Fifteenth International Conference on

COMPUTABILITY & COMPLEXITY IN ANALYSIS 2018

5-8 August 2018, Lake Kochel, Germany

Held in Honour of
Klaus Weihrauch's 75th birthday

Invited Speakers

Pieter Collins (Maastricht, Netherlands)
Daniel Graça (Faro, Portugal)
Rupert Hölzl (Munich, Germany)
Neil Lutz (Philadelphia, USA)
Ludovic Patey (Lyon, France)
Diogo Poças (Hamilton, Canada and Munich, Germany)
Matthias Schröder (Darmstadt, Germany)
Florian Steinberg (INRIA, France)

Programme Committee

Laurent Bienvenu (Montpellier, France)
Hannes Diener (Christchurch, New Zealand)
Denis Hirschfeldt (Chicago, USA)
Rupert Hölzl (Munich, Germany)
Zvonko Iljazović (Zagreb, Croatia)
Takayuki Kihara (Nagoya, Japan)
Elvira Mayordomo (Zaragoza, Spain)
Norbert Müller (Trier, Germany)
Ning Zhong (Cincinnati, USA)
Martin Ziegler, chair (Daejeon, Republic of Korea)

Organising Committee

Vasco Brattka, chair (Munich, Germany)
Peter Hertling (Munich, Germany)
Rupert Hölzl (Munich, Germany)
Philip Janicki (Munich, Germany)

Topics

Computable analysis
Complexity on real numbers
Constructive analysis
Domain theory and analysis
Effective descriptive set theory
Theory of representations
Computable numbers, subsets and functions
Randomness and computable measure theory
Models of computability on real numbers
Realizability theory and analysis
Reverse analysis
Real number algorithms
Implementation of exact real number arithmetic

Important Dates

Submission deadline: April 2, 2018
Notification of authors: May 7, 2018
Final version: June 4, 2018
Registration deadline: June 18, 2018

ASL Association for Symbolic Logic

DFG German Research Foundation

CCA NET

Photography by courtesy of Rupert Hölzl © 2018

http://cca-net.de/

Conference Programme

The conference takes place at

Georg-von-Vollmar Academy
Schloss Aspenstein
Am Aspensteinbichl 9-11
82431 Kochel am See
Bavaria, Germany

Phone: +49 8851 780
Fax: +49 8851 7823

Sunday, August the 5th

14:15 **Opening**

14:30 **Florian Steinberg (Invited Talk)**

A simple universal machine of type two and a formal proof of its correctness

15:30 **Coffee Break**

16:00 **Michal Konecný**

Verified exact real limit computation

16:30 **Franz Brauße, Norbert Müller, and Robert Rettinger**

Intensionality and multi-valued limits

17:00 **Eike Neumann**

On the computability of the Fréchet distance of surfaces in the bit-model of real computation

17:30 **Chansu Park**

Tensor calculus in exact real computation - From recursive analysis to abstract data types: a case study

18:00 **Dinner**

Monday, August the 6th

08:00 **Breakfast**

09:00 **Daniel Graça (Invited Talk)**

Computing the asymptotic behavior of continuous-time dynamical systems

10:00 **Coffee Break**

10:30 **Diogo Poças (Invited Talk)**

Analog computability with differential equations

11:30 **Christoph Spandl**

Computable series expansions in celestial mechanics

12:00 **Xiang Huang, Jack H. Lutz, and Andrei N. Migunov**

Algorithmic randomness in chemical reaction networks

12:30 **Lunch**

14:00 **Pieter Collins (Invited Talk)**

Rigorous numerical computing with Ariadne

15:00 **Florian Steinberg**

coqrep: A Coq library for representations

15:30 **Coffee Break**

16:00 **Svetlana Selivanova and Victor Selivanov**

Bit complexity of computing solutions of boundary-value problems for some systems of PDEs with guaranteed precision

16:30 **Ivan Koswara, Svetlana Selivanova, and Martin Ziegler**

Parallel complexity theory of linear ODEs and PDEs

17:00 **Prashant Batra and Vikram Sharma**

Complexity of a root clustering algorithm

17:30 **Bastian Dörig**

Bit-Complexity and rigorous implementation of the fast multipole method for Trummer's problem

18:00 **Dinner**

Tuesday, August the 7th

08:00 **Breakfast**

09:00 **Neil Lutz (Invited Talk)**

Effective dimensions of projected points

10:00 **Coffee Break**

10:30 **Zvonko Iljazovic**

Computability in a metric plane

11:00 **Zvonko Iljazovic and Bojan Pažek**

Computability of sets which can be approximated by 2-cells

11:30 **Lucija Validžic**

Computability of semicomputable polyhedra

12:00 **Dongseong Seon**

Computability of Haar averages

12:30 **Lunch**

14:00 **Ludovic Patey (Invited Talk)**

Never underestimate pigeons

15:00 **Philip Janicki**

Nearly computable numbers

15:30 **Coffee Break**

16:00 **Ruslan Kornev**

Computable metrics above the standard real metric

16:30 **Konrad Burnik and Zvonko Iljazovic**

Effective compactness and uniqueness of maximal computability structures

17:00 **Boat Trip to Conference Dinner**

Wednesday, August the 8th

08:00 **Breakfast**

09:00 **Matthias Schröder (Invited Talk)**

Co-Polish spaces in computable analysis

10:00 **Coffee Break**

10:30 **Matthew de Brecht**

A note on the descriptive complexity of the upper and double powerspaces

11:00 **Peter Hertling**

Trees describing topological Weihrauch degrees of multivalued functions

11:30 **Victor Selivanov**

The Wadge-like degrees of Borel functions

12:00 **Dimiter Skordev**

Moschovakis extension of multi-represented spaces

12:30 **Lunch**

14:00 **Rupert Hölzl (Invited Talk)**

Degrees of randomized computability

15:00 **Takayuki Kihara and Arno Pauly**

Separating Weihrauch degrees via Kleene's recursion theorem

15:30 **Coffee Break**

16:00 **Arno Pauly**

Weihrauch degrees of translations between representations of real numbers, revisited

16:30 **Closing**

A simple universal machine of type two and a formal proof of its correctness

Florian Steinberg

As a realistic theory of computing, computable analysis is based on a Turing machine model of computation. The most popular formulation of this machine model are Weihrauch's TTE machines [Wei00]. Another computationally equivalent formulation which is preferred for complexity theory of operators is that of oracle Turing machines whose oracle is an element of Baire space [KC12]. This talk takes the second perspective and considers a machine at type level two to have access to functional input by means of queries for its values on a specified input.

Mathematically such machines can be modeled by oracle Turing machines and are the standard for introducing computability on Baire space: For functional input $\varphi \in \mathbb{N}^{\mathbb{N}}$ and discrete input $n \in \mathbb{N}$ an oracle Turing machine M either produces a return value $M^{\varphi}(n)$ or diverges. A machine M is said to be defined on oracle φ if it converges for all inputs $n \in \mathbb{N}$ and is assigned the partial operator $F_M :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ whose domain are the functions the machine converges on and whose value $F_M(\varphi)$ on such a φ is given by $n \mapsto M^{\varphi}(n)$. An operator on Baire space is computable if it is a restriction of some F_M.

Due to the finitistic nature of the model of computation, any computable operator is continuous with respect to the natural topology on Baire space. Moreover, it is a well known fact that any continuous operator on Baire space is computable relative to an additional oracle. This basic fact is the basis for the function space construction used in computable analysis. Furthermore, it can be understood as stating the existence of a universal type-two machine. Using some standard pairing function $\langle \cdot, \cdot \rangle$ on Baire space this statement reads as follows: there is an oracle machine U such that for any continuous operator $F :\subseteq \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$ there exists a $\psi_F \in \mathbb{N}^{\mathbb{N}}$ such that $U^{\langle \psi_F, \cdot \rangle}$ computes F, i.e. for any φ from the domain of F and all $n \in \mathbb{N}$ it holds that

$$U^{\langle \psi_F, \varphi \rangle}(n) = F(\varphi)(n).$$

For a fixed machine U, a functional ψ_F as above is called an associate of the operator F. Sometimes it is called a Kleene associate, as the basic ideas behind the above go back to Kleene and Kreisel [Kle59, Kre59]. There are several ways to concretely construct universal machines. On one hand, one may rely on a universal Turing machine in type-one, i.e. use a relativized version of the universal machine from traditional computability theory. This allows to shift all computable parts of the execution of an operator away from the associate and towards the universal machine and is the construction most commonly used in computable analysis. On the other hand, one may keep most of the execution of an operator locked inside of the associate. This leads to a very simple form of the machine U.

The talk gives an explicit description of such a simple universal machine and discusses some of its advantages over more complicated ones as well as some of its drawbacks: The simplicity of the algorithm driving the machine allows a direct formulation in Coq and a fairly simple formal proof of the universality property. On the other hand, the simplicity of the universal machine leads to a loss in performance: The talk gives an argument, but not a proof, that one has to expect a quadratic overhead of executing a name of an operator in the universal machine over executing the operator itself. This universal machine is closely tied to ideas from higher-order complexity theory. The talk briefly discusses two such connections [BK02, Fér17].

Finally, the talk discusses an application: The universal machine was used to extend a basic library for computable analysis in Coq by a function space construction. The algorithm of the simple machine is directly executable inside of Coq. As a consequence, functions that have been constructed can be executed with minimal overhead. For the purpose of use for this library it was convenient to use a slightly generalized version of the universal machine. This illustrates one of the merits of proving the existence of a universal machine and similar statements formally, as the generalization came up naturally during this process. It turns out that the proof of universality does not only work for the natural numbers but can easily be lifted to more general input and output types. The assumptions needed about the types are that the input types are countable and the output types are inhabited.

References

[BK02] Samuel R. Buss and Bruce M. Kapron. Resource-bounded continuity and sequentiality for type-two functionals. *ACM Trans. Comput. Log.*, 3(3):402–417, 2002. Special issue on logic in computer science (Santa Barbara, CA, 2000).

[Fér17] Hugo Férée. Game semantics approach to higher-order complexity. *Journal of Computer and System Sciences*, 87:1 – 15, 2017.

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

[Kle59] Stephen Cole Kleene. Countable functionals. *Constructivity in Mathematics*, pages 81 – 100, 1959.

[Kre59] G. Kreisel. Interpretation of analysis by means of functionals of finite type. *Constructivity in Mathematics*, pages 101 – 128, 1959.

[Wei00] Klaus Weihrauch. *Computable analysis*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2000.

Verified exact real limit computation

Michal Konečný
Aston University, Birmingham, United Kingdom
m.konecny@aston.ac.uk

Consider a program for computing $\sqrt{x}$ for a real number $x > 1$ using Newton iteration:

```
1  sqrt_Newton x = lim (λε. sqrt_Newton_iter ε x x)
2
3  sqrt_Newton_iter ε x yₙ
4    | abs(yₙ - x/yₙ) <= ε   = yₙ  -- yₙ close to x/yₙ, ie yₙ close to √x
5    | otherwise             = let yₙ₊₁ = (yₙ + x/yₙ)/2 -- Newton step
6                              in sqrt_Newton_iter ε x yₙ₊₁ -- recursive call
```

where y_n and y_{n+1} are real numbers and ε is always a positive rational number. Our goal is to give programs such as `sqrt_Newton` an executable semantics and formally verify their semantics, eg that `sqrt_Newton` computes the exact $\sqrt{x}$ for any $1 < x$. Of particular interest is how to express and verify program termination in the presence of recursion and `lim`.

Function `sqrt_Newton` "computes" the exact $\sqrt{x}$ on RealRAM but this model cannot be faithfully implemented. With an approximate arithmetic such as floating-point arithmetic, `sqrt_Newton_iter` will typically not terminate for very small ε and verification is very difficult.

We will use exact real arithmetic [1, 2] to execute such programs exactly. In this example we stumble on the fact the comparison on line 4 is not computable due to a discontinuity at $|y_n - x/y_n| = \varepsilon$. A standard solution for this problem is to replace `abs(`y_n `- x/`y_n`) <=` ε with non-deterministic approximate comparison y_n `=`ε`=` `x/`y_n which returns true if the difference between the numbers is less than $\varepsilon/2$, returns false if the difference is over ε and returns either true of false if the difference is between $\varepsilon/2$ and ε. This comparison is deterministic on the level of operational semantics with real numbers encodings but since it is not extensional, it is non-deterministic at the level of denotational semantics with real numbers.

With this change, we can execute `sqrt_Newton` in exact real arithmetic. To formally verify its execution, we first encode a minimal programming language and its exact real operational semantics in a theorem prover. We base our formalisation on CIDR-F1, the first-order functional fragment of CIDR, a minimal imperative language for verifiable real number computation, currently under development with colleagues within CID, EU RISE project 731143.

We wish to express operational semantics as a function which we can use to extract verified executables from formalised programs. This means that the operational semantics is a potentially non-terminating function, which is not (well) supported in theorem provers such as Isabelle and Coq. To make the semantics a terminating function, we add a call stack size bound parameter:

$$\text{opsem: } (\text{FnName} \to \text{Term}) \to \text{Term} \to (\text{stackBound} : \mathbb{N}) \to \text{Input} \to \text{Val}^{\top}$$

and return $\top$ when a term evaluation makes too many nested calls, e.g. due to an infinite recursion. (In an imperative language we would also need to bound loops.) We specify that a program terminates by saying that for each valid input there is a stack size limit for which the operational semantics never returns $\top$.

The set Val includes real numbers represented e.g. by convergent "sequences" $\mathbb{Q}_+ \to \mathbb{Q}$. When computing `lim (`$\lambda\varepsilon$`. t` ε`)`, it is often the case that the closer is ε to zero, the more resources

`t` ε needs. A fixed bound on call stack size for a limit implies that the operational semantics will sometimes return "sequences" $\mathbb{Q}_+ \to \mathbb{Q}^\top$ that return $\top$ for very small ε. These are partial real numbers whose denotational semantics are compact real intervals. Our prototype Isabelle formalisation defines an abstract type of (constructive) partial real numbers, postulating their properties. For each partial real operation, we postulate that the operation is safe with regards to the corresponding real number operation and that the partial real operation has a (non-uniform) modulus of continuity, implying that it converges for exact inputs. For example:

$$x \in X \quad \wedge \quad y \in Y \ \wedge \ 0 \notin Y \qquad \Longrightarrow \quad x/y \in X/Y \qquad \text{(safety)}$$
$$\mathrm{wd}(X) \le \omega_{\div,L}(X,Y,\varepsilon) \quad \wedge \quad \mathrm{wd}(Y) \le \omega_{\div,R}(X,Y,\varepsilon) \quad \Longrightarrow \quad \mathrm{wd}(X/Y) \le \varepsilon \qquad \text{(modulus)}$$

We formalise also a non-deterministic denotational semantics of CIDR-F1:

$$\text{dsem: } (\text{FnName} \to \mathcal{P}(\text{Input} \times \text{Val}^\top)) \to \text{Term} \to (\text{stackBound} : \mathbb{N}) \to \mathcal{P}(\text{Input} \times \text{Val}^\top)$$

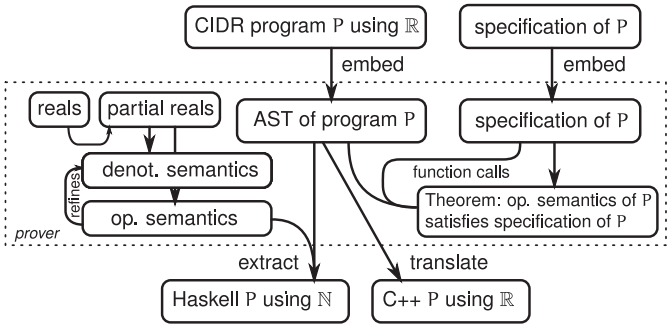
Assuming a certain input-output relation for all functions (typically formulated as pre- and post-conditions), if we can show that the derived input-output relation for the body of each function implies its own specification, we have proved that the assumed semantics is a safe approximation of the operational semantics. Thus we can specify the behaviour of our functions in terms of pre- and post-conditions and verify it using relatively simple inductive reasoning. The precondition contains a bound on the stack size.

As an example, we will review our formal proof of the following specification of `sqrt_Newton`:

Theorem 1. *For any real numbers* $1 < x$ *and* $0 < \varepsilon$*, the term* `sqrt_Newton x` *evaluated with a stack bound at least* $\max(1, 1 + \log_2(^{2(x-1)}/_{\varepsilon}))$ *returns a partial real number (interval)* R *that contains* $\sqrt{x}$ *and whose width is at most* ε*.*

The theorem gives an upper bound on an essential component on the complexity of the program and shows that the program computes $\sqrt{x}$ exactly if given unlimited resources.

The following sketch gives an overview of the components of our verification approach:



Our prototype has been implemented in Isabelle [3], currently without program extraction.

Our approach could be extended to the imperative part of CIDR, but stack bound alone would not suffice to get a total operational semantics; a bound on while-loop executions would be needed too. We plan to add an iRRAM-style iterative operational semantics to the formalisation. The iRRAM limit operator[1] has more complex behaviour than the one based on sequences.

References

[1] Norbert Th Müller. The iRRAM: Exact arithmetic in C++. In *Computability and Complexity in Analysis*, pages 222–252. Springer, 2001.

[2] Michal Konečný. AERN2. https://github.com/michalkonecny/aern2, 2018.

[3] Tobias Nipkow and Gerwin Klein. *Concrete Semantics, with Isabelle/HOL.* Springer, 2014.

Intensionality and Multi-valued Limits *

F. Brauße[1], N. Th. Müller[1], and R. Rettinger[2]

[1] Universität Trier, Germany
[2] Hochschule Dortmund, Germany

In the spirit of iRRAM [Mü01], i.e. abstracting programs from the concrete underlying computations on represented spaces, we consider a research programming language CCL based on Clerical [BPS17] which allows to concisely describe algorithms in ERC by providing an opaque data type for exact real numbers. We developed a compiler [Bra18] translating CCL to iRRAM/C++ which should be easily extendable to other Exact Real Computation backends, e.g. Aern2/Haskell [Kon]. A feature of such a small language usually is the possibility to formally specify the complete semantics. In ERC however, due to the necessity of multi-valuedness of the computations, even that is not straight-forward. Major work in this direction has been done by Bauer, Park and Simpson, who also implemented part of the semantics in Coq [Coq16]. We extend on this research by specifiying necessary and sufficient conditions for computing single-valued limits of multi-valued sequences in Coq and also by formalizing conditions for multi-valued sequences of Reals converging to multiple values in a multi-valued sense, i.e. a multi-valued limit. Examples for this requirement in the language are the computation of

- one of the complex square roots $\sqrt{\cdot} : \mathbb{C} \rightrightarrows \mathbb{C}$,
- one of the roots in $(0,1)$ of $f : \mathbb{R} \to \mathbb{R}$ continuous with $f(0) \cdot f(1) < 0$,
- given $(x_n)_n \in \mathbb{R}^\omega$, a real number x such that $\forall n : x \neq x_n$ [Wei00].

In order to compute such multi-valued convergent sequences, for each component the algorithm requires additional information on which of the multiple values to refine. We formally specifiy the semantics of this multi-valued limit operation by requiring the prover to provide an instance of a partial order on the subsidiary discrete hint provided to the computation of the fast sequence, thus establishing multi-valued convergence.

In addition to the specification in Coq, we plan to extend the compiler to actually produce proof obligations for given programs in the language, formulated in Coq. The correctness of such programs with respect to the Classical Reals[3] then establishes the correctness of the algorithm.

An example of a program verified in this manner is the computation of $\sqrt{\cdot} \in \mathbb{R}$, see fig. 1. Furthermore, we show that all computable functions can indeed be expressed in CCL. An interesting aspect with regard to this is the non-obvious computation of a modulus of convergence of a function given by a CCL expression.

* The research leading to these results has received funding from the DFG grants WERA MU 1801/5-1 and CAVER BE 1267/14-1. This project has received funding the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143.

[3] http://coq.inria.fr/distrib/8.6.1/stdlib/Coq.Reals.Reals.html

```
external abs(Real) -> Real
external bounded(Real, Int) -> Bool

function sqrt(Real x):
    lim n =>
        var y := 1.0 in (
        var two := 2.0 in
        var z := x/y in (
            while !bounded(abs(y-z), n)
            do (
                y := (y+z)/two;
                z := x/y
            )
        ); y)

do
    sqrt(2.0)
```

Fig. 1. CCL program computing the square root of 2 using Heron's method.

References

BPS17. Andrej Bauer, Sewon Park, and Alex Simpson. Clerical. https://github.com/andrejbauer/clerical, 2017.

Bra18. Franz Brauße. CCL – a compiler for the cclerical language. https://github.com/fbrausse/cclerical, March 2018.

Coq16. The Coq proof assistant, version 8.6.1, November 2016. URL: https://coq.inria.fr.

Kon. Michal Konečný. AERN – approximating exact real numbers. http://michalkonecny.github.io/aern/_site/index.html. [Online; accessed 02-April-2018].

Mü01. Norbert Th. Müller. The iRRAM: Exact Arithmetic in C++. *Lecture notes in computer science*, 2991:222–252, 2001.

Wei00. Klaus Weihrauch. *Computable analysis: an introduction*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.

On the computability of the Fréchet distance of surfaces in the bit-model of real computation *

Eike Neumann †
Aston University, Birmingham, UK

In 1906, Maurice Fréchet introduced a natural pseudometric for parametric curves [3], which he generalised in 1924 to parametric surfaces [4]. If $f\colon [0,1]^2 \to X$ and $g\colon [0,1]^2 \to X$ are continuous function in some metric space, then their Fréchet distance is given by:

$$\inf_{\varphi,\psi\in \mathrm{Aut}'([0,1]^2)} \max_{x\in[0,1]^2} d\left(f\left(\varphi(x)\right), g\left(\psi(x)\right)\right),$$

where $\mathrm{Aut}'([0,1]^2)$ denotes the set of orientation-preserving homeomorphisms of $[0,1]^2$.

The problem of computing the Fréchet distance of curves and surfaces has received considerable attention in Computational Geometry (see [1] and references therein). The problem whether the Fréchet distance of surfaces is computable has been an open problem for a long time (see [5]). Only some partial results were established by Alt and Buchin [2]. Recently Nayyeri and Xu [6, 7] have shown that the Fréchet distance of piecewise-linear surfaces of genus 0 that are "locally isometrically" immersed in $\mathbb{R}^3$ is computable in the real RAM model. They have even shown that the problem of deciding whether the Fréchet distance between two such surfaces is smaller than a given $\delta > 0$ is computable in PSPACE.

We show an analogous result in the bit-model with applies to arbitrary continuously parametrised surfaces in a computable metric space:

Theorem 1. *Let X be a computable metric space. There exists an algorithm which takes as input two continuous functions $f\colon [0,1]^2 \to X$, $g\colon [0,1]^2 \to X$ with values in X, and returns as output their Fréchet distance.*

The techniques used in the proof of Theorem 1 are quite different from those used by Nayyeri and Xu. The proof mainly relies on the following result which is of independent interest:

Theorem 2. *The closure $\overline{\mathrm{Aut}'([0,1]^2)}$ of the set of orientation-preserving automorphisms of the unit square $[0,1]^2$ is computable as a closed subset of the space $C([0,1]^2,[0,1]^2)$ of continuous self-maps of $[0,1]^2$.*

Theorem 2 is proved using the Brouwer mapping degree. The rest of the proof can be outlined as follows: For a map $\varphi\colon X \to Y$, let $\Gamma_\varphi \subseteq X \times Y$ denote its graph. Define a new distance function d_Γ on $C([0,1]^2,[0,1]^2)$ by

$$d_\Gamma(\varphi,\psi) = d_H(\Gamma_\varphi,\Gamma_\psi)$$

where d_H is the Hausdorff distance on the metric space $[0,1]^2 \times [0,1]^2$ with the product metric

$$d\left((x_0,y_0),(x_1,y_1)\right) = \max\left\{d(x_0,x_1), d(y_0,y_1)\right\}.$$

*Full paper [8] available at `https://arxiv.org/abs/1711.02161`.

†This work was supported by EU Horizon 2020 MSCA RISE project 731143. The majority of this work was undertaken while the author was visiting KAIST, Daejeon, Republic of Korea. The author would like to thank Martin Ziegler for bringing this problem to his attention.

For a function $\varphi\colon [0,1]^2 \to [0,1]^2$, let

$$F_{f,g}(\varphi) = \max_{x\in[0,1]^2} d\left(f(x), g(\varphi(x))\right)$$

denote its *Fréchet length*. It is easy to see that automorphisms with small graph distance have similar Fréchet length:

Lemma 1. *Let μ_f and μ_g be moduli of continuity of f and g respectively. Then we have the implication*

$$d_\Gamma(\varphi,\psi) \le 2^{-\mu_f(n+1)-\mu_g(n+1)} \to d\left(F_{f,g}(\varphi), F_{f,g}(\psi)\right) \le 2^{-n}.$$

By suitably "removing unnecessary oscillations" one can uniformly bound the Lipschitz constant of all automorphisms up to small perturbations in the graph-distance:

Lemma 2. *Let $\varphi\colon [0,1]^2 \to [0,1]^2$ be an automorphism. For all $n \in \mathbb{N}$ there exists an automorphism $\widetilde{\varphi}\colon [0,1]^2 \to [0,1]^2$ with $d_\Gamma(\varphi,\widetilde{\varphi}) < 2^{-n}$ such that $\widetilde{\varphi}$ has Lipschitz constant*

$$4^n \times 4^{4^n} \times \left(3 \times 4^n + 3\right) + 1.$$

Thus the Fréchet distance of two functions f and g is given up to error 2^{-n} by the minimum of $F_{f,g}(\varphi)$ where φ ranges over the compact set of all maps $\varphi \in \overline{\mathrm{Aut}'([0,1]^2)}$ with Lipschitz constant at most $4^n \times 4^{4^n} \times (3 \times 4^n + 3) + 1$. By Theorem 2 this set is uniformly computably compact in n. It follows that the minimum is uniformly computable in n. Hence the Fréchet distance is computable. Unfortunately, as opposed to the results by Nayyeri and Xu for the real RAM model we do not obtain any non-trivial complexity bounds.

References

[1] H. Alt. The Computational Geometry of Comparing Shapes. In S. Albers, H. Alt, and S. Näher, editors, *Efficient Algorithms*, volume 5760 of *Lecture Notes in Computer Science*, pages 235–248. Springer, Berlin, Heidelberg, 2009.

[2] H. Alt and M. Buchin. Can we compute the similarity between surfaces? *Discrete & Computational Geometry*, 43(1):78–99, 2010.

[3] M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti Circ. Mat. Palermo*, 22:1 – 74, 1906.

[4] M. Fréchet. Sur la distance de deux surfaces. *Ann. Soc. Polonaise Math.*, 3:4 – 19, 1924.

[5] M. Godau. *On the complexity of measuring the similarity between geometric objects in higher dimensions*. PhD thesis, Freie Universität Berlin, 1998.

[6] A. Nayyeri and H. Xu. On computing the Fréchet distance between surfaces. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, pages 55:1–55:15, 2016.

[7] A. Nayyeri and H. Xu. On the decidability of the Fréchet distance between surfaces. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1109–1120, 2018.

[8] E. Neumann. On the computability of the Fréchet distance of surfaces in the bit-model of real computation. `https://arxiv.org/abs/arXiv:1711.02161`, 2017.

Tensor Calculus in Exact Real Computation*

From Recursive Analysis to Abstract Data Types: A Case Study

Chansu Park, KAIST

Tensor Calculus is a generalization of classical Real Analysis which is concerned with continuous and differentiable functions of scalar, vector, and tensors. It is used in various fields of engineering and physics, including general relativity theory, quantum mechanics, statistical thermodynamics, classical mechanics, electrodynamics, solid mechanics, and fluid dynamics [3]. Most of them require precise computation, and Exact Real Computation (ERC) is a good solution. However, ERC has difficulties on tensor calculus; e.g., a computable real function may have an uncomputable derivative [2], and the boundary of the domain for surface integration might not be computable.

Simplicial Complex is a nice glueing of simplices. It can be finitely-parametrized, which is a special case of manifolds. While manifolds, yet, cannot be easily defined on ERC, simplicial complex is a feasible starting point for tensor calculus in ERC; interior points of the given simplicial complex can be parametrized uniquely.

To build tensor calculus on ERC, discrete enrichments should be exposed for uncomputable operations to establish computable methods, realized by explicit algorithms in practical object-oriented programming. In this paper, we use facet for simplicial complexes and Lipschitz-continuity for tensor fields as discrete enrichments. By adapting notions from Differential Geometry, we define 'k-times piecewise L-Lipschitz-continuously differentiable (p,q)-tensor fields' and canonical tensor operations including differentiation and integration. We introduce an Abstract Data Type for such objects that makes the internal representation and parametrization secure while rendering the above operations computable.

Definition 1. *a) A (closed)* simplex *of dimension d is the convex hull of $1+d$ points in $\mathbb{R}^D$:*

$$\mathrm{chull}(\vec{v}_0,\vec{v}_1,\ldots,\vec{v}_d) \;=\; \Big\{ \sum_{j=1}^{d} \lambda_j \vec{v}_j + \big(1-\sum_{j=1}^{d}\lambda_j\big)\vec{v}_0 \;\big|\; \lambda_j \geq 0, \sum_{j=1}^{d}\lambda_j \leq 1 \Big\}$$

such that the d vectors $\vec{v}_1-\vec{v}_0,\ldots,\vec{v}_d-\vec{v}_0 \in \mathbb{R}^D$ are linearly independent. $V(\vec{\vec{v}}) \subseteq \mathbb{R}^D$ denotes the d-dimensional linear subspace they span, abbreviating $\vec{\vec{v}} = (\vec{v}_0,\vec{v}_1,\ldots,\vec{v}_d) \in \mathbb{R}^{D\times(d+1)}$.
An inverse of unique affine transformation *$\varphi_{\vec{\vec{v}}}^{-1}$ maps $\vec{v} = \sum_{j=1}^{d}\lambda_j\vec{v}_j + \big(1-\sum_{j=1}^{d}\lambda_j\big)\vec{v}_0$ to $(\lambda_1,\ldots,\lambda_d) \in \mathrm{chull}(\vec{\vec{e}}) := \{(x_1,\ldots,x_d) : \forall i\ x_i \geq 0 \wedge \sum x_i \leq 1\}$.*
A simplicial complex *$\mathcal{S}$ is a finite set of simplices such that (i) every face of every $S \in \mathcal{S}$ again belongs to $\mathcal{S}$, and (ii) the intersection $S \cap S'$ of any two $S, S' \in \mathcal{S}$ is a face of both S and S'. A simplicial complex $\mathcal{S}$ of dimension d is* homogeneous *if (i) it contains a simplex of dimension d where (ii) every simplex in S of dimension less than d is a face of some simplex of dimension d in S.*

b) The Abstract Data Type Simplicial Complex *stores for any fixed homogeneous $\mathcal{S}$ (i) the ambient space dimension $D \in \mathbb{Z}$; (ii) a tuple $(\vec{v}_1,\ldots,\vec{v}_N)$ of pairwise distinct vectors in $\mathbb{R}^D$: the vertices of $\mathcal{S}$; and (iii) all those subsets $I \subseteq \{1,\ldots,N\}$ such that* chull $\big(\{\vec{v}_i : i \in I\}\big)$ *constitutes a face of $\mathcal{S}$.*

c) A (p,q)-tensor field T on simplicial complex $\mathcal{S}$ is k-times piecewise *L-Lipschitz continuously differentiable ($T \in \mathcal{T}'^{k,L}_{p,q}(\mathcal{S})$) if, for every* chull$(\vec{\vec{v}}) \in \mathcal{S}_d$ *and $\vec{k}$ with $|\vec{k}| = k$, the restriction $T\big|_{\mathrm{chull}(\vec{\vec{v}})}$ is k-times continuously differentiable and $\vec{\partial}^{\vec{k}}\big(T \circ \varphi_{\vec{\vec{v}}}\big)$ is L-Lipschitz continuous.*

d) The Abstract Data Type piecewise Lipschitz-continuously differentiable tensor field *stores for any $T \in \mathcal{T}'^{k,L}_{p,q}(\mathcal{S})$, apart from $\mathcal{S}$ itself according to (b): non-negative integers k,p,q,L and for each d-dimensional* chull$(\vec{\vec{v}}) \in \mathcal{S}$*, a (pointer to a) subroutine providing blackbox evaluation of $T_{\vec{i}\vec{j}}\big|_{\mathrm{chull}(\vec{\vec{v}})} \circ \varphi_{\vec{\vec{v}}} :$* chull$(\vec{\vec{e}}) \subseteq \mathbb{R}^d \to \mathbb{R}$.

*Shrinked version of a KAIST Technical Report (https://cs.kaist.ac.kr/research/techReport, CS-TR-2018-416). We thank Junhee Cho and Sewon Park for seminal discussions.

Theorem 2. *Abbreviate* $\mathcal{T}'^{k,*}_{p,q}(\mathcal{S}) := \bigcup_{L>0} \mathcal{T}'^{k,L}_{p,q}(\mathcal{S})$. *The following operations are computable with respect to enrichments from Definition 1b) and 1d):*

a) Pointwise tensor linear combination, tensor product, and tensor contraction.

b) Initializing a scalar field on $\mathcal{S}$ *from a given real function:* $\mathcal{C}^{k,L}\big(\overline{\supseteq \bigcup \mathcal{S}}\big) \to \mathcal{T}'^{k,L}_{0,0}(\mathcal{S})$.

c) Embedding $\mathcal{T}'^{k+1,*}_{0,0}(\mathcal{S}) \mapsto \mathcal{T}'^{k,*}_{0,0}(\mathcal{S})$.

d) Boundary complex $\mathcal{S} \mapsto \partial\mathcal{S}$.

e) Boundary restriction $\mathcal{T}'^{0,L}_{0,0}(\mathcal{S}) \ni T \mapsto T\big|_{\partial\mathcal{S}} \in \mathcal{T}'^{0,L}_{0,0}(\partial\mathcal{S})$

f) scalar field evaluation $\mathcal{T}'^{0,L}_{0,0}(\mathcal{S}) \times \bigcup\mathcal{S} \ni (T,\vec{x}) \mapsto T(\vec{x}) \in \mathbb{R}$

g) vector flux $\mathcal{T}'^{0,L}_{1,0}(\mathcal{S}) \ni T \mapsto T\cdot\vec{\nu} \in \mathcal{T}'^{0,L}_{0,0}(\partial\mathcal{S})$

h) scalar volume integration $\mathcal{T}'^{0,L}_{0,0}(\mathcal{S}) \ni T \mapsto \int_{\mathcal{S}} T(\vec{x})\,d\vec{x} \in \mathbb{R}$

i) Line integration of scalar fields with respect to continuously differentiable curves

$$\mathcal{T}'^{0,L}_{0,0}(\mathcal{S}) \times \mathcal{C}^{1,K}(\mathcal{S}) \ni (T,\vec{\gamma}) \mapsto \int_0^1 T\big(\vec{\gamma}(t)\big)\cdot\|\vec{\gamma}'(t)\|\,dt \in \mathbb{R}.$$

j) Flow integration of vector fields with respect to continuously differentiable curves

$$\mathcal{T}'^{0,L}_{1,0}(\mathcal{S}) \times \mathcal{C}^{1,K}(\mathcal{S}) \ni (R,\vec{\gamma}) \mapsto \int_0^1 R\big(\vec{\gamma}(t)\big)\cdot\vec{\gamma}'(t)\big)\,dt \in \mathbb{R}.$$

k) Gradient of tensor fields $\mathcal{T}'^{k+1,*}_{p,q}(\mathcal{S}) \ni T \mapsto \operatorname{grad} T \in \mathcal{T}'^{k,*}_{p,q+1}(\mathcal{S})$

l) Divergence of tensor fields $\mathcal{T}'^{k+1,*}_{p+1,q}(\mathcal{S}) \ni T \mapsto \operatorname{div} T \in \mathcal{T}'^{k,*}_{p,q}(\mathcal{S})$

Items of integration h), i), and j) are followed from [4, Theorem 6.4.1]; and items of differentiation k), and l) are based on the following particular case of [4, Corollary 6.4.8+Lemma 7.3.14]; cmp. [1, Lemma 3.4b]. These theorems heavily build on the following quantitative refinements of *finite differences*:

Lemma 3. *a) Let* $f : (a;b) \to \mathbb{R}$ *be differentiable with* L'*-Lipschitz continuous derivative. Then* $\big|f'(x) - \frac{f(x+\delta)-f(x)}{\delta}\big| \le \delta L'$ *for all* $x \in (a;b)$ *and all* $0 < \delta \le b-x$. *Moreover* f *is* L*-Lipschitz for* $L := |f(b)-f(a)|/|b-a| + 2L'\cdot|b-a|$.

b) Now let $f : (a;b) \to \mathbb{R}$ *be twice differentiable with* L''*-Lipschitz continuous second derivative* f''. *Then* $\big|f''(x) - \frac{f(x+2\delta)-2f(x+\delta)+f(x)}{\delta^2}\big| \le 2\delta L''$ *for all* $x \in (a;b)$ *and all* $0 < \delta \le (b-x)/2$. *Moreover* f' *is* L'*-Lipschitz for* $L' := 4\big|f(b) - 2f\big(\frac{a+b}{2}\big) + f(a)\big|/|b-a|^2 + 2|b-a|\cdot L''$.

c) Now consider second derivatives in the bivariate case: For $f : \operatorname{chull}^\circ\big((0,0),(0,1),(1,0)\big) \to \mathbb{R}$ *suppose all* $\partial_1^2 f, \partial_1\partial_2 f = \partial_2\partial_1 f, \partial_2^2 f$ *are* L''*-Lipschitz. Then*

$$\left|\partial_1\partial_2 f(x,y) - \left(\frac{f(x+2\epsilon^2,y+\epsilon) - f(x,y+\epsilon) - f(x+\epsilon^2,y) + f(x,y)}{4\epsilon^3} - \frac{f(x+2\epsilon^2,y+\epsilon) - 2f(x+\epsilon^2,y+\epsilon) + f(x,y+\epsilon)}{\epsilon^3}\right)\right| \le 4\big(\epsilon^2+1\big)\epsilon L''$$

holds for all $(x,y) \in \operatorname{chull}^\circ((0,0),(0,1),(1,0))$ *and every* $\epsilon > 0$, $[x;x+4\epsilon^2]\times[y;y+4\epsilon] \subseteq \operatorname{chull}^\circ((0,0),(0,1),(1,0))$; $\partial_1^2 f$ *and* $\partial_2^2 f$ *are analogous to Lemma 3b).*

References

[1] N. T. Müller. Subpolynomial complexity classes of real functions and real numbers. In L. Kott, editor, *Proceedings of the 13th International Colloquium on Automata, Languages, and Programming*, volume 226 of *Lecture Notes in Computer Science*, pages 284–293, Berlin, 1986. Springer.

[2] J. Myhill. A recursive function defined on a compact interval and having a continuous derivative that is not recursive. *Michigan Math. J.*, 18:97–98, 1971.

[3] H. Nguyen-Schäfer and J.-P. Schmidt. *Tensor Analysis and Elementary Differential Geometry for Physicists and Engineers*, volume 21 of *Mathematical Engineering*. Springer, Berlin, Heidelberg, 2014.

[4] K. Weihrauch. *Computable Analysis*. Springer, Berlin, 2000.

Computing the asymptotic behavior of continuous-time dynamical systems

Daniel Graça
Universidade do Algarve & Instituto de Telecomunicações
Portugal

One of the central aims of dynamical systems theory is to determine the asymptotic behavior of dynamical systems. In general this is a very hard task and several complementary approaches were developed to better tackle this problem. For example, one can use a qualitative approach where the "limit set" of a dynamical system is characterized by some of its properties (see e.g. [7]), or a probabilistic perspective (see e.g. [1]). One can take this problem one step further and try to understand up to which point this asymptotic behavior can be computed.

In this talk we will survey several results related to the computability of the asymptotic behavior of continuous-time dynamical systems defined with ordinary differential equations. Namely, we will consider the computability of the limit set of some classes of dynamical systems defined by ordinary differential equations in the plane and in the three dimensional space, since several well-known results (Peixoto's theorem, etc.) allow us to better understand the structure of the limit set in such cases. We will also look to some classical results from dynamical systems theory using a computability perspective. Namely the Hartman-Grobman theorem tells us that, near an hyperbolic equilibrium point, the flow behaves locally like its linearization. We will see that a computable counterpart of this result exists [6]. Another classical result is the stable manifold theorem, which states that near an hyperbolic equilibrium point the flow can be split into stable and unstable manifolds. We will also see that a computable counterpart of this result exists [5]. We will also present some results about the non-computability of the basin of attraction (i.e. the set of points which converge to some given attractor) for continuous-time analytic dynamical systems [4]. This talks surveys work done with Ning Zhong, Jorge Buescu, and Scott Dumas.

References

[1] L. Barreira. *Ergodic Theory, Hyperbolic Dynamics and Dimension Theory.* Springer, 2012.

[2] D. S. Graça, C. Rojas, and N. Zhong. Computing geometric Lorenz attractors with arbitrary precision. *Transactions of the American Mathematical Society*, 370:2955–2970, 2018.

[3] D. S. Graça and N. Zhong. Computability in planar dynamical systems. *Natural Computing*, 10:1295–1312, 2011.

[4] D. S. Graça and N. Zhong. An analytic system with a computable hyperbolic sink whose basin of attraction is non-computable. *Theory of Computing Systems*, 57:478–520, 2015.

[5] D. S. Graça, N. Zhong, and J. Buescu. Computability, noncomputability, and hyperbolic systems. *Applied Mathematics and Computation*, 219(6):3039–3054, 2012.

[6] D. S. Graça, N. Zhong, and H. S. Dumas. The connection between computability of a nonlinear problem and its linearization: the Hartman-Grobman theorem revisited. *Theoretical Computer Science*, 457(26):101–110, 2012.

[7] L. Perko. *Differential Equations and Dynamical Systems*. Springer, 3rd edition, 2001.

Analog Computability with Differential Equations

Diogo Poças[1], joint work with Jeffery Zucker[2]

[1] *Chair of Operations Research, Technical University of Munich, Germany, diogo.pocas@tum.de*
[2] *Dept. of Computing and Software, McMaster University, Hamilton, Canada*

Most of the physical processes arising in nature are modeled by differential equations, either ordinary (example: the spring/mass/damper system) or partial (example: heat diffusion). From the point of view of analog computability, the existence of an effective way to obtain solutions (either exact or approximate) of these systems is essential.

A pioneering model of analog computation is the General Purpose Analog Computer (GPAC), introduced by Shannon as a model of the Differential Analyzer and improved by Pour-El, Lipshitz and Rubel, Costa and Graça and others. The GPAC is capable of manipulating real-valued data streams. Its power is known to be characterized by the class of differentially algebraic functions, which includes the solutions of initial value problems for ordinary differential equations.

We address two limitations of this model. The first is its fundamental inability to reason about functions of more than one independent variable (the 'time' variable). In particular, the Shannon GPAC cannot be used to specify solutions of partial differential equations. The second concerns the notion of approximability, a desirable property in computation over continuous spaces that is however absent in the GPAC. In particular, the Shannon GPAC cannot be used to generate non-differentially algebraic functions such as the gamma function, which can be approximately computed in other models of computation.

To overcome these limitations, we extend the class of data types by taking channels carrying information on a general complete metric space X; for example the class of continuous functions of one real variable. We consider the original modules in Shannon's construction (constants, adders, multipliers, integrators) and add two new modules: a *differential* module which computes spatial derivatives, $u(t) \mapsto \partial_x u(t)$; and a *continuous limit* module which computes limits, $u(t) \mapsto \lim_{t\to\infty} u(t)$.

We then build networks using X-stream channels and the abovementioned modules. This leads us to a framework in which the specifications of such analog systems are given by fixed points of certain operators on continuous data streams, as considered by Tucker and Zucker. We study the properties of these analog systems and their associated operators. We present a characterization which generalizes Shannon's results, and show that some non-differentially algebraic functions such as the gamma function are generable by our model.

Computable Series Expansions in Celestial Mechanics

Christoph Spandl

Konzept Informationssysteme GmbH, D-88709 Meersburg, Germany
and
AIRBUS BSG Dornier Astronomie, D-88090 Immenstaad , Germany
spandl@online.de

Abstract

Celestial mechanics deals with solving the equations of motion for astronomical objects to obtain ephemeris data. The subject has a centuries-old tradition in mathematics. Typically, there are two preferred methods in celestial mechanics for solving the equations of motion: an analytical one based on series expansions (method of general perturbations) and a numerical one based on algorithms directly integrating the equations of motion (method of special perturbations). The second method applies numerical analysis, which commonly lacks a rigorous treatment. However, even the first method also commonly lacks a rigorous treatment. It is a widely used practice to neglect terms in the series expansion which are assumed to give no essential contribution to the solution without rigorous analysis. Even Gauss obeyed this commonly accepted practice and terminated his series expansions without error analysis when performing his calculations of perturbation for the planetoid Pallas in 1811. The first who seriously addressed the problem of convergence was Weierstrass in the second half of the nineteenth century. However, he failed and finally it was Poincaré at the end of then nineteenth century who showed the reasons why in his famous award essay on the three body problem [1]. Since the advent of electronic computers, the interest switched from general perturbations to special perturbations. Only a few authors address to the problem of error bounds and rigorous treatment in numerical simulations of celestial bodies (see e.g. [2]). Subject of the present work is in presenting methods for the exact solution in the field of celestial mechanics, as computable analysis and exact real arithmetic demands.

The subsequent treatment presents a rigorous method in the sense of computable analysis which can be seen simultaneously as analytic and numerical. It is an approach for solving systems of ordinary differential equations with rigorous error estimation. The method is based on Lie series expansion, which is formulated in detail by Wolfgang Gröbner [3]. Lie series expansion can be seen as a generalization of power series expansion. Elements of the application of the latter method for solving ODEs in the context of exact real arithmetic are already presented by Norbert Müller and Margarita Korovina [5]. For a general and complete treatment of the method, consider the system of ODEs

$$\dot{x} = f(t, x) \tag{1}$$

with $t \in \mathbb{R}$, $x \in \mathbb{R}^n$ and the initial condition $x(t_0) = x_0$. For some analytic function $F : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$, the k-th total derivation of $F(t, x(t))$, where $x(t)$ is a solution of the systems of ODEs, can be written as $\frac{d^k}{dt^k} F(t, x(t)) = (D^k)(t, x(t))$ with the differential operator

$$D = \partial_t + \sum_{i=1}^{n} f_i(t, x) \cdot \partial_{x_i}. \tag{2}$$

Hence the series expansion of $F(t, x(t))$, in the neighborhood of t_0, can be written as $F(t, x(t)) = \sum_{k=0}^{\infty} \frac{1}{k!} (D^k F)(t_0, x_0)(t - t_0)^k$. Introducing for some function $G : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$, analytic in the neighborhood of (t_0, x_0), the notation $G^k(t, x) := \frac{1}{k!}(D^k G)(t, x)$, then the solution of the systems of ODEs can be written as

$$x_i(t) = \sum_{k=0}^{\infty} Id_i^k(t_0, x_0) \cdot (t - t_0)^k \tag{3}$$

for $i = 1, \ldots, n$. Assume that each component of the vector field $f_i(t, x)$ is composed of elementary computable functions - e.g. addition, multiplication, sin, ... - then the functions $f_i^k(t, x)$ can be determined recursively out of $Id_i^l(t, x)$ for $i = 1, \ldots, n$, $l = 0, \ldots, k$. Using the relation $DId_i = f_i$ for $i = 1, \ldots, n$, the recursion

$$Id_i^{k+1}(t, x) = \frac{1}{k+1} \cdot f_i^k(t, x) \tag{4}$$

follows. This allows the computation of (3) up to some predefined order [6]. The method of Lie series supplies criteria for convergence. Furthermore, the Picard-Lindelöf approach for proving existence of a solution can be evaluated in addition for finding an enclosure for the solution as intermediate result. These techniques can be formulated in terms of computable analysis and leads to a computable and therefore rigorous procedure. The method can be used either to numerically integrate the equations of motion exactly or to derive an analytical expression for the flow in a neighborhood of (t_0, x_0) and additionally a rigorous error bound.

The method described so far deals with solving the exact equations (1). In the second part of the presentation, consider a system of DGLs $\dot{x} = \hat{f}(t, x)$ approximating the original equations (1). Let $\hat{x}(t)$ be the solution of the approximation with unchanged initial condition: $\hat{x}(t_0) = x_0$. Then a modified differential operator $\hat{D}$ replaces (2) which also leads to a series expansion of the form (3). According to Knapp [6], the residual of the approximation can be expressed by

$$\begin{aligned} x(t) &= \hat{x}(t) + \int_{t_0}^{t} (D - \hat{D})(\xi, \hat{x}(\xi))\, d\xi + ((DId)(\xi_1, x(\xi_1)) - (DId)(\xi_1, \hat{x}(\xi_1))) \cdot (t - t_0) \\ &= \hat{x}(t) + \Big(f(\xi_2, \hat{x}(\xi_2)) - \hat{f}(\xi_2, \hat{x}(\xi_2)) + f(\xi_1, x(\xi_1)) - f(\xi_1, \hat{x}(\xi_1)) \Big) \cdot (t - t_0) \end{aligned}$$

for some $\xi_1, \xi_2 \in [t_0, t]$. This formula allows a computable version of the above integration method based on a sequence of flow approximations. The approach is used in combination with the technique of normal forms, already introduced by Poincaré for solving problems in celestial mechanics [1]. Normal forms basically deal with series expansions of vector fields and transformations to obtain simple forms. Applied to f, the series expansion leading to a normal form is truncated after a finite step of transformations leading to a polynomial vector field $\hat{f}$ of "simplest" form. The order of the polynomial is successively increased. The complexity of the integration scheme of the first method and the second method with normal forms are compared. Some examples are given.

References

1. Henri Poincaré. *Les Méthodes Nouvelles de la Mécanique Céleste*. 3 volumes, Gauthier-Villars, Paris, 1892 - 1899.
2. Simon Portegies Zwart and Tjarda Boekholt. On the Minimal Accuracy Required for Simulating Self-Gravitating Systems by Means Of Direct N-Body Methods. *The Astrophysical Journal Letters*, 785:1–4, 2014.
3. Wolfgang Gröbner. *General Theory of the Lie Series*, pages 1–42. In Gröbner and Knapp [4], 1967.
   Wolfgang Gröbner. *Die Lie-Reihen und ihre Anwendungen*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1967.
4. Wolfgang Gröbner and Herbert Knapp, editors. *Contributions to the Method of Lie Series*. Bibliographisches Institut, Mannheim, 1967.
5. Norbert Müller and Margarita Korovina. Making Big Steps in Trajectories. *Electronic Proceedings in Theoretical Computer Science*, 24:106–119, 2010.
6. Herbert Knapp and Gerhard Wanner. *On the Numerical Treatment of Ordinary Differential Equations*, pages 43–98. In Gröbner and Knapp [4], 1967.
   Gerhard Wanner. *Integration Gewöhnlicher Differentialgleichungen*. Bibliographisches Institut, Mannheim, 1969.

Algorithmic Randomness in Chemical Reaction Networks *

Xiang Huang[1], Jack H. Lutz[1], and Andrei N. Migunov[1]

[1]Department of Computer Science, Iowa State University, Ames, IA 50011, USA,
{huangx,lutz,amigunov}@iastate.edu

A continuous-time Markov chain is a certain kind of stochastic process with a discrete state space that evolves probabilistically over continuous time. Its state changes are thus instantaneous *jump transitions* that occur at random real-valued times. One important class of continuous-time Markov chains consists of stochastic chemical reaction networks. Stochastic chemical reaction networks are known to be Turing universal [5] and are now used in molecular programming, DNA nanotechnology, and synthetic biology to model and specify the behaviors of natural and engineered molecular systems.

This talk develops the elements of the theory of algorithmic randomness in continuous-time Markov chains. Algorithmic randomness defines what it means for an *individual* trajectory of a computable continuous-time Markov chain to be random. This enables one to carry out *Kolmogorov's program* of replacing probabilistic laws stating that *almost every* trajectory has a given property with randomness laws stating that *every* random trajectory has the property.

Briefly, a continuous-time Markov chain is an ordered triple $C = (Q, \lambda, \pi)$ where Q is a countable set of *states*, $\lambda : Q \times Q \to [0, \infty)$ is a *rate matrix* satisfying $\lambda(q, q) = 0$ for each $q \in Q$, and the *state initialization* π is a probability measure on Q. We take Q to be of the form $\{0, 1, ..., n\}$ if Q is finite and to be $\mathbb{N}$ if Q is infinite, and we define C to be *computable* if λ and π are computable.

A *trajectory* of a continuous-time Markov chain C is a sequence $\boldsymbol{\tau} = ((q_0, t_0), (q_1, t_1), (q_2, t_2), ...)$ of ordered pairs $(q_i, t_i) \in Q \times (0, \infty)$. Intuitively, $\boldsymbol{\tau}$ denotes the turn of events in which, for each i, C enters the state q_i at time $\sum_{j<i} t_j$, remains in state q_i for the *sojourn time* t_i, and jumps to state q_{i+1} at time $\sum_{j \leq i} t_j$. When convenient, we write $\boldsymbol{\tau} = (\boldsymbol{q}, \boldsymbol{t})$ where $\boldsymbol{q} = (q_0, q_1, q_2, ...)$ and $\boldsymbol{t} = (t_0, t_1, t_2, ...)$. We write $\mathbb{T}[C]$ for the set of all trajectories of C.

The classical theory of continuous-time Markov chains [1, 2, 4] imposes a probability measure μ on $\mathbb{T}[C]$. We present here a formulation of *C-martingales* that bet on trajectories in $\mathbb{T}[C]$, and we prove a martingale characterization of the μ-measure 0 sets. This characterization says that a set $Z \subseteq \mathbb{T}[C]$ satisfies $\mu(Z) = 0$ if and only if there is a C-martingale d such that, for each $\boldsymbol{\tau} \in Z$, d *succeeds* (makes unbounded money betting) on $\boldsymbol{\tau}$. (This is an analog of Ville's theorem [6], which was proven in the simpler setting of Cantor space.)

When C is a continuous-time Markov chain that is computable, we define a trajectory $\boldsymbol{\tau} \in \mathbb{T}[C]$ to be *random* if no lower semicompoutable C-martingale succeeds on $\boldsymbol{\tau}$. An effective version of the above martingale characterization then yields a characterization of random trajectories in terms of Martin-Löf tests [3].We also give a characterization of random trajectories in terms of the Kolmogorov complexities of their approximations.

As an illustration, we prove that, in any stochastic chemical reaction network, *every* random trajectory $\boldsymbol{\tau} = ((q_0, t_0), (q_1, t_1), (q_2, t_2), ...)$ with bounded molecular counts has the *non-Zeno property* that $\sum_{i=0}^{\infty} t_i = \infty$.

References

[1] Krishna B. Athreya and Soumendra N. Lahiri. *Measure Theory and Probability Theory.* Springer, 2006.

*This research was supported in part by National Science Foundation Research Grants 1247051 and 1545028.

[2] Richard Durrett. *Essentials of Stochastic Processes*. Springer, 2016.

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

[4] Yurii A. Rozanov. *Probability Theory: A Concise Course*. Dover, 1977.

[5] David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with finite stochastic chemical reaction networks. *Natural Computing*, 7(4):615–633, 2008.

[6] Jean Ville. *Etude critique de la notion de collectif*. Gauthier-Villars Paris, 1939.

Rigorous Numerical Computing with ARIADNE

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

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

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

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

Fifteenth International Conference on
Computability and Complexity in Analysis
Kochelsee, 5-8 August 2018

Abstract

In this talk I will give an overview of the computational kernel of ARIADNE, a software tool originally developed for verification of hybrid nonlinear dynamic systems [2, 3]. Since this is challenging problem involving many fundamental problems of continuous mathematics, and requiring reliable solutions, the computational kernel is itself a general-purpose toolbox of rigorous numerical methods. Further, to achieve a clean interface and theoretically sound semantics, the toolbox is based on the framework of computable analyis.

The toolbox contains general-purpose numeric, algebraic, functional and geometric calculi [1, 4]. Generic classes are provided for the main types of continuous mathematics, including Boolean, Sierpinskian and Kleenean logical types, real numbers types including lower- and upper-reals, continuous and differentiable functions, and open, closed, compact and overt sets. For each type, `Effective` classes represent computable mathematical abstractions, whereas `Validated` classes represent finite-accuracy approximations; each supports the same computable operations. Effective objects typically have a symbolic representation, whereas different concrete implementations may be provided for validated objects to support data exchange and efficient computation.

The operations at the validated level are implemented using rigorous numerical methods, including interval arithmetic, automatic differentiation and affine and polynomial (Taylor) model arithmetic. More advanced operations are also provided, such as Newton and Krawczyk operators for the solution of parametrised algebraic equations, Taylor-Picard methods for the solution of differential equations, and interior-point methods for nonlinear optimisation and feasibility problems.

Throughout the talk I shall explain I will explain the main concepts used in the implementation, and give examples of the use of ARIADNE, from simple real-number arithmetic to systems verification.

Finally, I will give an overview of work in progress in extending the scope of the tool, including to piecewise-continuous, measurable and multivalued functions, and to solutions of differential inclusions.

References

[1] P. J. Collins. Computable analysis with applications to dynamic systems. Technical Report MAC-1002, Centrum Wiskunde en Informatica, May 2010.

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

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

[4] Pieter Collins, Milad Niqui, and Nathalie Revol. A validated real function calculus. *Math. Comput. Sci.*, 5:437–467, 2011.

coqrep: A Coq library for representations

Florian Steinberg

It is usually thought that each of proof of computability in computable analysis describes a concrete algorithm with rigorous semantics and can in principle directly be translated into a program executable on a digital computer. This is reflected in the existence of several software packages [10, 8, 4, and more]. The necessity to provide an algorithm that is correct at arbitrary precision makes it difficult for software based on computable analysis to compete with implementations based on machine numbers in terms of speed. Thus, the major application is reliable computation. In practice the transition from a mathematical proof to software is far from smooth. In building software by hand it is almost impossible to avoid bugs and the handling of continuous data can be subtle at times. Thus, automatic program extraction from formal proofs for computable analysis is an active field [5]. Unfortunately, extracted programs are known for being notoriously slow. Therfore, recently there has been an increased interest in both proving correctness of programs written in reliance on efficient software packages [11], and in verifying parts of the software itself. To make the proofs of correctness as reliable as possible, the use of proof assistants comes in handy.

Coq is one of the more popular proof assistants and traditionally centered around constructive reasoning. Constructive mathematics shares the acclaim of each proof being executable with computable analysis and Coq provides a machinery for program extraction and also allows direct execution inside of Coq. A strict restriction to constructive reasoning can at times be inconvenient and is not necessary for guaranteeing executability. Conveniently, in Coq it is possible to do classical reasoning via the introduction of Axioms, and, while this breaks executability in general, to keep track of where classical reasoning is used and suggest constructive reasoning in the important parts.

I present first steps I took towards providing a library for computable analysis in Coq. There exists work providing guidelines for how this should be done [2] and similar projects in different, mostly constructive, settings and environments [3, 7] that are helpful for orientation. I present some proofs of basic results formalized in this library, like the computability of the arithmetic operations on the reals, the incomputability of the limit operator but computability of its restriction to the efficiently convergent sequences. I also show how to prove computability of polynomial evaluation from the computability of the operations on the reals and discuss why the extracted algorithm is slow and how it could be optimized in the scope of the library. Finally I talk about some more abstract proofs like the correctness of the product and function space constructions. The library and basic examples are accessible on github [1].

The examples above outline some of the goals the library is designed around: Firstly, each proof of computability should be directly executable in Coq. Secondly, both high level reasoning and concretely specifying an algorithm and

proving it correct should be possible. And finally it should be as easy as possible to build interfaces with other libraries, and reuse proofs and algorithms. Some of these goals are debatable. For instance, due to the requirement to be able to directly execute inside of Coq, incomputability can currently only be proven by proving discontinuity. This is because it is possible to assume computationally void but consistent axioms which also makes executability fragile. I critically discuss some of the choices and possible ways to mend the deficiencies.

Coq provides well developed libraries for both constructive analysis [9] as well as for classical analysis [6]. The library CoRN for constructive analysis is for most parts executable and has been optimized for speed. It is currently strictly restrained to the use of constructive reasoning, but there seem to be plans to change this in the future. There are many other relevant libraries, for instance a library for interval computations. I describe first steps I took in attempt to build interfaces with some of these libraries. While the hope is to extract programs that produce results in acceptable time spans, there is no ambition to compete with existent efficient software in terms of speed. I hope that the library can be helpful in the task of verification.

References

[1] coqrep. `https://github.com/FlorianSteinberg/coqrep`.

[2] Andrej Bauer. Realizability as the connection between computable and constructive mathematics. `http://math.andrej.com/data/c2c.pdf`, 2005.

[3] Andrej Bauer and Christopher A. Stone. Rz: A tool for bringing constructive and computable mathematics closer to programming practice. In S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World*, pages 28–42, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[4] Luca Benvenuti, Davide Bresolin, Alberto Casagrande, Pieter Collins, Alberto Ferrari, Emanuele Mazzi, Alberto Sangiovanni-Vincentelli, and Tiziano Villa. Reachability computation for hybrid systems with Ariadne. *IFAC Proceedings Volumes*, 41(2):8960 – 8965, 2008. 17th IFAC World Congress.

[5] U. Berger. From coinductive proofs to exact real arithmetic: theory and applications. *Logical Methods in Comput. Sci.*, 7(1):1–24, 2011.

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

[7] Martìn Escardò and Chuangjie Xu. A constructive manifestation of the kleenekreisel continuous functionals. *Annals of Pure and Applied Logic*, 167(9):770 – 793, 2016. Fourth Workshop on Formal Topology (4WFTop).

[8] Michal Konečný. A Haskell library for Approximating Exact Real Numbers (AERN). `https://github.com/michalkonecny/aern2`.

[9] Robbert Krebbers and Bas Spitters. Computer certified efficient exact reals in Coq. In James Davenport, William Farmer, Josef Urban, and Florian Rabe, editors, *Intelligent Computer Mathematics*, volume 6824 of *Lecture Notes in Computer Science*, pages 90–106. Springer, 2011.

[10] Norbert Th. Müller. iRRAM - Exact Arithmetic in C++. http://irram.uni-trier.de/, https://github.com/norbert-mueller/iRRAM, 2017.

[11] Norbert Th. Müller, Sewon Park, Norbert Preining, and Martin Ziegler. On formal verification in imperative multivalued programming over continuous data types. *CoRR*, abs/1608.05787, 2016.

Bit Complexity of Computing Solutions of Boundary-value Problems for Some Systems of PDEs with Guaranteed Precision

S.V. Selivanova*
KAIST and S.L. Sobolev Institute of Mathematics SB RAS
V.L. Selivanov
A.P. Ershov Institute of Informatics Systems SB RAS
and Kazan Federal University

We establish upper bounds for the guaranteed-precision problem of finding solutions of boundary-value problems for symmetric hyperbolic systems

$$\begin{cases} A\frac{\partial \mathbf{u}}{\partial t} + \sum\limits_{i=1}^{m} B_i \frac{\partial \mathbf{u}}{\partial x_i} = f(t, x_1, \ldots, x_m), \\ \mathbf{u}|_{t=0} = \varphi(x_1, \ldots, x_m), \\ \Phi_i^{(1)}\mathbf{u}(t, x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_m) = 0, \\ \Phi_i^{(2)}\mathbf{u}(t, x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_m) = 0, \\ i = 1, 2, \ldots, m. \end{cases} \tag{1}$$

Here $A = A^* > 0$, $B_i = B_i^*$; $\Phi_i^{(1)}$, $\Phi_i^{(2)}$ are real rectangular matrices such that the boundary conditions are correctly posed and dissipative. For additional information on such systems see e.g. [1, 2].

Let $\mathbb{Q}, \mathbb{A}$ be respectively the ordered fields of rationals and of algebraic reals, equipped with natural binary encodings. Such encodings induce binary encodings of matrices and polynomials over these fields used in the following main result of this work. The formulation uses some rather standard notation from [1, 2].

Theorem 1. Let $m, n \geq 2$ be fixed positive integers and let T be a fixed positive rational number. Then

1. There is an EXPTIME algorithm which, for any matrices $A, B_1 \ldots, B_m \in M_n(\mathbb{A})$, polynomials $\varphi_1 \ldots, \varphi_n \in \mathbb{Q}[x_1 \ldots, x_m]$, $f_1 \ldots, f_n \in \mathbb{Q}[t, x_1 \ldots, x_m]$, rational matrices $\Phi_i^{(1)}$, $\Phi_i^{(2)}$ meeting the conditions above and for any precision $a \geq 1$, computes a spatial rational grid step h dividing 1, a time grid step τ dividing T and a rational (h, τ)-grid function $v : G \to \mathbb{A}$ such that

$$||\mathbf{u} - \widetilde{v\,|_H}||_{sL_2} < \varepsilon$$

where $\varepsilon = \frac{1}{a}$ and $H = [0, T] \times [0, 1]^m$.

*Supported by the National Research Foundation of Korea (grant NRF-2017R1E1A1A03071032) and the International Research & Development Program of the Korean Ministry of Science and ICT (grant NRF-2016K1A3A7A03950702)

2. The algorithm of item 1 becomes PTIME if the precision $a \geq 1$ is fixed and the input data for the algorithm are such that the following quantities are bounded by a fixed integer $M > 0$:

$$\max\{||A||_2, ||B_i||_2, ||\Phi_i^{(1)}||_2, ||\Phi_i^{(2)}||_2, \frac{\lambda_{max}(A)}{\lambda_{min}(A)}, ||(A^{-1}B_i)^2||_2,$$

$$\max_{i,j} ||A^{-1}B_iA^{-1}B_j - A^{-1}B_jA^{-1}B_i||_2, \sup_x ||\frac{\partial^2\varphi}{\partial x_i \partial x_j}(x), \sup_{t,x} ||\frac{\partial^2 f}{\partial x_i \partial x_j}(t,x)||_2\}.$$

Our approach makes a heavy use of some known and our own algorithms of computer algebra (exact computations with integers, rationals, algebraic reals and polynomials, polynomial-time computability of spectral decomposition of symmetric matrices and matrix pencils in the field of algebraic reals), together with notions and methods from numerical mathematics [1] and computable analysis [3, 4]. Altogether, our proofs demonstrate a fruitful mix of methods from symbolic and numerical computation.

References

[1] Godunov S.K., ed. Numerical Solution of Higher-dimensional Problems of Gas Dynamics (in Russian). Nauka, Moscow, 1976.

[2] Selivanova S., Selivanov V. Computing Solution Operators of Boundary-value Problems for Some Linear Hyperbolic Systems of PDEs. Logical Methods in Computer Science, 13 (4:13) 2017, pp. 1–31. Earlier version on arXiv:1305.2494 (2013).

[3] Weihrauch K. Computable Analysis. Berlin, Springer, 2000.

[4] Ziegler M., Brattka V. A computable spectral theorem. Proc. CCA-2001, Lecture Notes in Computer Science, v. 2064 (2001), 378–388.

Parallel Complexity Theory of Linear ODEs and PDEs*

Ivan Koswara, Svetlana Selivanova, and Martin Ziegler: KAIST

Abstract. We study the algorithmic cost of solving differential equations from the rigorous perspective of parallel complexity theory: linear ODEs, and surprisingly also certain linear PDEs, can be solved approximately up to absolute error 2^{-n} in space $\mathcal{O}(\log^2 n)$.

Recursive Analysis provides a rigorous algorithmic foundation to Numerics, that is, to operations on continuous data by means of approximations. It has a long history of thorough investigations regarding ordinary (ODEs) and partial differential equations (PDEs) with respect to computability and complexity [BGP11, KORZ14, KSZ17]. The present work focuses on parallel time, well-known [Bor77] to coincide with space (=memory). We consider computation of real functions on a compact domain by space-bounded Type-2 Machines and by Boolean circuits of bounded depth:

Definition 1. *Fix a total real function $f : X \subseteq \mathbb{R}^d \to [0;1]$.*

a) Consider a Type-2 Turing Machine with the usual one-way output tape and working tape(s), but the input tape(s) being read-only. *It is said to* compute *f if, given any sequence $\vec{a}_m$ of d-tuples of integers (in binary encoding $\mathrm{bin}(\vec{a})$) satisfying $|\vec{x} - \vec{a}_m/2^m| \le 2^{-m}$ for some $\vec{x} \in X$, it outputs a sequence $b_n \in \mathbb{Z}$ in binary such that $|f(\vec{x}) - b_n/2^n| \le 2^{-n}$.*

b) The computation from (a) runs in space *$s : \mathbb{N} \to \mathbb{N}$ if, until producing said $\vec{b}_n$, the machine accesses at most $s(n)$ different cells of the working tape: regardless of $\vec{x} \in X$ and of $(\vec{a}_m)_m$. If additionally f has a polynomial modulus of continuity, we say that f belongs to $\mathbb{R}\mathsf{SPACE}\big(s(n)\big)$. $\mathbb{R}\mathsf{PSPACE}$ is the class of real functions with exponential modulus of continuity computable using at most polynomial space.*

c) A modulus of continuity *of f is a non-decreasing mapping $\mu : \mathbb{N} \to \mathbb{N}$ such that $|f(\vec{x}) - f(\vec{x}')| \le 2^{-n}$ holds for all $\vec{x}, \vec{x}' \in X$ with $|\vec{x} - \vec{x}'| \le 2^{-\mu(n)}$.*

d) Consider a Boolean circuit C_n having n binary outputs and $m = \tilde{\mu}(n)$ binary inputs. Such a sequence (C_n) computes *f if C_n, on every (possibly padded) input $\mathrm{bin}(\vec{a})$ with $|\vec{x} - \vec{a}/2^m| \le 2^{-m}$ for some $\vec{x} \in X$, it outputs $\mathrm{bin}(b_n)$ such that $|f(\vec{x}) - b_n/2^n| \le 2^{-n}$.*

e) We say that the real function f is computable in polysize-depth *$t(n)$, written $\mathbb{R}\mathsf{DEPTH}\big(t(n)\big)$, if there exists a logspace-uniform sequence (C_n) of Boolean circuits with bounded fan-in of depth at most $t(n)$ and size (#gates) bounded polynomially in n. $\mathbb{R}\mathsf{NC}^i$ abbreviates $\mathbb{R}\mathsf{DEPTH}\big(\log^i n\big)$.*

Boolean circuits are a standard model of parallel computation whose depth corresponds to concurrent runtime. The well-known hierarchy [Vol99, Corollary 2.34] translates to our real setting:

$$\mathbb{R}\mathsf{NC}^1 \subseteq \mathbb{R}\mathsf{SPACE}(\log n) \subseteq \mathbb{R}\mathsf{NC}^2 \subseteq \mathbb{R}\mathsf{SPACE}(\log^2 n) \subseteq \mathbb{R}\mathsf{NC}^4 \subseteq \mathbb{R}\mathsf{P} \subseteq \mathbb{R}\mathsf{PSPACE} \ .$$

Note that we consider the *out*put length/precision $n \in \mathbb{N}$ as complexity parameter, both for Turing machines (Definition 1b) and for circuits (Definition 1e). In the classical setting of type-1 machines with finite inputs, any terminating computation using space $\mathcal{O}(\log n)$ necessarily makes at most a polynomial number $t(n)$ of steps. This fails for type-2 machines, as pointed out by Florian Steinberg (personal communication). There it is folklore [Wei03, Sch04, KSZ16] that every computation of a function f with compact domain X admits a runtime — and therefore also space — bound $t : \mathbb{N} \to \mathbb{N}$ depending only on the output precision n; moreover $\mu(n) := t(n+1) + 1$ constitutes a modulus of continuity of f. This explains us turning this into an explicit requirement in Item (b), and considering circuits of input size $\mathcal{O}\big(\mu(n)\big)$ for output size n in (d). We follow the classical conception of real numbers as 'streams' of approximations, both for input and output [Hoo90]: the alternative approach based on oracles [Ko91, KO14] involves a *stack* of query tapes to ensure closure under composition.

*Supported by the National Research Foundation of Korea (grant NRF-2017R1E1A1A03071032) and by the Korean Ministry of Science and ICT (grant NRF-2016K1A3A7A03950702)

Theorem 2. *Given $A \in [-1;1]^{d\times d}$ and $\vec{v} \in [-1;1]^d$, the solution to the first-order system of linear ordinary differential equations*

$$\tfrac{\partial}{\partial t}\vec{u}(t) \;=\; A\vec{u}(t), \quad t \in [0;1], \quad \vec{u}(0) = \vec{v} \tag{1}$$

is $\vec{u}(t) = \exp(tA)\vec{v} := \sum_k \frac{t^k}{k!} A^k\vec{v}$ and computable in polysize-depth $\mathcal{O}\big(\log(n)\cdot(\log d + \log n)\big)$.

The Abstract Cauchy Problem includes PDEs with differential operator A:

Theorem 3. *Fix a Banach space $\mathcal{B}$ with families of closed subspaces $\mathcal{V}_d \subseteq \mathcal{W}_d$ and linear operators $A_d : \mathcal{V}_d \to \mathcal{W}_d$ such that also all iterates $A_d^k : \mathcal{V}_d \to \mathcal{W}_d$ are well-defined and continuous. Then $u(t) = \exp(tA)\vec{v} \in \mathcal{W}_d$ is well-defined for all $\vec{v} \in \mathcal{V}_d$ and all $|t| \le R_d/2$ and solves Equation (1). Here $R_d := 1/\limsup_k \sqrt[k]{|||A_d^k|||/k!}$ with operator norm $|||A_d^k||| = \sup\{\|A_d^k\vec{v}\|/\|\vec{v}\| : \vec{v} \in \mathcal{V}_d\}$. Suppose furthermore that A_d is computable on $\mathcal{V}_d$'s unit ball in space $\mathcal{O}\big(s(d\cdot n)\big)$, $s(n) \ge \log n$. Then, given any unit $\vec{v} \in \mathcal{V}_d$ and $|t| \le R_d/2$, $u(t)$ is computable in space $\mathcal{O}\big(s(d\cdot n)\cdot\log(n)\big)$.*

Example 4. For each $d \in \mathbb{N}$ consider the normed vector space of real analytic functions

$$\mathcal{V}_d \;=\; \big\{f : [0;1] \to \mathbb{R} \;:\; \exists K \in \mathbb{N}\; \forall k \in \mathbb{N}\; |f^{(k)}|_\infty \le K\cdot k!\cdot d^k\big\}, \quad \|f\| \;:=\; \sum\nolimits_k |f^{(k)}|_\infty/k!^2$$

where $f^{(k)}$ denotes k-th iterated derivative and $|f|_\infty := \sup_x |f(x)|$. Then $\partial^k : \mathcal{V}_d \ni f \mapsto f^{(k)} \in \mathcal{V}_{2d} =: \mathcal{W}_d$ is well-defined and $e^d d^k k!$-Lipschitz (so $R_d = 1/d$) and computable in space $\mathcal{O}(\log n + \log d)$, cmp. [KMRZ15, §3.2].

Future work will extend this to Gevrey functions in several variables and more involved differential operators such as in linear hyperbolic systems and boundary conditions [SS17].

References

[BGP11] Olivier Bournez, Daniel S. Graça, and Amaury Pouly. Solving analytic differential equations in polynomial time over unbounded domains. In *Mathematical foundations of computer science 2011*, volume 6907 of *Lecture Notes in Comput. Sci.*, pages 170–181. Springer, 2011.

[Bor77] Borodin. Relating time and space to size and depth. *SIAM J. Comput.*, 6(4):733–744, 1977.

[Hoo90] H. James Hoover. Feasible real functions and arithmetic circuits. *SIAM J. Comput.*, 19(1):182–204, 1990.

[KMRZ15] Akitoshi Kawamura, Norbert Th. Müller, Carsten Rösnick, and Martin Ziegler. Computational benefit of smoothness: Parameterized bit-complexity of numerical operators on analytic functions and Gevrey's hierarchy. *J. Complexity*, 31(5):689–714, 2015.

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

[KO14] Akitoshi Kawamura and Hiroyuki Ota. Small complexity classes for computable analysis. In *Mathematical foundations of computer science 2014. Part II*, volume 8635 of *Lecture Notes in Comput. Sci.*, pages 432–444. Springer, Heidelberg, 2014.

[KORZ14] Akitoshi Kawamura, Hiroyuki Ota, Carsten Rösnick, and Martin Ziegler. Computational complexity of smooth differential equations. *Logical Methods in Computer Science*, 10:1:6,15, 2014.

[KSZ16] Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. Complexity theory of (functions on) compact metric spaces. In *Proc. 31st Ann. Symposium on Logic in Computer Science, LICS*, pages 837–846. ACM, 2016.

[KSZ17] Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. On the computational complexity of the Dirichlet problem for Poisson's equation. *Mathematical Structures in Computer Science*, 27:8:1437–1465, 2017.

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

[SS17] Svetlana Selivanova and Victor L. Selivanov. Computing solution operators of boundary-value problems for some linear hyperbolic systems of PDEs. *Logical Methods in Computer Science*, 13(4), 2017.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity*. EATCS Text in Theoretical Computer Science. Springer, 1999.

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

Complexity of a Root Clustering Algorithm

Prashant Batra
Hamburg University of Technology, Germany. batra@tuhh.de
Vikram Sharma
The Institute of Mathematical Sciences, HBNI, Chennai, India. vikram@imsc.res.in

The **exact root finding** (ERF) problem is to approximate the roots of a holomorphic function f in an input box B_0 to within some desired input accuracy. We assume that f and all its derivatives are represented by their "box"-versions, i.e., given a box $B \subset \mathbb{C}$, we can compute an overestimation of the range $f(B)$; moreover, for a sequence of boxes monotonically converging to a point the error in the overestimation goes to zero.

The ERF problem naturally generalizes the problem of approximating polynomial roots [Sch82]. However, not many algorithms are known in the literature for solving it [Hen74, GLSY05]. Very few of these algorithms have a complete complexity analysis. One such result states that the roots of a polynomial-time computable function are also poly-time computable [Ko91, Thm. 4.11]. But this result is not uniform and it assumes that all the roots are simple. A desirable goal is to devise **complete** algorithms, i.e., algorithms that do not make such niceness assumptions. However, working with a box-representation of f means that we cannot distinguish between a multiple root and a "cluster" of very close roots using finite precision. Therefore, we have to change the problem from approximating roots to approximating clusters. This is called the **root clustering problem** (RCP). Moreover, we cannot use comparisons with zero in devising such algorithms, as we can only compute absolute approximation in the computational model. We instead use a **soft zero test**, i.e., a comparison that only decides the sign of a *non-zero real number*. A complete algorithm based on soft zero tests for RCP was presented in [YSS13]. In this work, we bound the complexity of this subdivision-based algorithm and obtain a generalisation of similar results for the case of polynomials [BSS$^+$16].

To state our contribution, we introduce some notation. We consider a modified version of Smale's gamma-function $\gamma(f,z) := \sup_{k\geq 1} |f^{(k)}(z)/(k!f(z))|^{1/k}$ [GLSY05]. Let $Z(f) \subset \mathbb{C}$ be the set of roots (possibly empty) of f, and N be the number of roots in B_0. Let $\mathcal{C} \subset Z(f)$ be a cluster of roots of f as in [YSS13], $m_{\mathcal{C}}$ be its centroid and $r_{\mathcal{C}}$ the smallest disc centered at $m_{\mathcal{C}}$ that contains $\mathcal{C}$. Define $\gamma_{\mathcal{C}}(z) := \gamma(h,z)$, where h is the holomorphic function such that $f = h\prod_{\alpha\in\mathcal{C}}(z-\alpha)$. A cluster is called **strongly separated** if $r_{\mathcal{C}}\gamma_{\mathcal{C}}(m_{\mathcal{C}}) \leq 1$. Let $R_f(w)$ denote the radius of the largest disc centered at w on which the function $zf(z)$ is conformal, i.e., holomorphic and bijective. Using de Brange's theorem, we show that $\gamma(f,z) \leq 1/R_f(z)$, when $f(z) \neq 0$. Using the framework of continuous amortisation [Bur16], we derive our main result: The number of

subdivisions in the algorithm in [YSS13] is

$$O\left(\sum_{\mathcal{C}\in S_0} |\mathcal{C}|^2 + \int_{B_0 \setminus \cup_{\mathcal{C}\in S_0} D(m_{\mathcal{C}}, 1/\gamma_{\mathcal{C}}(m_{\mathcal{C}}))} \frac{dA}{(R_f(z))^2}\right), \tag{1}$$

where S_0 is the set of all strongly separated clusters in, roughly, the scaled box NB_0. This bound is a more precise analysis of the number of subdivisions than [Yak05], which contains an algorithm similar to [YSS13]; the key difference is that we account for boxes even when there are no roots in B_0. We also bound the precision required by the algorithm in [YSS13].

For the special case of the exponential function, using an estimate of Bloch on the radius of univalence, we show that $R_{\exp}(z)$ is always at least a constant, and hence the integral is upper bounded by $O(\text{Area}(B_0))$. *This bound is exponential in the input size*, namely, the dyadic coordinates of the corner vertices of B_0. We also show that this bound is tight. A similar analysis also applies for the sine function. Thus the algorithm in [YSS13] is in the worst-case an exponential time algorithm. When $f \in \mathbb{C}[x]$, the bound in (1) is polynomial in the input parameters, and is similar to known bounds in the literature [BSS$^+$16].

Our results, thus, establish a framework for developing uniform complexity results for RCP for holomorphic functions. We introduce some natural geometric parameters, such as $R_f(z)$, in the analysis of the algorithm; similar parameters have also been used in real analysis [KMRZ15]. Our analysis shows that the algorithm in [YSS13] is exponential in the worst case, and more work needs to be done for developing an efficient algorithm.

References

[BSS$^+$16] Ruben Becker, Michael Sagraloff, Vikram Sharma, Juan Xu, and Chee Yap. Complexity analysis of root clustering for a complex polynomial. In Proc. *41st Int'l Symp. Symbolic and Alge. Comp.* (ISSAC), 2016.

[Bur16] M. Burr. Continuous amortization and extensions: With applications to bisection-based root isolation. *J. Symb. Comput.*, 77:78–126, 2016.

[GLSY05] M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn. On location and approximation of clusters of zeros of analytic functions. *Found. Comput. Math.*, 5(3):257–311, 2005.

[Hen74] P. Henrici. *Applied and Computational Complex Analysis*, vol. 1. John Wiley & Sons, New York, 1974.

[KMRZ15] A. Kawamura, N. Müller, C. Rösnick, and M. Ziegler. Computational benefit of smoothness: Parameterized bit-complexity of numerical operators on analytic functions and Gevrey's hierarchy. *Journal of Complexity*, 31(5):689 – 714, 2015.

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

[Sch82] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Updated Manuscript, 2004. University of Tübingen.

[Yak05] J.-C. Yakoubsohn. Numerical analysis of a bisection-exclusion method to find zeros of univariate analytic functions. *J. of Compl.*, 21:652–690, 2005.

[YSS13] C. Yap, M. Sagraloff, and V. Sharma. Analytic root clustering: A complete algorithm using soft zero tests. In P. Bonizzoni et al., (eds.), *Computability in Europe* (CiE), Springer, Heidelberg, 2013.

Bit-Complexity and Rigorous Implementation of the Fast Multipole Method for Trummer's Problem*

Bastian Dörig

KAIST, South Korea and Technische Universität Darmstadt, Germany
doerig@mathematik.tu-darmstadt.de

Abstract. We will introduce a quasilinear, in terms of bit-cost, and implementation friendly version of the *Fast Multipole Method* (FMM) in order to solve the real version of Trummer's Problem, which is closely related to the N-Body Simulation Problem with pairwise Coloumb forces. While solving the differential equation for the previous mentioned problem numerically one has to solve Trummer's Problem in each iteartion of the Euler Method.

Keywords: Reliable Computation · N-Body Simulation · Bit Complexity.

1 Trummer's Problem

Definition 1 (Trummer's Problem) *Let $q \coloneqq (q_1, \ldots, q_N) \in \mathbb{R}^N$ be a vector of N real values, let $z \coloneqq (z_1, \ldots, z_N) \in \mathbb{R}^N$ be a vector of N real values and let $C(q,z)$ be the* Cauchy-Matrix *defined by $C(q,z)_{i,j} \coloneqq \frac{q_i}{z_i - z_j}$ for $i \neq j$ and $C(q,z)_{i,i} \coloneqq 0$. The real version of* Trummer's Problem *is the computation of the matrix vector product $C(q,z) \cdot q$ up to a given error 2^{-n}.*

The naive algorithm, multiplying each row of the matrix with the vector, has a quadratic bit-cost complexity. The first quasilinear, in terms of unit cost, solution to Trummer's problem was given by Gerasoulis et al. [2] by using polynomial interpolation and evaluation techniques. Further investigations [6] indicate that this has quadratic bit-complexity. When working with real numbers and real complexity theory [4, 5] the Fast Multipole Method [8] applied on a Well Separated Pair Decomposition [1] turns out to have quasilinear bit-complexity for solving Trummer's Problem.

2 The Fast Multipole Method

Reinterpretation of the i-th element of $C(q,z) \cdot q$ yields to evaluation of the function ${}^i\Phi : \mathbb{C} \to \mathbb{C} : z \mapsto \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{q_i q_j}{z - z_j}$ in the point $z = z_i$. In order to evaluate ${}^i\Phi$ in z we have to compute the N terms $\frac{q_1}{z - z_1}, \ldots, \frac{q_N}{z - z_N}$. The main idea of the *Fast Multipole Method* is to expand the ${}^i\Phi$ into Laurent series, manipulate them and evaluate them at the end. The manipulation includes: truncation to a reasonably order p, combination of multiple Laurent series, conversion of Laurent series into Taylor series and combination of multiple Taylor series. For some points $z_1, ..., z_n$ with the corresponding Laurent series L, the approximation error of the truncated Laurent series L_p depends on the truncation parameter p and the distance of the evaluation

* Special thanks to Martin Ziegler who gave me the opportunity to start my research on this project at KAIST and for the in depth discussions about real complexity theory.

point x to the other points z_i. In order to estimate the overall errorbound one has to cluster all points, such that the pairwise distance of all clusters can be estimated. This clustering can be done with the Well Separated Pair Decompostion [1]. The abstract procedure of the Fast Multipole Method to solve Trummer's Problem is sketched in algorithm 1.

Analysing the proposed algorithm one sees that the main steps *upcast*, *conversion* and *downcast* require the multiplication of Laurent and Taylor series. For the ease of implementation the multiplications of truncated series can be transformed into multiplications of long integers, which have quasilinear bit-complexity. These integers will be used throughout the complete algorithm, hence the data management becomes clearer. We keep track of all errorbounds on our own and use long integers to encode the real values, instead of relying on other frameworks for exact real arithmetic [7]. This should result in faster runtimes and tighter errorbounds. Analysation of the complete algorithm using the above representation in fact turns out be have quasilinear bit-cost complexity. The need to implement a Fast Fourier Transformation (FFT) to multiply polynomials and switch between different data structures is not necessary any more. N.B. The FFT is still mandatory to obtain quasilinear complexity, but the implementation is moved to the long integer library, which is available in most libraries, e.g. the GNU Multiple Precision Arithmetic Library [3] contains a hybrid algorithm which uses the FFT for very long integers.

Algorithm 1: The Fast Multipole Method

Input : N particles with charges $q_i \in \mathbb{R}$ and positions $z_i \in \mathbb{R}$ for $i = 1, \ldots, N$.
Output : $(C(q,z) \cdot q)_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{q_i q_j}{z_i - z_j} = {}^{i}\Phi(z_i)$ for $i = 1, \ldots, N$.

Compute WSPD: Compute the split tree $\mathcal{T}$ with associated WSPD $\mathcal{W}$;
Intialize: Compute the Laurent series ${}_{U}\mathfrak{L}(z)$ for each leaf $U \in \mathcal{T}$;
Upcast: Compute the Laurent series ${}_{U}\mathfrak{L}(z)$ by combining the Laurent series ${}_{V}\mathfrak{L}(z)$ and ${}_{W}\mathfrak{L}(z)$ for each node $U \in \mathcal{T}$ with the two corresponding children V, W;
Conversion: Convert the Laurent series ${}_{V}\mathfrak{L}(z)$ of V into a Taylor series ${}_{U}\mathfrak{T}(z)$ of the paired node U and vice versa for each pair $(U, V) \in \mathcal{W}$;
Downcast: Combine the old Taylor series ${}_{U}\mathfrak{T}(z)$ of U with the Taylor series ${}_{V}\mathfrak{T}(z)$ of the parent node V for each node $U \in \mathcal{T}$ with the corresponding parent V;
Evaluation: Evaluate the Taylor series ${}_{U}\mathfrak{T}(z_U)$ of U for each leaf $U \in \mathcal{T}$;

Bibliography

[1] Callahan, P.B.: Dealing with Higher Dimensions: The Well-separated Pair Decomposition and Its Applications. Ph.D. thesis, Baltimore, MD, USA (1995), uMI Order No. GAX95-33229

[2] Gerasoulis, A., Grigoriadis, M.D., Sun, L.: A fast algorithm for trummer's problem. SIAM Journal on Scientific and Statistical Computing **8**(1), s135–s138 (1987). https://doi.org/10.1137/0908017

[3] Granlund, T., Team, G.D.: GNU MP 6.0 Multiple Precision Arithmetic Library. Samurai Media Limited, United Kingdom (2015)

[4] Kawamura, A., Ziegler, M.: Invitation to Real Complexity Theory: Algorithmic Foundations to Reliable Numerics with Bit-Costs. ArXiv e-prints (Jan 2018)

[5] Ko, K.I.: Computational Complexity of Real Functions, pp. 40–70. Birkhäuser Boston, Boston, MA (1991). https://doi.org/10.1007/978-1-4684-6802-1, https://doi.org/10.1007/978-1-4684-6802-1

[6] Köhler, S., Ziegler, M.: On the stability of fast polynomial arithmetic. In: Proc. 8th Conference on Real Numbers and Computers. pp. 147–156 (1 Jul 2008)

[7] Müller, N., Ziegler, M.: From calculus to algorithms without errors. In: Hong, H., Yap, C. (eds.) Mathematical Software – ICMS 2014. pp. 718–724. Springer Berlin Heidelberg (2014)

[8] Reif, J.H., Tate, S.R.: N-body simulation i: Fast algorithms for potential field evaluation and trummer's problem. Tech. rep. (1996)

Effective Dimensions of Projected Points

Neil Lutz
University of Pennsylvania
nlutz@cis.upenn.edu

To what extent are fractal dimensions preserved by projection mappings? This is a fundamental question and a major area of inquiry within fractal geometry [1]. To be concrete, for all $E \subseteq \mathbb{R}^2$ and $\theta \in [0, 2\pi)$, define $\text{proj}_\theta E$ as the orthogonal projection of E onto a line through the origin in direction θ. We may ask what the Hausdorff dimension $\dim_H(\text{proj}_\theta E)$ of this projected set is. Marstrand's projection theorem [8], a classical result in fractal geometry, states that if E is analytic, then for almost all θ we have $\dim_H(\text{proj}_\theta E) = \min\{\dim_H(E), 1\}$, which matches the trivial upper bound.

In this talk, we will consider an effective and pointwise version of this question using the effective Hausdorff dimension of individual points [2, 4]. In particular, we will use the algorithmic information theoretic characterization of this quantity: The *(effective Hausdorff) dimension* of a point $z \in \mathbb{R}^2$ is

$$\dim(z) = \liminf_{r \to \infty} \frac{K_r(z)}{r} ,$$

where $K_r(z)$ is the minimum Kolmogorov complexity of any rational point in a disk of radius 2^{-r} centered on x [9]. We will ask, for $z \in \mathbb{R}^2$ and $\theta \in [0, 2\pi)$, what is the dimension $\dim(\text{proj}_\theta z)$ of the projected point?

In a series of recent papers [5, 6, 7], Don Stull and I have addressed this question, either directly or via its equivalent dual, which is determining the dimension of a point $(x, ax + b)$ given $a, b, x \in \mathbb{R}$. In this talk I will survey the results from those papers and explain the new effective dimension bounding technique that we developed. I will also discuss how such results can be—and have been—used in conjunction with a *point-to-set principle* [3] to prove new, entirely classical theorems.

References

[1] Kenneth Falconer, Jonathan Fraser, and Xiong Jin. Sixty years of fractal projections. In *Fractal geometry and stochastics V*, volume 70 of *Progr. Probab.*, pages 3–25. Birkhäuser/Springer, Cham, 2015.

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

[3] Jack H. Lutz and Neil Lutz. Algorithmic Information, Plane Kakeya Sets, and Conditional Dimension. *ACM Trans. Comput. Theory*, to appear.

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

[5] Neil Lutz and Donald M. Stull. Bounding the dimension of points on a line. In TV Gopal, Gerhard Jaeger, and Silvia Steila, editors, *Theory and Applications of Models of Computation: 14th Annual Conference, TAMC 2017, Bern, Switzerland, April 20-22, 2017, Proceedings*, pages 425–439, 2017.

[6] Neil Lutz and Donald M. Stull. Dimension spectra of lines. In *Unveiling Dynamics and Complexity — 13th Conference on Computability in Europe, CiE 2017, Turku, Finland, June 12-16, 2017, Proceedings*, pages 304–314, 2017.

[7] Neil Lutz and Donald M. Stull. Projection Theorems Using Effective Dimension. arXiv:1711.02124 [cs.CC], preprint.

[8] John M. Marstrand. Some fundamental geometrical properties of plane sets of fractional dimensions. *Proc. London Math. Soc. (3)*, 4:257–302, 1954.

[9] Elvira Mayordomo. A Kolmogorov complexity characterization of constructive Hausdorff dimension. *Inf. Process. Lett.*, 84(1):1–3, 2002.

COMPUTABILITY IN A METRIC PLANE

ZVONKO ILJAZOVIĆ

Let X be a set, let l be a nonempty subset of X and let $\leq$ and $\geq$ be two mutually reverse linear orders on l. Then we say that $(l, \{\leq, \geq\})$ is a line in X. For $A, B \in l$ the line segment $\overline{AB} \subseteq l$ in $(l, \{\leq, \geq\})$ is defined in an obvious way. Similarly, each $A \in l$ determines two subsets of l called half-lines or rays.

We examine a metric space (X, d) together with a family $\mathcal{L}$ of lines in X such that the following properties hold (usual axioms of geometry – axioms of incidence, Pasch's axiom, ...):

(i) for all $A, B \in X$, $A \neq B$, there exists a unique $(l, u) \in \mathcal{L}$ such that $A, B \in l$;
(ii) there exist $A, B, C \in X$ such that there exists no $(l, u) \in \mathcal{L}$ with property that $A, B, C \in l$;
(iii) if $A, B, C \in X$ and $(l, u) \in \mathcal{L}$ are such that $l \cap \overline{AB} \neq \emptyset$, then $l \cap \overline{AC} \neq \emptyset$ or $l \cap \overline{BC} \neq \emptyset$ (note that $\overline{AB}$ is well defined by (i));
(iv) if $A, B, C \in X$, then $d(A, B) = d(A, C) + d(C, B)$ if and only if $C \in \overline{AB}$;
(v) if $(l, u) \in \mathcal{L}$, $A \in L$ and r is a ray determined by A and (l, u), then for each $x \in \mathbb{R}$, $x > 0$, there exists $T \in r$ such that $d(T, A) = x$.

We will say that $(X, d, \mathcal{L})$ is a **metric plane**. We consider a metric plane as an abstract geometric structure. If we assume that certain additional properties (axioms) hold (among others Euclidean fifth postulate), we get a geometric structure which is actually the Euclidean plane (in that case it is possible to find an isometry between (X, d) and $(\mathbb{R}^2, p)$, where p is the Euclidean metric on $\mathbb{R}^2$).

If A, B are computable points in $\mathbb{R}^2$, then the line segment $\overline{AB}$ is a computable subset of $\mathbb{R}^2$. The question is what can be said about this and similar results in a metric plane, which is a more general structure than Euclidean plane. Of course, we first have to impose computability notions in the context of a metric plane.

We will do this in a very simple way. We will just fix an effective separating sequence α in (X, d), i.e. a sequence α such that (X, d, α) is a computable metric space. Then we have the notion of a computable point in (X, d, α), the notion of a computable set in (X, d, α) etc. (That a nonempty compact set S in (X, d) is computable in (X, d, α) means that we can for any $k \in \mathbb{N}$ effectively find a finite subset of $\{\alpha_i \mid i \in \mathbb{N}\}$ which is 2^{-k}-close to S in the sense of the Hausdorff metric.)

We prove the following: if $A, B \in X$, $A \neq B$, are computable points in (X, d, α), then the line segment $\overline{AB}$ (which is determined by the geometric structure $\mathcal{L}$) is a computable set in (X, d, α). We also examine computability in a metric plane of other standard geometric objects such as lines and triangles.

References

[1] Vasco Brattka and Gero Presser. Computability on subsets of metric spaces. *Theoretical Computer Science*, 305:43–76, 2003.
[2] Konrad Burnik and Zvonko Iljazović. Computability of 1-manifolds. *Logical Methods in Computer Science*, 10(2:8):1–28, 2014.
[3] Marian Pour-El and Ian Richards. Computability in Analysis and Physics. *Springer-Verlag, Berlin-Heielberg-New York*, 1989.
[4] Klaus Weihrauch. Computable Analysis *Springer, Berlin*, 2000.

COMPUTABILITY OF SETS WHICH CAN BE APPROXIMATED BY 2-CELLS

ZVONKO ILJAZOVIĆ AND BOJAN PAŽEK

We examine conditions under which a semicomputable set in a computable metric space is computable. We know that topology plays an important role in the description of such conditions. Some topological properties can force a semicomputable set to be computable or at least to have computable points. For example, if S is a semicomputable compact manifold with boundary such that ∂S is a semicomputable set, then S needs to be computable [4].

It is also known that a semicomputable continuum chainable from a to b, where a and b are computable points, must be computable [2, 5]. So the question under what conditions implication

$$\partial S \text{ semicomputable } \Longrightarrow S \text{ computable} \tag{1}$$

holds makes sense not just when S is a manifold (with boundary). Even when S is not a manifold, we can naturally consider certain subset of S as its boundary and ask whether (1) holds.

The following definition naturally arises. Let Δ and Σ be some topological spaces such that Σ is a subspace of Δ. We say that the topological pair (Δ, Σ) has **computable type** if for every computable metric space (X, d, α) and every embedding $f\colon \Delta \to X$ such that $f(\Delta)$ and $f(\Sigma)$ are semicomputable sets in (X, d, α) we have that $f(\Delta)$ is a computable set in (X, d, α).

So, if M is a compact manifold with boundary, then $(M, \partial M)$ has computable type; if K is a continuum chainable from a to b, then $(K, \{a, b\})$ has computable type.

It was proved recently in [6] that (D, W) has computable type, where D is the Warsaw disc and W is the Warsaw circle. The Warsaw circle W is defined by

$$W = (\{0\} \times [-2, 1]) \cup \{(x, \sin \frac{1}{x}) \mid 0 < x \leq 1\} \cup (\{1\} \times [-2, \sin 1]) \cup ([0, 1] \times \{-2\}),$$

see Figure 1. The Warsaw disc (see [8]) is the area of the plane bounded by the Warsaw circle (together with the Warsaw circle), see Figure 2.

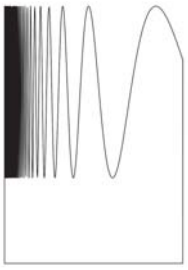


Figure 1.

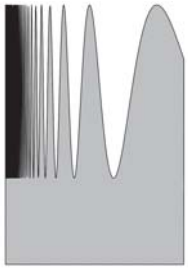


Figure 2.

Note that the Warsaw circle is not a manifold. Also, the Warsaw disc is not a manifold. However, these spaces "look like" a circle and a 2-cell respectively. The same can be said for spaces shown in Figures 3 and 4. It is naturally to ask the

following question: does (Δ, Σ) have computable type if Σ is the space shown in Figure 3 (the double Warsaw circle) and Δ is the space bounded by Σ? The same question can be asked for the spaces shown in Figure 4.

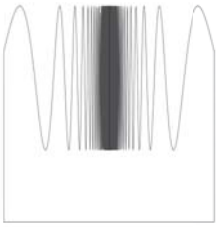


Figure 3.

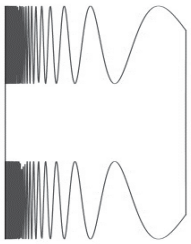


Figure 4.

As an answer to these questions, we have Theorem 1. First, we need the following notation.

Let $I := [0,1]$ and $I^2 := I \times I$. We set

$$S_1 := \{0\} \times I, \quad S_2 := I \times \{0\}, \quad S_3 := \{1\} \times I, \quad S_4 := I \times \{1\},$$
$$S = S_1 \cup S_2 \cup S_3 \cup S_4 \text{ and } \mathring{I}^2 := I^2 \setminus S.$$

If (X,d) is a metric space, $A \subseteq X$ and $\varepsilon > 0$, let $N_\varepsilon(A)$ denote the open ε-neighbourhood of A in (X,d). Let d_H denote the Hausdorff metric (on the set of all nonempty compact sets in (X,d)).

Theorem 1. *Suppose that Δ and Σ are compact subsets of $\mathbb{R}^2$ such that $\Sigma \subseteq \Delta$ and such that the following holds:*

(i) *there exist compact sets Σ_1, Σ_2, Σ_3 and Σ_4 such that*

$$\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \cup \Sigma_4, \quad \Sigma_1 \cap \Sigma_3 = \emptyset \quad \text{and} \quad \Sigma_2 \cap \Sigma_4 = \emptyset;$$

(ii) *for every $\varepsilon > 0$ there exists embedding $f\colon I^2 \to \mathbb{R}^2$ such that*

$$f(I^2) \subseteq \Delta, \quad \Sigma \cap f(\mathring{I}^2) = \emptyset,$$
$$\Delta \setminus f(I^2) \subseteq N_\varepsilon(f(S)) \quad \text{and} \quad d_H(f(S_i), \Sigma_i) < \varepsilon, \ \forall\, i \in \{1,2,3,4\}.$$

Then the topological pair (Δ, Σ) has computable type.

References

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

[2] Iljazović, Z.: Chainable and circularly chainable continua in computable metric spaces, *Journal of Universal Computer Science*, 15(6):1206–1235, 2009.

[3] Iljazović, Z.: Co-c.e. Spheres and Cells in Computable Metric Spaces, *Logical Methods in Computer Science*, Vol. 7(3:05):1–21, 2011.

[4] Iljazović, Z.: Compact manifolds with computable boundaries, *Logical Methods in Computer Science*, Vol. 9(4:19):1–22, 2013.

[5] Iljazović, Z., Pažek, B.: Computable intersection points, *Computability*, Vol. 7, no. 1, pp. 57-99, 2018.

[6] Iljazović, Z., Pažek, B.: Warsaw discs and semicomputability, *Topology and its Applications*, 239 (2018) 308–323.

[7] Miller, J.S., Effectiveness for Embedded Spheres and Balls, *Electronic Notes in Theoretical Computer Science*, volume 66, Elsevier, 2002, 127-138.

[8] Nadler, S.B.: *Continuum theory*, Marcel Dekker, Inc., New York, 1992.

Computability of semicomputable polyhedra

Lucija Validžić

University of Zagreb, Croatia

Semicomputable sets in computable metric spaces are a generalisation of sets of zero-points of computable functions $\mathbb{R}^n \to \mathbb{R}$. These sets need not be computable, not even in $\mathbb{R}^n$, but some topological properties of a set S can ensure that the implication

$$S \text{ semicomputable } \Rightarrow \ S \text{ computable} \tag{1}$$

holds. For example, if S is a compact manifold or a circularly chainable continuum which is not chainable, its semicomputability implies its computability. On the other hand, if S is a compact manifold with boundary, additional assumptions are needed for the same conclusion. In [4] it has been proved that a semicomputable compact manifold whose boundary is computable is necessarily computable. Motivated by this result, we focus on sets which have the topological type of a polyhedron and we investigate sufficient conditions under which the implication (1) holds. Topological 1-polyhedra are sets which are homeomorphic to a finite union of line segments in $\mathbb{R}^n$. We show that (1) is true if S is a topological 1-polyhedron whose boundary points (i.e. $x \in S$ which have a neighbourhood homeomorphic to $[0, 1)$, by a homeomorphism which maps 0 to x) are computable. When it comes to topological polyhedra of a higher dimension, we prove that (1) holds if S is a wedge of spheres. Moreover, we discuss boundary conditions which would lead to the same conclusion as in the case of 1-polyhedra.

References

[1] V. Brattka, *Plottable Real Number Functions and the Computable Graph Theorem*, SIAM Journal on Computing, Vol. 38, No. 1 : pp. 303-328, 2008.

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

[3] Z. Iljazović, L. Validžić, *Computable neighbourhoods of points in semicomputable manifolds,* Annals of Pure and Applied Logic, 168(4), pp. 840–859, 2017.

[4] Z. Iljazović, *Compact manifolds with computable boundaries*, Logical Methods in Computer Science 9(4:19), pp. 1–22, 2013.

[5] Z. Iljazović, L. Validžić, *Computability of a wedge of circles,* Rad Hrvatske akademije znanosti i umjetnosti: Matematičke znanosti (532:21), pp. 9–20, 2017.

[6] T. Kihara, *Incomputability of Simply Connected Planar Continua*, Computability 1(2), pp. 131–152, 2012.

[7] J.S. Miller, *Effectiveness for Embedded Spheres and Balls*, Electronic Notes in Theoretical Computer Science 66, pp. 127–138, 2002.

[8] M. Pour-El, I. Richards, *Computability in Analysis and Physics*, Springer-Verlag, Berlin-Heielberg-New York, 1989.

[9] E. Specker, *Der Satz vom Maximum in der rekursiven Analysis*, Constructivity in Mathematics (A. Heyting, ed.), North Holland Publ. Comp., Amsterdam, pp. 254–265, 1959.

[10] K. Weihrauch, *Computable Analysis*, Springer, Berlin, 2000.

Computability of Haar Averages

Dongseong Seon

KAIST, School of Computing

Definition 1. *For any measure space $(X, \mathcal{A}, \mu)$, average means the functional $C(X) \to \mathbb{R}$ s.t. maps f to $\int_X f d\mu$.*

It is known that the average over the real interval $[0, 1]$ is computable [1]. For averages over more general spaces, it is known that any probability measure can be represented by a probabilistic process; and if we receive it as an input, we can compute the average over topological space with respect to this measure [2]. However, Haar's theorem states that on any compact topological group, there exists the unique probability measure on the Borel subsets which is left-translation-invariant and regular. It implies that the measure with such conditions is represented implicitly by the space.

Definition 2. *For any compact topological group $(X, \circ)$, we call an average with the Haar probability measure as the Haar Average.*

In this paper, I prove that Haar Averages are computable under natural and mild assumption, which is more general than the average over the real interval $[0, 1]$ and different from accepting the measure as an input, by assuming that the space is a compact topological group. For the space hypothesis, [3] and [4] points out that the "width" property is important for compact metric sets and introduce notions called "separation bound" and "capacity". I followed these spirits and assumed that the sizes of maximal packings is computable where

Definition 3. *For any compact metric space (X, d) and its subset $T \subseteq X$,*

(1) T is called a n-packing if $x \in T, y \in T \Rightarrow x = y \vee d(x, y) > 2^{-n}$
(2) $\kappa_X : \mathbb{N} \to \mathbb{N}$ is called the sizes of maximal packings if

$$\kappa_X(n) := \max_{T \text{ is a } n\text{-packing}} |T|$$

(3) T is called a maximal n-packing if it is a n-packing and $|T| = \kappa_X(n)$.

Note that these definitions are very similar to [3] and thus natural, but slightly different. I also assumed the space is a computable metric space with bi-invariant metric, which means there is an encoding of some countable dense set and metric function, which takes this encodings as an input and outputs the distance, is computable.

Theorem 4. *Assume that the space $(X, d, \circ)$ is a compact topological group and a computable metric space with bi-invariant metric. Assume that the sizes of maximal packings is computable. Then the Haar Average functional is computable.*

I approximate the measure of sets using maximal packings, which is represented by the following lemma:

Definition 5. $\overline{B}_r(U) := \bigcup_{x \in U} \overline{B}_r(x)$, $\overline{B}_{-r}(U) := \{x \in U : \overline{B}_r(x) \subseteq U\}$

Lemma 6. *For measurable set $U \subseteq X, n \in \mathbb{N}$ and maximal n-packing T,*

$$\frac{\left|T \cap \overline{B}_{-2^{-n+1}}(U)\right|}{|T|} \leq \mu(U) \leq \frac{\left|T \cap \overline{B}_{2^{-n+1}}(U)\right|}{|T|}$$

But the above lemma is useless if we cannot approximate the number of points of T in $\overline{B}_{\pm 2^{-n+1}}(U)$. But closed balls have nice characteristics and one can make partitions from closed balls and approximate measures, by following lemmas:

Definition 7. *For any compact metric space (X, d) and its subset $T \subseteq X$,*

(1) We say a measurable set T is a point of continuity if $\lim_{r\to 0} \overline{B}_r(T)$ exists.
(2) We say r is a radius of continuity if r is a continuity point of the function that maps $r \in \mathbb{R}$ to $\mu(\overline{B}_r) \in \mathbb{R}$ (This is well-defined. See lemma 8).

Recall that a closed set S is called computable if for any $p \in X$, $d(S, p)$ is computable.

Lemma 8. *For any compact topological group $(X, d, \circ)$ with computable metric, if sizes of maximal packings is computable then*

(1) For any centers $p, q \in X, \mu(\overline{B}_r(p)) = \mu(\overline{B}_r(q))$
(2) The set of radiuses of continuity is dense in $\mathbb{R}$. Moreover, for any $a, b \in \mathbb{Q}$, we can effectively find some radius of continuity $r \in [a, b]$.
(3) Sets made with finite union, intersection, closure of complement of closed balls with computable center and radius are computable.
(4) Sets made with finite union, intersection, complement of points of continuity are also points of continuity.
(5) If measurable sets T and $\overline{T^c}$ are points of continuity and computable then $\mu(T)$ is computable.

Here's the overview of the proof: We compute some maximal n-packing T from its size. Next, compute radius of contiunity r big enough s.t. $X \subseteq \bigcup_{p\in T} \overline{B}_r(p)$. And then exclude overlapping parts from balls to make them disjoint, and form a partition. Since each fragment is formed by finite union, intersection and complement, we can compute its measure and multiply with the function value at each center. The total sum of these product approximates the Haar Average because we can control the error by the uniform continuity of the input function f.

Acknowledgement

Thanks for the financial support from National Research Foundation (NRF-2017R 1E1A1A03071032 and NRF-2016K1A3A7A03950702). Also thanks for my advisor Prof. Martin Ziegler for kind advice and supervision by suggesting the problem, showing related works and providing help for refining the structure of theorems and texts.

References

1. K. Ko, *Complexity theory of real functions.* Progress in theoretical computer science, Birkhäuser, 1991.
2. M. Schröder and A. Simpson, "Representing probability measures using probabilistic processes," *J. Complexity*, vol. 22, no. 6, pp. 768–782, 2006.
3. K. Weihrauch, "Computational complexity on computable metric spaces," *Math. Log. Q.*, vol. 49, no. 1, pp. 3–21, 2003.
4. A. Kawamura, F. Steinberg, and M. Ziegler, "Complexity theory of (functions on) compact metric spaces," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pp. 837–846, 2016.
5. K. Weihrauch, *Computable Analysis: An Introduction.* Texts in Theoretical Computer Science. An EATCS Series, Springer Berlin Heidelberg, 2000.

Never underestimate pigeons

Ludovic Patey

Institut Camille Jordan, Lyon, FRANCE
ludovic.patey@computability.fr,
WWW home page: http://ludovicpatey.com

The *infinite pigeonhole principle* for k-colors (RT^1_k) asserts, for every k-coloring of the integers, the existence of an infinite monochromatic set. This principle can be considered as a *problem*, whose instances are k-colorings of $\mathbb{N}$. A solution to an instance is an infinite monochromatic set. The analysis of a problem usually consists in proving lower bounds and upper bounds on the strength of the problem, in the former case by constructing an instance with only complex solutions, and in the latter case by showing that every instance admits a simple solution. The precise meaning of complex and simple depends on the chosen paradigm, eg, proof theory, computability theory, complexity theory.

At first, the pigeonhole principle seems trivial from a computational viewpoint, as for every computable k-coloring of the integers, one can find (although not uniformly) a computable infinite monochromatic set. However, the study of arbitrary instances of the pigeonhole principle happened to be surprisingly complicated, and was responsible for major advances in reverse mathematics, proof theory and computability theory. In particular, the pigeonhole principle can be seen as a bootstrap of Ramsey's theory, and its combinatorial analysis admits a translation into a computational analysis of Ramsey's theorem for pairs (RT^2_2). The pigeonhole principle is therefore at the heart of major open problems, such a the failure of the so-called *Big Five phenomenon* [4] in reverse mathematics with Seetapun's theorem [6], but also plays a central role in Liu's theorem [2], who proved that Ramsey's theorem for pairs does not imply weak König's lemma. The analysis of the first-order [5] and second-order [1] parts of the pigeonhole principle yielded new techniques which will likely be applicable in other areas of logics.

The most recent development was done by Monin and the author [3], who proved that pigeons do not jump high, and do not double jump high too/two, by introducing a new forcing notion with a fine control of the jump. Monin and the author obtained a similar result for every iteration of the jump, although the author could sadly not find a pun at every level of the hierarchy.

In this talk, we will survey the combinatorial and computational analysis of the pigeonhole principle, with a primary focus on reverse mathematics, and will introduce the major open problems related to it.

References

1. Cholak, P.A., Jockusch, C.G., Slaman, T.A.: On the strength of Ramsey's theorem for pairs. Journal of Symbolic Logic 66(01), 1–55 (2001)

2. Liu, L.: RT^2_2 does not imply WKL_0. Journal of Symbolic Logic 77(2), 609–620 (2012)
3. Monin, B., Patey, L.: Pigeons do not jump high (2018), to appear. Available at https://arxiv.org/abs/1803.09771
4. Montalbán, A.: Open questions in reverse mathematics. Bulletin of Symbolic Logic 17(03), 431–454 (2011)
5. Patey, L., Yokoyama, K.: The proof-theoretic strength of Ramsey s theorem for pairs and two colors (2015), submitted. Available at http://arxiv.org/abs/1601.00050.
6. Seetapun, D., Slaman, T.A.: On the strength of Ramsey's theorem. Notre Dame Journal of Formal Logic 36(4), 570–582 (1995)

NEARLY COMPUTABLE NUMBERS

PHILIP JANICKI

In this paper we introduce the concept of the nearly computably convergent sequences, which generalizes the well-known concept of the computably convergent sequences. 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. The set of the computable numbers is denoted by **EC**.

Definition 1.

(1) 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.
(2) A real number is called *nearly computable* if there is a computable sequence converging nearly computably to it.
(3) The set of all nearly computable numbers is denoted by **NC**.

It is not difficult to show that every computably convergent sequence is nearly computably convergent. So we have the implication that every computable number is nearly computable. On the other side, every nearly computable number is computably approximable by definition. We say that a real number is *computably approximable* if there is a computable sequence that converges to it. The set of the computably approximable numbers is denoted by **CA**. Not every computably approximable number is nearly computable. For example, for any computably enumerable set $A \subseteq \mathbb{N}$ that is not computable the number $x := \sum_{n \in A} 2^{-n}$ is computably approximable but not nearly computable. In fact, it is even a left-computable number. We say that a real number is *left-computable* if there is a computable increasing sequence converging to it. The set of the left-computable numbers is denoted by **LC**.

Theorem 2. *For a left-computable number $x \in [0; 1]$ the following are equivalent:*

(1) *There is a computable sequence converging nearly computably to x, that is $x \in \mathbf{NC}$.*
(2) *There is a computable increasing sequence converging nearly computably to x.*
(3) *Every computable increasing sequence converging to x converges nearly computably.*
(4) *For every computable increasing sequence $(x_n)_n$ converging to x the sequence $(x_{n+1} - x_n)_n$ converges computably to zero.*
(5) *Every computably enumerable prefix-free set $L \subseteq \{0, 1\}^*$ with $\sum_{w \in L} 2^{-|w|} = x$ is computable.*

The first four properties in Theorem 2 are equivalent even for arbitrary left-computable numbers. The equivalence of the fourth and the fifth property has been proved by Stephan and Wu in [5].

Definition 3. $\mathbf{LNC} := \mathbf{LC} \cap \mathbf{NC}$

Downey and LaForte have shown [2] that there is an incomputable left-computable number that satisfies the fifth condition of Theorem 2. Thus, $\mathbf{EC} \subsetneq \mathbf{LNC}$. If we look at the arithmetical properties of **NC**, we get to the following result.

Theorem 4. *The set* **NC** *is a real closed field.*

Since the set **LNC** is a subset of **LC** and has also incomputable members, it does not form a field, but there exists a smallest field containing all nearly computable left-computable numbers. In order to construct the arithmetical closure of **LNC**, we define the following set.

Definition 5. $\mathbf{DLNC} := \{\alpha - \beta \mid \alpha, \beta \in \mathbf{LNC}\}$

Theorem 6. *The set* **DLNC** *is a real closed field.*

This construction is similar to the one for the arithmetical closure of the left-computable numbers, which is the set of the weakly computable numbers introduced [1] by Ambos-Spies, Weihrauch and Zheng. A sequence $(x_n)_n$ is called *weakly computably convergent* if the series $\sum_{i=0}^{\infty} |x_{i+1} - x_i|$ converges. A real number is *weakly computable* if there is a computable sequence converging weakly computably to it. The set of the weakly computable numbers is denoted by **WC** and it is also a real closed field [3]. Furthermore [1], a real number x is weakly computable if and only if there exist left-computable numbers α and β with $x = \alpha - \beta$.

Definition 7. $\mathbf{WNC} := \mathbf{WC} \cap \mathbf{NC}$

Since **WC** and **NC** are real closed fields, the set **WNC** is a real closed field, too. Obviously **DLNC** is a subset of **WNC**, but the converse is not true. The proof of this theorem is an infinite injury priority construction.

Theorem 8. *There is an* $x \in \mathbf{WNC} \setminus \mathbf{DLNC}$.

In order to prove that **WNC** is a proper subset of **NC**, we will use the definition of the divergence bounded computable numbers that have been introduced [4] by Rettinger, Zheng, Gengler and von Braunmühl. For a function $f : \mathbb{N} \to \mathbb{N}$ we define [6] that a sequence $(x_n)_n$ *converges* f*-effectively* if for every $n \in \mathbb{N}$ there exist at most $f(n)$ non-overlapping pairs of indices $(i, j) \in \mathbb{N}^2$ with $i, j \geq n$ and $|x_i - x_j| \geq 2^{-n}$. Two pairs of indices $(i_1, j_1), (i_2, j_2) \in \mathbb{N}^2$ are *non-overlapping* if $i_1 < j_1 \leq i_2 < j_2$ or $i_2 < j_2 \leq i_1 < j_1$ holds. A real number is called f*-effectively computable* if there is a computable sequence converging f-effectively to it. It is called *divergence bounded computable* if it is f-effectively computable for a computable function $f : \mathbb{N} \to \mathbb{N}$. The set of all divergence bounded computable numbers is denoted by **DBC**. It has been proven [4] that **WC** is a proper subset of **DBC**.

Theorem 9. *There is an* $x \in \mathbf{NC} \setminus \mathbf{DBC}$.

The proof of this theorem is an infinite injury priority construction similar to the one used in [2]. Finally, we have the following hierarchy:

$$\mathbf{EC} \subsetneq \mathbf{LNC} \subsetneq \mathbf{DLNC} \subsetneq \mathbf{WNC} \subsetneq \mathbf{NC} \subsetneq \mathbf{CA}$$

References

[1] K. Ambos-Spies, K. Weihrauch, and X. Zheng. Weakly computable real numbers. *J. Complexity*, 16(4):676–690, 2000.

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

[3] A. Raichev. Relative randomness and real closed fields. *Electr. Notes Theor. Comput. Sci.*, 120:135–143, 2005.

[4] R. Rettinger, X. Zheng, R. Gengler, and B. von Braunmühl. Weakly computable real numbers and total computable real functions. In J. Wang, editor, *Computing and Combinatorics*, pages 586–595, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[5] 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.

[6] X. Zheng. Classification of computably approximable real numbers. *Theory of Computing Systems*, 43(3):603–624, Dec 2008.

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

Computable metrics above the standard real metric

R. A. Kornev

We study the notion of computable categoricity for the space of real numbers viewed as a metric space with the rationals as a dense subset. In other words, we try to construct computably inequivalent (in the sense of TTE [1]) computable metrics on the real line. In this setting, the following notions of computable reducibility of metrics are considered. Let (X, τ) be a Polish space with a dense subset W, let ρ and ρ' be complete metrics on X that induce topology τ. Say that ρ is computably reducible to ρ' ($\rho \leq_c \rho'$) if δ_ρ is computably reducible to $\delta_{\rho'}$, where δ_ρ and $\delta_{\rho'}$ are respective Cauchy representations of spaces (X, ρ, W) and (X, ρ', W). Furthermore, ρ is weakly reducible to ρ' ($\rho \leq_{ch} \rho'$) if there is a $(\delta_\rho, \delta_{\rho'})$-computable autohomeomorphism of X.

In [2] it was proved that there exists an infinite sequence of computable metrics on $\mathbb{R}$ below the standard metric $\rho_{\mathbb{R}}$ in the ordering $\leq_{ch}$ (in particular, $\mathbb{R}$ has computable dimension ω). In the present talk we show that a similar construction can be carried out to obtain metrics above $\rho_{\mathbb{R}}$, thus $\rho_{\mathbb{R}}$ is not maximal among all computable metrics. More precisely, the following result is established.

Theorem. *$\omega^{<\omega}$ is embeddable into the ordering $\leq_{ch}$ of computable metrics above $\rho_{\mathbb{R}}$; that is, there exists a sequence of computable metrics $(\rho_i)_{i\in\omega}$ on $\mathbb{R}$ such that $\rho_{\mathbb{R}} \leq_c \rho_i$ for all $i \in \omega$ and the ordering $((\rho_i)_{i\in\omega}, \leq_{ch})$ is isomorphic to $\omega^{<\omega}$.*

References

[1] K. Weihrauch, *Computable Analysis. An Introduction,* Springer-Verlag, Berlin/Heidelberg, 2000.

[2] R. Kornev, *Reducibility of computable metrics on the real line,* Algebra and Logic, **56** (4): 302–317, 2017.

EFFECTIVE COMPACTNESS AND UNIQUENESS OF MAXIMAL COMPUTABILITY STRUCTURES

KONRAD BURNIK AND ZVONKO ILJAZOVIĆ

A computability structure $\mathcal{S}$ on a metric space (X, d) is a set of sequences in X such that the following two properties hold:

(1) if $(x_i), (y_j) \in \mathcal{S}$, then the distances $d(x_i, y_j)$ can be effectively computed;
(2) if $(x_i) \in \mathcal{S}$ and (y_j) is a sequence in X which can be computed from (x_i), then $(y_j) \in \mathcal{S}$.

For example, if (X, d, α) is a computable metric space, then the set $\mathcal{S}_\alpha$ of all computable sequences in this space is a computability structure on (X, d). Computability structures of this form are called separable.

A computability structure $\mathcal{S}$ on (X, d) is called maximal if there is no computability structure $\mathcal{T}$ on (X, d) such that $\mathcal{S} \subseteq \mathcal{T}$ and $\mathcal{S} \neq \mathcal{T}$. Each separable computability structure is maximal, but a maximal computability structure need not be separable. For example, there is a unique separable computability structure on $[0, 1]$, but there are infinitely (uncountably) many maximal computability structures on $[0, 1]$.

If $\mathcal{S}$ is a computability structure on (X, d) and $x \in X$ such that $(x, x, x, \dots) \in \mathcal{S}$, then we say that x is a computable point in $\mathcal{S}$.

If (X, d) is a metric space and $a_0, \dots, a_n \in X$ such that $d(a_i, a_j)$ is a computable number for all $i, j \in \{0, \dots, n\}$, then we say that $a_0, \dots, a_n$ is an effective sequence in (X, d). If $a_0, \dots, a_n$ is an effective sequence in (X, d), then there exists a maximal computability structure $\mathcal{M}$ on (X, d) in which the points $a_0, \dots, a_n$ are computable. However, such a maximal computability structure need not be unique. The general question is: under what conditions such a maximal computability structure is unique. It is known that in the case of a subspace of Euclidean space geometric independence of points $a_0, \dots, a_n$ is important in the description of such conditions (that points $a_0, \dots, a_n$ are geometrically independent means that the vectors $a_1 - a_0, \dots, a_n - a_0$ are linearly independent). In particular, if $a_0, \dots, a_n$ is a geometrically independent effective sequence in $\mathbb{R}^n$ (where we consider the Euclidean metric on $\mathbb{R}^n$), then there exists a unique maximal computability structure on $\mathbb{R}^n$ in which these points are computable.

In a general metric space the notion of geometric independence of points does not make sense. The following observation gives an idea on how to proceed in the general case: if $a_0, \dots, a_n$ are geometrically independent points in $\mathbb{R}^n$, then for each $x \in \mathbb{R}^n$ the distances $d(x, a_0)$, $\dots$, $d(x, a_n)$ uniquely determine x. So we have the following natural definition.

Suppose (X, d) is a metric space, $n \in \mathbb{N}$ and $a_0, \dots, a_n$ is a finite sequence of points in X such that for all $x, y \in X$ the following implication holds:

$$\text{if } d(a_i, x) = d(a_i, y) \text{ for each } i \in \{0, \dots, n\}, \text{ then } x = y.$$

Then we say that $a_0, \dots, a_n$ is a **nice sequence** in (X, d).

Now the question is: if the finite sequence $a_0, \dots, a_n$ is nice and effective in (X, d), is a maximal computability structure $\mathcal{M}$ on (X, d) in which the given points are computable unique? It is easy to show that the answer is negative in general. However, under certain additional assumptions, we prove that $\mathcal{M}$ is unique.

A metric space (X, d) is said to be effectively compact if there exist an effective separating sequence α in (X, d) and a computable function $f : \mathbb{N} \to \mathbb{N}$ such that

$$X = B(\alpha_0, 2^{-k}) \cup \cdots \cup B(\alpha_{f(k)}, 2^{-k})$$

for each $k \in \mathbb{N}$. It is known that if (X, d) is effectively compact, then for each effective separating sequence α in (X, d) there exists such a computable function f.

Theorem 1. *Let (X, d) be an effectively compact metric space. Suppose $a_0, \ldots, a_n$ is a nice sequence in (X, d) and suppose that there exists a separable computability structure $\mathcal{S}$ on (X, d) in which $a_0, \ldots, a_n$ are computable points. Then $\mathcal{S}$ is a unique maximal computability structure on (X, d) in which $a_0, \ldots, a_n$ are computable points.*

We also examine certain subspaces of $\mathbb{R}^n$, where we consider metrics different from the Euclidean metric, for example, the supremum (maximum) metric, and we find conditions under which maximal computability structures on these spaces are unique.

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-Heielberg-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.

Co-Polish spaces in Computable Analysis

Matthias Schröder

March 29, 2018

We present and investigate the class of Co-Polish spaces. Co-Polish spaces play a big role in Type Two Complexity Theory. The name originates from the characterisation of a Co-Polish space X to be a regular sequential space for which the space $\mathsf{C}(\mathsf{X},\mathbb{R})$ of continuous real-valued functions on X equipped with the compact-open topology is a Polish space (i.e., separable and completely metrisable).

The pivotal property which exhibits Co-Polish spaces as suitable for parametrized complexity theory is their characterisation as an inductive limit of an increasing sequence of compact metrisable spaces. This fact allows the measurement of time complexity of functions between Co-Polish spaces in terms of two *discrete* parameters: on the one hand in terms of the output precision, as it is typical for complexity theory on the real numbers, and on the other hand, like in classical complexity theory, in terms of a discrete parameter on the input, namely the index of a compact metric subspace in which the input x lies.

As their major advantage, Co-Polish spaces have the property that one can assign to each element a natural number as their size (length), like in discrete complexity theory. An upper bound of this size can be computed from any name of x. By contrast, for treating general spaces occuring in Computable Analysis (like the function space $\mathbb{R}^{\mathbb{R}}$) one has to resort to defining the size (length) of an input as an element of the Baire space, as in A. Kawamura's and S. Cook's approach to complexity in Computable Analysis.

Examples of Co-Polish spaces are all locally compact separable metrisable spaces (e.g. the Euclidean space $\mathbb{R}$), Silva spaces known from the theory of locally convex vector spaces (i.e. the space of distributions with compact support) and, more generally, vector space duals of separable Banach spaces formed in the category of sequentially locally convex qcb-spaces.

A note on the descriptive complexity of the upper and double powerspaces

Matthew de Brecht*

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

Given a topological space X, $\mathbf{O}(X)$ denotes the lattice of open subsets of X with the Scott-topology, $\mathbf{K}(X)$ denotes the space of saturated compact subsets of X with the upper-Vietoris topology, and $\mathbf{A}(X)$ denotes the space of closed subsets of X with the lower-Vietoris topology (see [5]). If X is countably based and sober, then $\mathbf{A}(X)$, $\mathbf{K}(X)$, and $\mathbf{O}(\mathbf{O}(X))$ are all countably based and sober. These and other powerspaces frequently appear in computable analysis and the theory of represented spaces [1, 9, 8, 7, 4]. Our main result is about the complexity of $\mathbf{K}(X)$ and $\mathbf{O}(\mathbf{O}(X))$ when X is a countably based co-analytic sober space.

A subspace $S \subseteq X$ of a quasi-Polish space X is *analytic* (or $\mathbf{\Sigma}^1_1$) if there exists a continuous function $f\colon \mathbb{N}^{\mathbb{N}} \to X$ such that the range of f is equal to S (see [2]). A subspace of a quasi-Polish space is *co-analytic* (or $\mathbf{\Pi}^1_1$) if its complement is analytic. A countably based space is analytic (co-analytic) if it is homeomorphic to an analytic (co-analytic) subspace of a quasi-Polish space.

A space is a *Baire space* if the intersection of any countable sequence of dense open subsets is dense. A space is *completely Baire* if each of its closed subspaces is a Baire space. If X is a countably based completely Baire space, then so is every $\mathbf{\Pi}^0_2$-subspace of X ([3], Theorem 4.1).

Our main result is the following:

Theorem 1. *The following are equivalent for every countably based co-analytic sober space X:*

1. $\mathbf{K}(X)$ *is analytic,*
2. $\mathbf{K}(X)$ *is completely Baire,*
3. $\mathbf{O}(\mathbf{O}(X))$ *is analytic,*
4. $\mathbf{O}(\mathbf{O}(X))$ *is completely Baire,*
5. X *is quasi-Polish.*

□

It is well-known that a similar result (restricted to metrizable spaces) holds for the Vietoris powerspace (see Exercise 33.5 in [6]). Note that the Vietoris powerspace has a strictly finer topology than the powerspace $\mathbf{K}(X)$ defined here.

The proof of Theorem 1 will easily follow from previous results and Lemma 1 below. In the following, we will write $X \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(Y)$ to mean that X is homeomorphic to a $\mathbf{\Pi}^0_2$-subspace of Y. We also recall the definition of the countable space S_0 from [3]. The underlying set of S_0 is $\mathbb{N}^{<\mathbb{N}}$ (all finite strings of natural numbers), and a subbasis for the *closed* subsets of S_0 is given by sets of the form $\{\tau \in \mathbb{N}^{<\mathbb{N}} | \sigma \preceq \tau\}$, where $\sigma \in \mathbb{N}^{<\mathbb{N}}$ and $\preceq$ is the prefix relation. Note that the specialization order of S_0 is the inverse of the prefix relation.

Lemma 1. *If $S \subseteq \mathbb{N}^{\mathbb{N}}$ is co-analytic then $S \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{K}(S_0))$.*

Proof. For $\sigma \in \mathbb{N}^{<\mathbb{N}}$, let $\langle\sigma\rangle_0$ be the substring of σ of elements with even indices, and let $\langle\sigma\rangle_1$ be the odd elements. We write $\sigma \diamond \tau$ for the concatenation of σ and τ, and $|\sigma|$ for the length of σ. Our notation will treat $\mathbb{N}$ and $\mathbb{N}^2$ as subspaces of $\mathbb{N}^{<\mathbb{N}}$. Fix an enumeration $\{\tau_n\}_{n\in\mathbb{N}}$ of $\mathbb{N}^{<\mathbb{N}}$.

Sets of the form $\{K \in \mathbf{K}(S_0) \mid \tau_n \notin K\}$ (for $n \in \mathbb{N}$) form a subbase for the *open* subsets of $\mathbf{K}(S_0)$. It is easy to see that the elements of $\mathbf{K}(S_0)$ are precisely the well founded trees on $\mathbb{N}$, which is the underlying set of a standard example of a $\mathbf{\Pi}^1_1$-complete set [6] (note that the topology on $\mathbf{K}(S_0)$ is strictly weaker than the more standard zero-dimensional topology used in [6]).

We use the notation $\uparrow\sigma = \{x \in \mathbb{N}^{\mathbb{N}} | \sigma \preceq x\}$ for basic clopen subsets of $\mathbb{N}^{\mathbb{N}}$. Let $U \subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}}$ be open such that $S = \{x \in \mathbb{N}^{\mathbb{N}} \mid (\forall y \in \mathbb{N}^{\mathbb{N}})\langle x, y\rangle \in U\}$. For $x \in \mathbb{N}^{\mathbb{N}}$ define

$$\alpha(x) = \{0 \diamond \sigma \in \mathbb{N}^{<\mathbb{N}} \mid \langle\sigma\rangle_0 \preceq x \mathbin{\&} \uparrow\langle\sigma\rangle_0 \times \uparrow\langle\sigma\rangle_1 \not\subseteq U]\}$$
$$\beta(x) = \{1 \diamond n \in \mathbb{N}^2 \mid \tau_n \preceq x\} \cup \{2 \diamond n \in \mathbb{N}^2 \mid \tau_n \not\preceq x\}$$

* This work was supported by JSPS Core-to-Core Program, A. Advanced Research Networks and by JSPS KAKENHI Grant Number 18K11166.

Define $\gamma\colon S \to \mathbf{K}(S_0)$ so that $\gamma(x)$ is the saturation of $\alpha(x) \cup \beta(x)$. It is easy to see that γ is continuous and well-defined (the choice of U guarantees that $\alpha(x)$ does not contain an infinite branch if and only if $x \in S$). Furthermore, $\tau_n \preceq x$ if and only if $2 \diamond n \notin \gamma(x)$, hence γ is an embedding.

Let G be the set of all $K \in \mathbf{K}(S_0)$ such that the following hold ($n, m \in \mathbb{N}$ and $\sigma \in \mathbb{N}^{<\mathbb{N}}$):

1. $(\forall n)[1 \diamond n \in K \iff 2 \diamond n \notin K]$
2. $(\forall m, n$ satisfying $\tau_n \preceq \tau_m)[1 \diamond m \in K \Rightarrow 1 \diamond n \in K]$
3. $(\forall m, n$ satisfying $\tau_n \not\preceq \tau_m)[1 \diamond m \in K \Rightarrow 1 \diamond n \notin K]$
4. $(\forall m)(\exists n)[m \leq |\tau_n| \,\&\, 2 \diamond n \notin K]$
5. $(\forall \sigma$ satisfying $|\sigma| > 1)[1 \diamond \sigma \notin K \,\&\, 2 \diamond \sigma \notin K]$
6. $(\forall n > 2)\, n \notin K$
7. $(\forall \sigma, n$ satisfying $\tau_n = \langle\sigma\rangle_0)[0 \diamond \sigma \in K \iff (1 \diamond n \in K \,\&\, \uparrow\langle\sigma\rangle_0 \times \uparrow\langle\sigma\rangle_1 \not\subseteq U)]$

Each of the conditions above correspond to a countable intersection of finite boolean combinations of open subsets of $\mathbf{K}(S_0)$, hence G is a $\mathbf{\Pi}^0_2$-subspace of $\mathbf{K}(S_0)$. It is easy to verify that $\gamma(x) \in G$ for each $x \in S$.

Fix $K \in G$ and let $P = \{\tau_n \in \mathbb{N}^{<\mathbb{N}} \mid 1 \diamond n \in K\}$. The second and third conditions guarantee that P is closed under prefixes and linearly ordered by $\preceq$, and the first and fourth conditions guarantee that the lengths of the strings in P are unbounded. Hence there is a unique $x \in \mathbb{N}^{\mathbb{N}}$ such that $\tau_n \preceq x \iff \tau_n \in P \iff 1 \diamond n \in K$. Condition one now implies $2 \diamond n \in K \iff \tau_n \not\preceq x$, and using the fifth condition we have that the strings in K starting with 1 or 2 are exactly the strings in $\beta(x)$. The remaining non-empty strings in K must begin with 0 because of the sixth condition, and the seventh condition guarantees that these remaining strings are precisely the elements of $\alpha(x)$. The compactness of K implies $\alpha(x)$ has no infinite branch, hence $x \in S$ and $K = \gamma(x)$. Therefore, γ is a homeomorphism between S and the $\mathbf{\Pi}^0_2$-subspace G of $\mathbf{K}(S_0)$. □

Using Lemma 1, we can now prove Theorem 1 as follows. First note that if X is quasi-Polish, then $\mathbf{K}(X)$ and $\mathbf{O}(\mathbf{O}(X))$ are quasi-Polish (see [4, 5]), so items (1) through (4) all follow from (5). So assume that X is a countably based co-analytic sober space which is *not* quasi-Polish. Then X contains a $\mathbf{\Pi}^0_2$-subspace S which is homeomorphic to either $\mathbb{Q}$ or S_0 (see [3]). It was shown in ([4], Corollary 5.5) that $\mathbf{K}(\mathbb{Q})$ is not analytic, and Lemma 1 implies that $\mathbf{K}(S_0)$ is not analytic, hence $\mathbf{K}(S)$ is not analytic. Furthermore, $\mathbf{K}(S)$ is not completely Baire because $\mathbb{Q} \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{K}(\mathbb{Q}))$ ([5], Proposition 8) and $\mathbb{Q} \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{K}(S_0))$ (Lemma 1 above).

We have $\mathbf{K}(S) \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{K}(X))$ by ([4], Theorem 5.3). Sobriety of X implies $X \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{A}(X))$ ([5], Proposition 3), hence $S \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{A}(X))$, which implies $\mathbf{K}(S) \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{K}(\mathbf{A}(X)))$. Since $\mathbf{K}(\mathbf{A}(X))$ and $\mathbf{O}(\mathbf{O}(X))$ are homeomorphic ([5], Theorem 22) we have $\mathbf{K}(S) \mathrel{\widetilde{\in}} \mathbf{\Pi}^0_2(\mathbf{O}(\mathbf{O}(X)))$. Thus $\mathbf{K}(X)$ and $\mathbf{O}(\mathbf{O}(X))$ both have a $\mathbf{\Pi}^0_2$-subspace which is neither analytic nor completely Baire. Both of these properties are hereditary under $\mathbf{\Pi}^0_2$-subspaces, hence items (1) through (4) do not hold for X. This completes the proof of Theorem 1.

It would be interesting to see if an effective (light-faced) version of Lemma 1 holds, and also to have a full characterization of the $\mathbf{\Pi}^0_2$-subspaces of $\mathbf{K}(\mathbb{Q})$ and $\mathbf{K}(S_0)$. We also have the following question.

Question 1. If X is a QCB_0-space and $\mathbf{O}(X)$ is a countably based analytic space, then is $\mathbf{O}(X)$ quasi-Polish?

The author has learned from M. Schröder that his previous results ([10], Theorem 7.3) imply that the above question has a positive answer if X is Hausdorff, even when the "analytic" assumption is removed.

References

1. V. Brattka and K. Weihrauch. Computability on subsets of Euclidean space I: closed and compact subsets. *Theoretical Computer Science*, 219 (1999), 65–93.
2. M. de Brecht. Quasi-Polish spaces. *Annals of Pure and Applied Logic*, 164 (2013), 356–381.
3. M. de Brecht. A generalization of a theorem of Hurewicz for quasi-Polish spaces. *Logical Methods in Computer Science*, 14 (2018), 1–18.
4. M. de Brecht, M. Schröder, and V. Selivanov. Base-complexity classification of qcb_0-spaces. *Computability*, 5-1 (2016), 75–102.
5. M. de Brecht and T. Kawai. On the commutativity of the powerspace constructions. arXiv:1709.06226v1
6. A.S. Kechris. *Classical Descriptive Set Theory*. Springer, New York (1995).
7. E. Neumann. Continuous enclosures and best approximations of discontinuous operations. *Proceedings of the Twelfth International Conference on Computability and Complexity in Analysis*. (2015) 47–49.
8. A. Pauly. On the topological aspects of the theory of represented spaces. *Computability*, 5-2 (2016), 159–180.
9. M. Schröder. Admissible representations for continuous computations. Phd thesis, FernUniversität Hagen (2003).
10. M. Schröder. Spaces allowing Type-2 Complexity Theory revisited. *Math. Log. Quart.*, 50 (2004), 443–459.

Trees Describing Topological Weihrauch Degrees of Multivalued Functions

Peter Hertling
Fakultät für Informatik, Universität der Bundeswehr München,
85579 Neubiberg, Germany
Email: peter.hertling@unibw.de

In his PhD thesis [1, 2, 3] the author has considered functions defined on a subset of the Baire space with finite discrete range (one may call them partial k-partitions if k is an upper bound for the cardinality of the range) and has described the finite initial segments of the three hierarchies of such functions under

1. Wadge reducibility,
2. continuous strong Weihrauch reducibility, and
3. continuous Weihrauch reducibility

For the description he used certain trees that describe the discontinuities of such functions and three suitable reducibility relations between forests consisting of such trees. In addition he described the initial segment of the continuous Weihrauch degrees of functions defined on the Baire space with discrete range (not necessarily finite) by suitable trees and forests and a suitable reducibility relation on forests. The result for Wadge reducibility of partial k-partitions was extended by Selivanov first to all Δ^0_2-measurable k-partitions [6], later to all Δ^0_3-measurable k-partitions [7], and recently by Kihara and Montalbán to all Borel measurable functions $f : \mathbb{N}^{\mathbb{N}} \to B$ where B is any better-quasi-order [4].

We suggest definitions of continuous strong Weihrauch reducibility and of continuous Weihrauch reducibility on the set of functions $f :\subseteq \mathbb{N}^{\mathbb{N}} \to B$ where B is any quasi-order. Then we present descriptions of the corresponding topological strong Weihrauch degrees and of the topological Weihrauch degrees of Δ^0_2 measurable functions $f :\subseteq \mathbb{N}^{\mathbb{N}} \to B$, where B is any better-quasi-order, by suitable trees and forests and suitable reducibility relations on forests. We also consider Wadge degrees. Furthermore, we show that this leads to a similar description of the Wadge degrees, the topological strong Weihrauch degrees and the topological Weihrauch degrees of Δ^0_2 measurable multivalued functions $R :\subseteq X \to Y$ where X is any countably based T_0 space and Y is any finite discrete space (these degrees are defined in the usual sense via admissible representations [5] of countably based T_0 spaces and by continuous reduction

functions on the Baire space; in complete analogy to the nowadays more common strong Weihrauch degrees and Weihrauch degrees defined by computable reduction functions on the Baire space). At present we are working on extending this to Borel measurable functions with range in a better-quasi-order and to Borel measurable multivalued functions with finite discrete range, following [7, 4].

References

[1] P. Hertling. A topological complexity hierarchy of functions with finite range. Technical Report 223, Centre de recerca matematica, Institut d'estudis catalans, Barcelona, October 1993. Workshop on Continuous Algorithms and Complexity.

[2] P. Hertling. Topologische Komplexitätsgrade von Funktionen mit endlichem Bild. Technical Report 152, FernUniversität Hagen, December 1993.

[3] P. Hertling. *Unstetigkeitsgrade von Funktionen in der effektiven Analysis*. PhD thesis, Fachbereich Informatik, FernUniversität Hagen, 1996.

[4] T. Kihara and A. Montalbán. On the structure of the Wadge degrees of BQO-valued Borel functions. arXiv:1705.07802, 2017.

[5] C. Kreitz and K. Weihrauch. Theory of representations. *Theoretical Computer Science*, 38:35–53, 1985.

[6] V. L. Selivanov. Hierarchies of Δ^0_2–measurable k–partitions. *Mathematical Logic Quarterly*, 53(4–5):446–461, 2007.

[7] V. L. Selivanov. Extending Wadge theory to k-partitions. In J. Kari, F. Manea, and I. Petre, editors, *Unveiling Dynamics and Complexity*, volume 10307 of *Lecture Notes in Computer Science*, pages 387–399, Cham, 2017. Springer. 13th Conference on Computability in Europe, CiE 2017, Turku, Finland, June 12-16, 2017.

The Wadge-like Degrees of Borel Functions

V.L. Selivanov
A.P. Ershov Institute of Informatics Systems SB RAS
and Kazan Federal University

For subsets A, B of the Baire space $\mathcal{N} = \omega^\omega$, A is *Wadge reducible* to B ($A \leq_W B$), if $A = f^{-1}(B)$ for some continuous function f on $\mathcal{N}$. The quotient-poset of the preorder $(P(\mathcal{N}); \leq_W)$ under the induced equivalence relation $\equiv_W$ on the power-set of $\mathcal{N}$ is called *the structure of Wadge degrees* in $\mathcal{N}$. W. Wadge [8] characterised the structure of Wadge degrees of Borel sets (i.e., the quotient-poset of $(\mathbf{\Delta}^1_1(\mathcal{N}); \leq_W)$) up to isomorphism. In particular this quotient-poset is well-founded and has no 3 pairwise incomparable elements.

There are several reasons and several ways to generalise the Wadge theory. For example, one can consider other natural classes of reducing functions in place of the continuous functions. For any pointclass $\mathbf{\Gamma} \subseteq P(\mathcal{N})$, by $\mathbf{\Gamma}$-*function* we mean a function f on $\mathcal{N}$ such that $f^{-1}(A) \in \mathbf{\Gamma}$ for each $A \in \mathbf{\Gamma}$. Since the $\mathbf{\Gamma}$-functions are closed under composition and contain the identity function, we obtain the corresponding $\mathbf{\Gamma}$-reducibility $\leq_{\mathbf{\Gamma}}$. Among such reducibilities are $\mathbf{\Delta}^0_\alpha$-reducibilities, for each non-zero countable ordinal α. Note that $\mathbf{\Delta}^0_1$-reducibility coincides with the Wadge reducibility. We shorten the notation $\leq_{\mathbf{\Delta}^0_\alpha}$ to $\leq_\alpha$, so $\leq_1$ coincides with $\leq_W$. The $\mathbf{\Delta}^0_\alpha$-functions and $\mathbf{\Delta}^0_\alpha$-reducibilities (among the much larger class of so called Borel amenable reducibilities) were comprehensively investigated by L. Motto-Ros (see [4] and references therein). In particular, the quotient-posets of $(\mathbf{\Delta}^1_1(\mathcal{N}); \leq_\alpha)$ and $(\mathbf{\Delta}^1_1(\mathcal{N}); \leq_W)$ are isomorphic (in symbols, $(\mathbf{\Delta}^1_1(\mathcal{N}); \leq_\alpha) \simeq (\mathbf{\Delta}^1_1(\mathcal{N}); \leq_W)$).

Another extension of the Wadge theory is the extension from the case of subsets of $\mathcal{N}$ to the case of functions $A : \mathcal{N} \to Q$ to an arbitrary quasi-order Q. For such functions A, B, let $A \leq_W B$ mean that there is a continuous function f on $\mathcal{N}$ such that $A(x) \leq_Q B(f(x))$ for each $x \in \mathcal{N}$. The case of sets corresponds to the case when Q consists of two incomparable elements. Let $\mathcal{W}_Q = (\mathbf{\Delta}^1_1(Q^{\mathcal{N}}); \leq_W)$ where $\mathbf{\Delta}^1_1(Q^{\mathcal{N}})$ is the set of functions A such that $A^{-1}(q) \in \mathbf{\Delta}^1_1(\mathcal{N})$ for each $q \in Q$. A theorem of van Engelen, Miller and Steel (see Theorem 3.2 in [1]) states that if Q is a countable better quasi-order then $\mathcal{W}_Q$ is a better quasi-order. Although this theorem gives an important information about the quotient-poset of $\mathcal{W}_Q$, it is far from a characterisation.

Many efforts (see e.g. [2, 5, 6, 7] and references therein) to characterise the quotient-poset of $\mathcal{W}_Q$ were devoted to the particular case of k-*partitions of* $\mathcal{N}$ where Q is the anti-chain with k elements, for each $2 \leq k < \omega$. Our approach in [5, 6, 7] to this problem was to characterise the initial segments $(\mathbf{\Delta}^0_\alpha(k^{\mathcal{N}}); \leq_W)$ for bigger and bigger ordinals $2 \leq \alpha < \omega_1$. To achieve this, we defined the structures of iterated labeled forests with the so called homomorphism quasi-order and discovered useful properties of some natural operations on the iterated labeled forests.

An important progress was recently achieved in [3] where a full characterisation of the quotient-poset of $\mathcal{W}_Q$ for arbitrary countable better quasi-order Q is obtained, with a heavy use of the (suitably extended) iterated labeled forests and of methods of classical computability theory.

In this work, we unite the above-mentioned extensions of the Wadge theory by characterising the quotient-posets of $(\mathbf{\Delta}^1_1(Q^{\mathcal{N}}); \leq_\alpha)$, of their variations for several other Borel amenable reducibilities (which are extended to $Q^{\mathcal{N}}$ in the obvious way), and natural initial segments of such quotient-posets. For quasi-orders P and Q, let $P \simeq Q$ mean that the quotient-posets of P and Q are isomorphic. A typical result (extending the above-mentioned result of L. Motto Ros) may be formulated as follows:

Theorem 1 *For any countable better quasi-order Q and any non-zero countable ordinal α, $(\mathbf{\Delta}^1_1(Q^{\mathcal{N}}); \leq_\alpha) \simeq (\mathbf{\Delta}^1_1(Q^{\mathcal{N}}); \leq_W)$.*

We deduce this theorem from the following fact which follows by induction from the results (mentioned) in [3]: for all $\alpha, \beta, \gamma < \omega_1$ with $\alpha, \beta > 0$ we have $(\mathbf{\Delta}^0_{\alpha+\gamma}; \leq_\alpha) \simeq (\mathbf{\Delta}^0_{\beta+\gamma}; \leq_\beta)$. Among particular cases and variations of this result we mention the following: $(\mathbf{\Delta}^0_\omega(Q^{\mathcal{N}}); \leq_n) \simeq (\mathbf{\Delta}^0_\omega(Q^{\mathcal{N}}); \leq_W)$ for each $2 \leq n < \omega$; $(\bigcup_{k<\omega} \mathbf{\Delta}^0_k(Q^{\mathcal{N}}); \leq_n) \simeq (\bigcup_{k<\omega} \mathbf{\Delta}^0_k(Q^{\mathcal{N}}); \leq_W)$ for each $2 \leq n < \omega$; $(\mathbf{\Delta}^0_7(Q^{\mathcal{N}}); \leq_3) \simeq (\mathbf{\Delta}^0_5(Q^{\mathcal{N}}); \leq_W)$.

References

[1] van Engelen, F., Miller, A., Steel, J.: Rigid Borel sets and better quasiorder theory. Contemporary mathematics, 65 (1987), 199–222.

[2] Hertling, P.: Topologische Komplexitätsgrade von Funktionen mit endlichem Bild. Informatik-Berichte 152, Fernuniversität Hagen, 1993.

[3] Kihara, T., Montalban, A.: On the structure of the Wadge degrees of BQO-valued Borel functions. ArXiv: 1705.07802 v1 [Math.LO] 22 May 2017.

[4] Motto Ros, L.: Borel-amenable reducibilities for sets of reals, J. Symbolic Logic 74 (1) (2009), 27–49.

[5] Selivanov, V.L.: Hierarchies of $\mathbf{\Delta}^0_2$-measurable k-partitions. Math. Logic Quarterly, 53 (2007), 446–461.

[6] Selivanov, V.L.: Towards a descriptive theory of cb0-spaces. Mathematical Structures in Computer Science. v. 28 (2017), issue 8, 1553–1580. DOI: http://dx.doi.org/10.1017/S0960129516000177. Earlier version in: ArXiv: 1406.3942v1 [Math.GN] 16 June 2014.

[7] Selivanov, V.L.: Extending Wadge theory to k-partitions. J. Kari, F. Manea and Ion Petre (eds.) CiE 2017, LNCS 10307, 387–399, Berlin, Springer.

[8] Wadge, W.: Reducibility and Determinateness in the Baire Space. PhD thesis, University of California, Berkely, 1984.

Moschovakis extension of multi-represented spaces

Dimiter Skordev
Sofia University, Faculty of Mathematics and Informatics, Sofia, Bulgaria
skordev@fmi.uni-sofia.bg

The Moschovakis extension of a set B is the set B^* defined in [Mo69]. Assuming without loss of generality that no element of the set B is an ordered pair, one builds B^* as the closure of $B \cup \{o\}$ under formation of ordered pairs, where o (denoted by 0 in [Mo69]) is some object which does not belong to B and also is not an ordered pair. Certain relative computability notions for functions in B^* are introduced and studied in the mentioned paper. The functions considered there are, in general, multi-valued. In the case of single-valued functions, one of these notions, namely absolute prime computability, seems to be able to cover any reasonable kind of deterministic computability by means of programs using some given functions.

In [Sk18], a certain link between computability of the above-mentioned kind and TTE computability is indicated in the case when some representation of the set B is given. An appropriate related representation of B^* is used for that purpose. The present paper generalizes to the case of multi-representations some of the results from [Sk18] and extends them. Two kinds of TTE computability are considered. The first of them is the usual computability via realizations, with the restriction that only single-valued realizations may be used. The other computability we consider is a Brattka style one – it is in the sense of [Br03, Definition 7.1], appropriately generalized for the case of multi-representations.[1] We establish some statements which are in the same vein as [Br96, Theorem 31], [We00, Theorems 3.1.6 and 3.1.7], [Br03, Theorem 8.3] and the results in [We08]. For single-valued functions, we prove the TTE computability of any function which is absolutely prime computable in some TTE computable functions. A similar result holds for multi-valued functions, but with an analog of absolute prime computability.

Let γ be a multi-representation of the set B. We construct a multi-representation γ^* of the set B^* with the following properties, where the mentioned computability is the usual one:

- the identity function of B is both (γ, γ^*)- and (γ^*, γ)-computable;
- the ordered pair operation in B^* is $(\gamma^*, \gamma^*, \gamma^*)$-computable;
- the two unary partial functions in B^* which transform ordered pairs into their first and their second components are (γ^*, γ^*)-computable;
- the element o is γ^*-computable, and so is the mapping of B^* into $\mathbb{N}$ which maps o, the elements of B and all other elements of B^* into 0, 1 and 2, respectively.

We prove that any two multi-representations of the set B^* with these properties are equivalent.

When considering functions in B^*, it is not an essential restriction to confine oneself to unary ones. Let $\mathcal{F}$ be the set of all unary partial multi-valued functions in B^* (the set of the

[1] The corresponding definition reads as follows: if γ and γ' are multi-representations of X and X', respectively, and φ is a partial multi-valued function from X to X' then φ is said to be *Brattka style* (γ, γ')*-computable* if a computable partial mapping β of $\mathrm{dom}(\gamma) \times \mathbb{N}^{\mathbb{N}}$ into $\mathrm{dom}(\gamma')$ exists such that the following holds whenever $p \in \mathrm{dom}(\gamma)$ and $x \in \gamma(p) \cap \mathrm{dom}(\varphi)$:

- $(p, q) \in \mathrm{dom}(\beta)$ and $\gamma'(\beta(p, q)) \cap \varphi(x) \neq \varnothing$ for all $q \in \mathbb{N}^{\mathbb{N}}$;
- $\varphi(x) \subseteq \bigcup \{\gamma'(\beta(p, q)) \mid q \in \mathbb{N}^{\mathbb{N}}\}$.

single-valued ones can be regarded as a subset of $\mathcal{F}$). It is shown in [Sk92] that a function $\varphi \in \mathcal{F}$ is absolutely prime computable in some given functions $\psi_1, \ldots, \psi_l \in \mathcal{F}$ iff φ can be obtained from $\psi_1, \ldots, \psi_l$ and the functions π and δ from [Mo69] by finitely many applications of three natural operations, namely the usual composition in $\mathcal{F}$ and the following two ones: $\theta_1, \theta_2 \mapsto \lambda x.\theta_1(x) \times \theta_2(x)$ (this operation is called *combination* in [Sk92] and *juxtaposition* in [Br96, Br03, We08]) and $\theta_1, \theta_2 \mapsto \iota$, where $y \in \iota(x)$ iff a finite sequence $z_0, z_1, \ldots, z_n$ of elements of B^* exists such that $z_0 = x$, $z_n = y$,

$$z_i \in \mathrm{dom}(\theta_1) \cap \mathrm{dom}(\theta_2) \;\&\; \theta_2(z_i) \setminus (B \cup \{o\}) \neq \varnothing \;\&\; z_{i+1} \in \theta_1(z_i) \tag{1}$$

for all $i < n$, and $z_n \in \mathrm{dom}(\theta_2)$, $\theta_2(z_n) \cap (B \cup \{o\}) \neq \varnothing$ (the function ι is called *the iteration of* θ_1 *controlled by* θ_2 in [Sk92]). Making use of this characterization of absolute prime computability, we prove the (γ^*, γ^*)-computability of any function from $\mathcal{F}$ which is absolutely prime computable in some single-valued (γ^*, γ^*)-computable functions from $\mathcal{F}$.

To get a similar result for arbitrary functions in $\mathcal{F}$, we consider a new triple of operations, namely we replace the first and the last of the above three operations with certain modifications of them, the results of applying the modified operations being appropriate restrictions of the results of applying the original ones. The modified composition is one used in [Br96, We00, Br03, We08], and the modified iteration operation is $\theta_1, \theta_2 \mapsto \iota \restriction E$, where ι is the same as above, and E is the set of the elements x of B^* such that

- no infinite sequence $z_0, z_1, z_2, \ldots$ of elements of B^* exists with $z_0 = x$ and (1) holding for all i;
- $z_n \in \mathrm{dom}(\theta_2)$ & $\big(\theta_2(z_n) \setminus (B \cup \{o\}) \neq \varnothing \Rightarrow z_n \in \mathrm{dom}(\theta_1)\big)$ whenever $z_0, z_1, \ldots, z_n$ is a finite sequence of elements of B^* with $z_0 = x$ and (1) holding for all $i < n$.

It turns out that (γ^*, γ^*)-computability (including the Brattka style one) is preserved by each of the operations from the new triple.

Remark. A natural ternary operation of branching is omitted in both lists because it is expressible through the three listed operations.

The First Recursion Theorem from [Sk92] can be used to show the (γ^*, γ^*)-computability of certain least fixed points.

References

[Br96] Vasco Brattka. Recursive characterization of computable real-valued functions and relations. *Theor. Comput. Sci.*, **162**, 45–77, 1996.

[Br03] Vasco Brattka. Computability over topological structures. In: *Computability and Models*, eds. S. B. Cooper and S. S. Goncharov, Kluwer Academic/Plenum Publishers, 2003, 93-136.

[Mo69] Yiannis Moschovakis. Abstract first order computability. I. *Trans. Amer. Math. Soc.*, **138**, 427–464, 1969.

[Sk92] Dimiter Skordev. Computability in Combinatory Spaces. An Algebraic Generalization of Abstract First Order Computability. Kluwer Academic Publishers, 1992 (errata at `https://www.fmi.uni-sofia.bg/fmi/logic/skordev/errata_combinatory.htm`).

[Sk18] Dimiter Skordev. Moschovakis extension of represented spaces. Log. Meth. Comput. Sci. (submitted, `https://arxiv.org/abs/1801.09544`).

[We00] Klaus Weihrauch. Computable Analysis. An Introduction. Springer-Verlag, 2000.

[We08] Klaus Weihrauch. The computable multi-functions on multi-represented sets are closed under programming. *J. Univers. Comput. Sci.*, **14**, 801–844, 2008.

DEGREES OF RANDOMIZED COMPUTABILITY

RUPERT HÖLZL AND CHRISTOPHER P. PORTER

In this survey we discuss work of Levin and V'yugin on collections of sequences that are non-negligible in the sense that they can be computed by a probabilistic algorithm with high probability. More precisely, Levin and V'yugin introduced an ordering on collections of sequences that are closed under Turing equivalence. Roughly speaking, given two such collections $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}$ is less than $\mathcal{B}$ in this ordering if $\mathcal{A} \setminus \mathcal{B}$ is negligible. The degree structure associated with this ordering, the *Levin-V'yugin degrees* (or *LV-degrees*) can be shown to be a Boolean algebra, and in fact a measure algebra.

We demonstrate the interactions of this work with recent results in computability theory and algorithmic randomness: First, we recall the definition of the Levin-V'yugin algebra and identify connections between its properties and classical properties from computability theory. In particular, we apply results on the interactions between notions of randomness and Turing reducibility to establish new facts about specific LV-degrees, such as the LV-degree of the collection of 1-generic sequences, that of the collection of sequences of hyperimmune degree, and those collections corresponding to various notions of effective randomness. Next, we provide a detailed explanation of a complex technique developed by V'yugin that allows the construction of semi-measures into which computability-theoretic properties can be encoded. We provide examples of the uses of this technique by explicating and extending V'yugin's results about the LV-degrees of the collection of Martin-Löf random sequences and the collection of sequences of DNC degree, as well as results concerning atoms of the LV-degrees.

Fakultät für Informatik, Institut für Theoretische Informatik, Mathematik und Operations Research, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
E-mail address: r@hoelzl.fr
URL: http://hoelzl.fr

Department of Mathematics and Computer Science, Drake University, Des Moines, IA 50311, USA
E-mail address: cp@cpporter.com
URL: http://cpporter.com

Separating Weihrauch degrees via Kleene's recursion theorem

Takayuki Kihara
Department of Mathematical Informatics
Nagoya University, Nagoya, Japan
kihara@i.nagoya-u.ac.jp

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

Widely-used separation techniques for Weihrauch degrees are inherited from reverse mathematics (constructing ω-models) or descriptive set theory (using e.g. Σ^0_n-completeness). These techniques are insufficient to distinguish products of Weihrauch degrees. Some basic questions of this type can be resolved using cardinality arguments. Other suitable techniques include Hertling's level [5], the displacement lemma from [2], the small diameter lemma from [7] and the notion of finitely tolerance from [4]. While these techniques are easy to use in cases where they apply, they apply only to very specific situations.

Here we describe a template of how to use Kleene's recursion theorem to prove separations between Weihrauch degrees. This template takes some effort to instantiate, but is quite suitable to distinguish both parallel and sequential products. We had already employed the technique in [6] to show that $\mathrm{AoUC}_{[0,1]} \star \mathrm{AoUC}_{[0,1]} \not\leq_{\mathrm{W}} \mathrm{AoUC}^k_{[0,1]}$, albeit without making the template explicit. To formulate it, we use $\mathcal{M}(\mathbf{X}, \mathbf{Y})$ to denote the space of strongly continuous multifunctions from $\mathbf{X}$ to $\mathbf{Y}$ (see [3] for discussion).

Theorem 1. 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

$$f \not\leq_{\mathrm{W}} g$$

Proof. Assume that $f \leq_{\mathrm{W}} g$ via computable H, K. Let computable E be a realizer of e. Let $(\phi_n :\subseteq \mathbb{N} \to \mathbb{N})_{n\in\mathbb{N}}$ be a standard enumeration of the partial computable functions. By assumption, we can consider each ϕ_n to denote some element in $\mathbf{X}$. Let λ be a computable function such that $\phi_{\lambda(n)} = E(H(\phi_n), (v \mapsto K(\phi_n, v)))$. By Kleene's fixed point theorem, there is some n_0 with $\phi_{n_0} = \phi_{\lambda(n_0)}$. Inputting ϕ_{n_0} to f fails the assumed reduction witnesses. □

The strength of the theorem lies in the fact that it allows us to prove the non-existence of a reduction by constructing a particular computable function. It thus converts the typically difficult task of proving a negative to the easier task of proving a positive.

As an application, we will compare convex choice with products of the degree $\mathrm{Sort} : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$. The latter was defined in [8] via $\mathrm{Sort}(p) = 0^n1^\omega$ if p contains exactly n 0s, and $\mathrm{Sort}(p) = 0^\omega$ if p contains infinitely many 0s. Convex choice $\mathrm{XC}_n :\subseteq \mathcal{A}([0,1]^n) \rightrightarrows [0,1]^n$ takes a convex closed set as input, and returns some point in it. It was studied in [7].

Theorem 2. $\mathrm{C}_2 \times \mathrm{XC}^n_1 \not\leq_{\mathrm{W}} \mathrm{Sort}^n$.

In contrast, we can also show the following generalization of a result from [1]:

Theorem 3. $\mathrm{XC}_n \leq_{\mathrm{W}} \mathrm{Sort}^n$

Acknowledgements

We are grateful to Stéphane Le Roux for illuminating discussions leading to Theorem 3. Kihara's research was partially supported by JSPS KAKENHI Grant 17H06738, 15H03634, and the JSPS Core-to-Core Program (A. Advanced Research Networks). Pauly's research was partially supported by the MSCA-RISE project "CID - Computing with Infinite Data"(731143).

References

[1] Vasco Brattka, Rupert Hölzl & Rutger Kuyper (2017): *Monte Carlo Computability*. In Heribert Vollmer & Brigitte Vallee, editors: *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 66, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 17:1–17:14, doi:10.4230/LIPIcs.STACS.2017.17. Available at `http://drops.dagstuhl.de/opus/volltexte/2017/7016`.

[2] Vasco Brattka, Joseph Miller, Stéphane Le Roux & Arno Pauly (2012): *Connected Choice and Brouwer's Fixed Point Theorem*. http://arxiv.org/abs/1206.4809.

[3] Vasco Brattka & Arno Pauly (2018): *On the algebraic structure of Weihrauch degrees*. *Logical Methods in Computer Science*. Available at `http://arxiv.org/abs/1604.08348`. To appear.

[4] F. G. Dorais, D. D. Dzhafarov, J. L. Hirst, J. R. Mileti & P. Shafer (2016): *On uniform relationships between combinatorial problems*. *Transactions of the AMS* 368, pp. 1321–1359, doi:10.1090/tran/6465. ArXiv 1212.0157.

[5] Peter Hertling (1996): *Unstetigkeitsgrade von Funktionen in der effektiven Analysis*. Ph.D. thesis, Fernuniversität, Gesamthochschule in Hagen.

[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, doi:10.4230/LIPIcs.MFCS.2016.58.

[7] Stéphane Le Roux & Arno Pauly (2015): *Finite choice, convex choice and finding roots*. *Logical Methods in Computer Science*, doi:10.2168/LMCS-11(4:6)2015. Available at `http://arxiv.org/abs/1302.0380`.

[8] Eike Neumann & Arno Pauly (2018): *A topological view on algebraic computations models*. *Journal of Complexity* 44, doi:10.1016/j.jco.2017.08.003. Available at `http://arxiv.org/abs/1602.08004`.

Weihrauch degrees of translations between representations of real numbers, revisited

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

In recent years, the study of Weihrauch degrees has grown to a vibrant subfield. We point to [3] for a survey, and turn our attention towards the origins of this development. These are found in two technical reports by Weihrauch, [6] and [7] in which he introduced a precursor of Weihrauch reducibility. In [7] the problems studied were of a combinatorial nature and inspired by Brouwerian counterexamples. The usefulness of these problems (LPO^n and LLPO_n a.k.a. $\mathrm{C}_{\{0,\dots,n-1\}}$) has been demonstrated by their appearance in many subsequent works over the years.

The focus of [6] on the other hand was on problems of direct interest: The translations between various representations of real numbers. The translations between the Cauchy reals, the upper reals and the lower reals are highly relevant e.g. in computable measure theory. It has thus entered the standard canon that all non-trivial translations between these representations are equivalent to lim.

On the other hand, digit-based representations mostly appear as a bad example in introductions to computable analysis, and are forgotten afterwards. We shall thus revisit the results about the degrees of the translations between them (for various bases) and to them from [6], and place them into the context of the subsequent work. We will see that the relevant degrees have reappeared in the literature, too.

The digit-based representations are special cases of the weak separation representations. We will also consider the strong cut representations (which generalize the repeated fraction representation). Both types of representations are parameterized with a dense sequence. In the following, let $Q, P \subseteq \mathbb{R}$ be dense subsets with enumerations $\nu : \mathbb{N} \to Q$ and $\mu : \mathbb{N} \to P$. We will assume that ν and μ are computable. Of course, all our considerations relativize.

Definition 1. The weak separation representation δ_Q^{ws} induced by Q satisfies $\delta_Q(p) = x$ iff $x < \nu(n) \Rightarrow p(n) = 0$ and $\nu(n) < x \Rightarrow p(n) = 1$.

Recall from [4] that $\mathrm{C}_{\sharp \leq 2}$ denotes the problem of finding a path through a binary tree with two vertices per level (and thus either one or two infinite paths).

Theorem 2. The following are equivalent:

1. $\mathrm{C}_{\sharp \leq 2}$
2. $\mathrm{id} : \mathbb{R} \to (\mathbb{R}, \delta_Q^{\mathrm{ws}})$
3. $\mathrm{id} : (\mathbb{R}, \delta_P^{\mathrm{ws}}) \to (\mathbb{R}, \delta_Q^{\mathrm{ws}})$, provided that $Q \setminus P$ contains a dense c.e. set

It was shown in [4] that $\mathrm{C}_{\sharp\leq 2} <_{\mathrm{W}} \mathrm{C}_{\sharp\leq 2} \times \mathrm{C}_{\sharp\leq 2}$ (in fact, that taking more and more products produces an infinite ascending chain). Thus, converting more reals from the standard representation into the decimal reals is strictly harder than converting fewer. Since trivially $\mathrm{C}_{\sharp\leq 2} \leq_{\mathrm{W}} \mathrm{C}_{\{0,1\}^{\mathbb{N}}}$, we find that $\mathrm{C}_{\sharp\leq 2}$ is *low* by the uniform low basis theorem from [1]. This implies e.g. that limit computability w.r.t. the decimal representation as input already implies limit computability w.r.t. the standard representation.

If we ask for a weak separation representation on the output side, the situation is different. Note that all principles in Theorem 2 are cylinders, and that we are thus dealing with a strong Weihrauch degree. By using the jump introduced in [2] (recall that the jump $\mathbb{R}'$ of $\mathbb{R}$ is just the naive Cauchy representation), we thus obtain:

Corollary 3. $\lim <_{\mathrm{W}} \mathrm{C}_{\sharp\leq 2} \star \lim \equiv_{\mathrm{W}} \left(\mathrm{id} : \mathbb{R}' \to (\mathbb{R}, \delta_Q^{\mathrm{ws}})\right)$

Conversely, if we start with the lower or upper reals, no extra difficulty appears:

Proposition 4. Let ν be such that $\nu(n) < q$ is decidable for $n \in \mathbb{N}$, $q \in \mathbb{Q}$. Then:

$$\lim \equiv_{\mathrm{W}} \left(\mathrm{id} : \mathbb{R}_< \to (\mathbb{R}, \delta_Q^{\mathrm{ws}})\right) \equiv_{\mathrm{W}} (\mathrm{id} : \mathbb{R}_< \to \mathbb{R})$$

We can also say a few things about comparing different weak separation representations in case $Q \setminus P$ is not dense. Note that if $Q \setminus P$ is empty, the translation is trivially continuous.

Proposition 5. If $Q \setminus P$ is non-empty, then $\mathrm{C}_2 \leq_{\mathrm{W}} \left(\mathrm{id} : (\mathbb{R}, \delta_P^{\mathrm{ws}}) \to (\mathbb{R}, \delta_Q^{\mathrm{ws}})\right)$.

Proposition 6. If $Q \setminus P$ has Cantor-Bendixson rank at most n, then relative to some oracle:

$$\left(\mathrm{id} : (\mathbb{R}, \delta_P^{\mathrm{ws}}) \to (\mathbb{R}, \delta_Q^{\mathrm{ws}})\right) \leq_{\mathrm{W}} \mathrm{C}_{2^{n+1}}$$

The other type of representations of the reals we consider are the strong cut representations. Here, rather than considering enumerations of Dedekind cuts as just to obtain the lower reals, the upper reals or the Euclidean reals, the cuts are given as decidable sets here. Given some enumeration $\nu : \mathbb{N} \to Q$ of a dense countable set, we let ρ_Q^{sc} denote the strong cut representation. If $Q = \mathbf{Q}$, then ρ_Q^{sc} is equivalent to the continued fraction representation.

Our benchmark principle here is $\mathrm{Sort} : \{0,1\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ as introduced in [5]. This is defined by $\mathrm{Sort}(p) = 0^\omega$ if p contains infinitely many 0s, and $\mathrm{Sort}(p) = 0^n 1^\omega$ if p contains exactly n 0s.

Theorem 7. The following are equivalent:

1. Sort
2. $\mathrm{id} : \mathbb{R} \to (\mathbb{R}, \rho_Q^{\mathrm{sc}})$
3. $\mathrm{id} : (\mathbb{R}, \delta_Q^{\mathrm{ws}}) \to (\mathbb{R}, \rho_Q^{\mathrm{sc}})$
4. $\mathrm{id} : (\mathbb{R}, \delta_P^{\mathrm{ws}}) \to (\mathbb{R}, \rho_Q^{\mathrm{sc}})$, provided that $Q \setminus P$ contains a dense c.e. set

Again, since we know e.g. that $\mathrm{Sort} <_{\mathrm{W}} \mathrm{Sort} \times \mathrm{Sort}$ from [5], we can conclude that translating multiple numbers into the continued fraction representation is strictly harder than doing so for a single number.

References

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

[2] Vasco Brattka, Guido Gherardi & Alberto Marcone (2012): *The Bolzano-Weierstrass Theorem is the Jump of Weak König's Lemma. Annals of Pure and Applied Logic* 163(6), pp. 623–625. Also arXiv:1101.0792.

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

[4] Stéphane Le Roux & Arno Pauly (2015): *Finite choice, convex choice and finding roots. Logical Methods in Computer Science* Available at `http://arxiv.org/abs/1302.0380`.

[5] Eike Neumann & Arno Pauly (2018): *A topological view on algebraic computations models. Journal of Complexity* 44. Available at `http://arxiv.org/abs/1602.08004`.

[6] Klaus Weihrauch (1992): *The degrees of discontinuity of some translators between representations of the real numbers.* Informatik Berichte 129, FernUniversität Hagen, Hagen.

[7] Klaus Weihrauch (1992): *The TTE-interpretation of three hierarchies of omniscience principles.* Informatik Berichte 130, FernUniversität Hagen, Hagen.