CCA 2012

Ninth International Conference on

Computability and Complexity in Analysis

(extended abstracts)

June 24-27, 2012

Cambridge, UK

Edited by

Arno Pauly
Robert Rettinger
Klaus Weihrauch

Preface

The Ninth International Conference on Computability and Complexity in Analysis, CCA 2012, takes place on June 24-27, 2012 in Cambridge, UK. It is the 18th event in a series of workshops, seminars and conferences. For more information about CCA see **http://cca-net.de**.

The conference is concerned with Computable Analysis, the theory of computability and complexity over real-valued data. Computability theory studies the limitations and abilities of computers in principle. Computational complexity theory provides a framework for understanding the cost of solving computational problems, as measured by the requirement for resources such as time and space. In particular, computable analysis supplies an algorithmic foundation of numerical computation.

Scientists working in the area of computability and complexity over real numbers and over more general continuous data structures come from different fields, such as theoretical computer science, domain theory, logic, constuctive mathematics, computer arithmetic, numerical mathematics and all branches of analysis.

The conference programme consists of 18 contributed lectures and 7 invited talks. We would like to thank all authors for their contributions and the programme committee members and the additional referees for their careful refereeing work.

June 2012

Arno Pauly
Robert Rettinger
Klaus Weihrauch

Computability and Complexity in Analysis – CCA 2012

Invited Speakers

- Vasco Brattka (Cape Town, South Africa)
- Akitoshi Kawamura (Tokyo, Japan)
- Robert Lubarsky (Boca Raton, FL, US)
- Elvira Mayordomo (Zaragoza, Spain)
- Andre Nies (Auckland, New Zealand)
- Lawrence C. Paulson (Cambridge, UK)
- Helmut Schwichtenberg (Munich, Germany)

Programm Committee

- Rod Downey (Wellington, New Zealand)
- Martín Escardó (Birmingham, UK)
- Guido Gherardi (Bologna, Italy)
- Jack H. Lutz (Ames, USA)
- Peter Schuster (Leeds, UK)
- Victor Selivanov (Novosibirsk, Russia)
- Klaus Weihrauch, chair (Hagen, Germany)
- Ning Zhong (Cincinnati, USA)

Organizing Committee

- Anuj Dawar (Cambridge, UK)
- Martín Escardó (Birmingham, UK)
- Bjarki Holm (Cambridge, UK)
- Arno Pauly, chair (Cambridge, UK)
- Robert Rettinger (Hagen, Germany)

Contents

A Hierarchy of the Forced Retracing Computable Curves

(Abstract)

David Abdul-Malak[1], Megan Gillespie[1], and Xizhong Zheng[1,2⋆]

[1] Department of Computer Science and Mathematics
Arcadia University, Glenside, PA 19038, USA
{MGillespie,DAbdulMalak, ZhengX}@arcadia.edu

[2] Departments of Mathematics and Computer Science
Jiangsu University, Zhenjiang 212013, China
xzhx@ujs.edu.cn

A curve can be used to record the path of a particle motion. Therefore, we can define a curve as the image of a continuous function defined on, say, the unit interval. In this case, the continuous function is called a parametrization of the curve. However, under this definition, a curve can fill even a square (e.g., the Peano curve) which contradicts our intuition about a curve. If we consider only the curve of a finite length (so-called rectifiable curve), this cannot happen. On the other hand, if a particle motion is algorithmically determined, like the movement of a robot on the plane, its path is naturally called computable. Thus, we will call a curve computable if it has a computable parametrization $f : [0, 1] \to \mathbb{R}^2$. Where $f : [0, 1] \to \mathbb{R}^2$ is computable means that $f = (f_x, f_y)$ and both f_x and f_y are computable real functions.

If a curve does not intersect itself, i.e., it has an injective parametrization, then the curve is called simple. Surprisingly, Gu, Lutz and Mayordomo prove in [1, 2] that there exists a computable simple curve which does not have injective computable parametrization. In fact, they construct a curve Γ which is rectifiable, simple and even computable in polynomial time. But, any computable parametrization of Γ must retrace some part of the curve unboundedly many times. This means that any computable parametrization of Γ is forced to retrace the curve although Γ is simple and hence has a (non-computable) injective parametrization. The cures of this property will be called *forced retracing* curves.

In this paper, we investigate further how the number of retracing allowed by a computable parametrization could be related to the complexity of a forced retracing computable curve. To this end we first introduce the notion of *n-retracing* curves.

Definition 1. Let $C \subseteq \mathbb{R}^2$ be a rectifiable simple computable curve and let n be a natural number.

1. A parametrization $f : [0, 1] \to \mathbb{R}^2$ of C has n-retracing if there are at most $2n + 1$ disjoint intervals $[a_i, b_i] \subseteq [0, 1]$ $(1 \le i \le 2n + 1)$ such that $f[a_i, b_i] = f[a_1, b_1]$ for all i.
2. The curve C is called *n-retracing* if C has a computable parametrization of n-retracing. C is called *$*$-retracing* if it is n-retracing for some n.

The classes of all n-retracing curves and $*$-retracing curves are denoted by RC_n and RC_*, respectively. Then we have $RC_* = \bigcup_{n\in\mathbb{N}} RC_n$. By definition, if C is a 0-retracing curve, then it has a computable parametrization that is injective or monotone. In this case we call C *monotonically computable*, or *M-computable*. Thus, M-computable curves can be parametrized without retracing. The class of all M-computable curves is denoted by MC. On the other hand, a computable curve, like the curve Γ constructed in [1, 2], can have up to unboundedly many retracing. Let EC (effectively computable) be the class of all computable rectifiable simple cure. Then the following relations between the curve classes introduced so far, for all n, follow from the definition immediately.

$$MC = RC_0 \subseteq RC_n \subseteq RC_{n+1} \subseteq RC_* \subseteq EC.$$

⋆ Corresponding author. email: ZhengX@Arcadia.edu. He is supported by NSFC 61070231

The result of [1,2] shows actually that $RC_* \neq EC$. In this paper we will show that all other inclusion relations are also proper. Therefore there is an infinite hierarchy of all retracing curves. As the first step, we prove that there is a computable curve which is forced to retrace once.

Theorem 1. *There is a rectifiable simple computable 1-retracing curve C_1 which is not monotonically computable.*

Proof. (Sketch) The curve C_1 is constructed as the limit of an effectively convergent computable sequence (P_s) of rational polygons. The main idea is to code the halting problem K into the curve C_1 as shown in Figure 1.

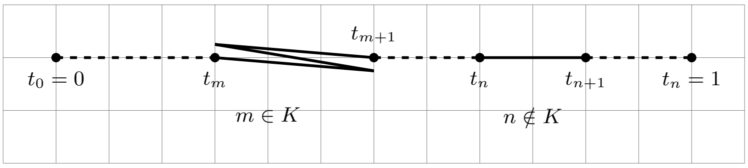

Fig. 1. Graph of the curve C

Fix a computable enumeration (K_s) of K, which is a computable sequence of finite sets such that $K = \lim K_s$ and $K_s \subseteq K_{s+1}$. The unit interval $[0,1]$ is divided into infinitely many subintervals $I_n := [t_n, t_{n+1}]$, where $t_n = 1 - 2^{-n}$. We define P_s as a rational polygon which encode the finite set K_s in the above way and let f_s be a computable 1-retracing parametrization of P_s such that, for any n, if $n \notin K_s$, then P_s is a line segment l_n connecting $(t_n, 0)$ and $(t_{n+1}, 0)$, and the function f_s retraces the segment l_n once evenly (passing through the line three times). Otherwise, if $n \in K_s$, then P_s has a zig-zag on the interval I_n and f_s passes through this part only once. The height of the zig-zag should be small enough go guarantee that the sequence (f_s) converges effectively. This can imply that C_1 is computable, simple and rectifiable as well.

Apparently C_1 is an 1-retracing computable curve. If, by contradiction, g is a computable injective parametrization of C_1, then we can show that $n \in K$ iff there are three disjoint intervals $[u_i, v_i]$ such that $g[u_i, v_i]$ pass the line $x = m_n$ for $i = 1, 2, 3$, where $m_n = (t_n + t_{n+1})/2$. Because the later can be determined effectively by the computability of g, this contracts to the non-computability of K. Therefore, g cannot be injective.

By similar constructions, we can further prove the following theorem.

Theorem 2. *1. For any n, there is an $(n+1)$-retracing computable curve which is not n-retracing.*
2. There is a computable curve which is not n-retracing for all n.

The second item of Theorem 2 is actually proved in [1,2] where the constructed curve Γ is even polynomial time computable and has continuous second order derivative. Actually, if we replace the polygons in our constructions by some smooth and polynomial time computable cures, we can also make the curves constructed above to be polynomial time computable and smooth.

References

1. X. Gu, J. H. Lutz, and E. Mayordomo. Points on computable curves. In *Proceedings of FOCS 2006*, pages 469–474. IEEE Computer Society Press, 2006.
2. X. Gu, J. H. Lutz, and E. Mayordomo. Curves that must be retraced. *Information and Computation* 209(2011), 992–1006
3. R. Rettinger and X. Zheng. Points on Computable Curves of Computable Length. *MFCS 2009* LNCS Vol. 5734, 736–743, Springer.
4. R. Rettinger and X. Zheng. On the Computability of Rectifiable Simple Curves. *CCA 2009* Informatik Berichte 353-7/2009, 243–253, Universität Hagen.
5. K. Weihrauch. *Computable Analysis, An Introduction.* Springer, Berlin Heidelberg, 2000.
6. G. T. Whyburn. What is a Curve?. *The American Mathematical Monthly*, 49(8):493-497, 1942.

Locating $\mathfrak{A}x$, where $\mathfrak{A}$ is a subspace of $\mathcal{B}(H)$

Douglas S. Bridges

Let H be a Hilbert space, $\mathcal{B}(H)$ the space of bounded operators on H, and $\mathfrak{A}$ a linear subspace of $\mathcal{B}(H)$. For each $x \in H$ write

$$\mathfrak{A}x \equiv \{Ax : A \in \mathfrak{A}\},$$

and, *if it exists*, denote the projection of H onto the closure $\overline{\mathfrak{A}x}$ of $\mathfrak{A}x$ by $[\mathfrak{A}x]$. Projections of this type play a very big part in the classical theory of operator algebras, in which context $\mathfrak{A}$ is normally a subalgebra of $\mathcal{B}(H)$. However, in the Bishop-style constructive setting of this paper, we cannot even guarantee that $[\mathfrak{A}x]$ exists. Our aim is to give sufficient conditions on $\mathfrak{A}$ and x under which $[\mathfrak{A}x]$ exists, or, equivalently, the set $\mathfrak{A}x$ is located, in the sense that

$$\rho(v, \mathfrak{A}x) \equiv \inf \{\|v - Ax\| : A \in \mathfrak{A}\}$$

exists.

We need some background on operator topologies. Specifically, in addition to the standard uniform topology on $\mathcal{B}(H)$, we need

- ▷ the **strong operator topology:** the weakest topology on $\mathcal{B}(H)$ with respect to which the mapping $T \rightsquigarrow Tx$ is continuous for all $x \in H$;
- ▷ the **weak operator topology:** the weakest topology on $\mathcal{B}(H)$ with respect to which the mapping $T \rightsquigarrow \langle Tx, y\rangle$ is continuous for all $x, y \in H$

These topologies are induced, respectively, by the seminorms of the form $T \rightsquigarrow \|Tx\|$ with $x \in H$, and $T \rightsquigarrow |\langle Tx, y\rangle|$ with $x, y \in H$. The unit ball

$$\mathcal{B}_1(H) \equiv \{T \in \mathcal{B}(H) : \|T\| \leqslant 1\}$$

of $\mathcal{B}(H)$ is classically weak-operator compact, but constructively the most we can prove is that it is weak-operator totally bounded. The evidence so far suggests that in order to make progress when dealing constructively with a subspace or subalgebra $\mathfrak{A}$ of $\mathcal{B}(H)$, it makes sense to add the weak-operator total boundedness of

$$\mathfrak{A}_1 \equiv \mathfrak{A} \cap \mathcal{B}_1(H)$$

to whatever other hypothesis we are making; in particular, it is known that $\mathfrak{A}_1$ is located in the strong operator topology—and hence $\mathfrak{A}_1 x$ is located for each $x \in H$—if and only if it is weak-operator totally bounded.

Recall that the **metric complement** of a subset S of a metric space X is the set $-S$ of those elements of X that are bounded away from X. When Y is a subspace of X, $y \in Y$, and $S \subset Y$, we define

$$\rho_Y (y, -S) \equiv \inf \{\rho (y, z) : z \in Y \cap -S\}$$

if that infimum exists.

Our main result is

Theorem 1 *Let $\mathfrak{A}$ be a uniformly closed subspace of $\mathcal{B}(H)$ such that $\mathfrak{A}_1$ is weak-operator totally bounded, and let x be a point of H such that $\mathfrak{A}x$ is closed and $\rho_{\mathfrak{A}x} (0, -\mathfrak{A}_1 x)$ exists. Then the projection $[\mathfrak{A}x]$ exists.*

Before proving this theorem, we discuss some general results about the locatedness of sets like $\mathfrak{A}x$, and we derive the following generalisation of the open mapping theorem, which leads to the proof of Theorem 1.

Theorem 2 *Let X be a Banach space,and C a located, bounded, balanced, and superconvex subset of X such that $\rho (0, -C)$ exists and $X = \bigcup_{n \geqslant 1} nC$. Then there exists $r > 0$ such that $B (0, r) \subset C$.*

Note that a bounded subset C of a Banach space X **superconvex** if for each sequence $(x_n)_{n \geqslant 1}$ in C and each sequence $(\lambda_n)_{n \geqslant 1}$ of nonnegative numbers such that $\sum_{n=1}^{\infty} \lambda_n$ converges to 1 and the series $\sum_{n=1}^{\infty} \lambda_n x_n$ converges, we have $\sum_{n=1}^{\infty} \lambda_n x_n \in C$. In that case, C is clearly convex.

Finally, we show, by a Brouwerian example, that the existence of $\rho_{\mathfrak{A}x} (0, -\mathfrak{A}_1 x)$ cannot be dropped from the hypotheses of Theorem 1.

Acknowledgement. This research was partially done when the author was a visiting fellow at the Isaac Newton Institute for the Mathematical Sciences, in the programme *Semantics & Syntax: A Legacy of Alan Turing*.

The strength of Ramsey's Theorem for coloring relatively large sets

Lorenzo Carlucci
Dipartimento di Informatica, University of Rome I "La Sapienza"
Rome, Italy
carlucci@di.uniroma1.it

Konrad Zdanowski
University of Cardinal Stefan Wyszyński
Warsaw, Poland
k.zdanowski@uksw.edu.pl

Abstract

We characterize the computational content and the proof-theoretic strength of an infinite Ramsey-type theorem due to Pudlàk and Rödl [6] and independently to Farmaki [2]. The theorem we analyze is as follows. For every infinite subset M of $\mathbf{N}$, for every coloring C of the exactly large subsets of M in two colors, there exists and infinite subset L of M such that C is constant on all exactly large subsets of L. An *exactly large* set is a set $X \subset \mathbf{N}$ such that $\mathrm{card}(X) = \min(X) + 1$. The notion of large set comes from the famous independence result for Peano Arithmetic due to Paris and Harrington [5]. The theorem can be equivalently formulated in terms of thin Schreier families [7] from Banach Space Theory (see, e.g., [1]). We prove that — over Computable Mathematics — this theorem is equivalent to closure under the ωth Turing jump. Natural combinatorial theorems at this level of complexity are rare. We give a complete characterization of the theorem from the point of view of Computability Theory and Reverse Mathematics. This nicely extends the current knowledge about the strength of Ramsey's Theorem [4, 8]. We conjecture that analogous results hold for generalizations of the theorem to larger ordinals.

References

[1] S. Argyros, S. Todorčević. Ramsey methods in Analysis. Birkhäuser, Berlin, 2005. 1

[2] V. Farmaki, S. Negrepontis. Schreier Sets in Ramsey Theory. *Transactions of the American Mathematical Society*, 360:2, 849–880, 2008. 1

[3] L. Harrington, J. Paris. A mathematical incompleteness in Peano Arithmetic, in *Handbook of Mathematical Logic*, J. Barwise (Ed.), 1133–1142, North-Holland, 1977. 1

[4] C. G. Jockusch. Ramsey's theorem and Recursion Theory. *The Journal of Symbolic Logic*, 37, 268–280, 1972. 1

[5] J. Paris, L. Harrington, A mathematical incompleteness in Peano Arithmetic. In J. Barwise (ed.), *Handbook of Mathematical Logic*, 1133–1142, North-Holland, 1977. 1

[6] P. Pudlàk, V. Rödl. Partition theorems for systems of finite subsets of integers. *Discrete Mathematics*, 39:1, 67–73, 1982. 1

[7] J. Schreier. Ein Gegenbeispiel zur Theorie der schwachen Konvergenz. *Studia Mathematica*, 2, 58–62, 1930. 1

[8] S. G. Simpson. Subsystems of Second Order Arithmetic. Springer-Verlag, 1999. 1

Conservatively Approximable Functions

Douglas Cenzer, University of Florida,
Gainesville, Florida 32611-8105, *cenzer@math.ufl.edu*

Keywords: symbolic dynamics, computable analysis, computability, effectively closed sets

An effectively closed subset (or Π^0_1 class) P of the Cantor space $2^{\mathbb{N}}$ is the set of infinite paths through some computable tree T. If the tree has no dead ends, then P is said to be decidable or computable. It is well known that the image of computable function on the Cantor space is a decidable Π^0_1 class. But most interesting Π^0_1 classes are *not* decidable. Thus we consider the problem of identifying a weaker notion of computable real functions which can have arbitrary Π^0_1 classes as the image.

A related problem arises in the study of effective dynamical systems. A Π^0_1 class P is said to be a *subshift* if it is closed under the shift operator σ, defined by $\sigma(X) = (X(1), X(2), \dots)$. The iteration of a computable function F on $2^{\mathbb{N}}$ create a dynamical system and to the Π^0_1 subshift of *itineraries* of F as follows. For each X, the sequence $(X, F(X), F(F(X)), \dots)$ is the *trajectory* of X. Given a fixed partition $U_0, \dots, U_{k-1}$ of $2^{\mathbb{N}}$ into clopen sets, the *itinerary* $It(X)$ of a point X is the sequence $(a_0, a_1, \dots) \in k^{\mathbb{N}}$ where $a_n = i$ iff $F^n(X) \in U_i$. Let $It[F] = \{It(X) : X \in 2^{\mathbb{N}}\}$. Note that $It[F]$ will be a closed set. We observe that, for each point X with itinerary $(a_0, a_1, \dots)$, the point $F(X)$ has itinerary $(a_1, a_2, \dots)$. It follows that $It[F]$ is a *subshift.*

Computable subshifts and the connection with effective symbolic dynamics were investigated by Cenzer, Dashti and King [1]. It was shown that for any computably continuous function $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$, $It[F]$ is a decidable Π^0_1 class and, conversely, any decidable Π^0_1 subshift P is $It[F]$ for some computable map F. In this paper, it is also shown that there exist Π^0_1 subshifts which have no computable elements and are therefore not decidable. Thus we consider again the problem of identifying a notion of weak computability which will produce arbitrary Π^0_1 subshifts.

A computable function F on 2^N may be represented by a function f on finite strings so that $F(X)$ is the limit of the sequence $f(x \restriction n)$. Here f has the property that whenever a string v extends a string u, then $f(v)$ also extends f(u). We introduce the notion of a conservatively approximable function in which the representation has the weaker property that $f(u*i)$ need only extend $f(w)$ for some string w with the same length as u. It is proved that the image of a conservatively approximable function is always a Π^0_1 class and that every Π^0_1 class P is the image of some conservatively approximable function.

We also consider the set of itineraries of conservatively approximable functions. We show that every Π^0_1 subshift is the set of itineraries of some conservatively approximable function and give conditions under which the set of itineraries of a conservatively approximable function will be a Π^0_1 class.

The notions of effectively closed sets and of computable type two functions for the real line and for other spaces have been studied intensively by Weihrach [2] and many others in the computable analylsis community. We also consider the notion of conservatively approximable functions on the real interval $[0, 1]$.

References

[1] D. Cenzer, A. Dashti and J.L.F. King, *Computable Symbolic Dynamics*, Math. Logic Quarterly 54 (2008), 524–533.

[2] K. Weihrauch, *Computable Analysis*, Springer (2000).

Computable Probability Theory and Stochastic Processes

Pieter Collins
Department of Knowledge Engineering, Maastricht University
Postbus 616, 6200 MD Maastricht, The Netherlands
pieter.collins@maastrichtuniversity.nl

Abstract

The aim of this paper is to present an elementary computable theory of probability, random variables and stochastic processes. The probability theory is baed on existing approaches using valuations and lower integrals. Various approaches to random variables are discussed, including the approach based on completions in a Polish space. We apply the theory to the study of stochastic dynamical systems in discrete-time, and give a brief exposition of the Wiener process as a foundation for stochastic differential equations. The theory is based within the framework of type-two effectivity, so has an explicit direct link with Turing computation, and is expressed in a system of computable types and operations, so has a clean mathematical description.

A Strong Turing Reduction for Additive BSS RAM's

Christine Gaßner
Ernst-Moritz-Arndt-Universität Greifswald, Germany
gassnerc@uni-greifswald.de

The Turing machine and several types of register machines are well-known models of computation that are used for describing the complexity of problems solved by computers. Whereas a Turing machine only processes a finite number of symbols, most register machines operate on real numbers, and in contrast to classical complexity theory based on the Turing model, the study of register machines leads in general to a non-uniform complexity theory. A combination of such models provides abstract machine models of computation over algebraic structures including the BSS model of computation over the reals. The resulting complexity theory is uniform, but the new model causes new questions and the reducibility relations need to be re-examined. Here we study some aspects concerning the Turing reduction related to additive and linear BSS RAM's over structures such as $\mathbb{R}^{\leq}_{\mathrm{add}} = (\mathbb{R}; \mathbb{R}; +, -; \leq)$, $\mathbb{R}^{=}_{\mathrm{add}} = (\mathbb{R}; \mathbb{R}; +, -; =)$, and $\mathbb{R}^{\leq}_{\mathrm{lin}} = (\mathbb{R}; 1; +, -, (\varphi_c)_{c \in \mathbb{R}}; \leq)$ with $\varphi_c(x) = cx$ for all $x \in \mathbb{R}$. The machines can be defined as in [Gaßner 2008] and can be assigned to classes as follows. The additive machines in $\mathsf{M}_{\mathrm{add}}$ and $\mathsf{M}^{=}_{\mathrm{add}}$ allow computations by means of the operations and relations in $\mathbb{R}^{\leq}_{\mathrm{add}}$ and $\mathbb{R}^{=}_{\mathrm{add}}$, respectively, and can execute the instruction $Z_j := c$ for any constant $c \in \mathbb{R}$. The machines in $\mathsf{M}^{1}_{\mathrm{add}}$, $\mathsf{M}^{1,=}_{\mathrm{add}}$, and $\mathsf{M}_{\mathrm{lin}}$ use only the constant 1. The classes $\mathrm{REC}_{\mathrm{add}}, \mathrm{REC}^{=}_{\mathrm{add}}, \ldots$ contain the problems recognizable (semi-decidable) by a machine in $\mathsf{M}_{\mathrm{add}}, \mathsf{M}^{=}_{\mathrm{add}}, \ldots$, respectively. $\mathrm{DEC}_{\mathrm{add}}, \ldots$ are the corresponding classes of decidable problems. In contrast to many models of computation over the integers the inclusions $\mathrm{REC}^{1,=}_{\mathrm{add}} \subset \mathrm{REC}^{1}_{\mathrm{add}} \subset \mathrm{REC}_{\mathrm{add}} \subset \mathrm{REC}_{\mathrm{lin}}$ and $\mathrm{DEC}^{1,=}_{\mathrm{add}} \subset \mathrm{DEC}^{1}_{\mathrm{add}} \subset \mathrm{DEC}_{\mathrm{add}} \subset \mathrm{DEC}_{\mathrm{lin}}$ are strict.

The Turing reduction over the reals is defined by machines using an oracle $\mathcal{O} \subseteq \mathbb{R}^{\infty}$. The *strong Turing reduction* with respect to the additive BSS model (here denoted by $\preceq$) can be performed by additive BSS oracle machines with the constant 1 and the order test, the *weak Turing reduction* (denoted by $\preceq^{r_1,\ldots,r_k}$) refers here to additive BSS oracle machines with real constants $r_1, \ldots, r_k$ and the order test. In order to point out common features and differences between the Turing machine, the additive BSS machines without irrational constants, and other BSS RAM's, we combine construction methods of the classical recursion theory with techniques for proving lower bounds of algebraic complexity, give some examples for different unsolvability degrees with respect to the strong Turing reduction, present hierarchies of decision problems below several halting problems $\mathbb{H}_{\rm add}$, $\mathbb{H}^1_{\rm add}$, and $\mathbb{H}^{1,=}_{\rm add}$ for the additive machines defined by

$$\mathbb{H}_{\rm add} =_{\rm df} \bigcup_{n\geq 1}\{(n\,.\,\mathbf{x}\,.\,{\rm code}(\mathcal{M})) \mid \mathbf{x}\in\mathbb{R}^n \;\&\; \mathcal{M}\in \mathsf{M}_{\rm add} \;\&\; \mathcal{M}(\mathbf{x})\downarrow\},$$

and so on, and investigate a weak Turing degree defined by the complete semi-decidable problem $\mathbb{H}_{\rm add}$. In particular, we present the following results.

1. For $\mathbb{L}_n = \{(x_1,\ldots,x_n)\in\mathbb{R}^n \mid (\exists(q_0,\ldots,q_{n-1})\in\mathbb{Q}^n)(q_0+\sum_{i=1}^{n-1} q_ix_i = x_n)\}$ there are problems $\mathbb{A},\mathbb{K}\subseteq\mathbb{N}$ with $\bigcup_{i\geq 1}\mathbb{L}_i \precneqq \bigcup_{i\in\mathbb{A}}\mathbb{L}_i \precneqq \bigcup_{i\in\mathbb{K}}\mathbb{L}_i \preceq \mathbb{H}^{1,=}_{\rm add}$.

2. The set $\mathbb{A}_{\rm alg}$ of algebraic numbers and $\mathbb{H}_{\rm add}$ define incomparable Turing degrees with respect to additive machines, and the same holds for $\mathbb{A}_{\rm alg}$ and $\mathbb{H}^{1,=}_{\rm add}$.

3. Let $p_1 = 2, p_2 = 3,\ldots$ be an enumeration of the prime numbers and let $\mathcal{K}$ be a machine recognizing $\{i\}\times(\mathbb{R}\setminus\{\sqrt{p_i}\})$ by checking, for any input (i,x), the condition $((x<\frac{r}{q}$ and $\frac{r^2}{q^2}<p_i)$ or $(x>\frac{r}{q}$ and $\frac{r^2}{q^2}>p_i))$ for all enumerated $(r,q)\in\mathbb{N}\times\mathbb{N}^+$ until the condition is satisfied. Then, for

$$\mathbb{H}_i =_{\rm df} \mathbb{H}^{1,=}_{\rm add}\cup\bigcup_{j\leq i}\{(2\,.\,(j,x)\,.\,{\rm code}(\mathcal{K})) \mid x\in\mathbb{R}\setminus\{\sqrt{p_j}\}\}\subseteq\mathbb{H}^1_{\rm add}$$

we get $\mathbb{H}^{1,=}_{\rm add} \precneqq \mathbb{H}_1 \precneqq \cdots \precneqq \mathbb{H}_k \precneqq \bigcup_{i\geq 1}\mathbb{H}_i \equiv \bigcup_{i\in\mathbb{A}}\mathbb{H}_i \equiv \bigcup_{i\in\mathbb{K}}\mathbb{H}_i \preceq \mathbb{H}^1_{\rm add}$ where $k\geq 2$.

4. For any oracle $\mathcal{O}\subseteq\mathbb{R}^\infty$, let $\mathsf{M}^1_{\rm add}(\mathcal{O})$ be the class of all additive oracle BSS RAM's using only the constant 1, the order test, and $\mathcal{O}$. For decomposing the weak Turing degree $\mathbb{H}_{\rm add}$ we use a sequence $(r_i)_{i\geq 1}$ of real numbers where $r_1 = 1$ and, for any $i\geq 2$, r_i is the representation of a special halting problem $\mathbb{H}_{\rm spec}(\mathsf{M}^1_{\rm add}(\mathbb{H}^{r_1,\ldots,r_{i-1}}_{\rm add}))\subseteq\mathbb{N}$ of the oracle machines in $\mathsf{M}^1_{\rm add}(\mathbb{H}^{r_1,\ldots,r_{i-1}}_{\rm add})$. ${\rm code}^{(i)}(\mathcal{M})$ is the sequence of the representations of the single symbols of the program of $\mathcal{M}$ where the real numbers $r_1,r_2,\ldots,r_i$ are encoded by the binary representations of their indices and any other real constant in $\mathbb{R}\setminus\{r_1,r_2,\ldots,r_i\}$ is encoded by itself. In this way we get

$$\mathbb{H}^{(i)}_{\rm add} =_{\rm df} \bigcup_{n\geq 1}\{(n\,.\,\mathbf{x}\,.\,{\rm code}^{(i)}(\mathcal{M})) \mid \mathbf{x}\in\mathbb{R}^n \;\&\; \mathcal{M}\in\mathsf{M}_{\rm add} \;\&\; \mathcal{M}(\mathbf{x})\downarrow\}.$$

Moreover, let $\mathsf{M}^{r_1,\ldots,r_i}_{\rm add}$ be the set of the additive BSS machines using only the constants $r_1,\ldots,r_i$ and let

$$\mathbb{H}^{r_1,\ldots,r_i}_{\rm add} =_{\rm df} \bigcup_{n\geq 1}\{(n\,.\,\mathbf{x}\,.\,{\rm code}^{(i)}(\mathcal{M}))\in\mathbb{H}^{(i)}_{\rm add} \mid \mathcal{M}\in\mathsf{M}^{r_1,\ldots,r_i}_{\rm add}\}.$$

Then, $\mathbb{H}_{\rm add} \equiv \mathbb{H}^{r_1}_{\rm add} \precneqq \cdots \precneqq \mathbb{H}^{r_1,\ldots,r_{i-1}}_{\rm add} \precneqq \mathbb{H}^{(i-1)}_{\rm add} \equiv^{r_i} \mathbb{H}^{(i)}_{\rm add} \equiv^{r_1,\ldots,r_i} \mathbb{H}_{\rm add}$ holds for any $i\geq 3$. Note that r_{i+1} is transcendent over $\mathbb{Q}(r_1,\ldots,r_i)$. Consequently, we can take theses constants in proofs if we need a sequence of real transcendental numbers $\tau_1,\tau_2,\ldots$ such that τ_{i+1} is transcendent over $\mathbb{Q}(\tau_1,\ldots,\tau_i)$.

I would like to thank the participants of the meeting "Real Computation and BSS Complexity" in Greifswald for the discussion.

[Gaßner 2008] Gaßner, C.: "A hierarchy below the halting problem for additive machines"; Theory of Computer Systems 43 (2008) (3), 464–470.

Computing Real Functions with Rudimentary Operators

Ivan Georgiev

Burgas Prof. Assen Zlatarov University,
Faculty of Natural Sciences,
Prof. Yakimov Str. 1, 8010, Burgas, Bulgaria,
ivandg@yahoo.com

Abstract. The class of general recursive operators can be defined by an inductive definition in the same spirit as the inductive definition of the total recursive functions. We narrow the class of total recursive functions to its small subclass $\mathcal{M}^2$ and define the corresponding class $\mathbb{RO}$ of rudimentary operators. We also define the class $\mathbb{MSO}$ of $\mathcal{M}^2$-substitutional operators which turns out to be a proper subclass of $\mathbb{RO}$. The aim of this paper is to apply a general characterization theorem of D. Skordev in [2] to obtain that the two classes $\mathbb{MSO}$ and $\mathbb{RO}$ have equivalent computational power for computing real functions in the sense of A. Grzegorczyk from [1] and exactly the same power possesses an approach of K. Tent and M. Ziegler from [6], which avoids the use of infinitistic names of real numbers.

Keywords: limited minimum operation, $\mathcal{M}^2$, rudimentary operator, $\mathcal{M}^2$-substitutional operator, computable real function

References

1. Grzegorczyk, A., Computable functionals. Fund. Math., 42, 168–202 (1955)
2. Skordev, D., On some computability notions for real functions. (submitted for publication in Computability Journal), http://www.fmi.uni-sofia.bg/fmi/logic/skordev/on-some-computability-notions-for-real-functions.htm
3. Skordev, D., Some subrecursive versions of Grzegorczyk's Uniformity Theorem. Math. Logic Quart., 50, 520–524 (2004)
4. Skordev, D., Uniform computability of real functions. In: 120 Years Faculty of Mathematics and Informatics, St. Kliment Ohridski University of Sofia, Proceedings, St. Kliment Ohridski Press, Sofia, 179–185 (2011)
5. Skordev, D., Weiermann, A., Georgiev, I., $\mathcal{M}^2$-computable real numbers. J. Logic and Computation (Advance Access published September 21, 2010), doi:10.1093/logcom/exq050
6. Tent, K., Ziegler, M., Computable functions of reals. Münster J. Math., 3, 43–66 (2010)

HYPERBOLIC SYSTEMS WITH NON-COMPUTABLE BASINS OF ATTRACTION

D.S. GRAÇA AND N. ZHONG

Abstract. Dynamical systems are fascinating mathematical objects. They can be defined with simple rules and yet their "time-evolution" can be highly complex and difficult to describe analytically. A particular challenge is the problem of characterizing basins of attraction. In recent years, as fast computers have become available, much effort has been devoted to developing algorithms for estimating basins of attraction of various attractors. It therefore becomes useful to know whether or not these sets can actually be generated by computers.

It is well known that for hyperbolic rational functions, there are (polynomial-time) algorithms for computing basins of attraction and their complements (Julia sets) with arbitrary precision; in other words, basins of attraction and Julia sets of hyperbolic rational functions are (polynomial-time) computable.

However, the question of computability remains open for (analytic) non-rational systems. In this paper we show that:

Main Theorem. There is an analytic and computable dynamical system with a hyperbolic sink s such that the basin of attraction of s is not computable.

Thus our result implies that no algorithmic characterization exists, in general, for a given basin of attraction.

In the case of discrete-time systems, we prove the result by encoding a well-known non-decidable problem into the basin of attraction of s. In the case of continuous-time systems, we prove the result by embedding a discrete-time system with a non-computable basin of attraction into a continuous-time system. The standard suspension method (see V. I. Arnold and A. Avez *Ergodic problems of classical mechanics*, W.A. Benjamin, 1968, S. Smale *Differentiable dynamical systems*, Bull. Amer. Math. Soc. 73 (1967), 747 – 817) for embedding a discrete-time system into a continuous-time system is not sufficient for our case; we instead develop a new method.

CEDMES/FCT, Universidade do Algarve, C. Gambelas, 8005-139 Faro & SQIG - Instituto de Telecomunicações, Lisboa, Portugal
E-mail address: dgraca@ualg.pt

Department of Mathematical Sciences, University of Cincinnati, Cincinnati, OH 45221-0025, U.S.A.
E-mail address: ning.zhong@uc.edu

Dimension spectra of random subfractals of self-similar fractals

Xiaoyang Gu* Jack H. Lutz† Elvira Mayordomo‡ § ¶ P. Moser‖ §

Abstract

The (constructive Hausdorff) dimension of a point x in Euclidean space is the *algorithmic information density* of x. Roughly speaking, this is the least real number $\dim(x)$ such that $r \times \dim(x)$ bits suffices to specify x on a general-purpose computer with arbitrarily high precisions 2^{-r}. The *dimension spectrum* of a set X in Euclidean space is the subset of $[0, n]$ consisting of the dimensions of all points in X.

The dimensions of points have been shown to be geometrically meaningful (Lutz 2003, Hitchcock 2003), and the dimensions of points in self-similar fractals have been completely analyzed (Lutz and Mayordomo 2008). Here we begin the more challenging task of analyzing the dimensions of points in random fractals. We focus on fractals that are randomly selected subfractals of a given self-similar fractal. We formulate the specification of a point in such a subfractal as the outcome of an infinite two-player game between a *selector* that selects the subfractal and a *coder* that selects a point within the subfractal. Our selectors are algorithmically random with respect to various probability measures, so our selector-coder games are, from the coder's point of view, games against nature.

We determine the dimension spectra of a wide class of such randomly selected subfractals. We show that each such fractal has a dimension spectrum that is a closed interval whose endpoints can be computed or approximated from the parameters of the fractal. In general, the maximum of the spectrum is determined by the degree to which the coder can *reinforce* the randomness in the selector, while the minimum is determined by the degree to which the coder can *cancel* randomness in the selector. This constructive and destructive interference between the players' randomnesses is somewhat subtle, even in the simplest cases. Our proof techniques include van Lambalgen's theorem on independent random sequences, measure preserving transformations, an application of network flow theory, a Kolmogorov complexity lower bound argument, and a nonconstructive proof that this bound is tight.

*Linkedin Corporation, 2029 Stierlin Court, Mountain View, CA 94043, USA Email: xgu@linkedin.com

†Department of Computer Science, Iowa State University, Ames, IA 50011, USA. Email: lutz@cs.iastate.edu
Research supported in part by National Science Foundation Grants 0652569 and 0728806, and by Spanish Government MEC Grant TIN 2005-08832-C03-02. Part of this work was done during a sabbatical at Caltech and the Isaac Newton Institute for Mathematical Sciences at the University of Cambridge.

‡Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, 50018 Zaragoza, Spain. Email: elvira(at)unizar.es

§Research supported in part by Spanish Government MEC Grants TIN2005-08832-C03-02, TIN2008-06582-C03-02, and TIN2011-27479-C04-01.

¶Part of this author's research was performed during a visit at Iowa State University, supported by Spanish Government (Secretaría de Estado de Universidades e Investigación del Ministerio de Educación y Ciencia) grant for research stays PR2007-0368.

‖Department of Computer Science, National University of Ireland, Maynooth. Maynooth, Co. Kildare. Ireland. Email: pmoser(at)cs.nuim.ie.

Computing a Solution of Feigenbaum's Functional Equation in Polynomial Time

Peter Hertling, Christoph Spandl
Computer Science Department,
Universität der Bundeswehr München, Germany
peter.hertling@unibw.de, christoph.spandl@unibw.de

Abstract

Independently, Feigenbaum [2] and Großmann and Thomae [3], observed that the behaviour of the points of bifurcations of certain parameterized classes of dynamical systems on an interval obeys certain universal laws that are governed by constants which are now called Feigenbaum constants. For detailed presentations of these notions the reader is referred to [1]. In particular the so-called first Feigenbaum constant $\alpha = -2.50290787\ldots$ is the inverse $1/g(1)$ of the value $g(1)$ at 1 of a solution g of Feigenbaum's functional equation which was explicitly constructed by Lanford [5]. We show that this solution function g is a polynomial time computable function. This implies that the first Feigenbaum constant is a polynomial time computable number.

Which real numbers are computable? This question was one of the motivations for Alan Turing to write his famous papers [6, 7], in which he developed the notion of a Turing machine and gave a definition of computable real numbers. A real number c is called *computable*, if there is an algorithm (a Turing machine) which, given an arbitrary $n \in \mathbb{N}$ computes a rational number q_n satisfying $|c - q_n| < 2^{-n}$. A real number c is called *polynomial time computable* [4] if there are a Turing machine M and a polynomial p with coefficients in $\mathbb{N}$ such that M, given the string 1^n for any $n \in \mathbb{N}$, computes in at most $p(n)$ steps a binary string $a = a_m \ldots a_0$ (where m is an arbitrary natural number) and a binary string $b = b_1 \ldots b_n$ such that

$$|c - a.b| < 2^{-n}$$

where $a.b = a_m \ldots a_0.b_1 \ldots b_n$ is a dyadic rational number. Finally, a sequence $(c_k)_{k\in\mathbb{N}}$ of real numbers is called *polynomial time computable* if there are a Turing machine M and a two-variate polynomial p with coefficients in $\mathbb{N}$ such that M, given 1^k01^n for $k, n \in \mathbb{N}$, computes in at most $p(k, n)$ steps a binary string $a = a_m \ldots a_0$ (where m is an arbitrary natural number) and a binary string $b = b_1 \ldots b_n$ such that

$$|c_k - a.b| < 2^{-n}.$$

In order to formulate our main result precisely we need to introduce some terminology. We closely follow Lanford [5]. In fact, this paper by Lanford is the basis of our analysis.

Let M be the set of all continuously differentiable functions $f : [-1, 1] \to [-1, 1]$ satisfying the following conditions:

1. $f(0) = 1$,
2. $x \cdot f'(x) < 0$ for $x \neq 0$, i.e., f is strictly increasing on $[-1, 0]$ and strictly decreasing on $[0, 1]$,
3. $f(-x) = f(x)$ for all x, i.e., f is even.

Furthermore, let $D \subseteq M$ be the set of all functions in M satisfying additionally the following conditions:

1. $0 < -f(1)$,
2. $-f(1) < f(f(1))$,
3. $f(f(f(1))) \leq -f(1)$.

It is easy to check that for any function $f \in D$, the function Tf, defined by

$$Tf(x) := \frac{1}{f(1)} \cdot f(f(f(1) \cdot x))$$

is an element of M. Lanford [5] showed the following result.

Theorem 0.1 ([5, Theorem 1 and Prop. 2]) *There exists a function g, analytic and even on the set $\{z \in \mathbb{C} \; : \; |z| < \sqrt{8}\}$ and with real values on real numbers, whose restriction to $[-1, 1]$ is an element of D and a fixed point of the operator T.*

The so-called first Feigenbaum constant α is given by $\alpha = 1/g(1)$. We show:

Theorem 0.2 *1. The sequence of Taylor coefficients around 0 of this analytic function g is a polynomial time computable sequence of real numbers.*

2. The number $\alpha = 1/g(1)$ is a polynomial time computable real number.

Our proof is based on Lanford's paper [5].

References

[1] P. Collet and J.-P. Eckmann. *Iterated Maps on the Interval as Dynamical Systems.* Progress in Physics. Birkhäuser, Boston, Massachusetts, 1980.

[2] M. J. Feigenbaum. Quantitative universality for a class of nonlinear transformations. *Journal of Statistical Physics*, 19:25–52, 1978.

[3] S. Großmann and S. Thomae. Invariant distributions and stationary correlation functions of one-dimensional discrete processes. *Zeitschrift für Naturforschung*, 32a:1353–1363, 1977.

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

[5] O. E. Lanford. A computer-assisted proof of the Feigenbaum conjectures. *Bulletin of the AMS*, 6:427–434, 1982.

[6] A. M. Turing. On computable numbers, with an application to the "Entscheidungsproblem". *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936.

[7] A. M. Turing. On computable numbers, with an application to the "Entscheidungsproblem". A correction. *Proceedings of the London Mathematical Society*, 43(2):544–546, 1937.

Computability of semi-computable compact manifolds

Zvonko Iljazović

A closed subset of Euclidean space $\mathbb{R}^n$ is said to be computable if it can be effectively approximated by a finite set of points with rational coordinates with arbitrary precision on an arbitrary bounded region of $\mathbb{R}^n$. A closed subset of $\mathbb{R}^n$ is said to be co-computably enumerable (co-c.e.) if its complement can be effectively covered by open balls. If S is a computable set, then S is co-c.e. On the other hand, a co-c.e. set need not be computable, even if it is very simple from the topological viewpoint: in each $\mathbb{R}^n$ we can find two points such that the segment determined by these points is co-c.e., but not computable. However, it is known that the implication

$$S \text{ co-computably enumerable} \Rightarrow S \text{ computable} \tag{1}$$

holds if S is a topological sphere, i.e. if S is homeomorphic to the unit sphere in some $\mathbb{R}^m$. This result is valid not just in Euclidean space, but in every computable metric space which is locally computable. We now show that the implication (1) holds not just for topological spheres, but for all compact manifolds. Actually, we show that in every computable metric space the implication

$$S \text{ semi-computable} \Rightarrow S \text{ computable} \tag{2}$$

holds if S is a compact manifold. That a set S is semi-computable means that we can effectively enumerate all rational open sets which cover S. In Euclidean space (and in each locally computable computable metric space) a compact set is co-c.e. if and only if it is semi-computable.

In order to prove the main result, we examine the notion of computability up to some set. The idea is to show that in each semi-computable compact manifold S each point has a neighborhood which is computable up to its complement and then, using compactness of S, to conclude that S is computable as a finite union of such neighborhoods. The same idea can be used to show a more general result: if S is a semi-computable compact manifold with boundary, then S is computable if its boundary ∂S is computable (or semi-computable). This is a generalization of the following known result: is S is a co-c.e. cell with co-c.e. boundary sphere, then S is computable.

Computational Complexity of Ordinary Differential Equations

Akitoshi Kawamura
University of Tokyo

How complex can the solution be to an ordinary differential equation given by a polynomial-time computable function? This question can be given a natural and precise sense in computational complexity theory by refining the standard definitions in Computable Analysis. In this talk, I will start with the definitions of polynomial-time computability of real functions and operators, and then present several results about the complexity of ordinary differential equations. It turns out that imposing stronger conditions on the input function (such as being Lipschitz continuous, smooth, analytic) makes the solution computationally simpler.

Uniform Polytime Computable Operators on Univariate Real Analytic Functions

A. Kawamura[1], N. Müller[2], C. Rösnick[3], and M. Ziegler[3]

[1] University of Tokyo, kawamura@is.s.u-tokyo.ac.jp
[2] Universität Trier, mueller@uni-trier.de
[3] TU Darmstadt, {roesnick,ziegler}@mathematik.tu-darmstadt.de

Recursive analysis as initiated by Turing (1937) explores the in-/computability of problems involving real numbers and functions by approximation up to prescribable absolute error 2^{-n}. Weihrauch's Type-2 Theory of Effectivity (TTE) extends this to mappings from/to the Cantor space of infinite binary strings encoding continuous universes in terms of so-called representations. Refining mere computability, classical complexity theory has successfully been transferred from the discrete to the real realm; see for instance the works of Harvey Friedman, Ker-I Ko, and Norbert Th. Müller in the 1980ies or, more recently and exemplarily, [4, 1]. However the common setting only covers real numbers x and (continuous) real functions f; operators $\mathcal{O}$ could be investigated merely in the non-uniform sense of mapping polytime computable functions to polytime ones — yielding strong lower bounds but weakly meaningful upper bounds for actual implementations of exact real number computation like `iRRAM`. As a major obstacle towards a uniform complexity theory for operators, the computable evaluation of a 'steep' function $f : x \mapsto f(x)$ requires more precision on x to approximate $y = f(x)$ than a 'shallow' one. More precisely as quantitative refinement of the sometimes so-called *Main Theorem* of recursive analysis, the (optimal) modulus of continuity of f constitutes a lower bound on its complexity — hence the evaluation operator cannot be computable on entire $C[0, 1]$ in time bounded by the output precision n only.

In [2] one of the authors and his advisor have proposed and exemplified a structural complexity theory for operators $\mathcal{O}$ from/to continuous functions f on Cantor space — given by approximations as so-called regular string functions, that is, mappings $\varphi :\subseteq \{0,1\}^* \to \{0,1\}^*$ whose output length $|\varphi(\vec{\sigma})|$ depends only, and monotonically, on the input length $|\vec{\sigma}|$. They consider Turing machines converting such φ (given as a function oracle) and $\vec{\tau} \in \{0,1\}^*$ to $\mathcal{O}(\varphi)(\vec{\tau})$ in time uniformly bounded by a second-order polynomial in the sense of Kapron&Cook (1996) depending on both $|\vec{\tau}|$ and $|\varphi|$. For real operators, a reasonable (second-order) representation of functions $f \in C[0, 1]$ as regular string functions φ amounts to $|\varphi|$ upper bounding f's modulus of continuity and renders evaluation (second-order) polynomial-time computable.

We further flesh out this theory by devising and comparing second-order representations and investigating the computational complexity of common (possibly multivalued) operators in analysis. Specifically, two rather different (multi-)representations are suggested for the space

$$\big\{(U, C, f|_C) : U \subseteq \mathbb{C} \text{ open}, \ C \subseteq U \text{ compact convex}, \ f : U \to \mathbb{C} \text{ analytic}\big\}$$

of functions analytic on (a complex neighbourhood of) a compact convex set and shown to be polytime equivalent. We then present second-order polytime algorithms computing some of the operators on this space that in [3] had been shown non-uniformly polytime computable.

References

1. I. Binder, M. Braverman, M. Yampolsky: "On the Computational Complexity of the Riemann Mapping", pp.221–239 in *Arkiv för Matematik* vol.**45:2** (2007).
2. A. Kawamura, S.A. Cook: "Complexity Theory for Operators in Analysis", pp.495–502 in *Proc. 42nd Ann. ACM Symp. on Theory of Computing* (STOC 2010).
3. K.-I. Ko: "*Computational Complexity of Real Functions*", Birkhäuser (1991).
4. R. Rettinger, K. Weihrauch: "The Computational Complexity of Some Julia Sets", pp.177–185 in *Proc. 35th Ann. ACM Symp. on Theory of Computing* (STOC 2003).

[1] The first author is supported in part by Kakenhi (Grant-in-Aid for Scientific Research) 23700009.

A Testable Abstract Data Type of Outer and Inner Real Approximations

Michal Konečný

We explore the foundations of an approach to reliable real number programming that supports denotational exact real semantics and is also efficient to execute as it is similar to iRRAM. Instead of building a minimalistic foundation for real number computation as do RealPCF or TTE, we provide a suitable abstraction of safely rounded interval arithmetic by formalising its key properties. These properties define an abstract data type (ADT) of real number approximations forming a family of approximate rounded fields that converges to the field of real numbers.

The ADT has a denotational semantics based on a continuous lattice within which it is easy to identify the field of real numbers, providing an alternative constructive definition of the real numbers. Thus one can build semantically transparent exact real algorithms over the ADT.

This ADT can serve as an interface that decouples exact real number algorithms of the iRRAM style from the underlying interval arithmetic, providing a formal specification of rounded interval arithmetic for verification and preventing any inappropriate use of interval arithmetic that violates its use as real number approximations, such as accessing the endpoints directly.

Moreover, the ADT is carefully constructed so that all the properties are testable using QuickCheck. Being able to "QuickCheck" an implementation of real number arithmetic against a complete axiomatisation of the real numbers is very valuable, whether or not formal verification is performed. QuickCheck provides counter-examples when a property violation is detected, speeding up localisation of mistakes. In an absence of formal verification, using QuickCheck with our ADT is a cheap way to gain a very high level of confidence in the reliability of an implementation of interval arithmetic.

A variant of this approach has been adopted in the AERN library, where both interval arithmetic and polynomial interval arithmetic implement a real number ADT. The polynomial arithmetic provides a data type of continuous real functions with real operations applied point-wise over the whole domain of the functions.

Figure 1 outlines, using an informal UML-like notation, the main concepts covered in this work and other related concepts. The ADT mentioned above corresponds to the box "in/out-rounded approximate reals" in the figure.

Note that in our approach we need to consider both outer- and inner-rounded interval operators whereas iRRAM uses only the outer-rounded operators. We

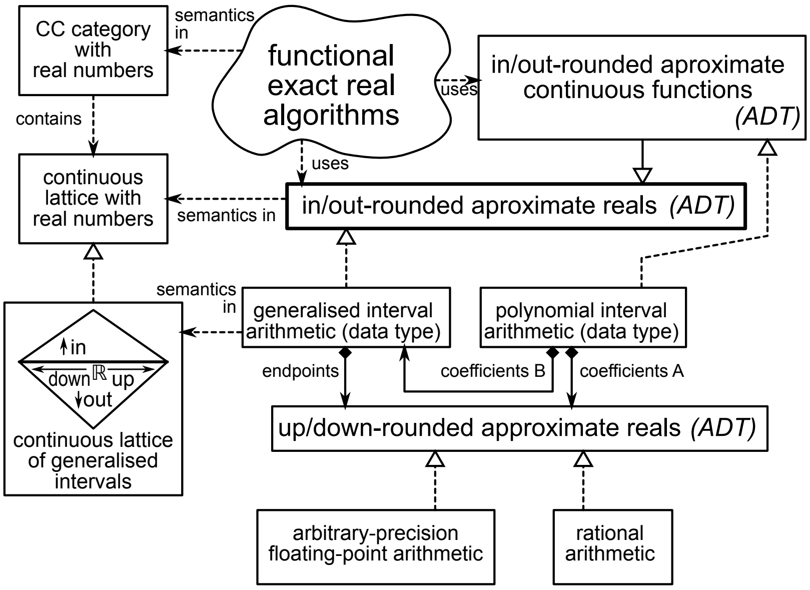

Figure 1: The main concepts covered in this paper and other related concepts.

require inner-rounded operators to be able to express rounded versions of properties such as commutativity and associativity. Also, the difference between a result obtained with outer-rounded operations and a result obtained using inner-rounded operations gives us a tangible measure of imprecision without having to refer to the exact result (i.e. result that would be obtained using exact interval operations).

A CONSTRUCTIVE VIEW OF CONTINUITY PRINCIPLES: ABSTRACT

ROBERT S. LUBARSKY

In constructive mathematics, distinctions can be drawn that cannot be made classically. This talk will examine principles, many fairly recent, that have come up within constructivism related to continuity, and their variants, with an eye toward open questions.

The first set of topics has to do with the boundedness principle BD-N [7]. A set of natural number A is said to be pseudo-bounded if every sequence (a_n) of members of A is eventually dominated by the identity function. BD-N is the assertion that every inhabited countable pseudo-bounded set is bounded. (Recall that a set is countable if it is the range of a function with domain the natural numbers. So it might not happen that every set of natural numbers is countable, as there may be some sets that just cannot be enumerated.) Over some basic theory of sets or of analysis, with Countable Choice, BD-N is equivalent to the continuity (that is, standard $\epsilon - \delta$ continuity) of every sequentially continuous map from a separable metric space to a metric space. Arguably BD-N is quite subtle, and hence identified only so late, because it is true in all the major schools of mathematics: Brouwer's intuitionism, Russian computably-based constructivism, and classical mathematics. However, it is independent of analysis and set theory [8, 9].

As weak as BD-N is, there have recently been identified principles that are strictly weaker, yet still not outright provable. One is the closure of anti-Specker spaces under Cartesian product [2, 5]. An anti-Specker space is one that does not allow for the existence of anything like the well-known Specker sequence from computable analysis. Since this is a form of compactness, one might well expect closure under product. Another is the Riemann Permutation Theorem [3, 4], a version of the introductory analysis theorem that any conditionally convergent sequence can be rearranged to converge to any desired real number. The last one we consider is that every partially Cauchy sequence is Cauchy, where a sequence is partially Cauchy if the diameters of arbitrarily long albeit finite sub-sequences go to 0. We will discuss the known independence proofs of these principles from standard base theories, and of BD-N from them, as well as the questions we would like to see addressed next [10].

The second set of topics is Brouwer's famous Fan Theorem and its topical variants. The Fan Theorem is the classical contrapositive of König's Lemma. The latter states that every infinite, finitely branching tree has an infinite branch, the former that every binary tree (i.e. subset of $2^{<\omega}$) with no infinite branch is finite. More commonly put, a bar B is defined to be a set of nodes (of $2^{<\omega}$) such that every infinite path contains a node from B. The Fan Theorem states that every bar is uniform. This could be viewed as the compactness of Cantor space.

Continuous functions from a compact metric space are uniformly continuous, so one would expect the Fan Theorem to yield some kind of implication from continuity to uniform continuity. This is indeed the case, in a very strong sense: some weakenings of the Fan Theorem are equivalent to the step from continuity to uniform continuity over certain spaces [1, 6]. These weakenings consist of restricting the relevant bars to a limited collection, namely to bars that are easily definable. As with BD-N, we will discuss the models showing non-implications among these variants of Fan [11], as well as the questions we would like to see addressed.

References

[1] Josef Berger, "The logical strength of the uniform continuity theorem," in Logical Approaches to Computational Barriers (eds. Beckmann, Berger, Löwe, and Tucker), Lecture Notes in Computer Science, Springer, 2006, p. 35 - 39

[2] Josef Berger and Douglas Bridges, "The anti-Specker property, a Heine-Borel property, and uniform continuity," **Archive for Mathematical Logic**, v. 46 (2008), p. 583-592

[3] Josef Berger and Douglas Bridges, "Rearranging series constructively," **Journal of Universal Computer Science**, v. 15 (2009), p. 3160-3168

[4] Josef Berger, Douglas Bridges, Hannes Diener, and Helmut Schwichtenberg, "Constructive aspects of Riemanns permutation theorem for series," in preparation

[5] Douglas Bridges, "Inheriting the anti-Specker property," preprint, University of Canterbury, NewZealand, 2009, submitted for publication

[6] Hannes Diener and Iris Loeb, "Sequences of real functions on [0, 1] in constructive reverse mathematics," **Annals of Pure and Applied Logic**, v. 157(1) (2009), p. 50-61

[7] Hajime Ishihara, "Continuity properties in constructive mathematics," **Journal of Symbolic Logic**, v. 57 (1992), p. 557-565

[8] Peter Lietz, "From constructive mathematics to computable analysis via the realizability interpretation," Ph.D. thesis, Technische Universität Darmstadt, 2004, http://www.mathematik.tu-darmstadt.de/ streicher/THESES/lietz.pdf.gz

[9] Robert Lubarsky, "The failure of BD-$\mathbb{N}$, and an application to the anti-Specker property," **Journal of Symbolic Logic**, to appear

[10] Robert Lubarsky and Hannes Diener, "Principles Weaker Than BD-$\mathbb{N}$," submitted for publication

[11] Robert Lubarsky and Hannes Diener, "Separating the Fan Theorem and Its Weakenings," available at math.fau.edu/lubarsky/pubs.html

Dept. of Mathematical Sciences, Florida Atlantic University, Boca Raton, FL 33431, USA

E-mail address: `Lubarsky.Robert@comcast.net`

Effective Dimension in Euclidean space

Elvira Mayordomo* †

Abstract

Effective dimension was introduced by Lutz [3,4] in order to quantitatively analyze complexity classes. The resulting concepts of effective dimension have turned out to be robust, since they have been shown to admit several equivalent definitions that relate them to well-studied concepts in computation, and they have proven very fruitful in investigating not only the structure of complexity classes but also in the modeling and analysis of sequence information. For a survey on the applications of effective dimension to the study of complexity classes see [2] and for the Information Theory connections see [5,1].

In this talk, we will survey the most recent developments in effective dimension, those that are back in fractal geometry. We will present the use of effective dimension in Euclidean space, its main robustness properties and all known applications in fractal geometry, very related to the new concept of "dimension of a point".

References

[1] R. Downey and D. Hirschfeldt. *Algorithmic randomness and complexity.* Springer-Verlag, 2010.

[2] J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. *The fractal geometry of complexity classes.* SIGACT News Complexity Theory Column, 36:24 - 38, 2005.

[3] J. H. Lutz. *Dimension in complexity classes.* SIAM Journal on Computing, 32:1236 – 1259, 2003.

[4] J. H. Lutz. *The dimensions of individual strings and sequences.* Information and Computation, 187:49 - 79, 2003.

[5] E. Mayordomo. *Effective fractal dimension in algorithmic information theory.* In New Compu- tational Paradigms: Changing Conceptions of What is Computable, pages 259 - 285. Springer- Verlag, 2008.

*Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, 50018 Zaragoza, Spain. Email: elvira(at)unizar.es

†Research supported in part by Spanish Government MEC Grant TIN2011-27479-C04-01.

COMPUTABLY CATEGORICAL SPACES

ALEXANDER G. MELNIKOV

Pour-El and Richards [2] were the first to study computable Banach spaces up to computable isometries. We observe that it is the category of computable metric spaces computable isometries are the natural morphisms, and introduce the notion of a *computably categorical metric space*:

Definition 0.1. We say that an uncountable metric space is *computably categorical* if every two computable structures on this space are equivalent up to a computable isometry.

Our definition is motivated by the classical notion of a computably categorical (autostable) *countable* algebraic structure due to Mal'cev [1] and Rabin [3]. We also observe that our definition can be easily modified to the case of computable Banach spaces: it is sufficient to restrict ourselves to computable isometries which are Banach space isomorphisms. Our approach in the case of Banach spaces is equivalent to the one from Pour-El and Richards [2] mentioned above.

We show that Cantor space, the Urysohn space, and every separable Hilbert space are computably categorical as metric spaces, but the space $\mathcal{C}[0, 1]$ of continuous functions on the unit interval with the supremum metric is not (as am metric space). We also characterize computably categorical metric subspaces of $\mathbb{R}^n$, and give a sufficient condition for a metric space to be computably categorical.

We hope that our research will help in establishing new links between computable model theory and computable analysis.

References

[1] Mal'cev, A. *Constructive algebra I*, Russian Math. Surveys **16** (1961), 77–129.

[2] Marian B., and Richards, J. Computability in Analysis and Physics. Perspectives in Mathematical Logic, Volume 1 Berlin: Springer-Verlag, 1989.

[3] Rabin, M. Computable algebra, general theory and theory of computable fields. Trans. Amer. Math. Soc., 95: 341 - 360, 1960.

Weak L^1-computability and Limit L^1-computability

Kenshi Miyabe*

Brattka, Miller and Nies [1] showed that some randomness notions are characterized by differentiability of some classes of functions. They also proposed to study which class corresponds to which randomness notion. Pathak, Rojas and Simpson [3] and independently Rute [4] showed that a real in the unit interval is Schnorr random if and only if the Lebesgue differentiation theorem for the point holds for all effective version of L^1-computable functions. Then its other randomness versions are of our interest. The author [2] gave several characterizations of the class of the effective version of L^1-computable functions. Then we also study its other randomness versions.

Let (X, d, α) be a computable metric space and μ be a computable measure on it. The following definition and result are by [2]. A *integral test for Schnorr randomness* is a nonnegative lower semicomputable function $f :\subseteq X \to \overline{\mathbb{R}}$ whose integral is computable. A function f is *L^1-computable with an effective code* if there exists a computable sequence $\{s_n\}$ of finite rational step functions such that $f = \lim_n s_n$ and $||s_{n+1} - s_n||_1 \leq 2^{-n}$ for all n.

Definition 1 ([2])**.** *Let $f :\subseteq X \to \mathbb{R}$ be a function whose domain is the set of Schnorr random points. Then f is L^1-computable with an effective code iff f is the difference between two integral tests for Schnorr randomness.*

The following is the Martin-löf randomness verions of this result. Recall that an *integral test* is a nonnegative lower semicomputable function $t : X \to \overline{\mathbb{R}}$ with $\int t d\mu < \infty$.

Definition 2. *A function $f :\subseteq X \to \mathbb{R}$ is weakly L^1-computable if there exists a computable sequence $\{s_n\}$ of finite rational step functions such that $f(x) = \lim_n s_n(x)$ and $\sum_n ||s_{n+1} - s_n||_1 < \infty$.*

Theorem 3. *Let $f :\subseteq X \to \mathbb{R}$ be a function whose domain is the set of Martin-Löf random points. Then f is weakly L^1-computable iff f is the difference between two integral tests.*

Similarly we can give the weak 2-randomness version.

The author gave another characterization of the effective L^1-computability via Schnorr layerwise computability. We say a function $f :\subseteq X \to \mathbb{R}$ is *Schnorr*

*Research Institute for Mathematical Sciences, Kyoto University, kmiyabe@kurims.kyoto-u.ac.jp

layerwise computable if there exists a Schnorr test $\{U_n\}$ such that the restriction $f|_{X\setminus U_n}$ is uniformly computable.

Theorem 4 ([2])**.** *Let $f :\subseteq X \to \mathbb{R}$ be a function whose domain is the set of Schnorr random points. Then f is Schnorr layerwise computable and its L^1-norm is computable iff f is the difference between two integral tests for Schnorr randomness.*

To study the other randomness versions of this result, we introduce Solovay reducibility for nonnegative lower semicomputable functions. Recall the following characterization of Solovay reducibility. For left-c.e. reals α and β, $\alpha \leq_S \beta$ iff there are a constant d and a left-c.e. real γ such that $d\beta = \alpha + \gamma$.

Definition 5. *Let f, g be nonnegative lower semicomputable functions. We say that f is Solovay reducible to g (denoted by $f \leq_S g$) if there exists a computable real d and a nonnegative lower semicomputable function h such that*

$$d \cdot g =_{\mathrm{WR}} f + h.$$

Recall that a *Solovay test for Schnorr randomness* is a sequence $\{U_n\}$ of uniformly c.e. open sets such that $\sum_n \mu(U_n)$ is computable.

Theorem 6. *A nonnegative lower semicomputable function f has a computable integral iff there exist a computable sequence $\{a_n\}$ of natural numbers and a Solovay test $\{U_n\}$ for Schnorr randomness such that $f \leq_S \sum_n a_n \cdot \mathbf{1}_{U_n}$ and $\sum_n a_n \mu(U_n)$ is computable.*

This theorem implies one implication of Theorem 4. Hence Solovay reducibility for nonnegative lower semicomputable functions will be a useful tool to study the relation between randomness notions and computability (like Schnorr layerwise computability).

References

[1] V. Brattka, J. S. Miller, and A. Nies. Randomness and differentiability. Submitted.

[2] K. Miyabe. An integral test for Schnorr randomness and its application. Submitted.

[3] N. Pathak, C. Rojas, and S. G. Simpson. Schnorr randomness and the Lebesgue Differentiation Theorem. To appear in Proceedings of the American Mathematical Society.

[4] J. Rute. Randomness, martingales and differentiability. In preparation, 2011.

Computability of Probability Distributions and Characteristic Functions

Takakazu Mori * Yoshiki Tsujii † Mariko Yasugi ‡

We have worked on mutual relationships between computability of probability distributions (Borel probability measures on the real line $\mathbb{R}$) and that of the corresponding probability distribution functions. As a sequel, we here deal with the same theme on probability distributions and the corresponding characteristic functions. We will show that the computability property as well as the effective convergence mutually transfer (under certain conditions) between those objects.

The characteristic function φ of a probability distribution μ is defined as the Fourier transformation, that is, $\varphi(t) = \int_R e^{itx}\mu(dx) = \mu(e^{it\cdot})$, and is characterized by the following properties: (Ci) φ is positive definite, (Cii) $\varphi(t)$ is continuous at 0, (Ciii) $\varphi(0) = 1$.

1 Preliminaries

Definition 1.1 (Computability of probability distributions [2]) *A sequence of probability distributions $\{\mu_m\}$ is said to be computable, if $\{\mu_m(f_n)\}$ is a computable double sequence of real numbers for any computable sequence of functions $\{f_n\}$ with a recursive compact support $L(n)$.*

We denote $\int_{\mathbb{R}} f(x)\mu(dx)$ with $\mu(f)$.

Definition 1.2 (Effective convergence of probability distributions [2]) *A sequence of probability distributions $\{\mu_m\}$ is said to converge effectively to a probability distribution μ if there exists a recursive function $\alpha(n,k)$ such that*

$$|\mu_m(f_n) - \mu(f_n)| < 2^{-k} \text{ if } m \geq \alpha(n,k) \tag{1}$$

holds for any computable sequence of functions $\{f_n\}$ with compact support.

Proposition 1.3 (Effective tightness of an effectively convergent sequence, Effectivization of Lemma 15.4 in [3]) *If a computable sequence of probability distributions $\{\mu_m\}$ effectively converges to a probability distribution μ, then there exists a recursive function $\alpha(k)$ such that $\mu_m(w^c_{\alpha(k)}) < 2^{-k}$ for all m.*

It also holds that $\mu_m([-\alpha(k)-1, \alpha(k)+1]^C) < 2^{-k}$ for all m.

$\{f_n\}$ is said to be effectively bounded, if there exists a recursive $M(n)$ such that $|f_n(x)| \leq M(n)$.

Proposition 1.4 ([2]) *If $\{\mu_m\}$ is computable, then it is* weakly sequentially computable*, that is, $\{\mu_m(f_n)\}$ is a computable sequence of reals for all effectively bounded computable sequence of functions $\{f_n\}$.*

Theorem 1.5 (Effective dominated convergence theorem for dx) *Let $\{g_{m.n}\}$ be a computable sequence of functions which converges effectively to $\{f_m\}$. Assume that there exists an effectively integrable computable sequence of functions $\{h_m\}$ such that $|g_{m,n}(x)| \leq h_m(x)$.*

Then $\{g_{m,n}\}$ is effectively integrable and $\{\int_{\mathbb{R}} g_{m,n}(x)dx\}$ converges effectively to $\{\int_{\mathbb{R}} f_m(x)dx\}$ as n tends to infinity effectively in m.

Theorem 1.6 *Let $\{f_n(x,y)\}$ be a computable sequence of binary functions and let $\{\mu_m\}$ be a computable sequence of probability distributions.*

*Faculty of Science, Kyoto Sangyo University: morita@cc.kyoto-su.ac.jp
†Faculty of Science, Kyoto Sangyo University: tsujiiy@cc.kyoto-su.ac.jp
‡Graduate School of Letters, Kyoto University: yasugi@cc.kyoto-su.ac.jp

(1) *If* $\{f_n(x,y)\}$ *is effectively bounded, then, as a function of* x, $\{\int_{\mathbb{R}} f_n(x,y)\mu_m(dy)\}$ *is an effectively bounded computable double sequence of functions of* x.

(2) *If there exists an effectively integrable computable function* $g(y)$ *such that* $|f_n(x,y)| \le g(y)$, *then* $\{\int_{\mathbb{R}} f_n(x,y)dy\}$ *is a computable sequence of functions of* x.

2 Characteristic functions

In the following, $\{\mu_m\}$ and μ will respectively denote a sequence of probability distributions and a probability distribution, and φ_m and φ will denote the corresponding characteristic functions.

Theorem 2.1 *If* $\{\mu_m\}$ *is computable, then* $\{\varphi_m\}$ *is uniformly computable.*

Theorem 2.2 (Effective Glivenko, cf, Theorem 2.6.4 in [1]) *Let* $\{\varphi_m\}$ *be computable. Then,* $\{\mu_m\}$ *converges effectively to* μ *if* $\{\varphi_m\}$ *converges effectively to* φ.

Theorem 2.3 (Effectivization of Theorem 2.6.3 in [1]) *Let* $\{\mu_m\}$ *and* μ *be computable. Assume that the corresponding probability distribution functions* $\{F_m\}$ *and* F *are Fine computable. Then,* $\{\varphi_m\}$ *converges effectively (compact-uniformly) to* φ *if* $\{\mu_m\}$ *converges effectively to* μ.

Theorem 2.4 *If* $\{\varphi_m\}$ *is computable, then* $\{\mu_m\}$ *is computable.*

Theorem 2.5 (Effective Bochner's theorem) *In order for* $\varphi(t)$ *to be a characteristic function of a computable probability distribution, it is necessary and sufficient that the following three conditions hold.*

(i) φ *is positive definite.* (ii′) φ *is computable.* (iii) $\varphi(0) = 1$.

3 De Moivre-Laplace Central Limit Theorem

Let $(\Omega, \mathcal{B}, \mathbb{P}, \{X_n\})$ be a realization of Coin Tossing with success probability p, that is, $(\Omega, \mathcal{B}, \mathbb{P})$ is a probability space and $\{X_n\}$ is an independent sequence of $\{0,1\}$-valued random variables with the same probability distribution $\mathbb{P}(X_n = 1) = p$ and $\mathbb{P}(X_n = 0) = q = 1 - p$.

The probability distribution of $S_m = X_1 + \cdots + X_m$ is the binomial distribution $\mu_m = \sum_{\ell=0}^{m} \binom{m}{\ell} p^\ell (1-p)^{m-\ell}\delta_\ell$ and its characteristic function $\varphi_m(t)$ is equal to $(pe^{it} + q)^m$.

Theorem 3.1 (Effective de Moivre-Laplace central limit theorem) *If* p *is a computable real number, then the probability distribution function of* $Y_m = \frac{X_1 + \cdots X_m - mp}{\sqrt{mp(1-p)}} = \sum_{\ell=1}^{m} \frac{X_\ell - p}{\sqrt{mpq}}$ *converges effectively to the standard Gaussian distribution.*

References

[1] Ito, K. An Introduction to Probability Theory. Cambridge University Press, 1978.

[2] Mori, T., M. Yasugi and Y. Tsujii. *Computability of probability distributions and probability distribution functions.* Proceedings of the Sixth International Conference on Computability and Complexity in Analysis (DROPS 20), 185-196, 2009.

[3] Tsurumi, S., Probability Theory. Shibunndou, 1964. (in Japanese)

“Almost everywhere” theorems and algorithmic randomness

André Nies
University of Auckland, New Zealand

Several important theorems in analysis assert a property for almost every real z. For instance, the Lebesgue density theorems says that for a measurable set E, almost every $z \in E$ has density 1 in E.

Now consider the case where the given objects are effective in some sense. How strong an algorithmic randomness notion for a real z is needed to make the theorem hold at z? Will the theorem in fact characterize the randomness notion?

In [2] we analyzed an effective version of the Lebsegue differentation theorem in this way. We showed that a real z is computably random if and only if each nondecreasing computable function on the unit interval is differentiable at z.

In the past 15 months there has been considerable progress when the given objects are effective in some weak sense, but not necessarily computable. In this case, usually Martin-Loef randomness of z is not enough, but the somewhat stronger notion of difference randomness (that is, ML-randomness together with Turing incompleteness) often suffices. In [1] we show that a ML-random real is difference random if and only if it has positive density in every effectively closed set containing it. We also use the concept of non-porosity to show that all Banach-Mazur computable functions satisfy the Denjoy alternative at difference random reals.

In [3] we consider nondecreasing functions g, called interval-r.e., where $g(b) - g(a)$ is left-r.e. uniformly in rationals a, b with $0 \leq a < b \leq 1$. We show that each such continuous function is the variation of a computable non-decreasing function. Forthcoming work with Bienvenu, Greenberg, Kucera, and Turetsky shows that a randomness notion of z slightly stronger than difference randomness makes these functions differentiable at z, and in particular ensures density 1 in in every effectively closed set containing the real.

[1] Bienvenu, Hoelzl, Miller and Nies. The Denjoy alternative for computable functions. Proceedings of STACS 2012.
[2] Brattka, Miller, and Nies. Randomness and differentiability. Submitted.
[3] Freer, Kjos-Hanssen, Nies and Stephan. Effective aspects of Lipschitz functions. Submitted.

OVERCOMING INTRACTABLE COMPLEXITY IN AN AUTOMATIC THEOREM PROVER FOR REAL-VALUED FUNCTIONS

PROF. LAWRENCE C. PAULSON

Real quantifier elimination, first established by Tarski [8] and later refined by Collins [2] and others, implies the decidability of first-order formulas involving the familiar arithmetic operations over the real numbers. Giving necessary and sufficient conditions for the existence of real roots of polynomials amounts to quantifier elimination. A classic example is the quadratic equation $ax^2 + bx + c = 0$, which has a real solution subject to surprisingly complicated conditions:

$$\exists x\, [ax^2 + bx + c = 0] \iff b^2 \geq 4ac \wedge (c = 0 \vee a \neq 0 \vee b^2 > 4ac).$$

We are accustomed to simply $b^2 \geq 4ac$ as a sufficient condition, but the full formula covers the degenerate cases $a = 0$ and $b = 0$. Even in this trivial example, eliminating a quantifier greatly increases the formula's Boolean complexity. Quantifier elimination is possible regardless of the degrees of the polynomials or the logical complexity of the formula. Given the tremendous power of this procedure, it is hardly surprising to learn that its complexity is intractable [4]: the length of the resulting quantifier-free formula can be doubly exponential in the number of quantified variables.

Researchers have made strenuous efforts to design efficient quantifier elimination procedures for well-behaved problem classes. Literature surveys include Dolzmann et al. [5] and Passmore [6]. The decision problem is called RCF, for "real-closed fields": fields that are elementarily equivalent to the field of real numbers.

Augmenting the language of polynomials with real-valued functions such as ln, exp, sin, cos, $\tan^{-1}$ obviously makes the decision problem even more difficult. Few decision procedures exist for such extended languages, regardless of complexity. This suggests the use of heuristic methods.

MetiTarski is an automatic theorem prover for first-order logic including polynomials and real-valued special functions. It solves problems in this extended language using a combination of resolution theorem proving and RCF decision procedures [1]. The key idea is to provide upper and lower bounds for each function of interest. Such bounds will typically be polynomials or rational functions obtained from power series or continued fraction expansions [3]. Inevitably, we need families of bounds, valid over various intervals, and trading accuracy against simplicity. Resolution uses these bounds (supplied as axioms) to reduce a problem involving special functions to problems involving rational functions, and ultimately to problems in RCF, which can then be solved by a decision procedure.

Despite the terrible complexity of real quantifier elimination, MetiTarski uses it as a subroutine. And in many cases, MetiTarski can prove difficult theorems in a couple of seconds. Here is an example:

$$\forall\, t > 0, v > 0$$
$$((1.565 + 0.313\, v)\, \cos(1.16\, t) + (.0134 + .00268\, v)\, \sin(1.16\, t))\, e^{-1.34\, t} - (6.55 + 1.31\, v)\, e^{-0.318\, t} + v \;\geq\; -10$$

MetiTarski actually outputs a proof, not one that a mathematician would want to read, but a detailed formal deduction in the resolution calculus.

The complexity of real quantifier elimination imposes strict limits on the number of variables allowed in a problem. This is largely dependent on the choice of RCF decision procedure. Early work used only QEPCAD, and theorems in more than two variables could seldom be proved. More recently, we have incorporated Mathematica, and proved theorems with up to 5 variables. The latest work uses the decision procedure Z3, which now supports non-linear arithmetic. With the help of heuristics specialised to MetiTarski [7], theorems with up to 9 variables can be proved.

Acknowledgements. The developers of MetiTarski include Behzad Akbarpour, James Bridge and Grant Passmore. The research was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/C013409/1, EP/I011005/1 and EP/I010335/1].

References

[1] Behzad Akbarpour and Lawrence Paulson. MetiTarski: An automatic theorem prover for real-valued special functions. *Journal of Automated Reasoning*, 44(3):175–205, March 2010.

[2] George E. Collins. Hauptvortrag: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In H. Barkhage, editor, *Automata Theory and Formal Languages*, volume 33 of *LNCS*, pages 134–183. Springer, 1975.

[3] A. Cuyt, V. Petersen, B. Verdonk, H. Waadeland, and W.B. Jones. *Handbook of Continued Fractions for Special Functions*. Springer, 2008.

[4] J. H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *J. Symbolic Comp.*, 5:29–35, 1988.

[5] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. Real quantifier elimination in practice. Technical Report MIP-9720, Universität Passau, D-94030, Germany, 1997.

[6] Grant Olney Passmore. *Combined Decision Procedures for Nonlinear Arithmetics, Real and Complex*. PhD thesis, University of Edinburgh, 2011.

[7] Grant Olney Passmore, Lawrence C. Paulson, and Leonardo de Moura. Real algebraic strategies for MetiTarski proofs. In Johan Jeuring, editor, *Conferences on Intelligent Computer Mathematics — CICM 2012*. Springer, 2012. In press.

[8] Alfred Tarski. A decision method for elementary algebra and geometry. Technical Report R-109, RAND Corporation, 1948.

Computer Laboratory, Univ. of Cambridge, England
E-mail address: lp15@cl.cam.ac.uk

Compactness and separation for represented spaces (Extended Abstract)

Arno Pauly
Clare College
University of Cambridge, United Kingdom
Arno.Pauly@cl.cam.ac.uk

Abstract

The effective notions of compactness and topological separation are studied in full generality for represented spaces. Each concept is characterized by a multitude of equivalent properties, each corresponding the computability or continuity of certain functions. In particular, admissibility is identified as the effective counterpart to T_0 separation. A synthetic approach allows simple proofs of strong results compared to the previous literature.

In the following, the main results of this work are listed. Note that for arbitrary represented spaces $\mathbf{X}$, $\mathbf{Y}$ we use $\mathcal{C}(\mathbf{X}, \mathbf{Y})$ for the function space from $\mathbf{X}$ to $\mathbf{Y}$, and $\mathcal{A}(\mathbf{X})$, $\mathcal{O}(\mathbf{X})$ and $\mathcal{K}(\mathbf{X})$ for the space of closed, open and saturated compact subsets of $\mathbf{X}$. A complete version is available at the arXiv as [4]. Most of these results are generalizations of known ones, e.g. in [5, 2, 1, 8, 7] using the mindset of synthetic topology [3].

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

1. $\mathbf{X}$ is (computably) compact, i.e. the map $\mathrm{IsEmpty}\,\mathbf{X} : \mathcal{A}(\mathbf{X}) \to \mathbb{S}$ is continuous (computable).

2. $\mathrm{IsFull}_{\mathbf{X}} : \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ defined via $\mathrm{IsFull}_{\mathbf{X}}(X) = 1$ and $\mathrm{IsFull}_{\mathbf{X}}(U) = 0$ otherwise is continuous (computable).

3. For every (computable) $A \in \mathcal{A}(\mathbf{X})$ the subspace $\mathbf{A}$ is (computably) compact.

4. The map $\mathrm{id} : \mathcal{A}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ is well-defined and continuous (computable).

5. $\subseteq: \mathcal{A}(\mathbf{X}) \times \mathcal{O}(\mathbf{X}) \to \mathbb{S}$ defined via $\subseteq (A, O) = 1$, iff $A \subseteq O$ is continuous (computable).

6. $\mathrm{IsCover} : \mathcal{C}(\mathbb{N}, \mathcal{O}(\mathbf{X})) \to \mathbb{S}$ defined via $\mathrm{IsCover}((U_n)_{n\in\mathbb{N}}) = 1$, iff $\bigcup_{n\in\mathbb{N}} U_n = X$ is continuous (computable).

7. FiniteSubcover $:\subseteq \mathcal{C}(\mathbb{N}, \mathcal{O}(\mathbf{X})) \rightrightarrows \mathbb{N}$ with dom(FiniteSubcover) $= \{(U_n)_{n\in\mathbb{N}} \mid \bigcup_{n\in\mathbb{N}} U_n = X\}$ and $N \in$ FiniteSubcover$((U_n)_{n\in\mathbb{N}})$ iff $\bigcup_{n=0}^{N} U_n = X$ is continuous (computable).

8. Enough $:\subseteq \mathcal{C}(\mathbb{N}, \mathcal{A}(\mathbf{X})) \rightrightarrows \mathbb{N}$ with $(A_i)_{i\in\mathbb{N}} \in$ dom(Enough) iff $\bigcap_{i\in\mathbb{N}} A_i = \emptyset$, and $N \in$ Enough$((A_i)_{i\in\mathbb{N}})$ iff $\bigcap_{i\leq N} A_i = \emptyset$ is continuous (computable).

9. For all $\mathbf{Y}$, the map $\pi_2 : \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{A}(\mathbf{Y})$ defined via $\pi_2(A) = \{y \in \mathbf{Y} \mid \exists x \in \mathbf{X}\ (x, y) \in A\}$ is continuous (computable).

10. For some non-empty $\mathbf{Y}$ (containing a computable point), the map $\pi_2 : \mathcal{A}(\mathbf{X} \times \mathbf{Y}) \to \mathcal{A}(\mathbf{Y})$ is continuous (computable).

Theorem 2. The following properties are equivalent for a represented space $\mathbf{X}$:

1. $\mathbf{X}$ is (computably) T_2, i.e. the map $x \mapsto \{x\} : \mathbf{X} \to \mathcal{A}(\mathbf{X})$ is well-defined and continuous (computable).
2. $\mathrm{id} : \mathcal{K}(\mathbf{X}) \to \mathcal{A}(\mathbf{X})$ is well-defined and continuous (computable).
3. $\cap : \mathcal{K}(\mathbf{X}) \times \mathcal{K}(\mathbf{X}) \to \mathcal{K}(\mathbf{X})$ is well-defined and computable.
4. $\kappa : \mathbf{X} \to \mathcal{A}(\mathbf{X})$ is well-defined and continuous (computable).
5. $\neq: \mathbf{X} \times \mathbf{X} \to \mathbb{S}$ defined via $\neq (x, x) = 0$ and $\neq (x, y) = 1$ otherwise is continuous (computable).
6. $\Delta_{\mathbf{X}} = \{(x, x) \mid x \in \mathbf{X}\} \in \mathcal{A}(\mathbf{X} \times \mathbf{X})$ (as a computable element)

Theorem 3. The following properties are equivalent for a represented space $\mathbf{X}$:

1. $\mathbf{X}$ is (computably) admissible (as in Schröder [6]).
2. $\kappa^{-1} : \mathbf{X}_\kappa \to \mathbf{X}$ is well-defined and continuous (computable).
3. $\mathbf{X}$ is T_0, and for any represented space $\mathbf{Y}$ the map $f \mapsto f :\subseteq \mathcal{C}(\mathcal{K}(\mathbf{Y}), \mathcal{K}(\mathbf{X})) \to \mathcal{C}(\mathbf{Y}, \mathbf{X})$ is continuous (computable).
4. $\mathbf{X}$ is T_0, and for any represented space $\mathbf{Y}$ the map $^{-1} :\subseteq \mathcal{C}(\mathcal{O}(\mathbf{X}), \mathcal{O}(\mathbf{Y})) \to \mathcal{C}(\mathbf{Y}, \mathbf{X})$ is continuous (computable).

References

[1] Vasco Brattka & Gero Presser (2003): *Computability on subsets of metric spaces*. *Theoretical Computer Science* 305(1-3), pp. 43 – 76.

[2] Vasco Brattka & Klaus Weihrauch (1999): *Computability on subsets of Euclidean space I: closed and compact subsets*. *Theoretical Computer Science* 219(1-2), pp. 65 – 93.

[3] Martin Escardó (2004): *Synthetic topology of datatypes and classical spaces. Electronic Notes in Theoretical Computer Science* 87.

[4] Arno Pauly (2012). *Compactness and Separation for Represented Spaces.* http://arxiv.org/abs/1204.3763.

[5] Matthias Schröder (2002): *Admissible Representations for Continuous Computations.* Ph.D. thesis, FernUniversität Hagen.

[6] Matthias Schröder (2002): *Extended admissibility. Theoretical Computer Science* 284(2), pp. 519–538.

[7] Klaus Weihrauch (2010): *Computable Separation in Topology, from T_0 to T_2. Journal of Universal Computer Science* 16(18), pp. 2733–2753.

[8] Klaus Weihrauch & Tanja Grubba (2009): *Elementary Computable Topology. Journal of Universal Computer Science* 15(6), pp. 1381–1422.

Acknowledgements

I would like to thank Vasco Brattka, Martin Escardó and Matthias Schröder for various discussions crucial for the formation of the present work.

Domain-represented spaces inside equilogical spaces

Matthias Schröder*
Universität der Bundeswehr, Munich, Germany

Abstract

The natural functor mapping domain-represented spaces into the category of equilogical spaces has the disadvantage of neither being full nor preserving function spaces. We define a useful subcategory of domain-represented spaces such that the aforementioned functor becomes full and preserves the cartesian closed structure. Moreover, we construct an embedding of the quasi-normal qcb-spaces into this subcategory which preserves binary products and function spaces.

1 Extended Abstract

There are several options to represent topological spaces via countably-based topological spaces appropriately. The Type Two Model of Effectivity (TTE) uses Baire space representations or Cantor space representations [9]. Domain Theory prefers to employ domains as the space of names, namely either countably-based Scott domains [1, 3] or, more generally, ω-continuous domains. Both approaches can be compared by considering equilogical spaces [1, 2, 8], which admit an arbitrary countably-based T_0-space as the space of representatives. The ensuing categories, namely the category $\mathsf{Rep}(\mathbb{N}^\mathbb{N})$ of Baire-represented spaces, the category DomRep of domain-represented spaces and the category $\omega\mathsf{Equ}$ of countably-based equilogical spaces, are all cartesian closed.

The fact that the Baire space and ω-Scott domains are allowed as representing spaces in equilogical spaces gives rise to two evident functors $E_{\mathtt{Rep}} : \mathsf{Rep}(\mathbb{N}^\mathbb{N}) \to \omega\mathsf{Equ}$ and $E_{\mathtt{DR}} : \mathsf{DomRep} \to \omega\mathsf{Equ}$. The second functor, for example, maps a domain-represented spaces X to the countably-based equilogical space that has the totality of X, i.e. the space of names, as its representing space. Both functors are not well-behaved: they fail to preserve function spaces; the second functor is not even full.

We motivate and introduce the class of *upwards-closed domain-represented spaces*. These consist of those domain-represented spaces such that the set of names of each element is upwards-closed in the representing domain. The corresponding subcategory $\mathsf{UpDomRep}$ of DomRep turns out to be cartesian closed. We prove that the restriction $E_{\mathtt{Up}}$ of $E_{\mathtt{DR}}$ to upwards-closed domain-represented spaces preserves function spaces and is full.

Finally we show that $\mathsf{UpDomRep}$ contains a copy of an important subcategory of qcb-spaces, namely the cartesian closed category QN of quasi-normal qcb-spaces [7]. The

*email: matthias.schroeder@unibw.de

respective embedding functor $E_{\mathtt{QN}}\colon \mathsf{QN} \to E_{\mathtt{DR}}$ preserves binary products and function spaces. Moreover it chimes with the functor $E_{\mathtt{DR}}$ in the sense that the diagram

$$\begin{array}{ccc} \mathsf{QCB}_0 & \xhookrightarrow{E_{\mathtt{QCB}}} & \omega\mathsf{Equ} \\ \uparrow \subseteq & & \uparrow E_{\mathtt{Up}} \\ \mathsf{QN} & \xhookrightarrow{E_{\mathtt{QN}}} & \mathsf{UpDomRep} \end{array}$$

commutes, where the cartesian closed embedding functor $E_{\mathtt{QCB}}$ is defined as in [4].

This investigation is motivated by the fact that the categories ωEqu and DomRep are both used to model some approaches to functional programming in Computable Analysis [2, 5, 6]. Our results show that it does not matter which of the categories ωEqu and DomRep is used to formalise those approaches.

References

[1] A. Bauer: "The Realizability Approach to Computable Analysis and Topology"; Ph.D. thesis, Carnegie Mellon University (2000).

[2] A. Bauer, M. Escardó, A. Simpson: "Comparing Functional Paradigms for Exact Real-number Computation"; Proc. ICALP '02, Lecture Notes in Computer Science 2380 (2002), 488–500.

[3] J. Blanck: "Domain Representations of Topological Spaces"; Theoretical Computer Science 247 (2000), 229-255.

[4] M. Escardó, J. Lawson, A. Simpson: "Comparing Cartesian closed Categories of Core Compactly Generated Spaces"; Topology and its Applications 143 (2004), 105–145.

[5] D. Normann: "Comparing Hierarchies of Total Functionals"; Logical Methods in Computer Science 1,2:4 (2005), 1–28.

[6] M. Schröder: "The sequential topology on $\mathbb{N}^{\mathbb{N}^{\mathbb{N}}}$ is not regular"; Mathematical Structures in Computer Science 19 (2009), 943–957.

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

[8] D. Scott, A. Bauer, L. Birkedal: "Equilogical Spaces"; Theoretical Computer Science 415 (2004), 35–59.

[9] K. Weihrauch: "Computable Analysis"; Springer, Berlin (2000).

Proofs, computations and analysis

Helmut Schwichtenberg
Munich, Germany

Abstract

Algorithms are viewed as one aspect of proofs in (constructive) analysis. The data for such algorithms are finite or infinite lists of signed digits -1, 0, 1 (i.e., reals as streams), or possibly non well-founded labelled (with lists of signed digits -1, 0, 1) ternary trees (representing uniformly continuous functions). A corresponding realizability interpretation of proofs is discussed. The main tools are (i) a distiction between computationally relevant and irrelevant logical connectives and (ii) simultaneous inductively/coinductively defined predicates.