinger. On computable approximations of Landau's constant. *Log. Methods Comput. Sci.*, 8(4):4:15, 11, 2012.

[Rit63] Robert W. Ritchie. Classes of predictably computable functions. *Trans. Amer. Math. Soc.*, 106(1):139–173, 1963.

[Sko08] Dimiter Skordev. On the subrecursive computability of several famous constants. *Journal of Universal Computer Science*, 14(6):861–875, 2008.

[SWG12] Dimiter Skordev, Andreas Weiermann, and Ivan Georgiev. $\mathcal{M}^2$-computable real numbers. *J. Logic Comput.*, 22(4):899–925, 2012.

[TZ10] Katrin Tent and Martin Ziegler. Computable functions of reals. *Münster J. Math.*, 3:43–65, 2010.

[Wei03] Klaus Weihrauch. Computational complexity on computable metric spaces. *Mathematical Logic Quarterly*, 49(1):3–21, 2003.

[Yos08] Masahiko Yoshinaka. Periods and elementary real numbers. *arXiv*, 0805.0349, 2008.

On the Relationship between Classical and Computable Topology

Pieter Collins
Department of Data Science and Knowledge Engineering, Maastricht University,
Bouillonstraat 8-10, Postbus 616, 6200MD, The Netherlands
pieter.collins@maastrichtuniversity.nl

Abstract

In this talk, we will look at some important concepts in classical topology, and discuss their computable counterparts. The main goals are to combine many important results which are scattered throughout the literature, and to find the most natural definitions for computable versions of classical properties. It turns out that many notions in computable topology coincide more closely with the corresponding notion in limit/sequential spaces than the classical topological notion. Further, for the class of spaces admitting an admissible representation, in many cases the definitions coincide; an important case are the various topologies on the open sets, including the Scott topology.

The material covered is essentially (a corrected version of) that of [Col10, Section 4].

Question 1. *When does a topological space have a suitable representation?*

A suitable representation of a topological space should encode "sufficient" information, which naturally coincides with continuity. However, it should also not encode "too much" information, which topologically means the map should be a quotient map, but computationally also means the representation must satisfy a further "admissibility" property [KW85]. Then a result of Schröder [Sch02] gives:

Theorem 1. *A topological space has an admissible quotient representation if, and only if it is a quotient of a countably-based space, or it is a sequential space with a countable sequential pseudobase.*

In particular, any topological quotient of a subspace of Σ^ω has an *admissible* quotient representation. We say a space is *representable* if it has an admissible quotient representation, and a *represented space* $\mathcal{X}$ is a topological space together with an admissible quotient representation.

Question 2. *What are the natural computable topologies on product and function spaces?*

Since tupling on Σ^ω is a computable operation, and continuous functions $\Sigma^\omega \to \Sigma^\omega$ have a natural representation in Σ^ω under which evalutation is computable (from the existence of universal Turing machines and Kleene's s-m-n theorem), there are natural computable structures on product and function spaces.

Theorem 2. *The canonical representations of product types $\mathcal{X}_1 \times \mathcal{X}_2$ and function types $\mathcal{Y}^\mathcal{X}$ induce the* sequentialisations *of the natural classical topologies.*

Question 3. *What is the natural notion of compactness in computable topology?*

Recall that a set is (countably) compact if any (countable) open cover has a finite subcover, and sequentiall compact if any sequence has a finite subsequence.

Theorem 3. *If $\mathcal{X}$ is a representable space, then compactness, countable compactness, and sequential compactness are equivalent.*

Question 4. *What is the natural topology on the type of open sets?*

The open sets $\mathcal{O}(\mathcal{X})$ of a topological space $\mathcal{X}$ are naturally associated with the space $\mathbb{S}^\mathcal{X}$, where $\mathbb{S}$ is the Sierpinski space. However, there are other topologies for the open sets of a space used in computability theory, most notably the *Scott* topology. Recall a directed collection of open sets $\mathcal{Q}$ is *Scott-open* if for all $U \in \mathcal{Q}$, there exists $V \in \mathcal{Q}$ such that $V \Subset U$, where $V \Subset U$ if every open cover of U has a finite subcover of V, and *ω-Scott open* if the property holds for countable open covers of U.

Theorem 4. *If $\mathcal{X}$ is a representable space, then the Scott topology and ω-Scott topology on $\mathcal{O}(\mathcal{X})$ coincide, and are the sequential topology of the function space $\mathbb{S}^{\mathcal{X}}$.*

Question 5. *What are the natural topologies and representations on closed and compact sets?*

Recall that a collection $\mathcal{Q}$ of open sets is a *filter* if $U_1 \cap U_2 \in \mathcal{Q} \iff U_1 \in \mathcal{Q} \wedge U_2 \in \mathcal{Q}$, and a *cofilter* if $U_1 \cup U_2 \in \mathcal{Q} \iff U_1 \in \mathcal{Q} \vee U_2 \in \mathcal{Q}$, and an *ultrafilter* if both. It is clear that the sets intersecting a given set are a cofilter, the supersets of a set are a filter, and sets containing a point are an ultrafilter. A space is *sober* if any Scott-open ultrafilter arises in this way. Sobriety is related to completeness for metric spaces. From [HM81], we have:

Theorem 5. *If $\mathcal{X}$ is a sober representable space, then there are natural bijections between the closed sets and the Scott-open cofilters $\mathcal{X}$, and between the saturated compact sets and the Scott-open filters of $\mathcal{X}$.*

Question 6. *What is a general, natural notion of local-compactness in the computable setting?*

Recall that a space is *locally-compact* if every point has a compact neighbourhood. A generalisation is the notion of *core-compactness*: A space is *core-compact* if for every open V and every $x \in V$, there exists an open set U such that $x \in U$ and $U \Subset V$. Combining results of [HL78, ELS04, EH02], we can show that local compactness is sufficient:

Theorem 6. *Any representable core-compact space is locally-compact. If $\mathcal{X}$ is core-compact, then $\mathcal{O}(\mathcal{X})$ is also locally-compact.*

References

[Col10] P. Collins. Computable analysis with applications to dynamic systems. Technical report, 2010. CWI Report MAC-1002.

[EH02] Martín Escardó and Reinhold Heckmann. Topologies on spaces of continuous functions. *Topology Proc.*, 26(2):545–564, 2002.

[ELS04] Martín Escardó, Jimmie Lawson, and Alex Simpson. Comparing Cartesian closed categories of (core) compactly generated spaces. *Topology Appl.*, 143(1-3):105–145, 2004.

[HL78] Karl H. Hofmann and Jimmie D. Lawson. The spectral theory of distributive continuous lattices. *Trans. Amer. Math. Soc.*, 246:285–310, 1978.

[HM81] K. Hofmann and M. Mislove. Local compactness and continuous lattices. In *Continuous Lattices*, volume 871 of *Lect. Notes Math.*, pages 209–248, 1981.

[KW85] C. Kreitz and K. Weihrauch. Theory of representations. *Theoretical Comput. Sci.*, 38:35–53, 1985.

[Sch02] Matthias Schröder. *Admissible Representations for Continuous Computations.* PhD thesis, Fachbereich Informatik, FernUniversität Hagen, 2002.

Semantics and Computation of the Evolution of Hybrid Systems with Ariadne

Pieter Collins,
Department of Data Science and Knowledge Engineering, Maastricht University,
Bouillonstraat 8-10, Postbus 616, 6200MD, The Netherlands
pieter.collins@maastrichtuniversity.nl

Davide Bresolin,
Dipartimento di Informatica, Università di Bologna,
Mura Anteo Zamboni 7, 40126 Bologna, Italy
davide.bresolin@unibo.it

Sanja Živanovic Gonzalez
Department of Mathematics and Computer Science, Barry University,
11300 NE 2nd Avenue, Miami Shores, FL 33161-6695, USA
szivanovic@barry.edu

Luca Geretti and Tiziano Villa
Dipartimento di Informatica, Università degli Studi di Verona,
Strada le Grazie 15, 37134 Verona, Italy
{luca.geretti,tiziano.villa}@univr.it

Abstract

In this talk we will present material on the semantics, computability, and algorithms for the evolution of hybrid dynamical systems, and an overview of the tool ARIADNE for verification of hybrid systems [1].

Hybrid systems are characterised by undergoing continuous evolution interspersed by discrete jumps. They exhibit all the complexities of finite automata, nonlinear dynamic systems and differential equations, and are extremely difficult to analyze. We will consider hybrid systems in which the continuous dynamics is given by a differential equation $\dot{x} = f(x)$, with discrete jumps $x' = r_i(x)$ which occur as soon as a guard condition $g_i(x) \geq 0$ is activated.

It is clear that the evolution of a hybrid system undergoes discontinuities, but since only continuous functions are computable, it is not clear to what extent, if any, it is possible to perform a rigorous analysis of a hybrid system. We will first show that we can define *lower* and *upper* semantics of evolution under which it is possible to compute reachable sets, and that away from discontinuity points (such as *grazing* or *corner collision* points), these semantics agree [2].

In order to perform reachability analysis, it is necessary to define the evolution over bounded initial sets of states. We show that this can be done using the operations of *range*, *compose*, *flow* and *solve* operations on functions. We will see that *constrained image sets* of the form

$$\{f(x) \mid x \in D \mid g(x) \in C\},$$

are sufficient to express the evolution exactly, except for the case of degenerate (non-transverse) crossings [3]. The flow operation is the most computationally demanding, and we will give some details of the implementation and efficiency considerations [4].

We will give examples of reachability analysis in ARIADNE, including electrical power converters and heating systems. Finally, we will outline some areas of active research, including differential inclusions [5] and modular reasoning.

References

[1] Davide Bresolin, Luca Geretti, Tiziano Villa, and Pieter Collins. An introduction to the verification of hybrid systems using ARIADNE. In Jan H. van Schuppen and Tiziano Villa, editors, *Coordination Control of Distributed Systems*, volume 456 of *Lecture Notes in Control and Information Sciences*, pages 339–346. Springer, 2015.

[2] Pieter Collins. Semantics and computability of the evolution of hybrid systems. *SIAM J. Control Optim.*, 49(2):890–925, 2011.

[3] Pieter Collins, Davide Bresolin, Luca Geretti, and Tiziano Villa. Computing the evolution of hybrid systems using rigorous function calculus. In *Proceedings of the 4th IFAC Conference on Analysis and Design of Hybrid Systems*, 2012.

[4] Luca Geretti, Davide Bresolin, Pieter Collins, and Tiziano Villa. Towards automation of reachability analysis for non-linear hybrid systems. Submitted to *Intern. J. Software Tools Tech. Transfer* special issue on Formal Methods and Automated Verification of Critical Systems, 2017.

[5] Sanja Živanović Gonzalez and Pieter Collins. Computing reachable sets of differential inclusions. In Jan H. van Schuppen and Tiziano Villa, editors, *Coordination Control of Distributed Systems*, volume 456 of *Lecture Notes in Control and Information Sciences*, pages 357–365. Springer, 2015.

Logical and computational aspects of Gleason's theorem in probability theory

Willem L. Fouché
Department of Decision Sciences,
School of Economic Sciences
University of South Africa
Pretoria

`To the memory of Johannes Heidema`

In this project, we explore the identification of the computational content and the implications of having constructive proofs for the important Gleason theorem in quantum logic [2] (1957). A constructive proof of a suitable classical reformulation of Gleason's theorem appears in Richman and Bridges [6] (1999).

We shall discuss some of the implications for quantum probability and logic of these meta-mathematical results. We also show that these results lead to interesting challenges of a combinatorial and computational nature.

Write E^n for the Hilbert space of dimension n over the reals and write S^{n-1} for the unit sphere in E^n .

In this case Gleason's theorem states that, if $n \geq 3$ and $p : S^{n-1} \to [0,1]$ is a frame function, meaning that p is in fact a function on the *rays* in H, (i.e., $p(-x) = p(x)$, for all relevant x), and for each frame (orthonormal basis) $f = \{e_i : 1 \leq i \leq n\} \subset S^{n-1}$, we have that $\sum_{\alpha \in f} p(\alpha) = 1$; then there is some density matrix (a quantum state) ρ on H such that $p(x) = (x, \rho x)$, for all $x \in S^{n-1}$.

A *(quantum) state* ρ or a *density matrix*, is a Hermitian, positive operator on H of trace 1. Being positive means that $(\rho x, x) \geq 0$ for all $x \in H$.

Thus a frame function p on S^{n-1} is necessarily a quadratic form of the form

$$p(x) = \Sigma_{i,j} r_{i,j} x_i x_j,$$

where $\rho = (r_{i,j})$ is a density matrix.

We shall explore the computational ramifications and physical implications of the following statement. The theorem is a consequence of the arguments in the paper by Hrushovski and Pitowsky[3].

Theorem 1 *We can algorithmically and explicitly construct a first-order statement* Π *in the theory* $\mathbf{R}$ *of real closed fields which is* `classically` *equivalent to Gleason's theorem for* E^n.

An analogous result holds for the first order theory $\mathbf{C}$ *for the field* $\mathbb{C}$ *of complex numbers and the version of Gleason's theorem for finite dimensional Hilbert spaces over* $\mathbb{C}$.

The theory $\mathbf{R}$ of a standard model of the reals $\mathbb{R}$, viewed as a field, can be axiomatised in a recursive manner if we look at $\mathbf{R}$ as the first order theory of $\mathbb{R}$ where the latter is also viewed as a totally ordered field via the order relation $>$ given by

$$x > 0 \leftrightarrow x \neq 0 \wedge \exists_y \, x = y^2.$$

As such it can be recursively axiomatised and admits quantifier elimitation and is therefore a decidable theory (Tarski).

Inspired by this theorem we formulate new problems of a Ramsey-theoretic nature for highly connected graphs towards understanding the complexity aspects of Gleason's theorem. The formulation of these problems was suggested by the deep combinatorial results in [4, 5] (1989, 2000) by Lovász, Saks and Schrijver. The author learnt about the significance of these combinatorial results for probability theory from the paper [1] by Abramsky and Brandenberger (2011).

References

[1] Abramsky, S., & Brandenberger, A. (2011) The Sheaf-Theoretic Structure of Non-Locality and Contextuality. New Journal of Physics, 13.

[2] Gleason, A. M. (1957). Measures on the closed subspaces of a Hilbert space. Journal of Mathematics and Mechanics, 6, 885-893.

[3] Hrushovski, E., & Pitowsky I. (2004). Generalizations of Kochen and Specker's theorem and the effectiveness of Gleason's theorem. Studies in History and Philosophy of Modern Physics, 35, 177-194.

[4] Lovász, L., Saks, M., & Schrijver, A. (1989). Orthogonal representations and connectivity of graphs. Linear Algebra and its Applications, 114, 439-454.

[5] Lovász, L., Saks, M. , & Schrijver, A. (2000). A correction: orthogonal representations and connectivity of graphs. Linear Algebra and its Applications, 313, 101-106.

[6] Richman, F., & Bridges, D. (1999). A constructive proof of Gleason's theorem. Journal of Functional Analysis, 162, 287-312.

Computing measures as a primitive operation

Christine Gaßner
University of Greifswald, Germany
gassnerc@uni-greifswald.de

Arno Pauly
Université libre de Bruxelles, Belgium
Arno.M.Pauly@gmail.com

Florian Steinberg
TU-Darmstadt, Germany
steinberg@mathematik.tu-darmstadt.de

We consider models of computation applicable to functions of type $f : \mathbb{R}^* \to \mathbb{R}^*$. Starting from a BSS-machine, we add as a primitive operator the ability to compute the Lebesgue measure of a given BSS-decidable subset of $[0,1]^d$. Let us call these machines BSS+λ machines. We will iterate this process once, i.e. add a primitive operator to compute the Lebesgue measure of a subset of $[0,1]^d$ decidable by a BSS+λ machine in order to introduce the BSS+λ+λ machines. As a third variant, we consider a machine that can use all computable functions as primitive operations, as well as $<$ as test. By adding the ability to compute the measure of a set decidable in this setting (which is just a Δ^0_2-set), we obtain the BSS+Cont+λ machines.

Alternatively, we can add the ability to compute a limit. The operator c-lim (for controlled limit) maps a program for a BSS-machine that computes a sequence of real numbers x_i with $|x_i, x_j| < 2^{-\min\{i,j\}}$ to the limit of this sequence. The operator u-lim (uncontrolled limit) accepts a program computing an arbitrary converging sequence of real numbers, and also outputs the limit. These operators correspond to the strongly analytic and to the weakly analytic machines going back to Hotz [3] – note that in our model c-lim and u-lim can be used multiple times, as opposed to only at the end as in analytic machines. By BSS+c-lim and BSS+u-lim we denote the respective machine models. For more detailed definitions of the models discussed so far, we refer to [4].

Our third model is a weakening of the non-deterministic Type-2 machines introduced by Ziegler [9] and further studied in [1]. We only require the model for $\mathbb{N}$ as advice space though: A weakly non-deterministic Type-2 machine with advice space $\mathbb{N}$ can guess some $n \in \mathbb{N}$ at the beginning of its computation, and then either output an infinite sequence which constitutes a valid solution to the computational task, or output finitely many symbols only which can then be arbitrary. For any input to the machine there has to be some $n \in \mathbb{N}$ that causes the machine to write an infinite output.

In [5] it was shown that BSS-machines and strongly analytic machines can be characterized by a complete Weihrauch degree, in the sense that every function computable in that model is Weihrauch reducible to the complete degree, and the complete degree has a representative computable in the model. In our context, we can establish an even stronger characterization:

Theorem 1. The following are equivalent for a function $f : \mathbb{R}^* \to \mathbb{R}^*$:

1. f is computable by a BSS+λ machine.
2. f is computable by a BSS+c-lim machine.
3. f is computable by a weakly non-deterministic Type-2 machine with advice space $\mathbb{N}$.

4. $f \leq_{\mathrm{W}} \Pi^0_2 - \mathrm{C}_{\mathbb{N}}$, where $\Pi^0_2 - \mathrm{C}_{\mathbb{N}} :\subseteq \Pi^0_2(\mathbb{N}) \rightrightarrows \mathbb{N}$ denotes Π^0_2-choice on $\mathbb{N}$.

The proof uses some results about higher-order choice principle in the Weihrauch lattice from [2].

Theorem 2. The following are equivalent for a function $f : \mathbb{R}^* \to \mathbb{R}^*$:

1. f is computable by a BSS+λ+λ machine.
2. f is computable by a BSS+u-lim machine.
3. f is computable by a BSS+Cont+λ machine.
4. $f \leq_{\mathrm{W}} \lim^{\diamond}$, with $^{\diamond}$ as defined in [5]

The techniques employed in proving these two theorems can also be adapted to address the effective relationship between measurability and integrability. Classically, any Borel measurable function $f : [0,1] \to [0,1]$ is integrable. Rather than dealing with all Borel measurable functions, we restrict our attention to the lowest non-trivial complexity and explore the Δ^0_2-measurable functions. These can be represented as the space $\mathcal{C}([0,1],[0,1]^{\nabla})$ as shown in [7, 6]. For a discussion of the space $\mathcal{L}_1([0,1],[0,1])$ of integrable functions we refer to [8].

Theorem 3. The following maps are Weihrauch equivalent:

1. $\lim :\subseteq (\mathbb{N}^{\mathbb{N}})^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$
2. $\lambda : \Delta^0_2([0,1]) \to \mathbb{R}$, $\lambda : \Delta^0_2([0,1]) \to \mathbb{R}_<$
3. $\int : \mathcal{C}([0,1],[0,1]^{\nabla}) \to \mathbb{R}$, $\int : \mathcal{C}([0,1],[0,1]^{\nabla}) \to \mathbb{R}_<$
4. $\mathrm{id} : \mathcal{C}([0,1],[0,1]^{\nabla}) \to \mathcal{L}_1([0,1],[0,1])$

References

[1] Vasco Brattka, Matthew de Brecht & Arno Pauly (2012): *Closed Choice and a Uniform Low Basis Theorem. Annals of Pure and Applied Logic* 163(8), pp. 968–1008.

[2] Vasco Brattka, Guido Gherardi, Rupert Hölzl, Hugo Nobrega & Arno Pauly. *Borel choice.* in preparation.

[3] Thomas Chadzelek & Günter Hotz (1999): *Analytic Machines. Theoretical Computer Science* 219, pp. 151–167.

[4] Christine Gaßner (2017). *Deterministic Operators for BSS RAM's I (extended abstract).* Preprint-Reihe Ag Berechenbarkeitstheorie über algebraischen Strukturen Greifswald. Available at `https://math-inf.uni-greifswald.de/fileadmin/uni-greifswald/fakultaet/mnf/mathinf/gassner/publikationen_/pdf_dateien/GassnerDeterministicOracles2017WithoutProofsI.pdf`.

[5] Eike Neumann & Arno Pauly (2016). *A topological view on algebraic computations models.* arXiv:1602.08004.

[6] Arno Pauly & Matthew de Brecht (2013). *Towards Synthetic Descriptive Set Theory: An instantiation with represented spaces.* http://arxiv.org/abs/1307.1850.

[7] Arno Pauly & Matthew de Brecht (2014): *Non-deterministic Computation and the Jayne Rogers Theorem. Electronic Proceedings in Theoretical Computer Science* 143. DCM 2012.

[8] Marian Pour-El & Ian Richards (1989): *Computability in analysis and physics.* Perspectives in Mathematical Logic. Springer.

[9] Martin Ziegler (2005): *Computability and continuity on the real arithmetic hierarchy and the power of type-2 nondeterminism.* In: Barry S. Cooper, Benedikt Löwe & Leen Torenvliet, editors: *Proceedings of CiE 2005, LNCS* 3526, Springer, pp. 562–571.

Complexity of some real numbers and functions with respect to the subrecursive class $\mathcal{M}^2$

Ivan Georgiev

Prof. d-r Asen Zlatarov University,
Faculty of Natural Sciences,
Prof. Yakimov Str. 1, 8010, Burgas, Bulgaria,
ivandg@yahoo.com

Abstract. The class $\mathcal{M}^2$ is a low subrecursive class of total functions in the natural numbers, contained in the third level $\mathcal{E}^2$ of Grzegorczyk's hierarchy. By choosing suitable representations of the real numbers and functions, we can evaluate their complexity with respect to the class $\mathcal{M}^2$. For example, results in [4] show that the numbers e and π are $\mathcal{M}^2$-computable and more generally all elementary functions of calculus preserve $\mathcal{M}^2$-computability of real numbers. Concerning real functions, there are some general reasons to consider two notions for relative computability - one uniform notion, based on a definition from [1] and one non-uniform notion, which we call conditional computabilty of real functions. Results from [3] show that all elementary functions of calculus are conditionally computable with respect to $\mathcal{M}^2$ and their restrictions to compact sets are uniformly computable with respect to $\mathcal{M}^2$. The aim of this talk is to go beyond elementary and study the complexity of the integration operator on real functions. Similar studies, but with respect to polynomial-time computability have much longer history (see Section 5.4 in [2]). Another paper, which has a close subject is [5], but the authors essentially use the bounded summation operation, which is not available in $\mathcal{M}^2$. Our main theorem is that the definite integral of a uniformly $\mathcal{M}^2$-computable analytic real function with $\mathcal{M}^2$-computable limits is itself $\mathcal{M}^2$-computable. We generalise this result to integrals with parameters and with varying limits, as well as to improper integrals. As an application, we show that the Euler-Mascheroni constant is $\mathcal{M}^2$-computable, which is an open problem from [4]. We also show some results on the uniform and conditional $\mathcal{M}^2$-computability of the gamma function and the Riemann zeta function.

Keywords: uniformly computable real functions, conditionally computable real functions, the subrecursive class $\mathcal{M}^2$, integration

References

1. Grzegorczyk, A., Computable functionals. Fund. Math., vol. 42 (1955), 168–202.
2. Ko, Ker-I, Complexity Theory of Real Functions, Progress in Theoretical Computer Science, Birkhäuser, Boston, 1991.
3. Skordev, D., Georgiev, I., On a relative computability notion for real functions. CIE Proceedings, Lecture Notes in Computer Science, vol. 6735 (2011), 270–279.
4. Skordev, D., Weiermann, A., Georgiev, I., $\mathcal{M}^2$-computable real numbers. Journal of Logic and Computation, vol. 22(4) (2012), 899–925.
5. Tent, K., Ziegler M., Computable functions of reals, Münster J. Math., vol. 3 (2010), 43–66.
6. Trefethen, L., Weideman J. A. C., The exponentially convergent trapezoidal rule. Society for Industrial and Applied Mathematics, vol. 56(3) (2014), 385–458.

Compact representations and orbit complexity

Robert Kenny

Orbit complexity, introduced for compact dynamical systems by A. A. Brudno [Bru82] and later developed on separable metric spaces by Galatolo [Gal00], in a sense measures the algorithmic information needed to encode initial segments of an orbit $\xi = (T^i x)_{i\in\mathsf{N}}$ up to a finite precision. In both approaches it mimics the definition(s) of topological entropy of T, and bears also a certain resemblance to characterisations of effective Hausdorff and packing dimensions by means of Kolmogorov complexity; for subshifts this can be made quite precise, see [Sim15], while in the metric space setting we may take the definitions of [Gal99] as a starting point.

Referring to the algorithmic dimensions just mentioned as 'point complexities', and following an idea of [Zie12], we introduce lower and upper orbit complexities $\underline{\mathcal{K}}$, $\bar{\mathcal{K}}$ and lower and upper point complexities $\underline{C}$, $\bar{C}$ with respect to representations δ in the sense of computable analysis (see [Wei00], although we use Baire space N^N for names). For instance, we define

$$\bar{\mathcal{K}}(\xi,\delta) := \sup_{\epsilon>0} \limsup_{n\to\infty} \frac{1}{n} \min\{C(p) \mid p = \langle p^{(0)},\dots,p^{(n-1)}\rangle^{(n)} \wedge (\forall i<n) p^{(i)} \in \delta^{-1}B(x_i;\epsilon)\},$$

where $\xi = (x_i)_{i\in\mathsf{N}} \in X^\mathsf{N}$ and[1] $C(p) := \sup_{N\in\mathsf{N}} C(p \upharpoonright N|N)$. It is worth noting some obstacles here, for instance in the '$\limsup_{\epsilon\searrow 0}$' defining upper point complexity $\bar{C}(x,\delta)$, one can equivalently take a sequence $(\epsilon_n)_{n\in\mathsf{Z}^+}$ such that $\frac{-\log_2\epsilon_n}{n}$ has a cluster point in $(0,\infty)$, whereas this is not known for $\underline{C}(x,\delta)$; consequently it may be more useful to change the definition to $\underline{C}_R$ where we constrain the radii ϵ to lie in a specified set $R \subseteq (0,\infty)$ with $0 \in \overline{R}$. Still, point complexities with respect to Cauchy representations ρ_ν and with respect to their defining dense partial sequences ν are equal, and we show a number of further desirable properties of the definitions.

More specifically: several reducibilities $(\leq_\mathrm{r}) = (\leq_\mathrm{cb}), (\leq_\mathrm{a}), (\leq'_\mathrm{a}), (\leq''_\mathrm{a})$ on

$$\mathcal{R}^X := \{\delta \mid \delta \text{ a representation of } X\}$$

are introduced, the simplest two of which are preorders. For each monotone $f : \mathsf{N}^* \to \mathsf{N}^*$, define $f_\omega :\subseteq \mathsf{N}^\mathsf{N} \to \mathsf{N}^\mathsf{N}$ by taking $\operatorname{dom} f_\omega$ as the set of infinite branches of N^* on which f is unbounded, $f_\omega(p) := \sup_{n\in\mathsf{N}} f(p\upharpoonright n)$ for all $p \in \operatorname{dom} f_\omega$ (see [Wei00, Defn 2.1.10(2)]). Also denote by $R^{(1)} \subseteq \mathsf{N}^\mathsf{N}$ the total computable functions $\mathsf{N}\to\mathsf{N}$, and the lookahead (or use function) of f by

$$\check{f} :\subseteq \mathsf{N}^\mathsf{N}\times\mathsf{N}\to\mathsf{N}, (p,m)\mapsto \mu l\,(|f(p\upharpoonright l)|\geq m) \quad (\operatorname{dom}\check{f} = \operatorname{dom} f_\omega\times\mathsf{N}).$$

Then we define $\delta_0 \leq_\mathrm{cb} \delta_1$ if there exists computable monotone f with f_ω a (δ_0,δ_1)-realiser of id_X also satisfying $(\forall p\in\operatorname{dom}\delta_0)(\exists b\in R^{(1)})(\forall m)(\check{f}(p,m)\leq b(m))$, and

$$\delta_0 \leq_\mathrm{a} \delta_1 :\iff (\forall\epsilon>0)\,(T_\epsilon : X \rightrightarrows X, x\mapsto\{y\mid d(x,y)<\epsilon\} \text{ is } (\delta_0,\delta_1)\text{-computable}).$$

Next, the (distributive lattice or upper semilattice) structure of $\mathcal{R}^X/\equiv_\mathrm{r}$ for $(\leq_\mathrm{r}) = (\leq_\mathrm{cb}), (\leq_\mathrm{a})$ is examined, and the inequalities are derived stating that $\underline{C}_{2R}(x,\delta_1) \leq \underline{C}_R(x,\delta_0)$ and $\bar{C}_{2R}(x,\delta_1) \leq \bar{C}_R(x,\delta_0)$ when $\delta_0 \leq_\mathrm{cb} \delta_1$.

The situation for nonincrease of orbit complexity under reducibility is more complicated. First, let $\delta_0 \leq''_\mathrm{a} \delta_1$ if for each $\epsilon>0$ there exists computable f such that $\operatorname{dom}\delta_0 \subseteq f_\omega^{-1}\operatorname{dom}\delta_1$ and $B_d(\delta_0(p);\epsilon) \ni (\delta_1\circ f_\omega)(p)$ for all $p\in\operatorname{dom}\delta_0$ with $\check{f}(p,m)\leq c(m,p\upharpoonright N(m))$ for all $p\in\operatorname{dom}\delta_0$, $m\in\mathsf{N}$, where $N:\mathsf{N}\to\mathsf{N}$ and $c :\subseteq \mathsf{N}\times\mathsf{N}^*\to\mathsf{N}^*$ are computable and independent of p, m.

[1] for this purpose we use conditional plain Kolmogorov complexity for strings over N.

Similarly let $\delta_0 \leq'_{\mathrm{a}} \delta_1$ if $\delta_0 \leq''_{\mathrm{a}} \delta_1$ via f, c, N with $N(m) = 1$ for all $m \in \mathsf{N}$. Then, denoting by δ' the representation of X^* defined by

$$n, \langle p^{(0)}, \ldots, p^{(n-1)} \rangle \in (\delta')^{-1}\{w\} :\iff (n = 0 \wedge w = \lambda) \vee \Big(n \geq 1 \wedge (\forall i < |w|)(p^{(i)} \in \delta^{-1}\{w_i\}) \Big),$$

we have:

Lemma 1. *If $\delta_i \in \mathcal{R}^X$ $(i < 2)$ have $\delta_0 \leq''_a \delta_1$ and $\operatorname{dom}\delta_0$ is co-c.e. compact then $\delta'_0 \leq'_a \delta'_1$.*

Proposition 1. *If X is a metric space, $\delta_i \in \mathcal{R}^X$ $(i < 2)$ with $\delta'_0 \leq'_a \delta'_1$ then any $\xi \in X^{\mathsf{N}}$ has $\underline{\mathcal{K}}(\xi, \delta_1) \leq \underline{\mathcal{K}}(\xi, \delta_0)$ and $\bar{\mathcal{K}}(\xi, \delta_1) \leq \bar{\mathcal{K}}(\xi, \delta_0)$.*

Still, one can do the following with $\leq_{\mathrm{a}}$ and related notions (most of which extends naturally to quasi-compact $\mathbf{T}_0$ spaces). First, for quasi-compact X, define the class $\mathcal{R}_{\mathrm{c}}^X$ consisting of δ such that, for each $r \in \mathsf{N}$, $[r] := \{1, \ldots, r\}$ and each finite open cover $(U_i)_1^r$, the operation $R : X \rightrightarrows [r], x \mapsto \{i \mid U_i \ni x\}$ is $(\delta, \delta_{\mathsf{N}}|^{[r]})$-computable. Also define

$$\mathcal{R}_{\mathrm{d}}^X := \{\delta \in \mathcal{R}^X \mid \overline{\delta(R^{(1)} \cap \operatorname{dom}\delta)} = X\}.$$

Then $\mathcal{R}_{\mathrm{d}}^X$ is $\leq_{\mathrm{a}}$-closed upward, $\mathcal{R}_{\mathrm{c}}^X$ is $\leq_{\mathrm{a}}$-closed downwards and closed under $\sqcup$ (so $\mathcal{R}_{\mathrm{c}}^X/\equiv_{\mathrm{a}}$ is an ideal in $\mathcal{R}^X/\equiv_{\mathrm{a}}$), and $\mathcal{R}_{\mathrm{c}}^X \cap \mathcal{R}_{\mathrm{d}}^X$ consists of at most one $\equiv_{\mathrm{a}}$-class (further results on the Boolean combinations of the sets $\mathcal{R}_{\mathrm{c}}$, $\mathcal{R}_{\mathrm{d}}$ will also be presented). In particular, among $\leq_{\mathrm{a}}$-degrees of Cauchy representations ρ_ν (for compact metric spaces), there is a least degree, wherein the orbit complexity with respect to ν is maximised for each $\xi \in X^{\mathsf{N}}$ (this generalises two results of [Gal00]). For the case of representations, we show any $\delta \in \mathcal{R}_{\mathrm{d}}^X$ and $\xi \in X^{\mathsf{N}}$ have orbit complexity bounded above by the Brudno orbit complexity $\bar{\mathcal{K}}(\xi)$, and present some results on bounding the orbit complexity below by this quantity for certain representations $\delta \in \mathcal{R}_{\mathrm{c}}^X$.

Theorem 1. *If X is a compact metric space and $\delta \in \mathcal{R}_c^X$ is proper (i.e. $\operatorname{dom}\delta$ is compact), then $\underline{\mathcal{K}}(\xi) \leq \underline{\mathcal{K}}(\xi, \delta)$ and $\bar{\mathcal{K}}(\xi) \leq \bar{\mathcal{K}}(\xi, \delta)$ for all $\xi \in X^{\mathsf{N}}$.*

Finally, we observe the following properties of orbit complexity, related to semicontinuity:

Proposition 2. *For any compact metric space X with $\delta \in \mathcal{R}^X$, $\epsilon, \eta > 0$ and sequence $(\xi^{(i)})_i \subseteq X^{\mathsf{N}}$ convergent to ξ with respect to d_∞, we have $\underline{\mathcal{K}}_{\epsilon+\eta}(\xi, \delta) \leq \liminf_{i\to\infty} \underline{\mathcal{K}}_\epsilon(\xi^{(i)}, \delta)$ and $\bar{\mathcal{K}}_{\epsilon+\eta}(\xi, \delta) \leq \limsup_{i\to\infty} \bar{\mathcal{K}}_\epsilon(\xi^{(i)}, \delta)$. Consequently, $\underline{k} : (X^{\mathsf{N}}, d_\infty) \to [0, \infty], \xi \mapsto \underline{\mathcal{K}}(\xi, \delta)$ is lower semicontinuous, and $\bar{k} : X^{\mathsf{N}} \to [0, \infty], \xi \mapsto \bar{\mathcal{K}}(\xi, \delta)$ has each $\bar{k}^{-1}(\kappa, \infty]$ empty or dense-in-itself $(0 < \kappa < \infty)$.*

References

[Bru82] A. A. Brudno. Entropy and the complexity of the trajectories of a dynamic system. *Trudy Moskov. Mat. Obshch.*, 44:124–149, 1982.

[Gal99] S. Galatolo. Pointwise information entropy for metric spaces. *Nonlinearity*, 12(5):1289–1298, 1999.

[Gal00] S. Galatolo. Orbit complexity by computable structures. *Nonlinearity*, 13(5):1531–1546, 2000.

[Sim15] Stephen G. Simpson. Symbolic dynamics: entropy = dimension = complexity. *Theory Comput. Syst.*, 56(3):527–543, 2015.

[Wei00] K. Weihrauch. *Computable analysis*. Springer-Verlag, Berlin, 2000.

[Zie12] Martin Ziegler. Real computation with least discrete advice: a complexity theory of nonuniform computability with applications to effective linear algebra. *Ann. Pure Appl. Logic*, 163(8):1108–1139, 2012.

Automatic Randomness Tests

Birzhan Moldagaliyev

Department of Mathematics
National University of Singapore

Abstract

In this paper we define a notion of automatic randomness tests (ART) which capture measure theoretic typicalness of infinite binary sequences within the framework of automata theory. An individual ART is found to be equivalent to a deterministic Büchi automaton recognizing ω-language of (Lebesgue) measure zero. A collection of ART's induce a notion of automatic random sequence. We provide a purely combinatorial characterization of an automatic random sequence in the form of a disjunctive property for sequences.

Introduction The theory of algorithmic randomness tries to explain what kind of properties make an individual element of a sample space to appear random. Mostly the theory deals with infinite binary sequences. There are three main paradigms to study randomness of an object:

1. Unpredictability
2. Incompressibility
3. Measure theoretical typicalness

First two paradigms were explored quite extensively in the context of automata theory. What seems to be lacking, though, is a study of automatic randomness from the measure theoretical typicalness point of view, which is our purpose.

Definitions Let us provide main definitions of our study. Let $\mathcal{U} = (U_i)_{i \in I}$ be an automatic family, a uniformly regular collection of languages. We say that $\mathcal{U}$ forms an *automatic randomness test (ART)* if

$$\liminf_{i \in I} \mu[U_i] = 0$$

In above definition $[U]$ refers to the collection of infinite sequences extending words in a language U. Furthermore, μ refers to Lebesgue measure on Cantor space. Corresponding nullsets are defined in the manner of Martin-Löf randomness tests. Given an ART $\mathcal{U} = (U_i)_{i \in I}$, let

$$F(\mathcal{U}) = \bigcap_{i \in I} [U_i]$$

be its *covering region*. An infinite sequence X is said to be covered by $\mathcal{U}$ if it belongs to the covering region of $\mathcal{U}$, i.e. $X \in F(\mathcal{U})$. A pair of ART's $(\mathcal{U}, \mathcal{V})$ is said to be equivalent if $F(\mathcal{U}) = F(\mathcal{V})$. Finally, we define a notion of a random sequence in parallel with the original definition by Martin-Löf: an infinite sequence X is said to be *automatic random (AR)* if X is not covered by any ART.

Properties Given above definitions, there are some resulting implications.

Theorem 1. *Let $\mathcal{C}$ be a class of the Cantor space, $\{0,1\}^{\mathbb{N}}$. The following are equivalent:*

1. *There is ART $\mathcal{U}$ such that $\mathcal{C} = F(\mathcal{U})$.*
2. *There is a deterministic Büchi automaton of measure zero such that $\mathcal{C} = L(M)$*

Theorem 2 (Characterization)**.** *Given an infinite sequence X the following are equivalent:*

1. *$X \in F(\mathcal{U})$ for some ART $\mathcal{U}$*
2. *X is accepted by a deterministic Büchi automaton of measure zero.*
3. *X is accepted by a deterministic Muller automaton of measure zero.*
4. *X is not a disjunctive sequence*

Theorem 3 (Combinatorial characterization of AR)**.** *An infinite sequence X is automatic random (AR) if and only if it is disjunctive.*

Theorem 4. *Suppose that a class $\mathcal{C}$ satisfies $\mathcal{C} = F(\mathcal{U})$ for some ART $\mathcal{U}$. Then there is an ART $\mathcal{V} = (V_j)_{j \in J}$ subsuming $\mathcal{U}$, such that $\mu[V_j] \leq \gamma^{|j|}$ for some $\gamma < 1$.*

Discussion We have defined and investigated properties of randomness tests in the context of automata theory. These investigations led to quite unexpected connections between randomness tests and ω-automata such as by Büchi and Muller. Furthermore, a purely combinatorial characterization of automatic randomness in the form of disjunctive property for sequences has been found. We believe that many more notions from the theory of algorithmic randomness could find their counterparts in the context of automata theory.

Continuous Enclosures of Discontinuous Problems II

Eike Neumann
Aston University, Birmingham, UK
neumaef1@aston.ac.uk

The main theorem of computable analysis asserts that any computable function is continuous. Many interesting problems encountered in practice turn out to be discontinuous, however. It hence is a natural question how much partial information may be obtained on the solutions of a discontinuous problem in a continuous or computable way. In [6] the notion of "partial information" was formalised based on approximation by compact sets in the upper Vietoris topology. This approach seemed to yield reasonable results for problems with locally compact space of solutions, but failed to faithfully reflect the computational content of certain discontinuous problems in functional analysis, such as finding fixed points of nonexpansive maps in separable Hilbert space. We develop an extension of this approach which is equivalent to our previous approach in the locally compact case, but which yields better results in general.

We will work in the category of *lower correspondences* on qcb_0-spaces. A lower correspondence $\Phi\colon X \rightsquigarrow Y$ is just a continuous map $\Phi\colon X \to \mathscr{V}(Y)$. Composition is defined by using that $\mathscr{V}$ is a monad. On second-countable Hausdorff spaces lower correspondences are essentially the same as computable multimaps [1], but the relationship is more subtle in general (see also [7]). If $\Phi\colon X \rightsquigarrow Y$ and $\Psi\colon X \rightsquigarrow Y$ are lower correspondences, we write $\Phi \leq \Psi$ if for all $x \in X$ we have $\Phi(x) \leq \Psi(x)$ with respect to the specialisation order. These definitions extend to relations $f\colon X \rightrightarrows Y$ with closed values by viewing them as (potentially discontinuous) functions $f\colon X \to \mathscr{V}(Y)$. Partial information on points of a qcb_0-space is naturally encoded in a corresponding *computable semilattice*:

Definition 1. An *effective semilattice* L is a qcb_0-space which admits compact meets and bounded overt joins with respect to the specialisation order. A *computable semilattice* is an effective semilattice where compact meets and bounded overt joins are uniformly computable. More explicitly, we require that the maps $\inf\colon \mathscr{K}(L) \to L$ and $\sup\colon \subseteq \mathscr{V}(L) \to L$, with $\mathrm{dom}(\sup) = \{A \in \mathscr{V}(L) \mid \exists x \in L.\, (x \geq A)\}$, be well-defined and computable.

The prototypical example of a computable semilattice is the space $\mathscr{K}^*_\bot(X)$ of non-empty compacts of a Hausdorff qcb-space X with a bottom element added. Our main definition is:

Definition 2. Let $f\colon X \rightrightarrows Y$ be a relation with closed values. An *enclosure* of f consists of two maps $F\colon X \rightsquigarrow L$ and $\xi_F\colon Y \to L$, where L is a computable semilattice, such that $\xi_F \circ f \geq F$. Let $F\colon X \rightsquigarrow L$ and $G\colon X \rightsquigarrow M$ be enclosures of f. Then F *tightens* G if there exists a map $\Phi\colon L \to M$ with $\Phi \circ F \geq G$ and $\Phi \circ \xi_F \leq \xi_G$.

The definition given in [6] corresponds - up to adding a bottom element - to the special case where L is always chosen to be $\mathscr{K}^*(Y)$. If $\xi_L\colon Y \to L$ is a map, then the set of all enclosures $F\colon X \rightsquigarrow L$ with $F \leq \xi_L \circ f$ has a greatest element, called the *principal enclosure* for (L, ξ_L). In the case where Y is locally compact Hausdorff we obtain a stronger result:

Theorem 3. *If Y is a locally compact Hausdorff space, then any relation $f\colon X \rightrightarrows Y$ has a tightest enclosure. It is given by the principal enclosure for $\left(\mathscr{K}^*(\hat{Y}), \imath\right)$ where $\hat{Y}$ is the one-point compactification of Y and $\imath\colon Y \to \mathscr{K}^*(\hat{Y})$ is induced by the usual embedding $i\colon Y \to \hat{Y}$.*

Theorem 3 can for instance be used to establish that the tightest enclosure of the problem of finding fixed points of continuous self-maps of the unit cube $[0,1]^n \subseteq \mathbb{R}^n$ is given by the *component cover representation* of the fixed point set, which was introduced by COLLINS [3] (see [4, 2] for related work).

Relations whose co-domain is not locally compact can be studied with the help of a certain refinement of strong Weihrauch equivalence: If we have maps $\alpha\colon X \rightsquigarrow A$, $\alpha'\colon A \rightsquigarrow X$, $\beta\colon B \to Y$, and $\beta'\colon Y \to B$ such that $\beta \circ g \circ \alpha \leq f$ and $\beta' \circ f \circ \alpha' \leq g$, then under certain additional assumptions on these maps we obtain a Galois connection between the principal enclosures of f and g. This can be used to calculate tightest enclosures of certain problems (and to prove their existence) by establishing a Galois connection with the enclosures of problems whose tightest enclosure is easy to calculate. We give two examples:

Theorem 4. *Consider the problem of finding a fixed point of a nonexpansive self-map of the unit ball in separable Hilbert space:*

$$\texttt{fix}\colon\ \subseteq C(B_{\ell^2}, B_{\ell^2}) \rightrightarrows B_{\ell^2},$$

where $\mathrm{dom}(\texttt{fix}) = \{f\colon B_{\ell^2} \to B_{\ell^2} \mid ||f(x) - f(y)|| \leq ||x - y||\}$. *Then the tightest enclosure is given by the correspondence which sends a nonexpansive map f to the weakly compact set of all its fixed points in the upper Vietoris topology induced by the weak topology on ℓ^2.*

The operator `fix` is well-defined thanks to the Browder-Göhde-Kirk theorem. The proof of Theorem 4 is a refinement of the proof of the strong Weihrauch equivalence of `fix` and a certain convex choice operator, which was established in [5].

Theorem 5. *Let $I \subseteq \mathbb{R}$ be an open interval. Consider the problem of numerical differentiation*

$$d\colon\ \subseteq C(I) \to C(I),\ f \mapsto f',$$

with $\mathrm{dom}(d) = C^1(I)$. *Then the tightest enclosure is given by the map which sends a function f to its L^1-weak derivative, represented in the singular representation of L^1, which was recently introduced by* STEINBERG *[10, Definition 2.4].*

The "singular" weak derivative in Theorem 5 contains strictly less information than the L^1-weak derivative, but strictly more information than the distributional derivative.

References

[1] V. Brattka and P. Hertling. Continuity and computability of relations. Technical report, FernUniversität Hagen, 1994.

[2] V. Brattka, S. Le Roux, J. S. Miller, and A. Pauly. Connected Choice and the Brouwer Fixed Point Theorem. *arXiv:1206.4809*, 2016.

[3] P. Collins. Computability and Representations of the Zero Set. *Electronic Notes in Theoretical Computer Science*, 221:37–43, 2008.

[4] J. Miller. Π^0_1*-classes in computable analysis and topology*. PhD thesis, Cornell University, 2002.

[5] E. Neumann. Computational problems in metric fixed point theory and their Weihrauch degrees. *Logical Methods in Computer Science*, 11(4), 2015.

[6] E. Neumann. Continuous enclosures and best approximations of discontinuous operations. In *Proceedings of the Twelfth International Conference on Computability and Complexity in Analysis (CCA 2015)*, pages 47–49, 2015.

[7] A. Pauly and M. Ziegler. Relative computability and uniform continuity of relations. *Journal of Logic and Analysis*, 5:1–39, 2013.

[8] M. Schröder. *Admissible Representations for Continuous Computations*. PhD thesis, FernUniversität Hagen, 2002.

[9] M. Schröder. Extended admissibility. *Theoretical Computer Science*, 284:519–538, 2002.

[10] F. Steinberg. Complexity theory for spaces of integrable functions. *arXiv:1612.06419*, 2017.

Evaluation strategies over continuous real functions

Eike Neumann Michal Konečný
Aston University, Birmingham, United Kingdom
{neumaef1,m.konecny}@aston.ac.uk

In theory (exact) continuous real functions often feature as first-class objects in computation, using a natural admissible representation of the space of continuous real functions. While all natural admissible representations on this space are computably equivalent, some are more suitable than others for practical computation. In particular, they may fail to be equivalent under the second-order polytime reduction by Kawamura and Cook [1]. We search for a data type of continuous functions suitable for practical computation and develop tools for comparing such data types.

In [4], we compared several representations of continuous functions $[-1,1] \to \mathbb{R}$ with respect to their polytime reducibility and practical performance on several benchmarks. We considered Cauchy sequences of (dyadic) polynomial, piecewise polynomial, rational and piecewise rational approximations, denoted Poly, PPoly, Frac and PFrac, respectively, as well as the representation traditionally used in computable analysis [6], denoted Fun. Building on [5, 2], we proved the following second-order polytime reductions: $\mathrm{Poly} < \mathrm{PPoly} \equiv \mathrm{Frac} < \mathrm{Fun}$.

In our benchmarks [4], we evaluated terms such as $\sin(10x + \sin(20x^2))$ in chosen representations, using best-effort implementations of operations such as pointwise multiplication, division and sine of continuous functions and then calculated the maximum and integral of the function over its domain $[-1,1]$. Typically, a variation of Fun did best in computing the maxima while a variation of PPoly did best in computing the integrals.

Fun performs well on maximisation because it supports local evaluation to a high accuracy, whereas Poly and PPoly, as implemented will enforce a high-accuracy evaluation over the whole domain. Inspired by this observation, we have added representations called LPoly, LPPoly and LFrac that *locally* use Poly, PPoly and Frac, respectively. While polytime equivalent to their counterparts, these representations consistently win in our benchmarks.

We have also developed some concepts to allow us to interpret what happens in the benchmarks and to draw appropriate conclusions from them. First, observe that the following two performance aspects of representations are sometimes in conflict:

- Ease of creating names for values ("writing names")
- Ease of extracting information about values from names ("reading names")

Ease of writing can be formalised as "easy" (*e.g.*, polytime) computability of a set of functions from scratch. The larger the set, the better. Ease of reading can be formalised as "easy" (*e.g.*, polytime) computability of functionals such as maximisation and integration that take a name of an object such as a continuous function and produce a result of a different type, such as a real number. Operations such as pointwise multiplication of functions involve both reading and writing of names.

Reducibility of names facilitates comparison of reading and writing in opposite directions:

- If representation A "easily" reduces to B but not vice versa, then A is "no worse" than B at reading and B is "no worse" than A at writing.
- If two representations are equivalent with respect to "easy" reducibility, then they are also equivalent in their "ease" of reading and writing.

These statements can be formalised using second-order polytime reducibility and computability. For example, in Fun it is "easier" than in Poly to write names because each Poly name can be polytime translated to a Fun name by evaluating the polynomial. On the other hand, in Poly it is strictly "easier" than in Fun to read names as we can compute maximisation and integration in polytime using Poly names but not Fun names [3]. Each representations offers a trade-off between these two aspects.

As both reading and writing is important, we should optimise for both. Thus we seek representations that are Pareto-optimal with respect to a set of relevant reading, writing and mixed operations. We formalise a term language over some signature of operations over a number of types, including the type of continuous functions. An **evaluation strategy** for a signature consists of representations for the types and algorithms for all the signature operations.

We reformulate some existing results as Pareto-optimality (ie maximality) of certain evaluation strategies over certain signatures. For example, with a certain comparison partial order on evaluation strategies, we have the following three results:

- Fun is maximal for the signature comprising constants for *all* computable real functions, function evaluation, range and common arithmetic operations.
- Poly and PPoly are maximal for the signature comprising constants for computable real *analytic* functions, function evaluation, range and common arithmetic operations. Moreover, Poly and PPoly dominate Fun on this signature.
- PPoly is maximal for the signature comprising constants for all computable real analytic functions, *pointwise maximum of functions*, function evaluation, range and common arithmetic operations. Moreover, PPoly dominates Poly and Fun on this signature.

These results in a way formalise the observation that Fun is better than Poly at writing and Poly and PPoly are better than Fun at reading.

References

[1] A. Kawamura and S. A. Cook. Complexity theory for operators in analysis. *ACM Transactions on Computation Theory*, 4(2):5, 2012.

[2] A. Kawamura, N. Müller, C. Rösnick, and M. Ziegler. Computational benefit of smoothness: Parameterized bit-complexity of numerical operators on analytic functions and Gevrey's hierarchy. *Journal of Complexity*, 31(5):689 – 714, 2015.

[3] K.-I. Ko and H. Friedman. Computational complexity of real functions. *Theoretical Computer Science*, 20:323–352, 1982.

[4] M. Konečný and E. Neumann. Representations for feasibly approximable functions. In *Proceedings of the Thirteenth International Conference on Computability and Complexity in Analysis (CCA 2016)*, pages 27–29, 2016.

[5] S. Labhalla, H.Lombardi, and E.Moutai. Espaces métriques rationnellement présentés et complexité, le cas de l'espace des fonctions réelles uniformément continues sur un intervalle compact. *Theoretical Computer Science*, 250:265–332, 2001.

[6] K. Weihrauch. *Computable Analysis - An Introduction.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2000.

Computable Operations on Compact Subsets of Metric Spaces with Applications to Fréchet Distance and Shape Optimization

Chansu Park[1], Ji-Won Park[1], Sewon Park[1], Dongseong Seon[2] and Martin Ziegler[1]

[1]KAIST School of Computing [2]Seoul National University,

The present work [PPP*17] generalizes the Theory of Computation from Euclidean unit cubes to compact metric spaces (X,d). Being separable, computing here naturally means approximation up to error 2^{-n} by a sequence (of indices w.r.t. a fixed partial enumeration $\xi :\subseteq \mathbb{N} \to X$) of some countable dense subset, thus generalizing the dyadic rationals $\mathbb{D} = \{a/2^n : a,n \in \mathbb{Z}\}$ canonically employed the real case. Of course the particular choice of said enumeration ξ heavily affects the computational properties it induces [BrPr03, Schr04].

Definition 1. *Fix a compact metric space (X,d).*

a) (X,d,ξ,D) is a presented (compact) metric space *if $\xi :\subseteq \mathbb{N} \to X$ is a partial dense enumeration and $D : \mathbb{N} \to \mathbb{N}$ strictly increasing such that, for every $m \in \mathbb{N}$, the closed balls $\overline{B}\big(\xi(u), 2^{-m-1}\big)$ with $u \in [2^{D(m)}] \cap \mathrm{dom}(\xi)$ cover X. (X,d,ξ,D) is* computably compact *if $\mathrm{dom}(\xi)$ and $D : \mathbb{N} \to \mathbb{N}$ are recursive and (X,d,ξ) constitutes a computable metric space* [BrPr03].

b) For another computable metric space (Y,e,υ), Computing *a relation $f \subseteq X \times Y$ means to convert any sequence $(u_m) \in \mathrm{dom}(\xi)$ with $u_m < 2^{D(m)}$ and $d\big(\xi(u_m),x\big) \leq 2^{-m}$ for some $x \in \mathrm{dom}(f) = \{x \mid \exists y : (x,y) \in f\}$ to a sequence $v_n \in \mathrm{dom}(\upsilon)$ with $e\big(\upsilon(v_n),y\big)$ for all n and some y with $(x,y) \in f$. The computation runs in time $T(n)$ if v_n appears after at most $T(n)$ steps, regardless of x and (u_m).*

c) For presented (X,d,ξ,D), a name *of compact non-empty $W \subseteq X$ is a sequence $\bar{A} = (A_m)$ of finite sets $A_m \subseteq [2^{D(m)}] \cap \mathrm{dom}(\xi)$ such that, for every $m \in \mathbb{N}$, the set $\xi[A_m] \subseteq X$ has Hausdorff distance at most 2^{-m} to W. W is* computable *if it has a name $\bar{A} = (A_m)$ which is uniformly recursive in the sense that the set $\prod_m \{m\} \times A_m \subseteq \mathbb{N} \times \mathbb{N}$ is decidable.*

These conditions allow us to (i) turn the hyper-space $\mathcal{K}(X)$ of non-empty compact subsets of X, equipped with the Hausdorff distance, into computably compact metric space, again; and (ii) similarly for the (by Arzelà-Ascoli compact) space $\mathcal{C}_\mu(W,Y)$ of partial equicontinuous functions $\Lambda : W \to Y$ having non-empty computable compact domain $W \subseteq X$ and recursive modulus of continuity μ. The latter proceeds by identifying such Λ with $\mathrm{graph}(\Lambda) \in \mathcal{K}(X \times Y)$ [Bra05]. We can thus generalize well-known results regarding computable function pre/images for the Euclidean, to arbitrary compact metric, spaces – including a hierarchy of higher types; which allows us to assert the computability of (iii) Fréchet Distances between curves / loops, as well as of (iv) constrained/Shape Optimization.

References

[AlBu10] H. Alt, M. Buchin: "Can We Compute the Similarity between Surfaces?", pp.78–99 in *Discr.&Comput.Geom.* vol.**43:1** (2010).

[Bra05] M. Braverman: "On the Complexity of Real Functions", pp.155–164 in *Proc. 46th Ann. IEEE Symp. on Foundations of Computer Science* (FoCS'05).

[BrPr03] V. Brattka, G. Presser: "Computability on subsets of metric spaces", pp.43–76 in *Theoretic.Computer Science* vol.**305** (2003).

[PPP*17] C. Park, J. Park, S. Park, D.-S. Seon, M. Ziegler: "Computable Operations on Compact Subsets of Metric Spaces with Applications to Fréchet Distance and Shape Optimization", `http://arXiv.org/abs/1701.08402` (2017)

[Schr04] M. Schröder: "Spaces Allowing Type-2 Complexity Theory Revisited", pp.443–459 in *Mathemat.Logic Quarterly* vol.**50** (2004).

Towards Vector Calculus in Exact Real Computation

Chansu Park and Martin Ziegler, KAIST

Abstract. We describe work in progress towards Vector Calculus as abstract data type in imperative Exact Real Computation: Computability investigations yield natural representations for (k-times) continuously differentiable curves, scalar and vector and tensor fields rendering arithmetic as well as common differentiation and (line, volume, and surface) integration operators on polytopes computable.

Exact Real Computation [6] is the paradigm of imperative programming over abstract data types for continuous data. It builds on the Kreitz-Weihrauch theory of *representations* [7], that is, on the formalization and comparison with respect to computability of encodings of objects from a continuous universe. The C++ library `iRRAM` [5] implements this paradigm, including data types for real numbers, polynomials, matrices, and analytic functions. The present work in progress aims at adding support for the objects commonly encountered in Multivariate Analysis and Continuum Mechanics [3]: continuously differentiable curves, scalar, vector, and tensor fields with operations such as gradients and (line, volume, and surface) integrals. We identify minimal discrete 'enrichment' [9, 8, 1] for the aforementioned objects to render said operations computable, and determine their parameterized complexity; cmp. [2].

Definition 1. *a) A* polytope $P \subseteq \mathbb{R}^d$ *is the non-empty convex hull of finitely many points. The dimension of the affine subspace spanned by P (or any subset of $\mathbb{R}^d$) is denoted by* $\dim(P)$. *A* face *of P is (either P itself or) its intersection $P \cap (\partial H^+)$ with the boundary of a closed halfspace H^+ containing P. The* face lattice $\mathcal{F}(P)$ *of P is the set of its faces, ordered with respect to inclusion and graded according to dimension.*

b) The data type polytope *stores for a polytope P, in addition to the coordinates of its vertices (i.e. 0-dimensional faces), the ambient dimension d and the leveled directed acyclic graph of said face lattice as discrete enrichment.*

c) For non-empty compact $C \subseteq \mathbb{R}^d$ and $\vec{k} = (k_1, \ldots, k_d) \in \mathbb{N}^d$ and convex $Y \subseteq \mathbb{R}^e$, $\mathcal{CL}^{\vec{k}}(C, Y)$ denotes the set of functions $f : P \to Y$ whose component-wise iterated partial derivative $\partial^{\vec{k}} f := \partial_1^{k_1} \cdots \partial_d^{k_d} f : C \to Y$ exists and is Lipschitz continuous.

d) The data type $\mathcal{CL}^{\vec{k}}$-vector field *stores for a mapping $f : P \to \mathbb{R}^e$ (a pointer to a function providing) 'black box' access as well as the integer tuple $\vec{k}$ and an integer upper bound to the Lipschitz constant of $\partial^{\vec{k}} f$ (w.r.t. the maximum norms on $\mathbb{R}^d$ and $\mathbb{R}^e$) as enrichment, in addition to the domain P according to b).*

e) Operations/methods on vector fields $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e)$ include:

i) constants $\mathbb{R}^e \ni \vec{y} \mapsto \left(\vec{x} \mapsto \vec{y}\right)$

ii) addition $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e) \times \mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e) \ni (f, g) \mapsto f + g \in \mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e)$

iii) scaling $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e) \times \mathbb{R} \ni (f, \lambda) \mapsto \lambda f \in \mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e)$

iv) inner product $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e) \times \mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e) \ni (f, g) \mapsto \sum_j f_j g_j \in \mathcal{CL}^{\vec{k}}(P, \mathbb{R})$

v) tensor product $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^a) \times \mathcal{CL}^{\vec{k}}(P, \mathbb{R}^b) \ni (f, g) = \left((f_i)_i, (g_j)_j\right) \mapsto (f_i \cdot g_j)_{i,j} \in \mathcal{CL}^{\vec{k}}(P, \mathbb{R}^{a \times b})$

vi) tensor contraction $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^{a \times b}) \ni f = (f_{i,j})_{i,j} \mapsto \left(\sum_\ell f_{i,\ell} f_{\ell,j}\right)_{i,j} \in \mathcal{CL}^{\vec{k}}(P, \mathbb{R}^{a \times c})$

vii) boundary restriction $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e) \times \mathcal{F}(P) \ni (f, F) \mapsto f_F \in \mathcal{CL}^{\vec{k}}(F, \mathbb{R}^e)$

viii) partial differentiation $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^e) \times \{1, \ldots, d\} \ni (f, j) \mapsto \partial_j f \in \mathcal{CL}^{\vec{k} - \vec{e}_j}(F, \mathbb{R}^e)$ *provided* $k_j > 0$

ix) divergence $\mathcal{CL}^{\vec{k}}(P, \mathbb{R}^d) \ni f = (f_j)_j \mapsto \sum_j \partial_j f_j \in \mathcal{CL}^{\vec{k} - (1, \ldots, 1)}(P, \mathbb{R})$ *provided* $\vec{k} \geq (1, \ldots, 1)$

x) *curl* $\mathcal{CL}^{\vec{k}}(P,\mathbb{R}^3) \ni f = (f_1,f_2,f_3) \mapsto \big(\partial_2 f_3 - \partial_3 f_2, \partial_3 f_1 - \partial_1 f_3, \partial_1 f_2 - \partial_2 f_1\big) \in \mathcal{CL}^{\vec{k}-(1,1,1)}(P,\mathbb{R})$ *for* $\vec{k} \geq (1,1,1)$ *and* $P \subseteq \mathbb{R}^3$

xi) *gradient* $\mathcal{CL}^{\vec{k}}(P,\mathbb{R}) \ni f \mapsto (\partial_j f_j)_j \in \mathcal{CL}^{\vec{k}-(1,\ldots,1)}(P,\mathbb{R}^d)$ *provided* $\vec{k} \geq (1,\ldots,1)$

xii) *curve integration* $\mathcal{CL}^{\vec{k}}(P,\mathbb{R}^d) \times \mathcal{CL}^1([0,1],P) \ni (f,\vec{\gamma}) \mapsto \int_0^1 f\big(\vec{\gamma}(t)\big) \cdot \vec{\gamma}'(t)\,dt \in \mathbb{R}$

xiii) *volume integration* $\mathcal{CL}^{\vec{k}}(P,\mathbb{R}) \ni f \mapsto \int_P f(\vec{x})\,d\vec{x} \in \mathbb{R}$

xiv) *surface integration* $\mathcal{CL}^{\vec{k}}(P,\mathbb{R}^d) \ni f \mapsto \int_{\partial P} f(\vec{x}) \cdot d\vec{\sigma}(\vec{x}) \in \mathbb{R}$

Note that (xii) is a composition of (iv), (viii), and (xiii); similarly (ix), (x), and (xi) build on (viii) and (i) to (iv). Restriction (vii) is mathematically trivial and easily ignored but for computability closely intertwined with our enrichment b).

Theorem 2. *With the enrichment from d), the operations in e) are computable.*

For integration (xii,xiii,xiv) this follows from [7, Theorem 6.4.1]; and differentiation (viii,ix,x,xi,xii) is based on the following particular case of [7, Corollary 6.4.8+Lemma 7.3.14]; cmp. [4, Lemma 3.4b]:

Lemma 3. *Let* $C \subseteq \mathbb{R}^d$ *be convex and* $f : C \to \mathbb{R}$ *differentiable with L-Lipschitz continuous derivative* ∇f*. Then* $\sup_{\vec{x}\in C} |f(\vec{x})| \leq \delta^2/(2L)$ *implies* $\sup_{\vec{x}\in C} |\nabla f(\vec{x})| \leq \delta$*. Moreover it holds*

$$\Big|\partial_j f(\vec{x}) \;-\; \frac{f(\vec{x}+\vec{e}_j\epsilon/L) - f(\vec{x})}{\epsilon/L}\Big| \;\leq\; \epsilon$$

If f *is computable in* regular *time* $t(n)$*, then* ∇f *is computable in time* $\mathcal{O}\big(t(2n + \log L)\big)$.

In the long term we want to generalize polytopal domains to compact differentiable oriented manifolds with boundary.

References

[1] Ambos-Spies, K., Brandt, U., Ziegler, M.: Real benefit of promises and advice. In: Bonizzoni, P., Brattka, V., Löwe, B. (eds.) Proc. 9th Conference on Computability in Europe. Lecture Notes in Computer Science, vol. 7921, pp. 1–11. Springer (2013)

[2] Kawamura, A., Müller, N., Rösnick, C., Ziegler, M.: Computational benefit of smoothness: parameterized bit-complexity of numerical operators on analytic functions and Gevrey's hierarchy. J. Complexity 31(5), 689–714 (2015)

[3] Marsden, J.E.: Vector Calculus. W.H. Freeman & Company (1976)

[4] Müller, N.T.: Subpolynomial complexity classes of real functions and real numbers. In: Kott, L. (ed.) Proceedings of the 13th International Colloquium on Automata, Languages, and Programming. Lecture Notes in Computer Science, vol. 226, pp. 284–293. Springer, Berlin (1986)

[5] Müller, N.T.: The iRRAM: Exact arithmetic in C++. In: Blanck, J., Brattka, V., Hertling, P. (eds.) Computability and Complexity in Analysis. Lecture Notes in Computer Science, vol. 2064, pp. 222–252. Springer, Berlin (2001), 4th International Workshop, CCA 2000, Swansea, UK, September 2000

[6] Müller, N.T., Ziegler, M.: From calculus to algorithms without errors. In: Hong, H., Yap, C. (eds.) Proc. 4th International Congress on Mathematical Software (ICMS). Lecture Notes in Computer Science, vol. 8592, pp. 718–724. Springer (2014)

[7] Weihrauch, K.: Computable Analysis. Springer, Berlin (2000)

[8] Ziegler, M.: Real computation with least discrete advice: a complexity theory of nonuniform computability with applications to effective linear algebra. Ann. Pure Appl. Logic 163(8), 1108–1139 (2012)

[9] Ziegler, M., Brattka, V.: Computability in linear algebra. Theoretical Computer Science 326(1–3), 187–211 (2004)

Reliable Diagonalization of Degenerate Matrices

Sewon Park and Martin Ziegler, KAIST

We consider the diagonalization of real symmetric square matrices with (not necessarily algebraic entries and) particular emphasis on the degenerate case: In general ill-posed and in fact provably uncomputable a problem, Recursive Analysis [ZiBr04] has established that some orthonormal basis of eigenvectors is computable in the sense of output approximation up to any desired precision — provided that, in addition to approximations to the $d \cdot (d+1)/2$ matrix' entires, its number k of *distinct* eigenvalues is known/provided. The present work explores the practical impact and quantitative efficiency of this qualitative and implicit result: We devise, combine, implement, evaluate, and compare four variants of rigorous and total algorithms guaranteed to (i) extract the coefficients of the matrix' characteristic polynomial, (ii) refine root enclosures of the latter using Pellet's Predicate until separating the k disjoint clusters to thus derive the individual eigenvalues' multiplicities d_j for $1 \leq j \leq k$, (iii) employ (some combination of) Trisection, Newton and/or Gräffe Iterations to improve said real root approximations to the precision required to finally (iv) determine an orthonormal base of eigenvectors to these eigenvalues. Algorithms (ii) and (iii) are based on, vary, and extend a recent series of contributions by Sagraloff, Sharma, Yap et al. (2013–2016), (iv) on own work (2016) using a variant of Gaussian Elimination that avoids total tests for real equality as these are known equivalent to the (complement of the) Halting Problem.

Our implementation builds on the C++ library `iRRAM`, providing an abstract data type `REAL` for rapid numerical prototyping with exact arithmetic and a subtly modified yet sound semantics of comparisons. Correctness is asserted by recovering the random eigenvector bases to a family or deliberately degenerate artificial test matrices; and efficiency is evaluated with respect to the three parameters output precision, matrix dimension, and eigenvalue separation. It confirms that the sophisticated combined algorithm [BSS*16] is indeed practical and asymptotically outperforms the two simpler variants — but can in turn be improved by employing trisection for medium precision.

The following lemmas 1, 2, and 3 illustrate the fundamental algorithmic ideas for (i) obtaining the coefficients of the matrix's characteristic polynomial, (ii) refining root enclosures of a polynomial, and (iii) improving real root approximations:

Lemma 1 (*Faddeev-Le Verrier* Algorithm). *For a complex square matrix $A \in \mathbb{C}^{d \times d}$ with characteristic polynomial $\chi_A(z) = z^d + p_1 z^{d-1} + \cdots + p_d \in \mathbb{C}[z]$, it holds*

$$\mathrm{trace}(T^k) \;+\; p_1 \cdot \mathrm{trace}(A^{k-1}) \;+\; \cdots \;+\; p_{k-1} \cdot \mathrm{trace}(A) \;+\; k \cdot p_k \;=\; 0, \qquad 1 \leq k \leq d. \tag{1}$$

Lemma 2 (Pellet Soft Predicate for Polynomial Root Clustering). *Let $f^{(j)}$ denote the j-th derivative of holomorphic f, defined on the complex open disc $D(c,r) = \{z \in \mathbb{C} : |c-z| < r\}$ with center c and radius r. For $\ell \in \mathbb{N}$, $L \geq 1$, $r > 0$, and $c \in \mathbb{C}$, consider the predicate*

$$T_\ell(f,c,r,L) \quad :\Longleftrightarrow \quad \left|f^{(\ell)}(c)/\ell!\right| \cdot r^\ell \;>\; L \cdot \sum\nolimits_{j \neq \ell} \left|f^{(j)}(c)/j!\right| \cdot r^j \;, \tag{2}$$

a) $T_\ell(f,c,r,1)$ implies that $D(c,r)$ contains exactly ℓ roots of f, counted with multiplicities [YSS13, Lemma 2]*; cmp.[Rump03].*

b) In particular if $f(z) = z^d + \sum_{j=0}^{d-1} p_j z^{d-j}$ is a monic polynomial of degree d, then all its lie in $D(0,R)$ for $R = R(p) := 1 + \max_j |p_j|$: the classical Cauchy Root Bound.

c) If both $D(c, r/11d)$ and $D(c, 18d^3 \cdot r)$ contain exactly ℓ roots of f (counted with multiplicities), then it holds $T_\ell\big(f,c,r,\frac{3}{2}\big)$ [BSS*16, Theorem 2].

This suggests repeatedly subdividing an initial real or complex interval according to Lemma 2b), dropping those which Lemma 2a) confirms to contain no roots, until arriving at k pairwise distinct sub-intervals that Lemma 2a) guarantees to contain roots with multiplicities summing up to d. Lemma 2c) guarantees all paths in this binary or 4-ary search tree to be finite, depending on the separation $\delta > 0$ of the non-coinciding roots. The following classical method increases said separation and thus accelerates the algorithm:

Lemma 3 (Dandelin-Lobachesky-Gräffe).

a) For $p(z) = a \cdot (z - z_1) \cdots (z - z_d) =: p_e(z^2) + z \cdot p_o(z^2) \in \mathbb{C}[z]$, $q(z) := (-1)^d \cdot \left(p_e^2(z) - z \cdot p_o^2(z)\right)$ *satisfies* $q(z) = a^2 \cdot (z - z_1^2) \cdots (z - z_d^2)$.

b) Suppose $D(0,1)$ *and* $D(0, 1+\delta)$ *contain the same number* ℓ *of roots of monic* p. *After applying* $\lceil \log_2 1/\delta \rceil + \lceil \log(1 + \log d) \rceil + 5$ *iterations of (a), the resulting polynomial* q *will have* p*'s roots from* $D(0, 1-\delta)$ *scaled to* $D(0, \frac{1}{11d})$ *and those from outside* $D(0, 1+\delta)$ *to outside of* $D(0, 18d^3)$ *such that* $T_\ell(q, 0, 1, \frac{3}{2})$ *from Lemma 2 holds* [BSS*15, Lemma 1].

We implement and empirically evaluate reliable diagonalization of symmetric matrices using four variants of the above approach: (a) Pellet Predicate with Gräffe Iteration, (b) Pellet Predicate with Gräffe Iteration until having separated the k root clusters, then refine the latter using *tri*section, (c) Pellet Predicate with Gräffe Iteration until having separated the k root clusters, then refine the latter using Interval Newton Iteration [BSS*16], (d) Pellet Predicate with Gräffe Iteration until having separated the k root clusters, then refine the latter using linearly-convergent trisection and finally switch to costly but eventually quadratically converging Interval Newton Iteration.

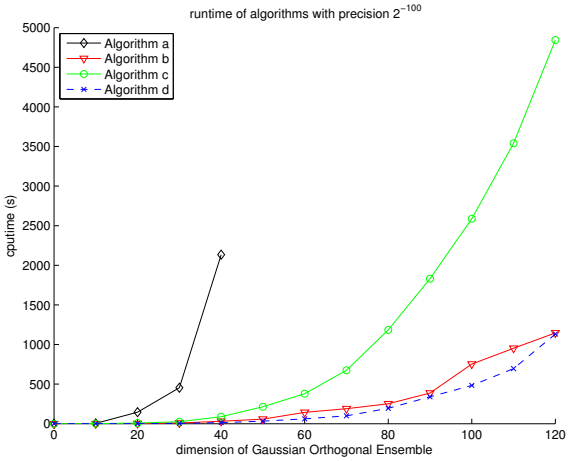


Fig. 1. Runtime in dependence on dimension for precision 2^{-100} of four algorithms (a)–(d).

References

[BSS*15] R. Becker, M. Sagraloff, V. Sharma, C. Yap: "A Near-Optimal Subdivision Algorithm for Complex Root Isolation based on the Pellet Test and Newton Iteration", `http://arXiv.org/abs/1509.06231`, (2015).

[BSS*16] R. Becker, M. Sagraloff, V. Sharma, J. Xu, C. Yap: "Complexity analysis of root clustering for a complex polynomial", pp.71–78 in *Proc. 41th International Symposium on Symbolic and Algebraic Computation*, (ISSAC'16).

[MPP*16] N. Müller, S. Park, N. Preining, M. Ziegler: "On formal verification in imperative multivalued programming over continuous data types", `http://arXiv.org/abs/1608.05787`.

[Müll01] N. Müller: "The irram: Exact arithmetic in c++", pp.222–252 in *Computability and Complexity in Analysis*, Springer, (2001).

[Rump03] S. M. Rump: "Ten methods to bound multiple roots of polynomials", pp.403–432 in *Journal of Computational and Applied Mathematics* 156(2), (2003).

[YSS13] C. Yap, M.. Sagraloff, V. Sharma: "Analytic root clustering: A complete algorithm using soft zero tests", pp. 434–444 in *Conference on Computability in Europe*, Springer (2013).

[ZiBr04] M. Ziegler, V. Brattka: "Computability in linear algebra", pp.187–211 in *Theoretical Computer Science* 326(1-3), (2004).

Complexity Theory using Hybrid Representations

Matthias Schröder, TU Darmstadt

May 24, 2017

There are several approaches to Complexity Theory in the representation based approach to Computable Analysis. Most concepts require to impose strong conditions on the used representations to ensure the existence of a time bound.

Weihrauch used *proper* representations to study complexity of functions on compact subsets of computable metric spaces [6]. In [3] *almost-compact* representations were introduced which allow for measurement of time complexity in terms of the output precision and a *discrete* parameter of the input. Kawamura and Cook proposed *second-order* representations [1]. This kind of representations can handle a much bigger class of spaces than proper or almost-compact representations.

Second-order representations use as their set of representitives the set of length-monotone functions on $\{\mathtt{0},\mathtt{1}\}^*$. The requirement of length-monotonicity leads to technical difficulties when defining natural representations for spaces. Examples of second-order representations for Hausdorff spaces of interest in Functional Analysis are typically constructed by coding a sequence of real numbers and a sequence of discrete information (cf. [1, 4]). The encoded real numbers can often be scaled down to numbers in the interval $[-1;1]$.

Therefore we propose to employ as space of representatives the product of the Hilbert cube with the Baire space, i.e. $\mathbb{H} := [-1;1]^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}}$. For endofunctions on $\mathbb{H}$ we consider the natural notion of computability induced by an effectively admissible proper TTE-representation $\varrho_{\mathbb{H}} \colon \tilde{\mathbb{H}} \to \mathbb{H}$, where $\tilde{\mathbb{H}} := \{-\mathtt{1},\mathtt{0},\mathtt{1}\}^{\mathbb{N}\times\mathbb{N}} \times \mathbb{N}^{\mathbb{N}}$. Time complexity for functions on $\mathbb{H}$ can be defined via oracle Turing machines with oracle space $\tilde{\mathbb{H}}$.

Definition 1 A *hybrid representation* of a space X is a partial surjection $\psi \colon \mathbb{H} \rightharpoonup \mathsf{X}$. A hybrid representation ψ is called *complete*, if it has a closed domain. It is called *admissible*, if the TTE-representation $\psi \circ \varrho_{\mathbb{H}}$ is admissible.

We present two examples of hybrid representations that are admissible and complete.

Example 2 (1) Let $\mathcal{C}[0;1]$ be the Polish space of continuous real-valued functions on $[0;1]$. We choose an effective numbering $(d_i)_i$ of the dyadic rationals in $[0;1]$ and define $(\vec{r},p) \in \mathbb{H}$ to be a name of $f \in \mathcal{C}[0;1]$, if $\forall i.f(d_i) = \vec{r}(i) \cdot p(0)$ and $k \mapsto p(k+1)$ is a modulus of continuity for f, meaning that $|x-y| \leq 2^{-p(k+1)}$ implies $|f(x)-f(y)| \leq 2^{-k}$.

(2) Let ℓ_2^* be the Hilbert space equipped with (the sequentialisation of) the weak*-topology. Here $\mathbb{H}_0 := [-1;1]^{\mathbb{N}} \times \mathbb{N}$ suffices as the space of representatives. We define $(\vec{r},m) \in \mathbb{H}_0$ to be a name of $x \in \ell_2^*$, if $\forall i.x(i) = \vec{r}(i) \cdot m$ and $\|x\|_2 \leq m$.

The main reason for preferring the Hilbert cube over $\mathbb{R}^{\mathbb{N}}$ in the definition of $\mathbb{H}$ are the following two propositions which are based on the compactness of $[-1;1]^{\mathbb{N}}$.

Proposition 3 *A hybrid representation ψ has a closed domain iff for every compact subset $K \subseteq \mathbb{N}^{\mathbb{N}}$ the set $\{(\vec{r},p) \in \mathrm{dom}(\psi) \,|\, p \in K\}$ is compact.*

Let X,Y be spaces equipped with complete hybrid representations $\psi_{\mathsf{X}},\psi_{\mathsf{Y}}$, and let M be an oracle Turing machine computing a realiser $g\colon \mathbb{H} \rightharpoonup \mathbb{H}$ for some function $f\colon \mathsf{X} \to \mathsf{Y}$. We call $t\colon \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^2 \to \mathbb{N}$ a *time bound* for M, if for all names $(\vec{r},p) \in \mathrm{dom}(\psi_{\mathsf{X}})$ and all $j,k \in \mathbb{N}$ the machine M produces $q(j)$ and a 2^{-k}-approximation to $\vec{s}(j)$ in at most $t(p,j,k)$ steps, where $(\vec{s},q)$ denotes $g(\vec{r},p)$. Proposition 3 implies:

Proposition 4 *Any oracle Turing machine realising a function between spaces equipped with complete hybrid representations has a continuous time bound $t\colon \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^2 \to \mathbb{N}$.*

A broad class of Hausdorff spaces enjoy an admissible complete hybrid representation.

Theorem 5

(1) *A metric space has an admissible complete hybrid representation iff it is Polish.*

(2) *A Hausdorff space X has an admissible complete hybrid representation with representing space $\mathbb{H}_0$ iff X is co-Polish (meaning that $\mathbb{R}^{\mathsf{X}}$ is Polish and X is regular).*

Theorem 6 *The category of Hausdorff qcb-spaces and continuous functions equipped with an admissible complete hybrid representation has countable products, countable co-products, and equalisers. But it is not closed under forming function spaces in* QCB.

The representations constructed for Theorem 5 satisfy a notion of admissibility which is stronger than the one in Definition 1. This notion forces to employ the Hilbert cube part of a representation and guarantees realisability of all continuous functions.

References

[1] A. Kawamura, S. Cook: *Complexity Theory for Operators in Analysis.* arXiv: 1305.0453v1 (2013).

[2] M. Schröder: *Admissible Representations for Continuous Computations.* Ph.D. Thesis, FernUniversität Hagen (2003).

[3] M. Schröder: *Spaces Allowing Type-2 Complexity Theory revisited.* Math. Log. Q. **50**, 443–459 (2004).

[4] F. Steinberg: *Computational Complexity Theory for Advanced Function Spaces in Analysis.* Ph.D. Thesis, Technische Universität Darmstadt (2017).

[5] K. Weihrauch: *Computable Analysis.* Springer, Berlin (2000).

[6] K. Weihrauch: *Computational Complexity on Computable Metric Spaces.* Math. Log. Q. **49**, 3–21 (2003).

The standard representation of the continuous functions from a Banach space perspective

Florian Steinberg*

One of the most commonly used frameworks for computational complexity of operators in analysis are the second-order representations as introduced by Kawamura and Cook [KC12]. One of the merits of this framework is that there is a second-order representation $\delta_\square$ of the continuous functions on the unit interval $C([0,1])$ that can be proven to be the minimal second-order representation such that evaluation is polynomial-time computable. Here, minimal means that any other representation of $C([0,1])$ such that evaluation is computable in polynomial time can be translated to $\delta_\square$ in polynomial time. This representation is accepted as the standard representation for computation on the continuous functions on the unit interval.

The paper [SS17] extends the framework of second-order representations. It introduces a notion of time-constructibility for second-order running times. The restriction to second-order representations is relaxed to regular representations, which avoid padding issues while maintaining polynomial-time computability of the length function. It introduces the notion of completeness of a representation as a condition that forces any computable mapping on the space to be computable in bounded time.

These extensions of the framework are put to use in defining a family of representations for Banach spaces that allow a Schauder basis as follows: Let $\mathbf{X}$ be an infinite dimensional separable Banach space that allows a Schauder-basis (e_i) and let $S : \omega^\omega \times \omega \to \omega$ be monotone and time-constructible such that for all $l \in \omega^\omega$ there exists an $l' \in \omega^\omega$ such that $S(l', \cdot) \geq l$. A string function $\varphi \in \mathcal{B}$ is a name of $x \in \mathbf{X}$ if and only if all of the following conditions hold:

(l) φ provides its length: For all $n \in \mathbb{N}$ it holds that $|\varphi(0^n)| = |\varphi|(n)$.

(a) φ encodes linear combinations that approximate x: For all $n \in \mathbb{N}$ there exists a linear combination of the first $S(|\varphi|, n)$ vectors e_i that approximates x with precision 2^{-n} and whenever $m \in \mathbb{N}$ is bigger than $\lceil \mathrm{lb}(S(|\varphi|, n+1)+1)\rceil + n + 1$ it holds that $\varphi(0\langle i, 1^n, 1^m\rangle) \in \mathbb{Z}$ and

$$\left\| \sum_{i=0}^{S(|\varphi|,n)} \frac{\varphi(0\langle i, 1^n, 1^m\rangle)}{2^m} e_i - x \right\| \leq 2^{-n+1}.$$

(o) φ provides an oracle for the norm: I.e. for all $n, m, N \in \mathbb{N}$ and integers $z_0, \ldots, z_N \in \mathbb{Z}$ it holds that $\varphi(1\langle\langle z_0, \ldots, z_N\rangle, N, 1^n, 1^m\rangle) \in \mathbb{Z}$ and

$$\left| \left\| \sum_{i=0}^{N} \frac{z_i}{2^m} e_i \right\| - \frac{\varphi(1\langle\langle z_0, \ldots, z_N\rangle, N, 1^n, 1^m\rangle)}{2^n} \right| \leq 2^{-n}.$$

These representations are proven to be admissible, regular and complete. Furthermore, a runtime bound of the metric is provided and a lower bound on the size of the sets of elements that have a short name is proven. The latter bound can

*Technische Universität Darmstadt; Email: steinberg@mathematik.tu-darmstadt.de

be understood as an optimality property: From a runtime of the metric an upper bound on the size of the sets that have short names can be extracted and the upper and lower bound are reasonably tight. Here, the sets at hand are compact and their 'size' is measured using the concept of metric entropy which also has applications in approximation theory [Lor66], constructive analysis [Bis67], proof mining [Koh08] and computable analysis [Wei03, KSZ16].

We consider the Banach space $\mathbf{X} := C([0,1])$ and prove that the standard representation is a special case of the above construction.

Theorem 1 *Let ξ denote the representation of $C([0,1])$ that arises if the Schauder basis (e_i) in the above construction is chosen to be the Faber-Schauder system [Fab10] and $S(l,n) := 2^{\max\{l(n),n\}}$. Then ξ is polynomial-time equivalent to $\delta_\square$.*

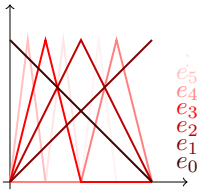


Figure 1: The Faber-Schauder system (e_i).

In a similar way, the representations of L^p-spaces introduced in [Ste17] are reproduced if the Faber-Schauder system is replaced by the Haar-system [Haa10].

The content of this extended abstract is mentioned without proof as an example in an extended abstract of [SS17] that has been accepted for the LICS conference 2017.

References

[Bis67] Errett Bishop. *Foundations of constructive analysis*. McGraw-Hill Book Co., New York-Toronto, Ont.-London, 1967.

[Fab10] Georg Faber. Über die orthogonalfunktionen des herrn haar. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 19:104–112, 1910. URL: http://eudml.org/doc/145217.

[Haa10] Alfred Haar. Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.*, 69(3):331–371, 1910. doi:10.1007/BF01456326.

[KC12] Akitoshi Kawamura and Stephen Cook. Complexity theory for operators in analysis. *ACM Trans. Comput. Theory*, 4(2):5:1–5:24, May 2012. doi:10.1145/2189778.2189780.

[Koh08] Ulrich Kohlenbach. *Applied proof theory: proof interpretations and their use in mathematics*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2008. doi:10.1007/978-3-540-77533-1.

[KSZ16] Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. Complexity theory of (functions on) compact metric spaces. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '16, pages 837–846, New York, NY, USA, 2016. ACM. doi:10.1145/2933575.2935311.

[Lor66] G. G. Lorentz. Metric entropy and approximation. *Bull. Amer. Math. Soc.*, 72(6):903–937, 11 1966. URL: http://projecteuclid.org/euclid.bams/1183528486.

[SS17] Matthias Schröder and Florian Steinberg. Bounded time computation on metric spaces and banach spaces. https://arxiv.org/abs/1701.02274, 2017. preprint.

[Ste17] Florian Steinberg. Complexity theory for spaces of integrable functions. https://arxiv.org/abs/1612.06419, 2017. to appear in Logical Methods in Computer Science.

[Wei03] Klaus Weihrauch. Computational complexity on computable metric spaces. *MLQ Math. Log. Q.*, 49(1):3–21, 2003. doi:10.1002/malq.200310001.

Polynomial running times for polynomial-time oracle machines

Akitoshi Kawamura and Florian Steinberg

Modern applications of second-order complexity theory almost exclusively use time-restricted oracle Turing machines to argue about the class of polynomial-time computable functionals [Lam06, FGH14, FZ15, KSZ16, SS17, etc.]. The acceptance of this model of computation goes back to a result by Kapron and Cook [KC96] that characterizes the class of basic feasible functionals introduced by Mehlhorn [Meh76]. Resource restricted oracle machines intuitively reflect what programmers require as efficiency if interpreted as programs with subroutine calls. I.e. the time taken to evaluate the subroutine is not counted towards the time consumption (the oracle query takes one time step) and more time is granted if the results of the queries are complicated. Superficially, the model is quite close to classical polynomial-time computability. The necessity to use of higher type objects as running times, however, introduces new difficulties. The framework introduced by Kawamura and Cook [KC12] addresses these problems by restricting to length-monotone string functions, forcing time-constructibility of second-order polynomials. In some situations this restriction seems inappropriate [BS17, SS17]. Thus, we investigate other solutions of the same problems.

Previous work by Cook [Coo91] defined a class of 'polynomial' oracle Turing machines avoiding the use of the length of a string function by only considering those oracle queries that turn up during a computation of an oracle Turing machine. We slightly adapt Cooks definition to the notion of a step count: Instead of bounding the total number of steps the machine may take by the maximum of the sizes of the oracle return values, we dynamically increase the number of allowed steps each time the machine encounters an answer bigger than any previous answer. This avoids higher-order objects as running-times. This differs from Cooks definition in that it makes clocking possible for time-constructible step counts. Having a polynomial step count is not enough: There are machines with polynomial step count that compute non polynomial-time functions on polynomial-time oracles. To forbid this, we require that the number of times the runtime of the machine is increased, that is the 'length revisions', are bounded independently of the input and the oracle. We call a functional 'strongly polynomial-time computable' if it is computed by an oracle machine that both allows a polynomial step count and has finite length revision. Strong polynomial-time computability implies polynomial-time computability.

We go on to compare the class of strongly polynomial-time computable functionals on Baire space to the class of functionals that are polynomial-time computable in the sense of second-order complexity theory in more detail: We prove that the notions of strong polynomial-time computability and polynomial-time computability coincide as long as the framework of Kawamura and Cook is used. That is, as long as one only considers functionals whose domain con-

sists of length-monotone string functions. For general functionals, this fails: We provide a polynomial-time computable total functional which is not strongly polynomial-time computable. Most work on complexity of operators in analysis is done in the framework of Kawamura and Cook. However, recent advances to construct minimal representations of the continuous functions resort to non length-monotone names and may lead to more natural examples of non strongly polynomial time computable operators [BS17]. Such operators may run for a long time without long oracle answers that provide evidence that this is justified.

Finally, we prove that strong polynomial-time computability is compatible with restrictions of the domain of an operator. This is in contrast to polynomial-time computable functionals, where incompatibility with respect to restrictions can be proven. The difference is due to the fact that strongly polynomial-time computability allows clocking and polynomial-time computability does not. The emphasis on compatibility with relativization and well-behavedness on restricted domains differentiates our approach from previous work from second-order complexity theory [Pez98], where partial functionals are only considered by means of total extensions and the closure properties are imposed on the total operators. A paper containing the results listed here can be found on the arXiv [KS17].

References

[BS17] Franz Braußе and Florian Steinberg. A minimal representation for the continuous functions. https://arxiv.org/abs/1703.10044, 2017.

[Coo91] Stephen A. Cook. Computational complexity of higher type functions. In *Proceedings of the International Congress of Mathematicians, Vol. I, II (Kyoto, 1990)*, pages 55–69. Math. Soc. Japan, Tokyo, 1991.

[FGH14] Hugo Férée, Walid Gomaa, and Mathieu Hoyrup. Analytical properties of resource-bounded real functionals. *J. Complexity*, 30(5):647–671, 2014. doi:10.1016/j.jco.2014.02.008.

[FZ15] Hugo Férée and Martin Ziegler. On the computational complexity of positive linear functionals on c[0;1], 2015. MACIS conference. URL: https://hugo.feree.fr/macis2015.pdf.

[KC96] B. M. Kapron and S. A. Cook. A new characterization of type-2 feasibility. *SIAM J. Comput.*, 25(1):117–132, 1996.

[KC12] Akitoshi Kawamura and Stephen Cook. Complexity theory for operators in analysis. *ACM Trans. Comput. Theory*, 4(2):5:1–5:24, May 2012. doi:10.1145/2189778.2189780.

[KS17] Akitoshi Kawamura and Florian Steinberg. Polynomial running times for polynomial-time oracle machines. https://arxiv.org/abs/1704.01405, 2017. arXiv preprint;.

[KSZ16] Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. Towards computational complexity theory on advanced function spaces in analysis. In Arnold Beckmann, Laurent Bienvenu, and Nataša Jonoska, editors, *Pursuit of the Universal: 12th Conference on Computability in Europe, Paris, France, June 27 - July 1, 2016, Proceedings*, pages 142–152. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-40189-8_15.

[Lam06] Branimir Lambov. The basic feasible functionals in computable analysis. *J. Complexity*, 22(6):909–917, 2006. doi:10.1016/j.jco.2006.06.005.

[Meh76] Kurt Mehlhorn. Polynomial and abstract subrecursive classes. *J. Comput. System Sci.*, 12(2):147–178, 1976. Sixth Annual ACM Symposium on the Theory of Computing (Seattle, Wash., 1974).

[Pez98] Elena Pezzoli. *On the computational complexity of type 2 functionals*, pages 373–388. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. doi:10.1007/BFb0028026.

[SS17] Matthias Schröder and Florian Steinberg. Bounded time computation on metric spaces and banach spaces. https://arxiv.org/abs/1701.02274, 2017. preprint; extended abstract accepted for LICS 2017 conference.

Author Index