Seventeenth International Conference on Computability and Complexity in Analysis

CCA 2020

September 9-11, 2020, virtual

(originally planned for Bologna, Italy)

Invited Speakers

- **Olivier Bournez** (Palaiseau, France)
- **Douglas Cenzer** (Gainesville, USA)
- **Eva Darulova** (Kaiserslautern, Germany)
- **Makoto Fujiwara** (Munich, Germany)
- **Mathieu Hoyrup** (Nancy, France)
- **Alberto Marcone** (Udine, Italy)
- **Eike Neumann** (Oxford, UK)
- **Arno Pauly** (Swansea, UK)

Scientific Programme Committee

- **Andrej Bauer** (Ljubljana, Slovenia)
- **Matthew de Brecht** (Kyoto, Japan)
- **Pietro Di Gianantonio** (Udine, Italy)
- **Daniel Graça** (Faro, Portugal)
- **Peter Hertling** (Munich, Germany)
- **Hajime Ishihara** (Ishikawa, Japan)
- **Zvonko Iljazović**, chair (Zagreb, Croatia)
- **Matthias Schröder** (Darmstadt, Germany)
- **Holger Thies** (Fukuoka, Japan)
- **Ning Zhong** (Cincinnati, USA)

Organizing Committee

- **Melissa Antonelli** (Bologna, Italy)
- **Francesco Bianchini** (Bologna, Italy)
- **Guido Boccali** (Bologna, Italy)
- **Giovanna Corsi** (Bologna, Italy)
- **Ugo Dal Lago** (Bologna, Italy)
- **Cecilia Di Florio** (Bologna, Italy)
- **Guido Fiorillo** (Bologna, Italy)
- **Guido Gherardi**, chair (Bologna, Italy)
- **Leonardo Mazzanti** (Bologna, Italy)
- **Eugenio Orlandelli** (Bologna, Italy)
- **Sofia Pierini** (Bologna, Italy)
- **Gioia Susanna** (Bologna, Italy)
- **Riccardo Zanichelli** (Bologna, Italy)

Invited Talks

Computability, Complexity and Programming with Ordinary Differential Equations and Finite Differences

Olivier Bournez

LIX, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris,
91128 Palaiseau Cedex, FRANCE
Olivier.Bournez@lix.polytechnique.fr

Ordinary Differential Equations (ODEs) appear to be a universally adopted and very natural way for expressing properties for continuous time dynamical systems. They are intensively used, in particular in applied sciences with an abundant literature about the hardness of solving ODEs with numerical methods.

In this talk, we will adopt a dual view: we consider ODEs as a way to program or to describe our mathematical/computer science world. We survey several results considering ODEs under this computational perspective, with a computability and complexity theory point of view.

We will both review some works related to classical (continuous) ODES, and their discrete counterparts, also called finite differences.

Continuous settings In the continuous settings, we will provide various reasons why we believe that polynomial ODEs should be considered as the continuous time analog of Turing machines for continuous-time computations, or should be used as a way to talk about computability, complexity and even mathematical logic.

We will mention applications in various fields: determining whether analog models of computation can compute faster than classical digital models of computation; solving complexity issues for computations with biochemical reactions in bioinformatics; machine independent characterizations of various computability and complexity classes such as P or NP, or proof of the existence of a universal polynomial ordinary differential equation whose solutions can approximate any continuous function if provided with a suitable well-chosen initial condition.

Discrete time settings We will also come to the discrete counterpart of ODEs: discrete ODEs. The associated derivative notion, also called *finite dif-*

ferences, has been widely studied in numerical optimization for function approximation and is reminiscent in *discrete calculus* for combinatorial analysis. Similarities between discrete and continuous statements have also been historically observed, under the terminology of *umbral* or *symbolic calculus* as early as in the 19th century. However, even if the underlying computational content of finite differences theory is clear and has been pointed out many times, no fundamental connections with algorithms and complexity have been exhibited so far.

We will present some recent results demonstrating that discrete ODEs is a very natural tool for algorithm design and to prove that complexity and computability notions can be elegantly and simply captured using discrete ordinary differential equations. We will illustrate this by providing a characterization of FPTIME, the class of polynomial time computable functions, and of its non deterministic analog FNP. To this aim, we will discuss how some notions from the analog world such as linearity of differential equations or derivatives along some particular functions (i.e. changes of variables) are representative of a certain computational hardness and can be used to solve efficiently some (classical, digital) problems, or revisit some constructions from implicit complexity.

References

[1] Olivier Bournez and Arnaud Durand. Recursion schemes, discrete differential equations and characterization of polynomial time computation. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th Int Symposium on Mathematical Foundations of Computer Science, MFCS*, volume 138 of *LIPIcs*, pages 23:1–23:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[2] Olivier Bournez and Amaury Pouly. A survey on analog models of computation. In Vasco Brattka and Peter Hertling, editors, *Handbook of Computability and Complexity in Analysis*. Springer, 2020.

RANDOMNESS EXTRACTION FROM A COMPUTABILITY-THEORETIC PERSPECTIVE

DOUGLAS CENZER AND CHRISTOPHER P. PORTER

The aim of this study is to analyze the rate of the extraction of randomness via various effective procedures using the tools of computability theory and algorithmic randomness. Our starting point is a classic problem posed by von Neumann in [vN51], namely that of extracting unbiased randomness from the tosses of a biased coin. Von Neumann provides an elegant solution to the problem:

(1) Toss the biased coin twice.
(2) If the outcome is HH or TT, then discard these tosses.
(3) If the outcome is HT, then output **H**.
(4) If the outcome is TH, then output **T**.

Notice in the case that the coin comes up heads with probability p,

- the probability of HH is p^2,
- the probability of TT is $(1-p)^2$, and
- the probability of HT (and that of TH) is $p(1-p)$.

It follows from the independence of the events H and T that with probability one the derived sequence will be an infinite sequence in which the events **H** and **T** occur with probability $1/2$.

It is well-known that von Neumann's procedure is rather inefficient, since on average $\frac{1}{p(1-p)}$ biased bits are required to produce one unbiased bit when the biased coin comes up heads with probability $p \in (0,1)$.

However, a number of improvements have been found. For instance, in [Per92], Peres studies a sequence of procedures obtained by iterating von Neumann's procedure and studies the associated extraction rate of each such procedure. Under this procedure, the extraction rate is shown to be

$$\lim_{k\to\infty} \limsup_{n\to\infty} \frac{E(|\phi_k(x_1, x_2, \ldots, x_n)|)}{n} = H(p),$$

where $H(p) = -p\log(p) - (1-p)\log(1-p)$ is the entropy associated with the underlying source.

In this article, we study a definition of the extraction rate for Turing functionals that accept their input with probability one (referred to as *almost total* functionals). In particular, we can formalize certain randomness extraction procedures as Turing functionals and study the behavior of these functionals when applied to algorithmically random sequences.

Date: August 6, 2020.

For a number of such functionals, it is known that almost every sequence attains the extraction rate; here we provide a sufficient level of algorithmic randomness that guarantees this result.

We consider three main examples here:

(1) functionals defined in terms of maps on $2^{<\omega}$ that we call block maps, which generalize von Neumann's procedure,
(2) functionals derived from certain trees called discrete distribution generating trees (or DDG trees, for short), introduced by Knuth and Yao [KY76] in the study of non-uniform random number generation, and
(3) a procedure independently developed by Levin [LZ70] and Kautz [Kau91] for converting biased random sequences into unbiased random sequences.

Notably, our analysis of the extraction rates of these three classes of examples draw upon the machinery of effective ergodic theory, using certain effective versions of Birkhoff's ergodic theorem (and, in the case of the Levin-Kautz procedure, an effective version of the Shannon-McMillan-Breiman theorem from classical information theory due to Hoyrup [Hoy12]). We also consider extraction rates in the case of algorithmically random continuous functions on 2^ω, a class of functions first introduced by Barmpalias, Brodhead, Cenzer, Remmel, and Weber in [BBC+08].

References

[BBC+08] George Barmpalias, Paul Brodhead, Douglas Cenzer, Jeffrey B. Remmel, and Rebecca Weber. Algorithmic randomness of continuous functions. *Arch. Math. Logic*, 46(7-8):533–546, 2008. 2

[Hoy12] Mathieu Hoyrup. The dimension of ergodic random sequences. In *29th International Symposium on Theoretical Aspects of Computer Science*, volume 14 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 567–576. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2012. 2

[Kau91] Steven M. Kautz. *Degrees of random sets*. ProQuest LLC, Ann Arbor, MI, 1991. Thesis (Ph.D.)–Cornell University. 2

[KY76] Donald E. Knuth and Andrew C. Yao. The complexity of nonuniform random number generation. In *Algorithms and complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1976)*, pages 357–428, 1976. 2

[LZ70] L. Levin and A.K. Zvonkin. The complexity of finite objects and the development of the concepts of information and randomness of means of the theory of algorithms. *Uspekhi Mat. Nauk*, 25:85–127, 1970. 2

[Per92] Y. Peres. Iterating von neumann's procedure for extracting random bits. *Ann. Statist.*, 20:590–597, 1992. 1

[vN51] John von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, pages 36–38, 1951. 1

Towards a Sound Approximating Compiler for Numerical Kernels

Eva Darulova

Max Planck Institute for Software Systems, eva@mpi-sws.org

Computing resources are fundamentally limited and sometimes an exact solution may not even exist. Thus, when implementing real-world systems, approximations are inevitable, as are the errors they introduce. The magnitude of errors is problem-dependent but higher accuracy generally comes at a cost in terms of memory, energy or runtime, effectively creating an accuracy-efficiency tradeoff. Many applications are robust to some amount of noise, which allows one to *deliberately* reduce their accuracy or quality in return for better resource usage. Indeed, accepting a small error or a slightly suboptimal output can result in relatively large efficiency improvements [13].

To take advantage of this tradeoff, we need to ensure that the computed results are sufficiently accurate, otherwise we risk disastrously incorrect results or system failures. Approximation errors propagate through a program in complex ways, so that navigating the tradeoff between accuracy or quality of an application and its efficiency is highly nontrivial.

Despite their pervasiveness and inherent difficulty, programming with approximations today is largely a manual and ad-hoc process. A developer typically starts from an exact specification or program, and repeatedly introduces approximations until some resource bound is met, e.g. the program uses sufficiently little memory, runs fast enough, or is satisfactorily energy-efficient. In this process, she needs to decide which kind of approximations to apply, where to apply them, and check that the computed results remain acceptable. As little rigorous automated tool support exists, the current programming practice with approximations requires significant human effort, is error-prone and often results in suboptimal solutions.

Ideally, approximations would be hidden similarly to how compilers today hide low-level machine details and automatically optimize programs' performance. That is, the user should write her program as if infinite resources were available and specify only an overall error bound on the program's result. It would then be the task of an automated synthesis tool to select and apply individual approximations such that the overall specified error bound is met, while at the same time resource consumption is minimized.

We have realized this vision for straight-line numerical kernels that, for instance, frequently appear in embedded systems (e.g. implementing controller) and machine learning (e.g. neural networks). With our open-source tool Daisy [4], the user specifies the ideal program as a real-valued function, for which Daisy fully automatically generates an efficient finite-precision implementation that is guaranteed to meet a given end-to-end error bound.

In this talk, we give an overview of Daisy's verification and optimization procedures. Daisy automatically optimizes the arithmetic by selecting a suitable evaluation order that minimizes finite-precision rounding error [6, 3]; it chooses different precisions for different operations to minimize running time or memory [3, 12]; and it selects application-specific polynomial approximations of elementary functions (e.g. sine and exponential) to improve running time over general-purpose library implementations [7, 9]. In order to guarantee soundness, we develop several different static analysis techniques [5, 8, 11, 10], targeting different kinds of errors, making Daisy applicable in a variety of domains. For some of the computed error bounds we provide a certification framework [2, 1] that independently verifies them using the interactive theorem provers Coq and HOL4, providing additional correctness guarantees.

References

1. Bard, J., Becker, H., Darulova, E.: Formally Verified Roundoff Errors Using SMT-based Certificates and Subdivisions. In: Formal Methods (FM) (2019)
2. Becker, H., Zyuzin, N., Monat, R., Darulova, E., Myreen, M.O., Fox, A.C.J.: A Verified Certificate Checker for Finite-Precision Error Bounds in Coq and HOL4. In: Formal Methods in Computer Aided Design (FMCAD) (2018)
3. Darulova, E., Horn, E., Sharma, S.: Sound Mixed-Precision Optimization with Rewriting. In: International Conference on Cyber-Physical Systems (ICCPS) (2018)
4. Darulova, E., Izycheva, A., Nasir, F., Ritter, F., Becker, H., Bastian, R.: Daisy - Framework for Analysis and Optimization of Numerical Programs (Tool Paper). In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS) (2018)
5. Darulova, E., Kuncak, V.: Sound Compilation of Reals. In: Principles of Programming Languages (POPL) (2014)
6. Darulova, E., Kuncak, V., Majumdar, R., Saha, I.: Synthesis of Fixed-Point Programs. In: Embedded Software (EMSOFT) (2013)
7. Darulova, E., Volkova, A.: Sound Approximation of Programs with Elementary Functions. In: Computer Aided Verification (CAV) (2019)
8. Izycheva, A., Darulova, E.: On Sound Relative Error Bounds for Floating-Point Arithmetic. In: Formal Methods in Computer Aided Design (FMCAD) (2017)
9. Izycheva, A., Darulova, E., Seidl, H.: Synthesizing Efficient Low-Precision Kernels. In: Automated Technology for Verification and Analysis (ATVA) (2019)
10. Lohar, D., Darulova, E., Putot, S., Goubault, E.: Discrete Choice in the Presence of Numerical Uncertainties. IEEE Trans. on CAD of Integrated Circuits and Systems **37**(11) (2018)
11. Lohar, D., Prokop, M., Darulova, E.: Sound Probabilistic Numerical Error Analysis. In: Integrated Formal Methods (IFM) (2019)
12. Salamati, M., Salvia, R., Darulova, E., Soudjani, S., Majumdar, R.: Memory-Efficient Mixed-Precision Implementations for Robust Explicit Model Predictive Control. ACM Trans. Embedded Comput. Syst. **18** (2019)
13. Stanley-Marbell, P., Alaghi, A., Carbin, M., Darulova, E., Dolecek, L., Gerstlauer, A., Gillani, G., Jevdjic, D., Moreau, T., Cacciotti, M., Daglis, A., Jerger, N.E., Falsafi, B., Misailovic, S., Sampson, A., Zufferey, D.: Exploiting Errors for Efficiency: A Survey from Circuits to Applications. ACM Comput. Surv. **53**(3) (2020)

Weihrauch and constructive reducibility between existence statements

Makoto Fujiwara

Meiji University / LMU München

makotofujiwara@meiji.ac.jp

A large amount of mathematical statements are of the logical form

$$\forall f(A(f) \to \exists g B(f,g)), \tag{1}$$

where f and g may be tuples. Such statements are often called **existence statements** since they argue the existence of some objects. One can see existence statements represented as sentences of the form (1) as problems to be solved. In such a context, any f such that $A(f)$ holds is called an **instance** of the problem and g is called a **solution** to the instance. Mutual computability between existence statements have been investigated extensively in computable analysis and classical reverse mathematics. The investigations usually employ the following reduction: a Π^1_2 sentence P of the form (1) is reducible to another Π^1_2 sentence Q of the form (1) if there exist Turing functionals Φ and Ψ such that whenever f is an instance of P, then $\Phi(f)$ is an instance of Q, and whenever g' is a solution to $\Phi(f)$, then $g := \Psi(f, g')$ is a solution to f. This is a particular case of **Weihrauch reducibility** for Π^1_2 sentences with Baire space as their represented spaces (see [1, Appendix]).

On the other hand, there is another development on the relation between existence statements from constructive mathematics, where existence is shown by giving a construction of the witness entirely in the proofs. Ishihara and others have developed reverse mathematics first informally in Bishop's constructive mathematics, and later formally in two-sorted intuitionistic arithmetic.

Interestingly, there are several corresponding results between constructive reverse mathematics and mutual computability in classical reverse mathematics and computable analysis (based on classical logic). In this decade, several attempts have been done to reveal the proper relation between them. In this talk, I would like to introduce my work [2, 3] among them.

We consider the primitive recursive variants of Weihrauch reduction between existence statements P and Q formalized as sentences $\forall f(A_1(f) \to \exists g B_1(f,g))$ and $\forall f(A_2(f) \to \exists g B_2(f,g))$ in the context of finite-type arithmetic $\mathsf{E\text{-}HA}^\omega$ (or Feferman's restriction $\widehat{\mathsf{E\text{-}HA}}^\omega\restriction$ thereof) as follows:

- For a finite-type arithmetic S^ω containing $\mathsf{E\text{-}HA}^\omega$, P **is Gödel-primitive-recursive Weihrauch reducible to** Q **in** S^ω if there exist closed terms s and t (of suitable types) in $\mathbf{T}$ such that S^ω proves

$$\forall f(A_1(f) \to A_2(sf)) \wedge \forall f, g' \left(B_2(sf, g') \wedge A_1(f) \to B_1(f, tfg')\right). \tag{2}$$

– For a finite-type arithmetic S^ω containing $\widehat{\mathsf{E\text{-}HA}}^\omega\restriction$, P **is Kleene-primitive-recursive Weihrauch reducible to** Q **in** S^ω if there exist closed terms s and t (of suitable types) in $\mathbf{T_0}$ such that S^ω proves (2).

In addition, P **is normally reducible to** Q **in** S^ω if S^ω proves

$$\forall f\,(A_1(f) \to \exists f'\,(A_2(f') \wedge \forall g'\,(B_2(f',g') \to \exists g B_1(f,g)))) \,.$$

The notions of Gödel/Kleene-primitive-recursive Weihrauch reducibility is a natural restriction of formalized Weihrauch reducibility where Turing functionals for the reduction are replaced by primitive recursive functionals in the sense of Gödel/Kleene. The normal reducibility requires a proof of $\mathrm{Q} \to \mathrm{P}$ in a standard manner. Since intuitionistic finite-type arithmetic with a choice principle roughly corresponds to Bishop's constructive mathematics, one may regard the normal reducibility in a nearly intuitionistic finite-type arithmetic as a sort of constructive reducibility. Then we show a meta-theorem stating that the primitive-recursive Weihrauch reducibility verifiably in a fragment of classical finite-type arithmetic is equivalent to the normal reducibility in the corresponding (nearly) intuitionistic finite-type arithmetic for all existence statements formalized with $\exists$-free (containing neither $\exists$ nor $\vee$) formulas [3]. Thus constructive reducibility can be captured by the primitive-recursive variant of Weihrauch reducibility with an additional restriction on the verification theory (which has not been took into account in computable analysis). Of course, the Weihrauch reductions between concrete existence statements are not always primitive recursive (in the sense of Gödel/Kleene). In addition, there are many existence statements which are not formalized with $\exists$-free formulas. Nonetheless, there seem to be fairly many examples to which the meta-theorem is applicable. In fact, the Weihrauch reduction between concrete existence statements can be verified usually in a weak fragment of classical finite-type arithmetic. We observe this in some typical examples from computable analysis and constructive reverse mathematics.

In addition, we also characterize the property that "P is Gödel/Kleene-primitive-recursive Weihrauch reducible to the parallelization of Q (verifiably in a weak fragment of classical finite-type arithmetic)" in some natural context of constructive reverse mathematics [2].

References

1. Dorais, F.G., Dzhafarov, D.D., Hirst, J.L., Mileti, J.R., Shafer, P.: On uniform relationships between combinatorial problems. Trans. Amer. Math. Soc. **368**(2), 1321–1359 (2016)
2. Fujiwara, M.: Parallelizations in Weihrauch reducibility and constructive reverse mathematics. In: Anselmo, M., Della Vedova, G., Manea, F., Pauly, A. (eds.) Beyond the Horizon of Computability. pp. 38–49. Springer International Publishing, Cham (2020)
3. Fujiwara, M.: Weihrauch and constructive reducibility between existence statements. Computability (2020), to appear

Descriptive complexity on represented spaces

Antonin Callard and Mathieu Hoyrup
–
antonin.callard@gmail.com, mathieu.hoyrup@inria.fr

1 Introduction

It follows from the work of Schröder [Sch02b, Sch02a] that the QCB-spaces, i.e. the quotients of countably-based spaces, are at the same time a class of topological spaces with very good closure properties (they form a cartesian closed category), and the class of spaces where computable analysis can be developed (they are the admissibly represented spaces). Therefore, those spaces can be studied under the light of both topology and representations [Pau15, PdB15]. In particular, these two viewpoints lead to different measures of descriptive complexity of subsets of the space.

Traditionally, descriptive complexity is topological: it measures the difficulty of describing sets of *points* in terms of open sets, using boolean operations. Representations provide an alternative way, by measuring the descriptive complexity of the corresponding set of *names* instead. We call it *symbolic complexity*. Algorithms are more directly confronted to this notion of complexity, because they work with names rather than points.

It was proved by Brattka [Bra05] that topological and symbolic complexity coincide on Polish spaces, and by de Brecht [dB13] that they actually coincide on any countably-based space. In recent works [CH20, Hoy20], we investigate what happens on other spaces and address the following questions:

- On which spaces, and for which complexity levels do topological and symbolic complexity coincide?
- What properties of a space explain the differences between topological and symbolic complexity?
- What does the symbolic complexity of set tells us about the set?
- How to describe the sets in a given symbolic complexity class?

We mainly focus on the class of co-Polish spaces [Sch04, dBPS19], in particular the space of real polynomials, and the space of open subsets of a Polish space. In each case, we obtain a precise picture of the relationship between topological and symbolic complexity.

Our results suggest that the mismatch between topological and symbolic complexity reflects the disagreement between topological and sequential aspects of the space. It is in agreement with the fact that admissibly represented spaces, or equivalently QCB-spaces, are better behaved in the category of sequential spaces than in the category of topological spaces.

References

[Bra05] Vasco Brattka. Effective Borel measurability and reducibility of functions. *Mathematical Logic Quarterly*, 51(1):19–44, 2005.

[CH20] Antonin Callard and Mathieu Hoyrup. Descriptive complexity on non-Polish spaces. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, volume 154 of *LIPIcs*, pages 8:1–8:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[dB13] Matthew de Brecht. Quasi-Polish spaces. *Ann. Pure Appl. Logic*, 164(3):356–381, 2013.

[dBPS19] Matthew de Brecht, Arno Pauly, and Matthias Schröder. Overt choice. *Computability*, ?, 2019.

[Hoy20] Mathieu Hoyrup. Descriptive complexity on non-polish spaces II. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 132:1–132:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Pau15] Arno Pauly. On the topological aspects of the theory of represented spaces. *Computability*, 5(2):159–180, 2015.

[PdB15] Arno Pauly and Matthew de Brecht. Descriptive set theory in the category of represented spaces. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 438–449. IEEE Computer Society, 2015.

[Sch02a] Matthias Schröder. *Admissible Representations for Continuous Computations*. PhD thesis, FernUniversität Hagen, 2002.

[Sch02b] Matthias Schröder. Extended admissibility. *Theor. Comput. Sci.*, 284(2):519–538, 2002.

[Sch04] Matthias Schröder. Spaces allowing type-2 complexity theory revisited. *Math. Log. Q.*, 50(4-5):443–459, 2004.

THE WEIHRAUCH LATTICE AROUND ATR_0 AND $\mathbf{\Pi}^1_1$-CA_0

ALBERTO MARCONE

The classification of mathematical problems in the Weihrauch lattice is a line of research that blossomed in the last few years, starting with [GM09, BGM12]. Initially this approach dealt mainly with statements which are provable in ACA_0 and showed that usually Weihrauch reducibility is more fine-grained than reverse mathematics.

In the last few years the study of multi-valued functions arising from statements laying at higher levels (such as ATR_0 and $\mathbf{\Pi}^1_1$-CA_0) of the reverse mathematics spectrum started as well. The multi-valued functions studied so far include those arising from the perfect tree theorem [KMP20, Hir19], from comparability of well-orders [KMP20, Goh20], from determinacy of open and clopen games [KMP20], from König's duality theorem [Goh19], from various forms of choice [AdK19], and from the open and clopen Ramsey theorem [MV20].

At this level often a single theorem naturally leads to several multi-valued functions of different Weihrauch degree, depending on how the theorem is "read" from a computability viewpoint. A case in point is the perfect tree theorem: it can be read as the request to produce a perfect subtree of a tree with uncountably many paths (leading to the multi-valued function PTT_1), or as the request to list all paths of a tree which does not contain a perfect subtree (yielding List). Similarly, the clopen Ramsey theorem leads to the multi-valued function $\mathbf{\Delta}^0_1 - \mathsf{RT}$ that associates to every clopen subset of $[\mathbb{N}]^{\mathbb{N}}$ an infinite homogenous set on either side, and to the multi-valued function $\mathsf{FindHS}_{\mathbf{\Delta}^0_1}$ producing for each clopen subset which has an infinite homogeneous sets on one side a homogenous set on that side. $\mathbf{\Sigma}^0_1 - \mathsf{RT}$, $\mathsf{FindHS}_{\mathbf{\Sigma}^0_1}$, and $\mathsf{FindHS}_{\mathbf{\Pi}^0_1}$ are defined similarly starting from the open Ramsey theorem.

In [KMP20, MV20] we showed that

- $\mathsf{List} \equiv_{\mathrm{W}} \mathbf{\Delta}^0_1 - \mathsf{RT} \equiv_{\mathrm{W}} \mathsf{UC}_{\mathbb{N}^{\mathbb{N}}}$,
- $\mathsf{PTT}_1 \equiv_{\mathrm{W}} \mathsf{FindHS}_{\mathbf{\Delta}^0_1} \equiv_{\mathrm{W}} \mathsf{FindHS}_{\mathbf{\Pi}^0_1} \equiv_{\mathrm{W}} \mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$,

where $\mathsf{UC}_{\mathbb{N}^{\mathbb{N}}}$ and $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$ are respectively unique choice and choice on the Baire space. On the other hand, $\mathbf{\Sigma}^0_1 - \mathsf{RT} <_{\mathrm{W}} \mathsf{FindHS}_{\mathbf{\Sigma}^0_1}$ and both functions cannot be computed, not even allowing arithmetic reductions, using $\mathsf{C}_{\mathbb{N}^{\mathbb{N}}}$.

In this talk I discuss some of these results and present work in progress (joint with Vittorio Cipriani) dealing with various forms of the Cantor-Bendixson theorem.

References

[AdK19] Paul-Elliot Anglès d'Auriac and Takayuki Kihara. A comparison of various analytic choice principles, 2019. arXiv:1907.02769.

[BGM12] Vasco Brattka, Guido Gherardi, and Alberto Marcone. The Bolzano-Weierstrass theorem is the jump of weak Kőnig's lemma. *Ann. Pure Appl. Logic*, 163(6):623–655, 2012.

[GM09] Guido Gherardi and Alberto Marcone. How incomputable is the separable Hahn-Banach theorem? *Notre Dame J. Form. Log.*, 50(4):393–425 (2010), 2009.

[Goh19] Jun Le Goh. *Measuring the Relative Complexity of Mathematical Constructions and Theorems*. PhD thesis, Cornell University, 2019.

[Goh20] Jun Le Goh. Embeddings between well-orderings: computability-theoretic reductions. *Ann. Pure Appl. Logic*, 171(6):102789, 17, 2020.

[Hir19] Jeffry L. Hirst. Leaf management. *Computability*, 2019.

[KMP20] Takayuki Kihara, Alberto Marcone, and Arno Pauly. Searching for an analogue of ATR_0 in the Weihrauch lattice. *The Journal of Symbolic Logic*, page 1–37, 2020.

[MV20] Alberto Marcone and Manlio Valenti. The open and clopen Ramsey theorems in the Weihrauch lattice, 2020. arXiv:2003.04245.

Email address: alberto.marcone@uniud.it

Decision Problems for Linear Recurrences in Computable Analysis

Eike Neumann

University of Oxford, UK

eike.neumann@cs.ox.ac.uk

A real linear recurrence sequence is a sequence $(u_k)_k$ of real numbers satisfying a linear recurrence relation of the form $u_{k+n} = c_1 u_{k+n-1} + \cdots + c_n u_k$. The Skolem Problem asks to decide whether such a sequence, given by an integer vector $c \in \mathbb{Z}^n$ of coefficients and an integer vector $u \in \mathbb{Z}^n$ of initial values, has a zero. It is generally believed to be decidable for linear recurrences of any order, but known to be decidable only up to order four [3, 4]. Two closely related decision problems of note are the Positivity Problem, which asks for given a linear recurrence to decide whether all of its terms are positive, and the Ultimate Positivity Problem, which asks for given a linear recurrence to decide whether all but finitely many of its terms are positive. In [2] it was shown that both Positivity and Ultimate Positivity are decidable up to order five. At the same time it was shown that a feasible algorithm for solving either Positivity or Ultimate Positivity at order six would entail major breakthroughs in the field of Diophantine approximation, making it highly unlikely for existing mathematical methods to allow for further progress to be made on these problems.

For a computable analyst it is natural to study these problems for real number inputs which are given as fast converging Cauchy sequences of rationals. More formally, we assume that we are given a linear recurrence as an element of the represented space $\coprod_{n\in\mathbb{N}} \mathbb{R}^{2n}$, where a point $(c,u) \in \coprod_{n\in\mathbb{N}} \mathbb{R}^{2n}$ represents the linear recurrence with coefficients c and initial values u.

Deciding membership for a subset $A \subseteq X$ of a represented space is clearly equivalent to computing its characteristic function $\chi_A \colon X \to \{0,1\}$. Since computable functions are continuous this is only possible if the characteristic function is constant on each connected component of X. This makes the real number versions of the above decision problems undecidable for trivial

reasons. The next best thing one can ask for is the existence of a computable function $\tilde{\chi}_A : X \to \{0, 1, \bot\}$, where $\bot$ is below 1 and 0 in the specialisation order, such that $\tilde{\chi}_A(x) \leq \chi_A(x)$ in the specialisation order for all $x \in X$ and $\tilde{\chi}_A(x) \geq g(x)$ for all computable $g : X \to \{0, 1, \bot\}$ with $g(x) \leq \chi_A(x)$. Call a set A *maximally partially decidable* if such a function $\tilde{\chi}_A$ exists. Note that if the space X is discrete then a subset of X is maximally partially decidable if and only if it is decidable.

I will show that the Skolem Problem, the Positivity Problem, and the Ultimate Positivity Problem with real number inputs are maximally partially decidable. I will also show that the maximal partial algorithms for deciding the Positivity Problem and the Ultimate Positivity Problem return a value different from $\bot$ almost everywhere. The full paper is available as [1].

References

[1] E. Neumann. Decision problems for linear recurrences involving arbitrary real numbers. `https://arxiv.org/abs/2008.00583`, 2020.

[2] J. Ouaknine and J. Worrell. Positivity problems for low-order linear recurrence sequences. In C. Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 366–379. SIAM, 2014.

[3] R. Tijdeman, M. Mignotte, and T. Shorey. The distance between terms of an algebraic recurrence sequence. *Journal für die reine und angewandte Mathematik*, 349:63–76, 1984.

[4] N. Vereshchagin. Occurrence of zero in a linear recursive sequence. *Mat. Zametki*, 38(2):177 – 189, 1985.

An update on Weihrauch complexity

Arno Pauly
Department of Computer Science
Swansea University, Swansea, UK
Arno.M.Pauly@gmail.com

I will give an update on progress in the study of Weihrauch degrees since the survey [2] in 2017, and the Dagstuhl meeting in 2018 [1], and highlight some open questions. No claim of completeness or unbiasedness is made.

Algebraic structure of the Weihrauch degrees In logical terms, the algebraic structure of the Weihrauch degrees is $(\mathfrak{W}; 0, 1, \top; \sqcap, \sqcup, \times, \star, \to, ^*, \widehat{\ }, ^\diamond)$, with the constants 0, 1, $\top$ being *vacuously true*, *true* and *false* respectively. The $\sqcap$-operator is *or*, and $\sqcap$, $\times$, $\star$ are all logical *and*s, the addittive, multiplicative and sequential *and* respectively. The right-residual to $\star$ is $\to$, we do not have additional implications. The unary "bang" operators *, $\widehat{\ }$, $^\diamond$ provide access to finitely many parallel instances, countably-many parallel instances and finitely-many consecutive instances. A persistent open question on these operations was recently solved:

Theorem 1 (Westrick [8])**.** $1 \leq_{\mathrm{W}} f$ and $f \equiv_{\mathrm{W}} f \star f$, iff $f \equiv_{\mathrm{W}} f^\diamond$.

Open Question 2. What can we say about $\mathrm{Th}(\mathfrak{W}; 0, 1, \top; \sqcap, \sqcup, \times, \star, \to, ^*, \widehat{\ }, ^\diamond)$? Is its quantifier-free part finitely axiomatisable? What about fragments of the signature?

Off the beaten track There is a canonic scaffold of familiar degrees in the Weihrauch lattice, the closed choice principles and the levels of the effective Baire hierarchy, which has served well for many classifications of computational problems. Recently however we have also studied some Weihrauch degrees that lie very much not in this explored part of the lattice. One such example is found in *overt choice* for incomplete countably-based spaces. This principle is given overt (aka positive) information about a closed subset, and needs to output a point in this set.

Theorem 3 (de Brecht, Schröder & P. [3])**.** Let $\mathrm{VC}_{\mathbb{Q}}$ denote overt choice on the rationals. Then

1. If $f : \mathbf{X} \rightrightarrows \mathbb{N}$ satisfies $f \leq_{\mathrm{W}} \mathrm{VC}_{\mathbb{Q}}$, then f is computable.
2. $\mathrm{VC}_{\mathbb{Q}}$ is non-uniformly computable.
3. $\mathrm{VC}_{\mathbb{Q}} \not\leq_{\mathrm{W}} \mathrm{C}_{\mathbb{R}}$

For overt choice $\mathrm{VC}_{\mathbf{X}}$ for non-Fréchet-Urysohn coPolish spaces $\mathbf{X}$ we only know that $\mathrm{LPO} \leq_{\mathrm{W}} \mathrm{VC}_{\mathbf{X}} \leq_{\mathrm{W}} (\widehat{\mathrm{isEmpty} : \mathcal{A}(\mathbb{N}^{\mathbb{N}}) \to \mathbb{S}})$ – which is perfectly compatible with $\mathrm{VC}_{\mathbf{X}}$ being a familiar Weihrauch degree, and is a remarkably weak classification.

Open Question 4 (de Brecht, Schröder & P. [3])**.** Can we classify overt choice for non-Fréchet-Urysohn coPolish spaces more precisely?

Another example for this is found in the principle DS that takes as input a linear order with a descending sequence, and then finds such a sequence. This principle sits, in a sense, beside the Baire hierarchy, as revealed by the following theorem:

Theorem 5 (Goh, P. & Valenti)**.**
1. $\lim <_{\mathrm{W}} \mathrm{DS} <_{\mathrm{W}} \mathrm{C}_{\mathbb{N}^{\mathbb{N}}}$
2. $\mathrm{UC}_{\mathbb{N}^{\mathbb{N}}} |_{\mathrm{W}} \mathrm{DS}$
3. If $f : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is a function with $f \leq_{\mathrm{W}} \mathrm{DS}$, then $f \leq_{\mathrm{W}} \lim$

An either-or power of Ramsey's theorem Let $\mathrm{NON} : \{0,1\}^{\mathbb{N}} \rightrightarrows \{0,1\}^{\mathbb{N}}$ be defined as $q \in \mathrm{NON}(p)$ iff q is not computable relative to p. The following shows, in a sense, that Ramsey's theorem for pairs has both the power to force non-computable solutions, and to exhibit functional discontinuity – but it cannot do both at once:

Theorem 6 (Dzhafarov, Goh, Hirschfeldt, Patey & P. [4])**.** $\mathrm{NON} \times \mathrm{LPO} \not\leq_{\mathrm{W}} \mathrm{RT}^2_2$

Belated closure under composition It may seem intuitive that natural Weihrauch degrees should either be closed under composition, or more and more iterations should give ever increasing power. We do know a few counterexamples to this. Let AoUC be the degree of finding a solution to $bx = a$ where $0 \leq a \leq b$, and let $\mathrm{List}_{\{0,1\}^{\mathbb{N}}}$ be the problem "given a countable closed subset of Cantor space, find an enumeration of its elements". We know:

Theorem 7 (Kihara & P. [6])**.** $\mathrm{AoUC}^* <_{\mathrm{W}} \mathrm{AoUC}^* \star \mathrm{AoUC}^* \equiv_{\mathrm{W}} \mathrm{AoUC}^{\diamond}$.

Theorem 8 (Kihara, Marcone & P. [5])**.** $\mathrm{List}_{\{0,1\}^{\mathbb{N}}} <_{\mathrm{W}} \mathrm{List}_{\{0,1\}^{\mathbb{N}}} \star \mathrm{List}_{\{0,1\}^{\mathbb{N}}} \star \mathrm{List}_{\{0,1\}^{\mathbb{N}}} \equiv_{\mathrm{W}} \mathrm{UC}_{\mathbb{N}^{\mathbb{N}}}$

Open Question 9. Is there a square-root operator on the Weihrauch degrees?

Separation techniques To make proving the non-existence of reductions more feasible, techniques that instead let us prove positive results are very useful. One such technique is based on the recursion theorem:

Theorem 10 (Kihara & P. [7])**.** Let $\mathbf{X}$ have a total precomplete representation. Let $f : \mathbf{X} \rightrightarrows \mathbf{Y}$ and $g : \mathbf{U} \rightrightarrows \mathbf{V}$ be such that there exists a computable $e : \mathbf{U} \times \mathcal{M}(\mathbf{V}, \mathbf{Y}) \rightrightarrows \mathbf{X}$ such that if $x \in e(u, k)$ and $v \in g(u)$, then $k(v) \not\subseteq f(x)$. Then $f \not\leq_{\mathrm{W}} g$.

References

[1] Vasco Brattka, Damir D. Dzhafarov, Alberto Marcone & Arno Pauly (2019): *Measuring the Complexity of Computational Content: From Combinatorial Problems to Analysis (Dagstuhl Seminar 18361). Dagstuhl Reports* 8(9), pp. 1–28. Available at `http://drops.dagstuhl.de/opus/volltexte/2019/10327`.

[2] Vasco Brattka, Guido Gherardi & Arno Pauly (2017). *Weihrauch Complexity in Computable Analysis.* arXiv 1707.03202.

[3] Matthew de Brecht, Arno Pauly & Matthias Schröder (2020): *Overt choice. Computability* Available at https://arxiv.org/abs/1902.05926.

[4] Damir D. Dzhafarov, Jun Le Goh, Denis. R. Hirschfeldt, Ludovic. Patey & Arno Pauly (2020): *Ramsey's theorem and products in the Weihrauch degrees. Computability* 9(2).

[5] Takayuki Kihara, Alberto Marcone & Arno Pauly (202X): *Searching for an analogue of ATR in the Weihrauch lattice. Journal of Symbolic Logic* .

[6] Takayuki Kihara & Arno Pauly (2016): *Dividing by Zero – How Bad Is It, Really?* In: Piotr Faliszewski, Anca Muscholl & Rolf Niedermeier, editors: *41st Int. Sym. on Mathematical Foundations of Computer Science (MFCS 2016)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 58, Schloss Dagstuhl, pp. 58:1–58:14.

[7] Takayuki Kihara & Arno Pauly (2019). *Convex choice, finite choice and sorting.* arXiv 1905.03190.

[8] Linda Westrick (202X): *A note on the diamond operator. Computability* To appear, arXiv:2001.09372.

Contributed Talks

Clerical: an Imperative Language for Verified Real-number Computation *

Andrej Bauer[1], Sewon Park[2], Alex Simpson[1]

[1] University of Ljubljana, Slovenia [2] KAIST, South Korea

Imperative programming is a natural and ubiquitous programming style, which supports a well-established precondition-postcondition-based program verification methodology. Given the potential usefulness of exact-real-number computation, one would like to extend the imperative programming style to naturally accommodate both this style of computation and its verification.

On the programming side, this has been most comprehensively addressed by Müller's iRRAM [3], which provides a C++ library for exact real-number computation, based on the abstraction of *real-valued random access memory*: program variables are thought of as storing exact real values. This allows a natural imperative style of programming for real-number computation. The verification side is less established. Nevertheless, Brauße *et. al.* study a programming language fragment (influenced by iRRAM) from the viewpoint of program verification [1]. Formal rules are provided for the derivation of Hoare triples, with illustrative examples that include the verification of a root-finding program.

Our work began as an attempt to fill a lacuna in [1]. Programs in *op. cit.* are able to accept real numbers as input values stored in variables, but the language does not itself provide any mechanism for assigning arbitrary computable real numbers to variables. Because of this, one cannot compose programs by using one program to set real-number variables to pass as input to another program. In contrast, the iRRAM library does provide such functionality in the form of limit operators, which take functions encoding sequences of approximations as their arguments.

In this paper we propose *Clerical* (Command-Like Expressions for Real Infinite-precision Calculations) as a streamlined imperative language for real-number computation, that combines real-valued variables with a limit operation. We provide this language with formal semantics in all forms: operational, denotational and axiomatic, with the latter providing verification principles.

Our limit operation is given by a construction of the form $\texttt{lim}(n\!:\!\mathsf{Z}, e)$, where e is an expression that defines a real value parametrically in an integer variable n (Z is the type of integers). On the assumption that the induced function $n \mapsto e$ defines a (rapid) Cauchy sequence for nonnegative n, the limit operation defines its limit as a value of type R, the type of reals. The above formulation of `lim` allows us to avoid including first-class functions in the language, in keeping with our aim to have a simple imperative language. (In iRRAM, the limit operator is applied to a function argument.)

In order to be able to define a useful class of limits in the above way, it is essential that the expression language is rich enough for e to be able to define interesting functions $n \mapsto e$. To achieve this, we make our expressions *command-like*, in that they are allowed to contain loops and perform arbitrary computation. Indeed, in Clerical, expressions subsume *commands*, with the latter being defined as expressions of unit type. In general, such command-like expressions may have side effects. Nevertheless, it is important to distinguish between uses of expressions in which side-effects are allowed (e.g., when they are used as commands), and uses in which *pure* (i.e., side-effect-free) expressions are required. For example, the expression e in $\texttt{lim}(n\!:\!\mathsf{Z}, e)$ is required to be pure, as the value of the expression e, on different values of n, must not depend on the strategy for evaluating such approximating expressions, which is considered implementation-specific, and hence left unspecified. Further, our language includes a nondeterministic guarded case operation (see below), and the expressions used as guards must also be pure, so that there is no interference between guard expressions being evaluated in parallel. In Clerical, purity is achieved, by permitting pure expressions to read from but not write to the shared store, but allowing them to dynamically create their own local read-write store to use for calculation purposes.

Nondeterminism is unavoidable in real-number computation [2]. A primitive construct for nondeterminism has to be provided by the language, in order to overcome the inherent partiality in order comparisons between exact real numbers. Our source of nondeterminism is given by Dijkstra's guarded nondeterministic case construction, which has the form $\texttt{case}\ b_1 \Rightarrow c_1 \mid b_2 \Rightarrow c_2\ \texttt{end}$ where b_1, b_2 are pure Boolean expressions and c_1, c_2 are (possibly side-effecting) expressions. The intended meaning is that c_1 may execute if e_1 is true, and c_2 may execute if b_2 is true. When b_1 and b_2 both hold, either branch may

* This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143, and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2016K1A3A7A03950702)

execute nondeterministically. Even when one of the evaluations of the guards does not terminate, if the other holds, the corresponding branch executes. For example, we can write a program for an approximate test $x > y$ with precision 2^{-n} by $\mathsf{case}\ x > y + 2^{-n} \Rightarrow \mathsf{true} \mid y > x + 2^{-n} \Rightarrow \mathsf{false}\ \mathsf{end}$.

The nondeterminism of the case construct and the presence of the limit operator make the specification of semantics for Clerical nontrivial. Our approach is to only specify semantics for expressions whose value is well-defined independently of implementation choices, such as the strategy for evaluating the sequence of expressions approximating a limit. Accordingly, some expressions are declared to be semantically *ill-defined*; for example, those in which the limit operator is applied to expressions that do not specify a rapid Cauchy sequence. Expressions that are semantically *well-defined* are, in general, nondeterministic, and different nondeterministic branches may evaluate to values of the appropriate type, or alternatively fail to terminate. Accordingly, when A is the set of values of a type, we define the denotation of an expression of that type to be: $\emptyset$ if the expression is ill-defined; and the set of values (including $\bot$ for nontermination) obtained along all execution branches otherwise. The set of all such possible denotations is: $\mathcal{P}_\star(A) := \{S \subseteq A \cup \{\bot\} \mid S \text{ is finite or } \bot \in S\}$. We endow this with a partial order

$$X \sqsubseteq Y \quad \Leftrightarrow \quad (\bot \notin X \text{ and } X = Y) \text{ or } (\bot \in X \text{ and } X \subseteq Y \cup \{\bot\}) \text{ or } (\bot \in X \text{ and } Y = \varnothing)$$

giving a minor variant of the *Plotkin powerdomain* [4] over the flat domain $A \cup \{\bot\}$. Using the above domain, the denotation of a while loop $\mathtt{while}\ b\ \mathtt{do}\ c\ \mathtt{end}$ can be defined as usual as the least fixed point of a continuous operator. (This turns out to be possible, even though the denotation of nondeterministic case expressions $\mathtt{case}\ b_1 \Rightarrow c_1 \mid b_2 \Rightarrow c_2\ \mathtt{end}$ is not monotone in the guard expressions b_1 and b_2.)

For our verification calculus, we write $\{\phi\}\ e\ \downarrow\{y{:}\tau \mid \psi\}$ to express a *total correctness* specification: for any initial state satisfying the precondition ϕ, the expression e of type τ is well defined and it necessarily terminates in a value y that satisfies the postcondition ψ in the final state. Our axiomatic semantics consists of inference rules for proving such specifications. In the case of Clerical, proofs of total correctness sometimes require partial correctness assertions to be proved as subgoals. Thus we also provide proof rules for *partial correctness specifications* $\{\phi\}\ e\ \{y{:}\tau \mid \psi\}$, meaning: for any state satisfying ϕ, the expression e is well defined and every possible termination value $y{:}\tau$ satisfies ψ in the final state.

The following example illustrates why rules for both partial and total correctness are required.

$$\frac{\{\phi_i\}b_i\downarrow\{y{:}\mathsf{B} \mid y = \mathsf{true}\}\quad \{\theta_i\}b_i\{y{:}\mathsf{B} \mid y = \mathsf{false}\}\quad \{(\phi_1 \vee \phi_2) \wedge \neg\theta_i\}c_i\downarrow\{y{:}\tau \mid \psi\}\quad \text{for } i = 1, 2.}{\{\phi_1 \vee \phi_2\}\ \mathtt{case}\ b_1 \Rightarrow c_1 \mid b_2 \Rightarrow c_2\ \mathtt{end}\ \downarrow\{y{:}\tau \mid \psi\}}$$

Here B is the type of Booleans. By partial correctness, θ_i contains only states in which b_i is $\{\mathsf{false}\}$, $\{\mathsf{false}, \bot\}$ or $\{\bot\}$. Hence, $\neg\theta_i$ contains all states for which b_i contains true as a possible value. By total correctness, if ϕ_i holds, b_i is $\{\mathsf{true}\}$. So the case expression necessarily continues via c_1 or c_2, for any state satisfying $\phi_1 \vee \phi_2$. Moreover, it is only possible for the branch c_i to be taken in the case that the state also satisfies $\neg\theta_i$.

Operational semantics is a method of mathematically specifying operational behaviour while abstracting away from arbitrary implementation details. In the case of Clerical, we define the operational semantics as a *small-step* transition relation of single computation steps between expressions. Our semantics does not commit to any specific implementation of the real-number datatype. It also does not enforce any particular protocol for evaluating the parallel guards in a case expression. Instead, we impose only the general constraint that infinite execution sequences have to be *fair*: computation cannot continue indefinitely inside just one guard of a case expression as long as it is possible for computation steps to also be made in the other guard. For well-defined expressions, the denotational semantics can be recovered operationally as the set of result values arising from all finite completed execution sequences, together with $\bot$ if there exists a fair infinite execution sequence.

References

1. Brauße, F., Collins, P., Kanig, J., Kim, S., Konečnỳ, M., Lee, G., Müller, N., Neumann, E., Park, S., Preining, N., et al.: Semantics, logic, and verification of "exact real computation". arXiv preprint arXiv:1608.05787 (2016)
2. Luckhardt, H.: A fundamental effect in computations on real numbers. Theoretical Computer Science **5**(3), 321 – 324 (1977). https://doi.org/10.1016/0304-3975(77)90048-2, http://www.sciencedirect.com/science/article/pii/0304397577900482
3. Müller, N.T.: The iRRAM: Exact arithmetic in C++. In: International Workshop on Computability and Complexity in Analysis. pp. 222–252. Springer (2000)
4. Plotkin, G.D.: A powerdomain construction. SIAM Journal on Computing **5**(3), 452–487 (1976)

THE DISCONTINUITY PROBLEM

VASCO BRATTKA

Abstract. We present a natural candidate for the simplest natural unsolvable problem and we call this problem the *discontinuity problem*.

1. Summary

There is a simplest discontinuous function $\mathsf{LPO} : \mathbb{N}^{\mathbb{N}} \to \{0,1\}$, which is the test for the zero sequence, i.e., the characteristic function of the set $\{0^{\mathbb{N}}\}$, also known as *limited principle of omniscience*. In fact, a function $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is discontinuous, if and only if LPO is continuously Weihrauch reducible to F [Wei00, Lemma 8.2.6].

However, there are many discontinuous and hence non-computable multi-valued problems $f :\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}$ to which LPO is not reducible. Examples are the *intermediate value theorem* IMT or *weak Kőnig's lemma* WKL [BGP17]. Hence, it is natural to ask whether there is a simplest discontinuous, i.e., unsolvable, multi-valued problem $f :\subseteq \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}$. More generally, by a *problem* $f :\subseteq X \rightrightarrows Y$ we mean a multi-valued map on represented spaces X, Y that admits a realizer. We call a problem *continuous*, if it has a continuous realizer and *computable* or *solvable*, if it has a computable realizer.

So far, the literature on Weihrauch complexity has known the *non-computability problem* $\mathsf{NON} : \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}, p \mapsto \{q \in \mathbb{N}^{\mathbb{N}} : q \not\leq_{\mathrm{T}} p\}$ and the *all-or-co-unique choice problem* $\mathsf{ACC}_{\mathbb{N}}$ (also known as $\mathsf{LLPO}_{\mathbb{N}}$) as the most simple (but incomparable) discontinuous problems [BHK17, BGP17].

We claim that the problem

$$\mathsf{DIS} : \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}, p \mapsto \{q \in \mathbb{N}^{\mathbb{N}} : \mathsf{U}(p) \neq q\}$$

is the simplest "natural" total such problem, where $\mathsf{U} :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ is a universal computable function. Indeed, DIS is a common lower bound of NON and $\mathsf{ACC}_{\mathbb{N}}$. This claim is witnessed by a number of further results.

Theorem 1.1. *A problem $f :\subseteq X \rightrightarrows Y$ is effectively discontinuous if and only if $\mathsf{DIS} \leq^*_{\mathrm{W}} f$.*

Here $\leq^*_{\mathrm{W}}$ denotes the continuous version of Weihrauch reducibility and f is called *effectively discontinuous* if there is a continuous function $D : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ that witnesses discontinuity of f in the sense that for every potential realizer Φ_p of f the value $D(p)$ is an input for f on which f is not correctly realized by Φ_p. Here Φ denotes some standard representation of the continuous functions $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ with the property that $\Phi_p(q) = U\langle p, q\rangle$.

The theorem above shows that DIS is at least the simplest effectively discontinuous problem in terms of Weihrauch reducibility. The theorem can be proved in Zermelo-Fraenkel set theory ZF with the axiom of dependent choice DC. If one adds the axiom of determinacy AD [Mos09], then one can go further than that.

Theorem 1.2. *In $\mathsf{ZF} + \mathsf{DC} + \mathsf{AD}$ every total problem $f : \mathbb{N}^{\mathbb{N}} \rightrightarrows \mathbb{N}^{\mathbb{N}}$ is either continuous or effectively discontinuous.*

Date: June 1, 2020.

Hence, in $\mathsf{ZF}+\mathsf{DC}+\mathsf{AD}$ the problem DIS is actually the simplest total discontinuous problem on Baire space. In particular, this implies that DIS is the simplest discontinuous problem among all (also partial) problems with respect to the continuous version of total Weihrauch reducibility [BG18]. However, with the full axiom of choice AC it is easy to construct counterexamples to this claim.

Example 1.3. In $\mathsf{ZFC}=\mathsf{ZF}+\mathsf{AC}$ there exists a total discontinuous problem $f:\mathbb{N}^{\mathbb{N}}\rightrightarrows\mathbb{N}^{\mathbb{N}}$ that is not effectively discontinuous.

ZFC and $\mathsf{ZF}+\mathsf{DC}+\mathsf{AD}$ are known as alternative but incompatible axiomatic settings for mathematics [Mos09]. Hence, it is actually a matter of the axiomatic setting of how "natural" DIS appears as a simplest unsolvable problem.

Independent of the axiomatic setting DIS happens to have a number of further very natural properties, among those the following.

Theorem 1.4. $\widehat{\mathsf{DIS}}\equiv_{\mathrm{sW}}\mathsf{NON}$.

That is, the *parallelization* $\widehat{\mathsf{DIS}}$ of DIS is equivalent to the non-computability problem, which leads to the suggestive slogan that "non-computability is the parallelization of discontinuity". In fact, studying the discontinuity problem reveals further insights into the nature of the relation between continuity and computability of which we have just scratched the surface.

One further relation between the discontinuous problems mentioned here is that it turns out that DIS is the summation of $\mathsf{ACC}_{\mathbb{N}}$. Here *summation* is a newly introduced interior operator in the Weihrauch lattice that plays a dual rôle to parallelization. Hence, the three weakest discontinuous problems discussed here, $\mathsf{DIS},\mathsf{ACC}_{\mathbb{N}}$ and NON are all related to each other in a very natural way.

References

[BG18] Vasco Brattka and Guido Gherardi. Weihrauch goes Brouwerian. arXiv 1809.00380, 2018.

[BGP17] Vasco Brattka, Guido Gherardi, and Arno Pauly. Weihrauch complexity in computable analysis. arXiv 1707.03202, 2017.

[BHK17] Vasco Brattka, Matthew Hendtlass, and Alexander P. Kreuzer. On the uniform computational content of computability theory. *Theory of Computing Systems*, 61(4):1376–1426, 2017.

[Mos09] Yiannis N. Moschovakis. *Descriptive Set Theory*, volume 155 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, Rhode Island, second edition, 2009.

[Wei00] Klaus Weihrauch. *Computable Analysis*. Springer, Berlin, 2000.

Faculty of Computer Science, Universität der Bundeswehr München, Germany and Dept. of Mathematics & Appl. Math. University of Cape Town, South Africa[1]

Email address: Vasco.Brattka@cca-net.de

[1]Vasco Brattka is supported by the National Research Foundation of South Africa.

Computability and adjunctions of Euclidean spaces

Matea Čelar and Zvonko Iljazović

A compact set S in a computable topological space $(X, \mathcal{T}, (I_i))$ is computably enumerable (or overt) if we can effectively enumerate all base elements I_i which intersect S. A compact set S is semicomputable if we can effectively enumerate all finite unions of base elements which contain S. We say that a set is computable in a computable topological space if it is computably enumerable and semicomputable. This definition corresponds with the classical definition of a computable set in the Euclidean space and, more generally, in computable metric spaces.

We examine topological conditions under which semicomputability of a set implies its computable enumerability and, therefore, computability. For certain sets S it is sufficient to study such conditions locally around each point $x \in S$. This follows from the fact that computable enumerability is, in a way, a local property: every compact locally computably enumerable set is computably enumerable. A set is locally computably enumerable if it is computably enumerable at every point x. A set S is computably enumerable at x if there exists a neighborhood U of x in S and a c.e. set $\Omega \subseteq \mathbb{N}$ such that

$$I_i \cap U \neq \emptyset \;\Rightarrow\; i \in \Omega \quad \text{and} \quad i \in \Omega \;\Rightarrow\; I_i \cap S \neq \emptyset.$$

In this case we say that U is computably enumerable up to S. For example, if a point $x \in S$ has a neighborhood in S homeomorphic to $\mathbb{R}^n$, then S is computable at x. This implies that every semicomputable manifold in a computable topological space is computable. Similarly, it holds that a semicomputable manifold with semicomputable boundary is computable [2].

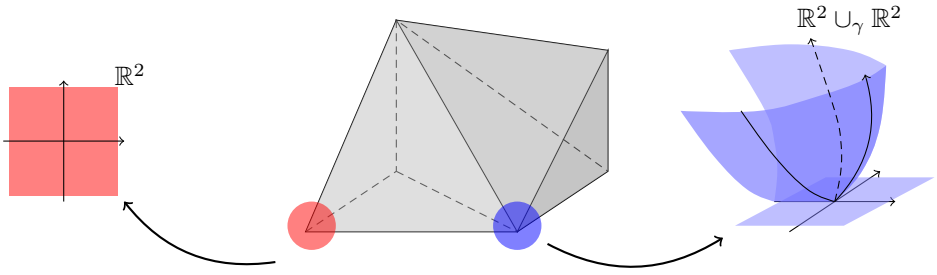

Figure 1: Neighborhoods of points in a polyhedron

We want to expand this result to a more general family of sets – for example, sets with topological type of polyhedra. Let P be a semicomputable topological polyhedron in a computable topological space. Some points in P have neighborhoods homeomorphic to $\mathbb{R}^n$ or to upper half-space $\mathbb{H}^n = \{(x_1, \ldots, x_n) \mid x_n \geq 0\}$. The results mentioned above guarantee that P is computably enumerable at every such point. However, other points in P have neighborhoods homeomorphic to multiple copies of $\mathbb{R}^n$ or $\mathbb{H}^n$ attached (or "glued together") along some subspace (Figure 1). In certain such cases, a similar result holds.

The concept of attaching two spaces together is formalized by the notion of adjunction space. Let X and Y be topological spaces, let A be a subspace of X and let $f : A \to Y$ be a continuous function. By $X \cup_f Y$ we denote the space obtained from the disjoint union of X and Y by identifying a with $f(a)$ for all $a \in A$, and we call this space adjunction space. We have the following theorem.

Theorem 1. *Let $n \geq 2$ and let A be a closed subset of $\mathbb{R}^{n-1} \times \{0\}$ such that $(0, \ldots, 0) \in A$. Let Y be a locally compact Hausdorff space and let $\gamma : A \to Y$ be an embedding such that $\gamma(A)$ is closed in Y. Suppose $(X, \mathcal{T}, (I_i))$ is a computable topological space and $f : \mathbb{R}^n \cup_\gamma Y \to X$ is an embedding such that $f(\mathbb{R}^n \cup_\gamma Y)$ is an open subset of a semicomputable set S. Then the set $f([-1, 1]^n)$ is c.e. up to S.*

As a consequence we get the following result: if P is a space obtained by gluing the boundaries of unit cubes ∂I^n and ∂I^m together along their k-dimensional faces A and B, then each semicomputable set S (in a computable topological space) homeomorphic to P is computable.

Acknowledgements

This work has been fully supported by Croatian Science Foundation under the project 7459 CompStruct.

References

[1] V. Brattka and G. Presser. Computability on subsets of metric spaces. *Theoretical Computer Science*, 305:43–76, 2003.

[2] Z. Iljazović and I. Sušić. Semicomputable manifolds in computable topological spaces. *Journal of Complexity*, 45:38–114, 2018.

[3] Z. Iljazović. Computability of graphs. *Mathematical Logic Quarterly*, 66:51–64, 2020.

[4] K. Weihrauch and T. Grubba. Elementary Computable Topology. *Journal of Universal Computer Science*, 15:1381-1422, 2009.

Weihrauch reducibility around Π^1_1-CA_0

Vittorio Cipriani
Universitá degli studi di Udine
Dipartimento di Scienze Matematiche, Informatiche e Fisiche (DMIF)
Via delle Scienze 206, 33100 Udine, Italy
cipriani.vittorio@spes.uniud.it

Abstract

In this paper, using the framework of Weihrauch reducibility, we aim to provide a fine analysis of some mathematical theorems that, among the *big five* axiom systems in reverse mathematics, are equivalent to Π^1_1-CA_0 ([Sim09]). Recently, there has been growing interest in finding possible counterparts to ATR_0 in the Weihrauch degrees. In [KMP18] several candidates have been examined, and in [dK19] and [Goh19] the authors continued in this direction. Indeed, it is natural to proceed in investigating "stronger" principles from the reverse mathematics point of view. For example, the natural function that represents Π^1_1-CA_0 in the Weihrauch degrees is the one that maps a countable sequence of trees to the characteristic function of the set of indices corresponding to well-founded trees. Recently in [Hir18], the author showed its Weihrauch equivalence with PK, the function that takes as input a tree and outputs its perfect kernel. Our goal is to study the Weihrauch degrees of multivalued functions arising from statements equivalent to Π^1_1-CA_0. This is joint work with Alberto Marcone and Manlio Valenti.

References

[dK19] Paul-Elliot Angls d'Auriac and Takayuki Kihara. A comparison of various analytic choice principles, 2019.

[Goh19] Jun Le Goh. Some computability-theoretic reductions between principles around ATR_0, 2019.

[Hir18] Jeffry L. Hirst. Leaf management, 2018.

[KMP18] Takayuki Kihara, Alberto Marcone, and Arno Pauly. Searching for an analogue of atr in the weihrauch lattice, 2018.

[Sim09] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Logic. Cambridge University Press, 2 edition, 2009.

ON COMPUTABILITY OVER METRIC BUNDLES

Rafael A. García*

***A*bstract.** This paper introduces and investigates TTE-style computability over metric bundles over a topological space T.

***K*eywords** metric bundle · computable topological space · bundle of computable metric spaces · computable nundle of metric spaces

1 Introduction

The Category of Metric Bundles on a topological space T is a generalization of the Category of Metric Spaces. [2] presents an analytical construction of the Metric Bundle of the Upper Semicontinuous Functions, and shows that this bundle plays the similar role to the *real numbers object* in the Category of Metric Spaces, and that contains, as a section, the distance between two arbitrary sections of a metric bundle on T. [5, 3] are the recommended references on bundles of uniform spaces.

The Type-2 Theory of Effectivity (TTE for short) offers the fundamentals for computable analysis and has allowed us to study computability on both points and functions sets. The main idea is to define computations on real numbers via representations. This work assumes that the reader is familiarized with TTE. [6, 4] are complete references on TTE.

This paper introduces the definition of *Bundle of Computable Metric Spaces*, explores the effects of imposing computability conditions on the base space of a metric bundle, demostrates a theorem of existence of T_0 effective bundles of metric spaces, and demostrates the admissibility of a representation for these bundles.

This paper presents some results from the author's PhD-thesis [1]. The work studies the effectivity content in the category of metric bundles in the framework of TTE. The goal is to import, develop and study a notion of computability in the category of metric bundles over a topological space T.

For brevity, the proofs are systematically omitted. Complete proofs are available at [1].

2 Bundles of computable metric spaces

Definition 1 Let $p : G \longrightarrow T$ be a surjective function. A *selection* for p is a function $\alpha : Q \longrightarrow G$, with $Q \subseteq T$, such that $p \circ \alpha$ is the identity map on Q. If $Q = T$, α is a *global selection*. If T is a topological space and Q is an open subset of T, α is a *local selection*. When both G and T are topological spaces, a continuous selection is called a *section* for p.

Definition 2 A function $d : G \times G \longrightarrow [0, +\infty]$ such that, for all u, v, $w \in G$, $d(u, v) = +\infty \iff p(u) \neq p(v)$, $d(u, v) = 0 \iff u = v$, $d(u, v) = d(v, u)$, and, $d(u, v) \leq d(u, w) + d(w, v)$, is called a *metric* for p. If $\alpha : Q \longrightarrow G$ is a selection for p, $\mathcal{T}_\epsilon(\alpha) = \{u \in G \mid p(u) \in dom(\alpha) \wedge d(u, \alpha(p(u))) < \epsilon\}$ is called the *ϵ-tube around α*. Note that, for each $t \in T$, $G_t := p^{-1}(t)$ is a metric space, when endowed with the restriction of d to $G_t \times G_t$.

Definition 3 Let G and T be topological spaces, let $p : G \longrightarrow T$ be a continuous surjective function and let d be a metric for p, such that for every $u \in G$ and every $\epsilon > 0$, there is a local section α such that $u \in \mathcal{T}_\epsilon(\alpha)$. Then (G, p, T) is called a *bundle of metric spaces*, provided that the collection of sets $\mathcal{T}_\epsilon(\alpha)$, where $\epsilon > 0$ and α runs throughout the local sections for p, is a base for the topology of G. The space T is called the *base space*, $G_t = p^{-1}(t)$ is called the *fiber* above the point t, for each $t \in T$, and G is called the *fibered space*.

The next existence theorem is a particular case of the existence theorem of uniform bundles [5], and allow us to construct metric bundles (bundles of metric spaces) when the data are provided by a topological space T, a set G, and a family Γ of local selections for a surjective function $p : G \longrightarrow T$.

Theorem 1 *Let T be a topological space, $p : G \longrightarrow T$ be a surjective function, d be a metric for p and Γ be a family of local selections for p. Assume that for every $u \in G$ and every $\epsilon > 0$ there exists $\alpha \in \Gamma$ such that $u \in \mathcal{T}_\epsilon(\alpha)$, and, for every $\alpha, \beta \in \Gamma$, the function $\Phi_{\alpha\beta} : dom(\alpha) \cap dom(\beta) \longrightarrow \overline{\mathbb{R}}$, defined by $\Phi_{\alpha\beta}(t) = d(\alpha(t), \beta(t))$, is upper semicontinuous. Then G can be equipped with a topology $\mathfrak{T}$ such that the family $\mathcal{B} = \{\mathcal{T}_\epsilon(\alpha_Q) \mid \epsilon > 0 \wedge \alpha \in \Gamma \wedge Q \subseteq dom(\alpha)\}$, with $\alpha_Q = \alpha \restriction_Q$, is a base for $\mathfrak{T}$; every $\alpha \in \Gamma$ is a section; and (G, p, T) is a bundle of metric spaces.*

*Faculty of Engineering, Design and Innovation. Politécnico Grancolombiano. rgarcia@poligran.edu.co

Definition 4 ***(Bundle of Computable Metric Spaces - BCMS).*** Let (G, p, T) be a metric bundle. Then (G, p, T) is called a *Bundle of Computable Metric Spaces* if and only if, for all $t \in T$, there exists Q_t and ν_t such that $\mathfrak{G}_t = (G_t, d \restriction_{G_t \times G_t}, Q_t, \nu_t)$ is a computable metric space, where $G_t = p^{-1}(t)$ and $d_t = d \restriction_{G_t \times G_t}$.

The term BCMS will be used to refer to Metric Bundles whose fibers are computable metric spaces. It is worth noting that no computability conditions are imposed on the base space or on the fibered space. Some useful and non-trivial examples of bundles of computable metric spaces are presented in [1].

3 Metric Bundles with Computable Base

This section explores the effects of imposing computability conditions on the base space X of the metric bundle (G, p, X). It will be assumed that $\mathfrak{X} = (X, \tau, \beta, \nu)$ is a computable T_0 space with standard representation $\delta_{\mathfrak{X}}$. Theorems 4 and 6 are the main results.

Definition 5 ***(Cauchy representation for*** G ***according to*** Γ***).*** If G has cardinality not greater than the continuum, $p : G \longrightarrow X$ is a surjective function, d is a metric for p, Γ is a countable family of local selections for p such that for each $u \in G$ and $n \in \mathbb{N}$ there exists $\alpha \in \Gamma$ with $u \in \mathcal{T}_{2^{-n}}(\alpha)$, and, a notation ν_Γ for Γ, then define $\delta_{\Gamma C}$, the *Cauchy representation for G according* Γ by $\delta_{\Gamma C}(q) = u$, if and only if, there exists $a \in dom(\delta_{\mathfrak{X}})$ and $z_0, z_1, \cdots \in dom(\nu_\Gamma)$, such that $q = \langle a, \iota(z_0)\, \iota(z_1) \ldots \rangle$, $\delta_{\mathfrak{X}}(a) = p(u)$, and, $d(u, \nu_\Gamma(z_n)(p(u))) < 2^{-n}$ for all $n \in \mathbb{N}$.

In the following it will be assumed that G, p, d, Γ and ν_Γ are satisfy the conditions impossed in the previous definition of $\delta_{\Gamma C}$. Note that G is a set wiht cardinality is not greater than the continuum, and that p is a $(\delta_{\Gamma C}, \delta_{\mathfrak{X}})$-computable function since $p = \pi_1 \circ \delta_{\Gamma C}$.

Theorem 2 *If d is $(\delta_{\Gamma C}, \delta_{\Gamma C}, \rho)$-computable then (G, p, X) is a bundle of computable metric spaces.*

Theorem 3 *If $\nu_{\mathbb{N}}$ is a standard notation for $\mathbb{N}$, then, there exists a function $(\delta_{\Gamma C}, \nu_{\mathbb{N}}, \nu_\Gamma)$-computable function SEL such that, if $u \in G$ and $n \in \mathbb{N}$ are given, then SEL returns a ν_Γ-name for a selection γ in Γ such that $u \in \mathcal{T}_{2^{-n}}(\gamma)$. In other words, the first condition of the definition of bundle of computable metric spaces is a computable condition.*

Now, it will be understood that a bundle of metric spaces (G, p, X) is a *Computable Bundle of Metric Spaces - CBMS* if the fibered space G is a computable topological space. In this sense, the next theorem is a *Theorem of Existence of T_0 Effective Bundles of Metric Spaces*.

Theorem 4 *Let $\mathfrak{X} = (X, \tau, \beta, \nu)$ be a T_0 effective topological space, G be nonempty set, $p : G \longrightarrow X$ be a surjective function, d be a metric for p, Γ be a countable collection of local selections for p, and ν_Γ be a notation for Γ with recursively enumerable domain. Assume that for all $u \in G$ and $n \in \mathbb{N}$ there exists a local selection $\gamma \in \Gamma$ such that $u \in \mathcal{T}_{2^{-n}}(\gamma)$, and, for all $\gamma, \zeta \in \Gamma$ the function $\Phi_{\gamma\zeta} : dom(\gamma) \cap dom(\zeta) \longrightarrow \mathbb{R}$, defined by $\Phi_{\gamma\zeta}(t) = d(\gamma(t), \zeta(t))$, is upper semicontinuous. Then G can be equipped with a topology τ_G such that the family $\beta_G = \{\mathcal{T}_{2^{-n}}(\gamma_Q) \mid n \in \mathbb{N} \wedge \gamma \in \Gamma \wedge Q$ is a basic open subset of $dom(\gamma)\}$ is a base for τ_G; every $\gamma \in \Gamma$ is a section; (G, p, X) is a bundle of metric spaces; there exists a notation ν_G for β_G with recursively enumerable domain and such that $\mathfrak{G} = (G, \tau_G, \beta_G, \nu_G)$ is a T_0 effective topological space with standard representation δ_G; and, the projection p is $(\delta_G, \delta_{\mathfrak{X}})$-computable.*

Theorem 5 $\delta_G \leq \delta_{\Gamma C}$.

Theorem 6 *$\delta_{\Gamma C}$ is τ_G-admissible.*

References

[1] Rafael García. *Sobre computabilidad en campos de espacios métricos*. PhD thesis, Universidad Nacional de Colombia, 2018. Mención Meritoria.

[2] Rafael García, Edilberto Reyes, and Januario Varela. A semicontinuous continuum. *Boletín de Matemáticas, Nueva Serie*, XII(1):1–18, 2005.

[3] Clara Neira and Januario Varela. On separation axioms of uniform bundles and sheaves. *Applied General Topology*, 5(2):155–171, 2004.

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

[5] Januario Varela. On the existence of uniform bundles. *Revista Colombiana de Matemáticas*, XXIX:95–101, 1995.

[6] Klaus Weihrauch. *Computable Analysis: An Introduction*. Springer-Verlag New York, Inc., 2000.

Happy Birthday iRRAM*

Considerations for the Future of *Exact Real Computation*

Jiman Hwang, Svetlana Selivanova, and Martin Ziegler

KAIST, School of Computing

Abstract. Twenty years ago at CCA 2000, iRRAM was released and presented to the public: An opportunity to celebrate, to recollect lessons learned — and to start planning for the next major revision.

1 Introduction

Among software packages for Reliable Computing [1,7], `iRRAM` stands out for supporting real numbers as entities that can be operated on exactly, i.e., without rounding errors. Unlike intervals or floating point numbers (which violate for instance the distributive law), `iRRAM` thus realizes the intuitive Blum-Shub-Smale Model, aka *real-RAM* in Computational Geometry — with the minor caveat of comparisons being necessarily partial (which can be 'mended' with the multivalued *choose* [5]).

This semantics is theoretically justified [2], while hiding the underlying cumbersome Turing machine model. `iRRAM` thus provides (rare) evidence that Computable Analysis indeed is relevant to practice! Common criticism claims that real numbers as infinite objects cannot be realized on inherently finite digital computers; however this seems valid only on the operational level:

1.1 Exact Real Computation

Denotationally an Abstract Data Type `REAL` suffices to only appear as, namely behave indistinguishable from, actual exact real numbers. And for any finite execution of a user program, this can be achieved by approximations — as long as their precision is chosen suitably high. Indeed, traditional numerical programming leaves it to the user (program) to determine and set this as working precision, for example in `MPFR`. In general this requires taking into account both tail/truncation errors and error propagation. The latter can be traced using interval arithmetic to see whether the initial precision was sufficient — a posteriori. Some libraries, like `core` [18] or `realLib` [9], record the user operations in a possibly huge dynamic data structure: which can then be analyzed to automatically determine a sufficient precision parameter first, and finally evaluate said data structure approximately in said working precision. `iRRAM` trades costly memory for runtime by repeatedly and transparently restarting the user program with progressively increased working precisions until the result satisfies the desired error bound.

The above libraries constitute different approaches to the same goal, to realize the paradigm of *Exact Real Computing* [14]: to enable the user writing numerical programs similar to proofs in Calculus, such as of the Intermediate Value Theorem or the Banach Fixedpoint Theorem. Such proofs build on exact real numbers and

* Supported by the National Research Foundation of Korea (grant 2017R1E1A1A03071032), by the International Research & Development Program of the Korean Ministry of Science and ICT (grant 2016K1A3A7A03950702), and by the NRF Brain Pool program (grant 2019H1D3A2A02102240). We thank Franz Brausse for helpful advice and feedback.

their logical and analytic properties; they thus (can and do) ignore how the reals in turn are approximated by Dedekind Cuts or as Cauchy Sequences of rationals. Similarly, a program' operating on integers should be written and run without knowing (needing to know) how a data type like `BigInt` is implemented on top of the hardware. This added level of abstraction [12, p.169] in the end simplifies numerical software development, reduces programming mistakes, and supports elegant formal verification.

20 years after its 'official' release [13] during a phase where *many* libraries for reliable computing were conceived, `iRRAM` prevails as arguably the only one in active development supporting *Exact Real Computation*: Congratulations, Norbert!

2 Future Perspectives

Exact Real Computation consists of three layers: (i) on top, programs in user space can (and should only) use the semantics of real numbers as exact entities; (iii) on the bottom resides some reliable variable-precision arithmetic; and mediating between both, layer (ii) controls the execution of the user program (i) and its operations in carefully chosen finite precision(s) using (iii).

2.1 Modular Lower Layer

For efficiency purposes, `iRRAM` currently integrates (ii) and (iii), using hardware `double` for the initial precision $p \approx -50$ and `MPFR` hardcoded for higher precisions. The transition from hardware to software incurs a significant penalty in performance. This could be improved by introducing one or several additional stages for intermediate precisions, see the appendix. Further acceleration might be gained from support by suitable FPGAs/ASICs as 'hybrid' between off-the-shelf CPU and software [11].

More generally the lower layer (iii) could be implemented in various ways. A former version of `iRRAM` already did let the user choose between `MPFR` and `GMP` at compile time. Beyond arithmetic on single numbers, some libraries offer highly optimized `double` precision matrix operations [15]; others are tailored towards dynamical systems [1].

For `iRRAM` 3 we consider layer (iii) to be modularized and exchangeable: with interface specifications of both mandatory and optional, fixed and/or variable precision operations from lower to higher types.

2.2 Continuous Abstract Data Types

`iRRAM` 2 supports vectors/matrices and polynomials: continuous abstract data types built on finitely many real numbers. This has been extended to tensors [4], that is, higher-order matrices subject to basis in/co/variance. Analytic functions [6] are a continuous abstract data type over countably infinitely many reals.

As opposed to the above examples of implementations in user space, (Lipschitz-) continuous functions are supported intrinsically by means of `C++` templates. Currying currently seems delicate due to `iRRAM`'s internal precision control?

`iRRAM` 3 should add support for compact Euclidean subsets: using 'soft' membership predicates rather than distance functions [3]. This would allow in the next step to approach (solution operators for) partial differential equations with boundary conditions [8,16]. Random sampling of *continuous* objects, such as Brownian Motion, is another goal [10].

References

1. A. Balluchi, A. Casagrande, P. Collins, A. Ferrari, T. Villa, and A.L. Sangiovanni-Vincentel. Ariadne: a framework for reachability analysis of hybrid automata. In *Proc. 17th Int. Symp. on Mathematical Theory of Networks and Systems*, 2006.
2. Vasco Brattka and Peter Hertling. Feasible real random access machines. *Journal of Complexity*, 14(4):490–526, 1998.
3. Mark Braverman. *Computational Complexity of Euclidean Sets*. PhD thesis, University of Toronto, 2004.
4. Martin Ziegler Chansu Park, Sewon Park. Tensor calculus in exact real computation. KAIST School of Computing, Feb. 2018. CS-TR-2018-416.
5. Martín Hötzel Escardó, Martin Hofmann, and Thomas Streicher. On the non-sequential nature of the interval-domain model of real-number computation. *Mathem. Structures in Computer Science*, 14(6):803–814, 2004.
6. Akitoshi Kawamura, Florian Steinberg, and Holger Thies. Parameterized complexity for uniform operators on multidimensional analytic functions and ODE solving. In Lawrence S. Moss, Ruy J. G. B. de Queiroz, and Maricarmen Martínez, editors, *Logic, Language, Information, and Computation - 25th International Workshop, WoLLIC 2018, Bogota, Colombia, July 24-27, 2018, Proceedings*, volume 10944 of *Lecture Notes in Computer Science*, pages 223–236. Springer, 2018.
7. M. Konecny, W. Taha, J. Duracz, A. Duracz, and A. Ames. Enclosing the behavior of a hybrid system up to and beyond a zeno point. In *Proc. 1st IEEE Int. Conf. on Cyber-Physical Systems, Networks, and Applications*, pages 120–125, 2013.
8. I. Koswara, S. Selivanova, and M. Ziegler. Computational complexity of real powering and improved solving linear differential equations. In *Proc. 14th International Computer Science Symposium in Russia*, volume 11532 of *LNCS*, 2019.
9. Branimir Lambov. RealLib: an efficient implementation of exact real arithmetic. In Tanja Grubba, Peter Hertling, Hideki Tsuiki, and Klaus Weihrauch, editors, *Computability and Complexity in Analysis*, volume 326 of *Informatik Berichte*, pages 169–175. FernUniversität in Hagen, July 2005. Proccedings, Second International Conference, CCA 2005, Kyoto, Japan, August 25–29, 2005.
10. Hyunwoo Lee. Random sampling of continuous objects. Master's thesis, KAIST School of Computing, 2020.
11. Fritz Mayer-Lindenberg. Mathematical abstraction in a simple programming tool for parallel embedded systems. In Victor Malyshkin, editor, *Parallel Computing Technologies - 15th International Conference, PaCT 2019, Almaty, Kazakhstan, August 19-23, 2019, Proceedings*, volume 11657 of *Lecture Notes in Computer Science*, pages 32–50. Springer, 2019.
12. Norbert Müller and Christian Uhrhan. Some steps into verification of exact real arithmetic. *NASA Formal Methods*, pages 168–173, 2012.
13. Norbert Th. Müller. The iRRAM: Exact arithmetic in C++. In Jens Blanck, Vasco Brattka, and Peter Hertling, editors, *Computability and Complexity in Analysis*, volume 2064 of *Lecture Notes in Computer Science*, pages 222–252, Berlin, 2001. Springer. 4th International Workshop, CCA 2000, Swansea, UK, September 2000.
14. Norbert Th. Müller and Martin Ziegler. From calculus to algorithms without errors. In *Proc. 4th International Congress on Mathematical Software (ICMS)*, volume 8592 of *LNCS*, pages 718–724. Springer, 2014.
15. Siegfried M. Rump. INTLAB - interval laboratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, 1999.
16. S. Selivanova and V. Selivanov. Computing solution operators of boundary-value problems for some linear hyperbolic systems of PDEs. Logical Methods in Computer Science, 13 (4), 2017.
17. David H Bailey Yozo Hida, Xiaoye S Li. Library for double-double and quad-double arithmetic, 2007.
18. Jihun Yu, Chee Yap, Zilin Du, Sylvain Pion, and Hervé Brönnimann. The design of core 2: A library for exact numeric computation in geometry and algebra. In Komei Fukuda, Joris van der Hoeven, Michael Joswig, and Nobuki Takayama, editors, *Mathematical Software – ICMS 2010*, pages 121–141. Springer, 2010.

A Accelerating Intermediate Precision(s)

iRRAM currently uses native hardware double for its initial iteration at precision $p \approx -50$, and then switches to the variable-precision software library MPFR. The latter, although asymptotically optimal and highly optimized for intermediate precisions, incurs a significant setback in performance: the reliability comes at a high price!

In order to smoothen the transition, we suggest to introduce an additional phase for intermediate precision(s). Indeed several contemporary hardware floating point units already support precisions beyond the double's 53-bit mantissae, such as *x87* extended (64 bit) or *Power ISA v3* (113 bit). And on systems without such native support, 113-bit mantissae can be simulated by combining regular 53-bit doubles [17].

A.1 Our Measurements

We have explored the performance of combining two and four hardware-supported doubles for precisions $p \approx -100$ and $p \approx -200$, respectively, in comparison with MPFR. Our experiments indicate that two doubles (ddouble or DD) outperform MPFR by a factor of about 6. Four doubles (qdouble or QD) on the other hand show no benefit compared to MPFR.

We have performed two tests: matrix multiplication and polynomial multiplication. In the first test, inputs are two matrices of a fixed size $n \times n$, $A = [a_{ij}]$ and $B = [b_{ij}]$. We measure the time for computing $AB = [\sum_{k=1}^{n} a_{ik}b_{kj}]_{ij}$, where each element is implemented by DD, QD and MPFR, respectively. For observing relative performance of DD(QD) to MPFR, the ratios of the time taken by MPFR over that by DD(QD), are plotted on Fig 1*a* (1*b*).

Similarly to the matrix test, the polynomial test takes the coefficients of two polynomials of order n, $(a_0, \cdots, a_n)$ and $(b_0, \cdots, b_n)$, and computes $(c_0, \cdots, c_{2n})$, where $c_k = \sum_{i+j=k} a_i b_j$ for all k. The times for acquiring $(c_0, \cdots, c_{2n})$ are measured for each implementation by DD, QD and MPFR, by which the corresponding ratios are calculated and plotted on Fig 1*c* and 1*d*.

In both tests, we consider possible optimization in MPFR depending on subtle different precisions and the size of the mantissa of DD and QD. To this end, MPFR with precisions 96, 100, 104, 106, 112, and 128 bits are compared with DD, and MPFR with precisions 192, 200, 208, 212, and 224 bits are compared with QD. See *benchmark* at https://github.com/realcomputation/DOUBLESTAR for the detail.

A.2 Conclusion and Perspective

Numerical applications tend to come with a trade-off between performance and reliability. The current version of iRRAM exhibits a performance penalty of (very roughly) about factor 20. Boiling this overhead down in iRRAM 3 to, say, factor 3 will significantly extend its 'marketability'. Our experiments indicate that this is indeed feasible: by introducing a new hybrid iteration between the initial plain hardware-based double precision $p \approx -50$ and software-based MPFR for precisions $p \to -\infty$.

When incorporating this new phase into iRRAM it seems important, though, to keep other possible sources of overhead in check: For example, checking the current precision p in order to branch and call an external library with parameter passing easily becomes as expensive as the actual calculation itself—when performed for each single number/variable separately, but amortizes out on compound array/vector/matrix data types.

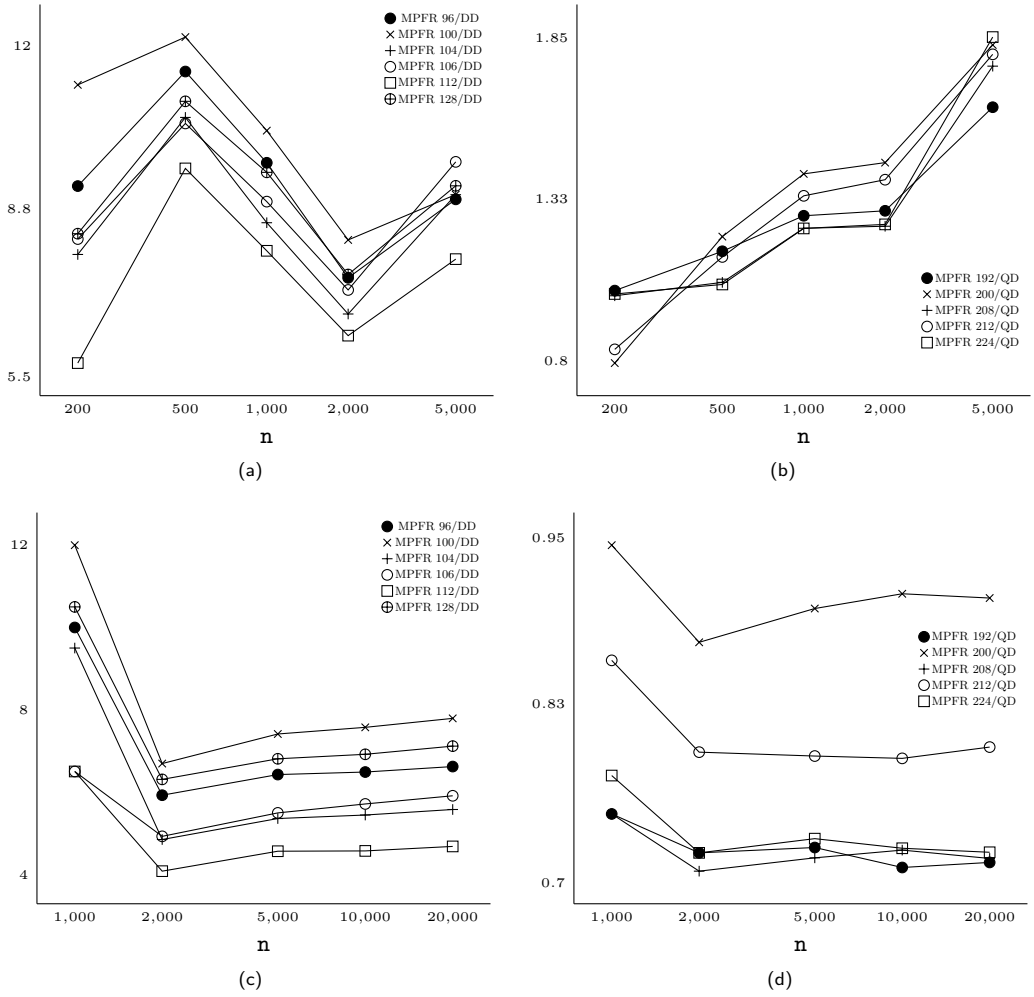

Fig. 1: Comparisons of the elapsed CPU time ratio for (a) Matrix mul.(MPFR/DD), (b) Matrix mul.(MPFR/QD), (c) Polynomial mul.(MPFR/DD) and (d) Polynomialmul.(MPFR/QD). Tested on Intel i7-7700, 16GB and Ubuntu 18 x64, and compiled with gcc7.5.

Future work will explore more closely the range of precisions between $p \approx -100$ and $p \approx -200$ where `MPFR` catches up on multiple `double`s. Specifically we intend to experiment with three `double`s (`tdouble/TD`) for $p \approx -150$; and with two `extended`s for $p \approx -128$.

Isometries and the equivalence of the effective separating sequences*

Zvonko Iljazović and Lucija Validžić

A *computable metric space* is a triple (X, d, α) such that (X, d) is a metric space and α a dense sequence in (X, d) such that the function $\mathbb{N}^2 \to \mathbb{R}$, $(i, j) \mapsto d(\alpha_i, \alpha_j)$ is computable. We say that α is an *effective separating sequence* in (X, d). Using the sequence α, we can define the notions of a computable point, a computable sequence and a computable compact set in (X, d, α). Now the question is, are those notions uniquely determined by the metric space itself i.e., if β is another effective separating sequence in (X, d), are computable points, computable sequences and computable compact sets in (X, d, β) the same as those in (X, d, α)?

In order to examine this problem closely, we define a relation on the set of the effective separating sequences in (X, d). We say that α and β are *equivalent* ([1]), $\alpha \sim \beta$, if α is a computable sequence in (X, d, β). If $\mathcal{S}_\alpha$ and $\mathcal{S}_\beta$ are the sets of computable sequences in (X, d, α) and (X, d, β), it can be shown (see [1]) that $\alpha \sim \beta$ if and only if $\mathcal{S}_\alpha = \mathcal{S}_\beta$, so $\sim$ is an equivalence relation. Moreover, since the computable points and the computable compact sets are determined by the computable sequences, our problem now comes down to finding the circumstances under which all effective separating sequences on a metric space are equivalent. The set of sequences which are computable with respect to a fixed effective separating sequence is a *computability structure* on a metric space. So our previous question can be rephrased as: Is a computability structure on a metric space unique?

The answer is in general negative. For example, if $x \in \mathbb{R}$ is a noncomputable number and α a computable sequence of real numbers which is dense in $(\mathbb{R}, d)$, where d is the Euclidean metric, then $(\alpha_i + x)$ is an effective separating sequence but x is computable in $(\mathbb{R}, d, (\alpha_i + x))$ and x is not computable in $(\mathbb{R}, d, \alpha)$. On the other hand, it is known that the computability structure on $[0, 1]$ (equipped with the Euclidean metric) is unique (Example 10 in [1]).

One obvious difference between $\mathbb{R}$ and $[0, 1]$ is that $[0, 1]$ is compact, but there are examples of segments in $\mathbb{R}$ which do not have a unique computability structure, so compactness is not a property that is strong enough to imply uniqueness of a computability structure.

We say that a computable metric space (X, d, α) is *effectively compact* (or *computably compact*) if (X, d) is complete and there exists a computable function $\varphi : \mathbb{N} \to \mathbb{N}$ such that $X = \bigcup_{i=0}^{\varphi(k)} B(\alpha_i, 2^{-k})$, for each $k \in \mathbb{N}$.

A simple example of an effectively compact computable metric space which does not have a unique computability structure is (S, d, α), where S is a unit circle in $\mathbb{R}^2$, d the Euclidean metric on S and α a computable dense sequence in S. If we take a noncomputable point $x \in S$, there is a rotation f with the center $(0, 0)$ such that $f(0, 1) = x$. Now the sequence $f \circ \alpha$ is an effective separating sequence and x is computable with respect to $f \circ \alpha$, so $\alpha \not\sim f \circ \alpha$. Notice that one obvious difference between S and $[0, 1]$ is that S has infinitely many isometries. In [1] it is shown that if there are only finitely many isometries of the underlying metric space, then the computability structure in an effectively compact computable metric space is unique. We improve this result by the following Theorem:

Theorem 1. *Let (X, d, α) be an effectively compact computable metric space and K a computable compact set in (X, d, α) such that there are only finitely many isometries $f : X \to X$ such that $f(K) \subseteq K$. If β is an effective separating sequence in (X, d) such that K is computable in (X, d, β), then $\alpha \sim \beta$.*

And as a consequence:

*This work has been fully supported by Croatian Science Foundation under the project 7459 CompStruct.

Proposition 1. *Assume that (X, d, α) is an effectively compact computable metric space and $x_0, \ldots, x_n$ are computable points in (X, d, α) such that there are only finitely many isometries $f : X \to X$ such that $f(x_i) = x_i$, $i = 0, \ldots, n$. If β is an effective separating sequence such that $x_0, \ldots, x_n$ are computable points in (X, d, β), then $\alpha \sim \beta$.*

It is easy to see that if α is an effective separating sequence in (X, d) and f is an isometry of (X, d), then $f \circ \alpha$ is an effective separating sequence. Moreover f maps the computability structure induced by α to the computability structure induced by $f \circ \alpha$. Now if a metric space does not have a unique computability structure, the question is, are all computability structures on that metric space actually an isometric image of a fixed computability structure?

We say that two effective separating sequences α and β are *equivalent up to an isometry* if there exists an isometry of the metric space such that $\alpha \sim f \circ \beta$. So the question now becomes, under which circumstances are all effective separating sequences equivalent up to an isometry?

In [1] it is shown that the Euclidean space $(\mathbb{R}^n, d)$ has that property, but also that there are metric spaces which do not (for example, a segment $[0, \gamma]$, for $\gamma > 0$ left computable, but not computable - Example 6 in [1]). Using Proposition 1, we can show that all effective separating sequences in the unit circle are equivalent up to an isometry. Let (S, d, α) be a previously mentioned example of the unit circle in $\mathbb{R}^2$, β an effective separating sequence in S and b a computable point in (S, d, β). Then there exists a rotation f such that $f(0, 1) = b$. Therefore, b is computable with respect to β as well as $f \circ \alpha$, so since there are only two isometries which fix b, by Proposition 1, we conclude that $\beta \sim f \circ \alpha$. We generalise this result to arbitrary sets in $\mathbb{R}^2$ with infinitely many isometries:

Theorem 2. *Assume that X is a subset of $\mathbb{R}^2$, d is the Euclidean metric on X and that there are infinitely many isometries of the metric space (X, d). If α is a sequence such that (X, d, α) is an effectively compact metric space and β is an effective separating sequence in (X, d), then α and β are equivalent up to an isometry.*

Together with the result from [1] it follows that if a set in $\mathbb{R}^2$ admits a structure of an effectively compact computable metric space, then it follows that all effective separating sequences in that set are equivalent up to an isometry.

References

[1] Zvonko Iljazović. Isometries and Computability Structures. *Journal of Universal Computer Science*, 16(18):2569–2596, 2010.

[2] Zvonko Iljazović and Lucija Validžić. Maximal computability structures. *Bulletin of Symbolic Logic*, 22(4):445–468, 2016.

[3] Alexander Melnikov. Computably isometric spaces. *Journal of Symbolic Logic*, 78:1055–1085, 2013.

[4] Marian Pour-El and Ian Richards. Computability in Analysis and Physics. *Springer-Verlag, Berlin-Heidelberg-New York,* 1989.

[5] Klaus Weihrauch. Computable Analysis. *Springer, Berlin*, 2000.

[6] M. Yasugi, T. Mori and Y. Tsujji. Effective properties of sets and functions in metric spaces with computability structure. *Theoretical Computer Science*, 219:467–486, 1999.

[7] M. Yasugi, T. Mori and Y. Tsujji. Computability structures on metric spaces. *Combinatorics, Complexity and Logic* Proc. DMTCS96 (D.S. Bridges et al), Springer, Berlin, 351–362, 1996.

NEARLY COMPUTABLE NUMBERS AND THE SOLOVAY REDUCTION

PHILIP JANICKI

Universität der Bundeswehr München

In this paper we are concerned with left-computable numbers that are nearly computable and how they behave under the Solovay reduction. We will only consider sequences of rational numbers. Let x be a real number. We say that a sequence $(x_n)_n$ *converges computably* to x if there is a computable function $u : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$ and for all $i \geq u(n)$ the inequality $|x - x_i| < 2^{-n}$ holds. A real number is called *computable* if there is a computable sequence that converges computably to it. On the other hand, a sequence $(x_n)_n$ is called *nearly computably convergent* if it is convergent and for every computable increasing function $s : \mathbb{N} \to \mathbb{N}$ the sequence $(x_{s(n+1)} - x_{s(n)})_n$ converges computably. A real number is called *nearly computable* if there is a computable sequence converging nearly computably to it. Furthermore, a real number is called *left-computable* if there exists a computable increasing sequence converging to it. It is not difficult to show that every computable number is both left-computable and nearly computable, but the converse is not true, which can be derived from [5] and [9]. The set of the left-computable numbers is a widely investigated real number class and of particular interest both to computable analysis and to algorithmic information theory. For left-computable numbers there is a natural reducibility relation due to Solovay [8], which leads to a classification of left-computable real numbers according to their Solovay degrees.

Definition 1 (Solovay, 1975, [8])**.** Let x and y be left-computable numbers. We say that x is *Solovay-reducible* to y (denoted by $x \leq_S y$) if there exist computable increasing sequences $(x_n)_n$ and $(y_n)_n$ of rational numbers converging to x and y, respectively, and a constant $c > 0$ such that for all $n \in \mathbb{N}$ we have:

$$x - x_n \leq c \cdot (y - y_n)$$

It is obvious that the Solovay reduction is reflexive, and it is easy to see that it is transitive. For two left-computable numbers x and y with $x \leq_S y$ and $y \not\leq_S x$ we also write $x <_S y$. As usual, one defines an equivalence relation $\equiv_S$ on the left-computable numbers by $x \equiv_S y$ if, and only if, $x \leq_S y$ and $y \leq_S x$. Its equivalence classes are called *Solovay degrees*. There is a smallest Solovay degree, the set of computable numbers [1]. And there is also a largest Solovay degree, namely the set of left-computable numbers that are random [2, 8, 1, 6] in the sense of Martin-Löf [7]. Furthermore $\leq_S$ is an upper semilattice: The $\leq_S$-supremum of the Solovay degrees of two left-computable numbers x and y is the Solovay degree of the left-computable number $x + y$ [1]. For further results on Solovay degrees see [4] and the monograph [3].

If one wishes to understand the computability-theoretic complexity of left-computable numbers that are nearly computable with regard to the Solovay reduction, little is known so far. One important new result is the fact that the set of nearly computable left-computable numbers is closed downwards under the Solovay reduction.

Theorem 2. *Let x and y be left-computable numbers with $x \leq_S y$. If y is nearly computable, then x is also nearly computable.*

E-mail address: philip.janicki@unibw.de.

Since there are left-computable numbers that are not nearly computable, a left-computable number that is nearly computable, cannot be random. This result has already been published [9] by Stephan and Wu. So all nearly computable Solovay degrees are strictly below $[\Omega]_S$. In this case one may ask if there also exists a non-random Solovay degree dominating all nearly computable degrees. Our next result, which is the main result of this paper, gives a negative answer to this question. It says that every left-computable number that is neither computable nor random has an incomparable nearly computable left-computable counterpart. The proof is by an infinite injury priority argument.

Theorem 3. *For every left-computable number x with $0 <_S x <_S \Omega$ there exists a nearly computable left-computable number y with $x \not\leq_S y$ and $y \not\leq_S x$.*

Corollary 4. *The Solovay degree $[\Omega]_S$ is the only degree dominating all nearly computable degrees.*

Since there are left-computable computable numbers that are neither computable nor random, we get a strengthening of Downey's and LaForte's theorem.

Corollary 5 (Downey, LaForte, 2002, [5])**.** *There exists a left-computable number that is nearly computable, but not computable.*

Finally, the sum of two nearly computable numbers is a nearly computable number as well, and the $\leq_S$-supremum of two incomparable nearly computable degrees is also a nearly computable degree that is strictly above them. So there is no largest Solovay degree that is nearly computable.

Corollary 6. *For every nearly computable left-computable number x there exists a nearly computable left-computable number y with $x <_S y$.*

References

[1] C. S. Calude, P. H. Hertling, B. Khoussainov, and Y. Wang. Recursively enumerable reals and Chaitin Ω numbers. *Theoretical Computer Science*, 255:125–149, 2001.

[2] G. J. Chaitin. A theory of program size formally identical to information theory. *J. ACM*, 22(3):329–340, July 1975.

[3] R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer New York, 2010.

[4] R. G. Downey, D. R. Hirschfeldt, and A. Nies. Randomness, computability, and density. In *SIAM Journal of Computation*, pages 1169–1183. Springer-Verlag, 2002.

[5] R. G. Downey and G. L. LaForte. Presentations of computably enumerable reals. *Theoretical Computer Science*, 284(2):539–555, 2002.

[6] A. Kucera and T. Slaman. Randomness and recursive enumerability. *SIAM J. Comput.*, 31(1):199–211, Jan. 2002.

[7] P. Martin-Löf. The definition of random sequences. *Information and control*, 9(6):602–619, 1966.

[8] R. Solovay. Draft of a paper (or series of papers) on Chaitin's work ... done for the most part during the period of Sept. - Dec. 1974. *Unpublished notes*, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 215pp., 1975.

[9] F. Stephan and G. Wu. Presentations of K-trivial reals and Kolmogorov complexity. In *CiE*, volume 3526 of *Lecture Notes in Computer Science*, pages 461–469. Springer, 2005.

Raising Small Polynomials with Real Coefficients to Exponential Power

Ivan Koswara[1], Gleb Pogudin[2], Svetlana Selivanova[1], Martin Ziegler[1]

Supported by the National Research Foundation of Korea (grant 2017R1E1A1A03071032) and by the International Research & Development Program of the Korean Ministry of Science and ICT (grant 2016K1A3A7A03950702) and by the NRF Brain Pool program (grant 2019H1D3A2A02102240).

{chaoticiak,sseliv,ziegler}@kaist.ac.kr , pogudin@cims.nyu.edu

[1] KAIST, School of Computing, 291 Daehak-ro, Daejeon, Republic of Korea

[2] Courant Institute, New York University, 251 Mercer st., NY, USA

Keywords: computational complexity, computational analysis, exponential dimension

Motivated by the application on solving partial differential equations using difference schemes, we look into the problem of raising a small (constant-size) polynomial into a huge (exponential) power, and ask the computational complexity of this problem. Specifically, consider the following problem:

Definition 1. *Fix a multi-variate polynomial $p \in \mathbb{C}^{d\times d}[z_1, \ldots, z_v]$ with complex matrix coefficients. Suppose that when $|z_i| \leq 1$ for all i, we have $\|p(\vec{z})\| \leq 1$. Let $E, I_1, \ldots, I_v, s, t, n$ be naturals. The goal is to determine the (s,t)-th entry of the coefficient of $\prod_j z_j^{I_j}$ in p^E up to error 2^{-n}.*

As an algorithmic problem, the input consists of the binary representations of $E, I_1, \ldots, I_v$, and the unary representations of s, t, n. The output is an integer multiple of 2^{-n}. It is emphasized that E is of order $O(2^n)$, hence the name "exponential power".

Our main results are as follows:

Theorem 2. *a) There exists some $k = poly(n)$ for which it's enough to have the elements of coefficients of p be given up to error 2^{-k}.*

b) When $d = 1$, the problem is poly-time computable with respect to n.

c) In general, the problem is in GapP*.*

Item a) guarantees that our problem is "well-posed"; if k had been super-polynomial in n, it wouldn't make sense to analyze the time complexity with respect to n, since it would take more than $poly(n)$ time to even read the input.

For b), arity 1, this uses two results from other fields. Chernoff bound is used to show that when the desired coefficient is not within a small interval, the result is close enough to 0 that it can be approximated by 0. Otherwise, Stirling approximation is used to approximate binomial coefficient to a respectable accuracy. Larger arities are simply product of terms in arity 1.

For c), the answer we seek is generally in the form of an exponential-size summation, but the term being added is a term in the form of b). Thus the counting ability of GapP is only used to add up the summation.

There are several corollaries of these results in the form of powering exponential-size matrices to exponential power. (1) Difference schemes often lead to such matrices, making these corollaries useful to determine the time complexity of them.

Corollary 3. *a) Let M be a $2^n \times 2^n$ circulant real matrix with only 2 nonzero bands. If $|M^k| \leq 1$ for all natural k, then computing any particular element of M^{2^n} up to error 2^{-n} can be done in poly-time with respect to n.*

b) The direct sum and the Kronecker product of matrices in the form of a) also has poly-time computable powers, as long as the powers are bounded.

[1] I. Koswara, S. Selivanova, and M. Ziegler, *Computational complexity of real powering and improved solving linear differential equations*, in Proc. 14th International Computer Science Symposium in Russia, vol. 11532 of LNCS, 2019.

Complexity and Coding Theory of Hilbert Spaces*

What is a *Polynomial-Time* Computable $\mathcal{L}^2$ Function?

Donghyun Lim, Svetlana Selivanova, and Martin Ziegler

KAIST, School of Computing

Abstract. Already for real numbers, different representations (binary, rational Cauchy, dyadic Cauchy, signed digit) induce different notions of computability, and complexity even more; similarly for continuous real functions: Weierstrass/Chebyshev polynomials, piecewise linear/rational approximations etc. We pick up on [5] and compare three natural representations of (a natural compact subset of) $\mathcal{L}^2[0;1]$: approximations by step/piecewise-linear functions, Fourier series, and the coefficient sequence w.r.t. any fixed orthonormal basis (such as the Haar Wavelets).

The Hilbert space $\mathcal{L}^2[0;1]$ of square-integrable complex-valued functions $f : [0;1] \to \mathbb{C}$ is of importance to partial differential equations. Adequate representations and notions of computation have well been identified; but suitable notions of computational complexity are still under exploration [5].

Complexity considerations in one integer parameter n apply to total computations on compact spaces [6,3,2]; hence for simplicity we here focus on the subset $\mathcal{L}^2_1[0;1]$ of $\mathcal{L}^2[0;1]$, compact according to the Fréchet-Kolmogorov Theorem, of all $f \in \mathcal{L}^2[0;1]$ with $\|f\|_2 \leq 1$ satisfying

$$\forall \varepsilon > 0 : \quad \|\tau_\varepsilon f - f\|_2 \ \leq \ \varepsilon \tag{1}$$

with the cyclic shift operator $\tau_r : \mathcal{L}^2[0;1] \to \mathcal{L}^2[0;1]$, $\big(\tau_r f\big)(t) = f(t+r \bmod 1)$.

Mathematics suggests many natural ways of encoding $f \in \mathcal{L}^2_1[0;1]$ via (indices to) some fixed dense sequence: polynomials with dyadic coefficients, piecewise constant functions with dyadic step points/values, piecewise linear functions, Fourier sequences, or coefficient sequences with respect to some other fixed orthonormal basis.

This choice raises the question which encoding (formalized as *representation*, over Cantor or over Baire space) to base complexity investigations of partial differential equations on—and whether it makes a difference. A simple but important constraint has been identified in [4, Lemma 3.1.13]:

Fact 1 *If a surjection $\delta : X \rightrightarrows Y$ has modulus of continuity μ, then the Kolmogorov entropies of X and Y satisfy: $\eta_Y \leq \eta_X \circ \mu$.*

* This work was supported by the National Research Foundation of Korea (grant 2017R1E1A1A03071032) and by the International Research & Development Program of the Korean Ministry of Science and ICT (grant 2016K1A3A7A03950702) and by the NRF Brain Pool program (grant 2019H1D3A2A02102240). We particularly thank Akitoshi Kawamura and Florian Steinberg for patient discussions.

Since quantitative continuity is well-known related to computational cost [6,2], this suggests discarding representations with 'unnecessarily' large modulus; but still leaves too many contingencies.

For continuous functions, polynomial-time computability of the evaluation functionals $(t, f) \mapsto f(t)$ is a common criterion for complexity-theoretically suitable representations [1, Lemma 4.7]; but for $f \in \mathcal{L}^2_1[0;1]$, evaluation $f(t)$ is in general not even defined. Instead we compare the computational cost of other natural functionals $\mathcal{L}^2_1[0;1] \to [0;1]$, and in particular the cost of converting between the different representation.

Encoding via Step/Piecewise-Linear Functions

Step functions are well-known to be dense in the integrable ones, and commonly employed in computability considerations, for instance of partial differential equations. However a step function $\bar{f}_n$ approximating a given $f \in \mathcal{L}^2_1[0;1]$ up to error $1/2^n$ may well violate Equation (1).

We therefore consider piecewise linear functions instead.

Encoding via Fourier Sequences

Recall the Hilbert space ℓ^2 of square-summable complex-valued sequences $\bar{z} = (z_k)$, equipped with the 'energy' norm $\|\bar{z}\|_2 = \sqrt{\sum_k |z_k|^2}$. According to Parseval's Identity, $\mathcal{L}^2[0;1]$ is isometric with ℓ^2 via the Fourier transform. Consider the subset ℓ^2_1 of all $\bar{z} \in \ell^2$ with $\|\bar{z}\|_2 \leq 1$ such that $\bar{z}' := (2\pi i k z_k)_k$ also satisfies $\|\bar{z}'\|_2 \leq 1$. This is isometric to $\mathcal{L}^2_1[0;1]$ via the Fourier transform.

Encoding via Orthonormal Coordinates

Generalizing the Fourier transform, fix an arbitrary orthonormal basis $\varphi_{\vec{k}}$ of $\mathcal{L}^2$ (not of $\mathcal{L}^2_1$: recall that no orthonormal sequence can contain a converging subsequence.) and let $\hat{f}_{\vec{k}} := \langle f, \varphi_{\vec{k}} \rangle \in \mathbb{C}$ denote the induced 'coordinate' of f, with the abbreviation $\langle f, g \rangle = \int_0^1 f(t) \cdot \bar{g}(t)\, dx$ for the inner product.

References

1. Akitoshi Kawamura and Stephen A. Cook. Complexity theory for operators in analysis. *TOCT*, 4(2):5:1–5:24, 2012.
2. Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. Complexity theory of (functions on) compact metric spaces. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 837–846. ACM, 2016.
3. Matthias Schröder. Spaces allowing type-2 complexity theory revisited. *Mathematical Logic Quarterly*, 50(4,5):443–459, 2004.
4. Florian Steinberg. *Computational Complexity Theory for Advanced Function Spaces in Analysis*. PhD thesis, TU Darmstadt, 2016.
5. Florian Steinberg. Complexity theory for spaces of integrable functions. *Logical Methods in Computer Science*, 13(3), 2017.
6. Klaus Weihrauch. Computational complexity on computable metric spaces. *Mathematical Logic Quarterly*, 49(1):3–21, 2003.

Algorithmically Optimal Outer Measures

Jack H. Lutz
Iowa State University

Neil Lutz
Iowa State University

Algorithmic fractal dimensions, which quantify the density of algorithmic information in individual points [10, 1, 12], have recently been used to prove new theorems [16, 14, 17, 15, 11] about their classical forerunners, the Hausdorff and packing dimensions of sets. Since algorithmic fractal dimensions are products of the theory of computing, and since the aforementioned new theorems are entirely classical (not involving logic or the theory of computing), these developments call for a more thorough investigation of the relationships between algorithmic and classical fractal dimensions.

In this paper, we establish direct connections between algorithmic and classical fractal dimensions. Aside from the presence versus absence of algorithms, the most striking difference between algorithmic fractal dimensions and classical fractal dimensions is that the algorithmic dimensions are usefully defined for individual points in Euclidean space, while the classical Hausdorff and packing dimensions vanish on individual points. To bridge this gap, we examine the classical *local dimensions* (also called *pointwise dimensions*) of outer measures at individual points in Euclidean spaces [5]. These local fractal dimensions have been studied at least since the 1930s and are essential tools in multifractal analysis [6, 4].

Outer measures, introduced by Carathéodory [2] in the "prehistory" of Hausdorff dimension [7] (defining what later became known as the 1-dimensional Hausdorff measure), are now best known for their role in Carathéodory's program [3] to generalize Lebesgue measure to a wide variety of settings [19]. However, it is the role of outer measures in local fractal dimensions that are of interest here.

The second author observed [13] that a particular, very nonclassical outer measure $\boldsymbol{\kappa}$, defined in terms of Kolmogorov complexity, has the property that the classical local fractal dimensions of $\boldsymbol{\kappa}$ coincide exactly with the algorithmic fractal dimensions at every point in $\mathbb{R}^n$. This property of $\boldsymbol{\kappa}$ is analogous to Levin's coding theorem [8, 9], which pertains to a particular, very nonclassical subprobability measure $\mathbf{m}$ on strings. Levin's theorem says that if we substitute $\mathbf{m}$ for the probability measure p in the classical Shannon *self-information* [18] $\log 1/p(x)$, then the resulting quantity $\log 1/\mathbf{m}(x)$ is essentially the prefix Kolmogorov complexity (i.e., the algorithmic information content) of the string x.

Levin defined $\mathbf{m}$ as an optimal lower semicomputable subprobability measure, so the above analogy leads us to investigate here the algorithmic optimality properties of $\boldsymbol{\kappa}$ and other outer measures on Euclidean spaces. We first investigate outer measures that are *globally optimal*, a property that is closely analogous to the optimality property of Levin's $\mathbf{m}$.

Definition. An outer measure μ on $\mathbb{R}^n$ is *globally optimal* if μ is strongly finite and lower semicomputable and if, for every strongly finite, lower semicomputable outer measure ν on $\mathbb{R}^n$, there is a constant $\beta \in (0, \infty)$ such that, for all $E \subseteq \mathbb{R}^n$, $\mu(E) \geq \beta \cdot \nu(E)$.

We prove that globally optimal outer measures on $\mathbb{R}^n$ exist. As it turns out, the outer measure $\boldsymbol{\kappa}$ is not globally optimal. We prove this fact, and we introduce and investigate the more general and more subtly defined class of *locally optimal* outer measures on $\mathbb{R}^n$.

Definition. Let μ and ν be outer measures on $\mathbb{R}^n$. We say that μ *dominates* ν *on dyadic cubes* if there is a function $\beta : \mathbb{N} \to (0, \infty)$ such that $\beta(r) = 2^{-o(r)}$ as $r \to \infty$ and, for every $r \in \mathbb{N}$ and every r-dyadic cube Q, $\mu(Q) \geq \beta(r) \cdot \nu(Q)$.

Definition. An outer measure μ on $\mathbb{R}^n$ is *locally optimal* if μ is strongly finite and lower semicomputable and if, for every strongly finite, lower semicomputable outer measure ν on $\mathbb{R}^n$, μ dominates ν on dyadic cubes.

We prove that the outer measure $\boldsymbol{\kappa}$ is locally optimal. Our main theorem establishes that *every* locally optimal outer measure μ on a Euclidean space $\mathbb{R}^n$ has the property that the classical local fractal dimensions of μ coincide exactly with the algorithmic dimensions at every point in $\mathbb{R}^n$. We discuss implications of our results, especially for the point-to-set principles that have enabled the new classical results mentioned in the first paragraph of this document.

References

[1] Krishna B. Athreya, John M. Hitchcock, Jack H. Lutz, and Elvira Mayordomo. Effective strong dimension in algorithmic information and computational complexity. *SIAM Journal on Computing*, 37(3):671–705, 2007.

[2] Constantin Carathéodory. Über das lineare maß von punktmengen—eine verallgemeinerung des längenbegriffs. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, pages 404–426, 1914. Translated as On the Linear Measure of Point Sets—a Generalization of the Concept of Length. In *Classics on Fractals*, Gerald A. Edgar (Ed.). Addison Wesley, 1993.

[3] Constantin Carathéodory. *Vorlesungen über reele Funktionen.* Teubner, Leipzig, 1918.

[4] Kenneth J. Falconer. *Techniques in Fractal Geometry.* Wiley, 1997.

[5] Kenneth J. Falconer. *Fractal Geometry: Mathematical Foundations and Applications.* Wiley, third edition, 2014.

[6] Otto Frostman. Potential d'equilibre et capacité des ensembles avec quelques applications a la theórie des fonctions. *Meddelanden från Lunds Universitets Matematiska Seminarium*, 3:1–118, 1935.

[7] Felix Hausdorff. Dimension und äusseres Mass. *Mathematische Annalen*, 79:157–179, 1919.

[8] Leonid A. Levin. On the notion of a random sequence. *Soviet Mathematics Doklady*, 14(5):1413–1416, 1973.

[9] Leonid A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.

[10] Jack H. Lutz. The dimensions of individual strings and sequences. *Information and Computation*, 187(1):49–79, 2003.

[11] Jack H. Lutz, Neil Lutz, and Elvira Mayordomo. The dimensions of hyperspaces. arXiv:2004.07798, 2020.

[12] Jack H. Lutz and Elvira Mayordomo. Dimensions of points in self-similar fractals. *SIAM Journal on Computing*, 38(3):1080–1112, 2008.

[13] Neil Lutz. *Algorithmic Information, Fractal Geometry, and Distributed Dynamics.* PhD thesis, Rutgers University, 2017.

[14] Neil Lutz. Fractal intersections and products via algorithmic dimension. In *42nd International Symposium on Mathematical Foundations of Computer Science August 21–25, 2017, Aalborg, Denmark, Proceedings*, 2017.

[15] Neil Lutz. Fractal intersections and products via algorithmic dimension. arXiv:1612.01659v4 [cs.CC], 2019.

[16] Neil Lutz and D. M. Stull. Bounding the dimension of points on a line. *Proceedings of the 14th Annual Conference on Theory and Applications of Models of Computation, TAMC 2017, Bern, Switzerland*, 2017.

[17] Neil Lutz and D. M. Stull. Projection theorems using effective dimension. In *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, pages 71:1–71:15, 2018.

[18] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3–4):379–423, 623–656, 1948.

[19] Terence Tao. *An Introduction to Measure Theory.* Graduate Studies in Mathematics. American Mathematical Society, 2011.

Characterising Continuously Realizable Multifunctions

Matthias Schröder

July 20, 2020

Multifunctions play an important rôle in Computable Analysis [2, 8]. They form an appropriate substitute for ordinary functions that fail to be computable just for continuity reasons. For example, the precise test "$x < y$" on the real numbers is discontinuous and thus incomputable. However, it can be often replaced by the computable *finite precision test* $(x <_\varepsilon y)$ defined by

$$(x <_\varepsilon y) := \begin{cases} \{\mathtt{tt}\} & \text{if } x \le y - \epsilon \\ \{\mathtt{tt}, \mathtt{ff}\} & \text{if } y - \varepsilon < x < y \\ \{\mathtt{ff}\} & \text{if } x \ge y \end{cases} \qquad \text{for all } x, y \in \mathbb{R}, \varepsilon > 0$$

(cf. [2]). Another prominent example is zero-finding of polynomials. There is no computable function which maps $(a, b, c) \in \mathbb{R}^3$ to a zero of the polynomial $x \mapsto x^3 + ax^2 + bx + c$, because any such function is discontinuous. Nevertheless, the multifunction mapping (a, b, c) to all zeros of the corresponding polynomial is computable [8]; the trick is to allow the computed zero to depend on the representatives of the coefficients.

Ordinary (single-valued) functions are known to be continuously realizable w.r.t. admissible representations if, and only if, they are sequentially continuous [7]. This fact is referred to as the Main Theorem of Computable Analysis in [8]. Our aim is to find a similar characterisation of continuously realizable multifunctions in terms of some form of ordinary mathematical continuity.

Computability and continuously realizability of multifunctions

A multifunction F between admissibly represented spaces X and Y is a binary relation between the underlying sets of X and Y, written as $F\colon \mathsf{X} \rightrightarrows \mathsf{Y}$. The intuition behind this is to view the elements of $F[x] := \{y \in \mathsf{Y} \mid (x, y) \in F\}$ as the legitimate results for an input x. In the following we assume any multifunction to be total, meaning $F[x] \neq \emptyset$ for all $x \in \mathsf{X}$. A *realizer* of F is a function $g\colon \mathbb{N}^\mathbb{N} \rightharpoonup \mathbb{N}^\mathbb{N}$ satisfying

$$\delta_\mathsf{Y}(g(p)) \in F[\delta_\mathsf{X}(p)] \quad \text{for every } p \in \mathrm{dom}(\delta_\mathsf{X}).$$

Here δ_X and δ_Y denote the respective representations of X and Y. If F has a computable realizer, then F is called *computable*; if F has a continuous realizer, then F is called *continuously realizable* [1, 8]. A multifunction $H\colon \mathsf{X} \rightrightarrows \mathsf{Y}$ *tightens* F, if $\emptyset \neq H[x] \subseteq F[x]$ for all $x \in \mathsf{X}$. If a tightening of F is continuously realizable, then so is F.

Multifunctions on computable metric spaces

Any computable metric space X has an open fiber-compact representation ϕ_X which is computably equivalent to the Cauchy representation (cf. [5]). Fiber-compactness means that the "fiber" $\phi^{-1}\{x\}$ is compact for every $x \in \mathsf{X}$. Any computable multifunction $F\colon \mathsf{X} \rightrightarrows \mathsf{Y}$ has a computable tightening with compact images, namely $\hat{g}[x] := \delta_\mathsf{Y} g[\phi^{-1}\{x\}]$ for any computable $(\phi_\mathsf{X}, \delta_\mathsf{Y})$–realizer g of F.

One can view $\hat{g}$ as a set-valued function $\bar{\hat{g}}$ from X to the space $\mathrm{K}_+(\mathsf{Y})$ of non-empty compact subsets of Y. As ϕ_X is an open function, the tightening $\bar{\hat{g}}$ is lower semi-continuous. This means

that $\bar{\hat{g}}$ is continuous with respect to the lower Vietoris topology on $\mathrm{K}_+(\mathsf{Y})$, which is induced by the subbasic sets $\Diamond V := \{K \in \mathrm{K}_+(\mathsf{Y}) \,|\, K \cap V \neq \emptyset\}$, where V varies over the open subsets of Y. These considerations yield the forward implication of V. Brattka's and P. Hertling's characterisation of continuously realizable multifunctions between separable metric spaces.

Theorem 1 (Brattka & Hertling [1]) *Let X, Y be separable metric spaces. Any multifunction $F\colon \mathsf{X} \rightrightarrows \mathsf{Y}$ is continuously realizable if, and only if, F has a lower semi-continuous tightening $h\colon \mathsf{X} \to \mathrm{K}_+(\mathsf{Y})$.*

The backward implication of Theorem 1 is based on the fact that compact overt choice $\mathrm{KVC}_{\mathsf{Y}}\colon \mathrm{K}_+(\mathsf{Y}) \rightrightarrows \mathsf{Y}$ selecting an element of the given input set is continuous realizable (even computable) for computable metric spaces (cf. [1]). The next lemmas show that the requirements that X and Y be separable metrisable cannot be omitted in Theorem 1.

Lemma 2 *Let Y be a non-Fréchet-Urysohn Hausdorff QCB-space. Then compact overt choice $\mathrm{KVC}_{\mathsf{Y}}\colon \mathrm{K}_+(\mathsf{Y}) \rightrightarrows \mathsf{Y}$ is not continuously realizable with respect to κ_+.*

Here κ_+ is a standard representation of $\mathrm{K}_+(\mathsf{Y})$ that is admissible w.r.t. the lower Vietoris topology (cf. [7]). The Co-Polish space [3] of polynomials is an example of a non-Fréchet-Urysohn space and therefore not metrisable.

Lemma 3 *Let X be a QCB_1-space without a countable base. Then there is a continuously realizable multifunction $F\colon \mathsf{X} \rightrightarrows \mathbb{N}$ that lacks a lower semi-continuous tightening $h\colon \mathsf{X} \to \mathrm{K}_+(\mathbb{N})$.*

Multifunctions with quasi-normal target spaces

We will now consider quasi-normal spaces Y as target spaces of multifunctions. Quasi-normal spaces [6] form a cartesian closed subcategory of Hausdorff QCB-spaces that contain all separable metric spaces, all Co-Polish spaces [3] and many spaces relevant in Functional Analysis.

In view of Lemma 2 we equip $\mathrm{K}_+(\mathsf{Y})$ with a representation κ_{+b} that is slightly more informative than the positive representation κ_+ by yielding additionally negative information of some compact superset. More precisely, we define κ_{+b} by

$$\kappa_{+b}\langle p, s\rangle = K \;:\Longleftrightarrow\; \kappa_+(p) = K \;\&\; K \subseteq \kappa_-(s)\,,$$

where κ_- is a standard negative representation of the compact subsets of Y which is admissible w.r.t. the upper Vietoris topology (cf. [7]). For quasi-normal spaces Y, κ_{+b} turns out to be informative enough to admit continuous realizability of compact overt choice $\mathrm{KVC}_{\mathsf{Y}}$ for Y.

The convergence relation induced by κ_{+b} is strictly sharper than the one induced by the lower Vietoris topology, in the sense that κ_{+b} admits strictly less convergent sequences (unless Y is compact). The space $\mathcal{K}_{+b}(\mathsf{Y})$ equipped with this convergence relation satisfies the axioms of a limit space, but it is in general not a topological space. We call sequentially continuous functions to $\mathcal{K}_{+b}(\mathsf{Y})$ *boundedly lower semi-continuous*. Importantly, κ_{+b} is an admissible representation for $\mathcal{K}_{+b}(\mathsf{Y})$, hence boundedly lower semi-continuous functions to $\mathrm{K}_+(\mathsf{Y})$ can be continuously realized with respect to κ_{+b} (cf. [7]).

Our main result is the following generalisation of Theorem 1 to quasi-normal target spaces.

Theorem 4 *Let X be a separable metric space and let Y be a quasi-normal space. Then a total multifunction $F\colon \mathsf{X} \rightrightarrows \mathsf{Y}$ is continuously realizable if, and only if, F has a boundedly lower semi-continuous tightening $h\colon \mathsf{X} \to \mathcal{K}_{+b}(\mathsf{Y})$.*

Lemma 3 shows that Theorem 4 cannot be generalised to non-metrisable quasi-normal source spaces X, because countably-based quasi-normal spaces are metrisable. D. Lim and S. Park observed in [4] that the powerset $\wp_+(\mathsf{Y})$ of Y cannot be equipped with a topology such that any multifunction $F\colon \mathsf{X} \rightrightarrows \mathsf{Y}$ is continuously realizable if, and only if, F is continuous viewed as a set-valued function $\overline{F}\colon \mathsf{X} \to \wp_+(\mathsf{Y})$, unless X is discrete. This justifies the attempt to characterise continuous realizable multifunctions via continuous set-valued tightening functions into a restricted family of subsets of target space.

References

[1] Vasco Brattka, Peter Hertling: *Continuity and Computability of Relations.* Informatik Berichte **164**, FernUniversität Hagen (1994)

[2] Vasco Brattka, Peter Hertling: *Feasible real random access machines.* Journal of Complexity **14**(4), 490–526 (1998)

[3] Matthew de Brecht, Arno Pauly, Matthias Schröder: *Overt choice.* To appear in Journal of Computability. doi: 10.3233/COM-190253. Available as arXiv: 1902.05926v1 (2019)

[4] Donghyun Lim, Sewon Park: *Shortcomings of the Topological and Realizability Approach on Multifunctions.* Draft (April 2020)

[5] Matthias Schröder: *Spaces Allowing Type-2 Complexity Theory revisited.* Mathematical Logic Quarterly 50(4/5), pp. 443–459 (2004)

[6] Matthias Schröder: *An Effective Tietze-Urysohn Theorem for QCB-Spaces.* Journal of Universal Computer Science **15**(6), pp. 1317–1336 (2009)

[7] Matthias Schröder: *Admissibly Represented Spaces and QCB-Spaces.* To appear as a chapter in the Handbook of Computability and Complexity in Analysis, Springer Verlag. Available as arXiv: 2004.09450v1 (2020)

[8] Klaus Weihrauch: *Computable Analysis.* Springer, Berlin (2000)

ON THE DESCRIPTIVE COMPLEXITY OF SALEM SETS

MANLIO VALENTI

The notion of Salem sets arises naturally in the context of geometric measure theory and the theory of fractal dimensions. It is based on the notions of Hausdorff dimension and Fourier dimension.

The first one is a fundamental notion in geometric measure theory and can be found in almost every textbook in the field. It describes the "size" of a set by means of the diameter of sets in an open cover.

When working with Borel subsets of $\mathbb{R}^n$, Frostman's lemma characterizes the Hausdorff dimension of a set by means of the existence of finite Radon measures supported on the set with certain regularity properties (see e.g. [1]). In particular, for a Borel subset $A \subset \mathbb{R}^n$, the Hausdorff dimension $\dim_{\mathcal{H}}(A)$ of A is the supremum of all $s \in [0, n]$ s.t.

$$(\exists c > 0)(\exists \mu \in \mathbb{P}(A))(\forall x \in \mathbb{R}^n)(\forall r > 0)(\mu(B(x, r)) \leq cr^s),$$

where $\mathbb{P}(A)$ denotes the set of probability measure with support in A and $B(x, r)$ denotes the ball of center x and radius r.

Interestingly, it turns out that the Fourier transform can be used to compute lower bounds for the Hausdorff dimension. Indeed, it can be shown that every probability measure whose Fourier transform decays sufficiently quickly satisfies the condition above (see [2]). In particular this leads to the notion of Fourier dimension, which can be defined as the supremum of all $s \in [0, n]$ s.t.

$$(\exists c > 0)(\exists \mu \in \mathbb{P}(A))(\forall x \in \mathbb{R}^n)(|\widehat{\mu}(x)| \leq c|x|^{-s/2}),$$

where $\widehat{\mu}$ denote the Fourier transform of μ.

For Borel subsets of $\mathbb{R}^n$ we always have $\dim_{\mathrm{F}}(A) \leq \dim_{\mathcal{H}}(A)$. A set for which the two notions coincide is called a Salem set.

In this work we investigate the descriptive complexity of the family of closed Salem sets.

We first work in the hyperspace $F([0, 1])$ of closed subsets of $[0, 1]$, endowed with the Vietoris topology. We show that the family of closed Salem subsets of $F([0, 1])$ is $\mathbf{\Pi}^0_3$-complete. We show also that, for each fixed $p \in (0, 1]$, the family of closed sets A with $\dim_{\mathcal{H}}(A) \geq p$ is $\mathbf{\Pi}^0_3$-complete. Using the same argument we can show that we obtain the same complexity if we replace the Hausdorff with the Fourier dimension. Moreover, for each fixed $p \in [0, 1)$, the family of closed sets A with $\dim_{\mathcal{H}}(A) > p$ is $\mathbf{\Sigma}^0_2$-complete (again we obtain the same complexity if we replace the Hausdorff with the Fourier dimension).

It is not trivial to generalize these results to higher dimensions: while the upper bounds rely on the compactness of the ambient space (and so the same proofs for the 1-dimensional case yield the upper bound when working in $[0, 1]^n$), the hardness results are more delicate. Indeed, the notion of Fourier dimension is critically dependent on the ambient space. It is well-known that, if $m < n$, then every subset

of a m-dimensional hyperplane has null Fourier dimension when seen as a subset of $[0,1]^n$.

Using some tools from harmonic analysis, we can prove that the complexities remain unchanged when replacing $[0,1]$ with $[0,1]^n$.

We then turn our attention to the family of closed Salem subsets of $\mathbb{R}^n$. Notice that, while there is a canonical choice for the topology on the hyperspace $F([0,1]^n)$, this is not the case for $\mathbb{R}^n$. We therefore consider both the Fell and the Vietoris topology for $F(\mathbb{R}^n)$ (the two topologies coincide when working on a compact space).

In this case, while the hardness' results follow from the results for $[0,1]^n$, the upper bounds require extra care, as we are relaxing the compactness of the ambient space. We prove that the complexity of the collection of closed Salem subsets is again $\mathbf{\Pi}^0_3$-complete, both if we endow $F(\mathbb{R}^n)$ with the Fell or the Vietoris topology.

All the reductions in this work are effective, hence the same proofs yield a Π^0_3-completeness result for the family of closed Salem subsets of $[0,1]^n$ or $\mathbb{R}^n$, both with the Vietoris and the Fell topology.

This is joint work with Alberto Marcone.

References

1. Kenneth Falconer, *Fractal geometry, mathematical foundations and applications*, 3 ed., Wiley, 2014.
2. Pertti Mattila, *Geometry of sets and measures in Euclidean spaces: Fractals and rectifiability*, Cambridge University Press, 1995.

Dipartimento di Scienze Matematiche, Informatiche e Fisiche, University of Udine, Udine, Italy

Email address: manlio.valenti@uniud.it