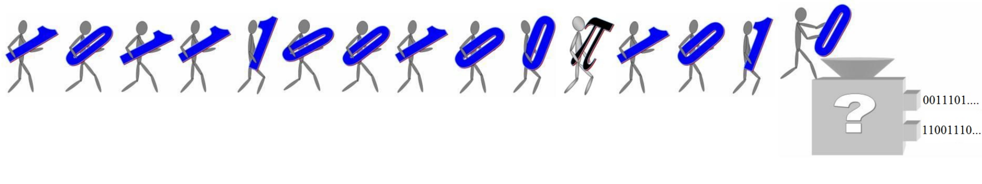
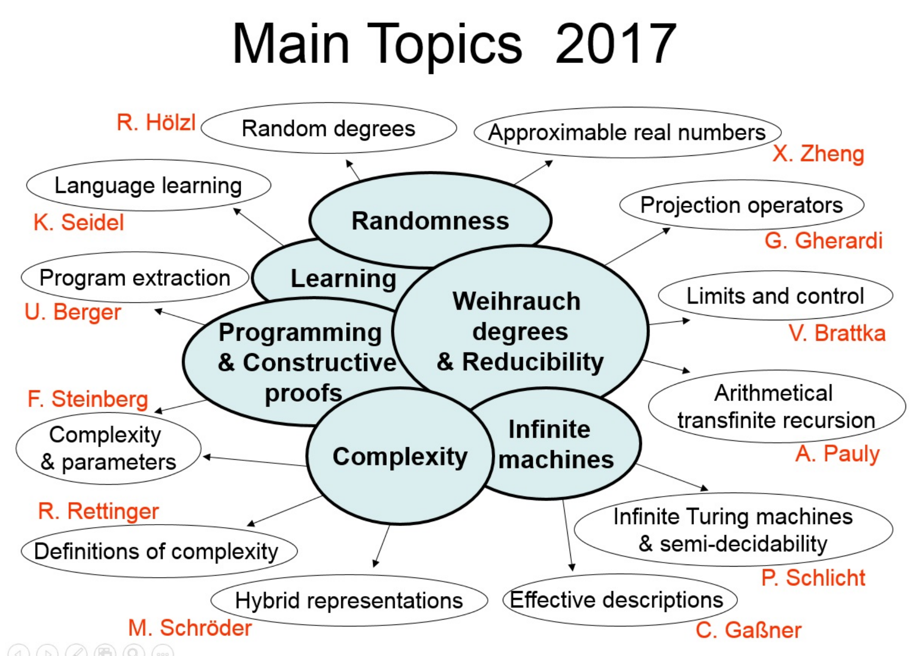
Meeting

# COMPUTABILITY AND REDUCIBILITY

**6 – 11 August 2017, Hiddensee / University Greifswald**


**Ulrich Berger**
&
**Vasco Brattka**
&
**Christine Gaßner**
&
**Guido Gherardi**
&
**Rupert Hölzl**
&
**Arno Pauly**
&
**Robert Rettinger**
&
**Philipp Schlicht**
&
**Matthias Schröder**
&
**Karen Seidel**
&
**Florian Steinberg**
&
**Xizhong Zheng**


Main Topics 2017
R. Hölzl
Random degrees
Approximable real numbers
X. Zheng
Language learning
K. Seidel
Randomness
Projection operators
G. Gherardi
Program extraction
U. Berger
Learning
Weihrauch degrees & Reducibility
Limits and control
V. Brattka
Programming & Constructive proofs
F. Steinberg
Arithmetical transfinite recursion
A. Pauly
Complexity & parameters
Complexity
Infinite machines
R. Rettinger
Infinite Turing machines & semi-decidability
Definitions of complexity
P. Schlicht
Hybrid representations
Effective descriptions
M. Schröder
C. Gaßner

# Programme

**Sunday, August 6, 2017**

8.00 pm
Welcome meeting

**Monday, August 7, 2017**

9.30 am
Karen Seidel
*Language learning from a computational viewpoint*
10.30 am
Ulrich Berger
*Concurrent program extraction in computable analysis*
2.30 pm
Arno Pauly
*Welcher Weihrauch-Grad entspricht* $\mathrm{ATR}_0$*?*
7.30 pm
Meeting in the restaurant ,,Buhne XI"

**Tuesday, August 8, 2017**

9.30 am
Guido Gherardi
*Projection operators in computable analysis*
10.30 am
Vasco Brattka
*Ein Limit-Kontrollsatz mit Anwendungen*
2.30 pm
Philipp Schlicht
*Unendliche Turingmaschinen und erkennbare Mengen*
3.30 pm
Christine Gaßner
*Effective descriptions of mathematical objects and the BSS–RAM model*

**Wednesday, August 9, 2017**

9.30 am
Matthias Schröder
*Hybrid representations as a tool for Complexity Theory*
10.30 am
Robert Rettinger
*On the definition of computational complexity in analysis*
2.30 pm
Florian Steinberg
*Complexity in analysis and parameters*

**Thursday, August 10, 2017**

9.30 am
Xizhong Zheng
*On the computability and reducibility of approximable real numbers*
10.30 am
Rupert Hölzl
*Rank and randomness*
2.30 pm
Discussions
New projects and open problems

**Friday, August 11, 2017**

10.00 am
Closing discussion

# Open Problems

**Vasco Brattka**: Path Intersection Theorem

Das Path Intersection Theorem besagt, dass zwei stetige Kurven $f, g : [0,1] \to [0,1]^2$ mit $f(0) = (0,1)$, $f(1) = (1,0)$, $g(0) = (0,0)$, $g(1) = (1,1)$ einen Schnittpunkt haben, d.h. range$(f)$ hat einen nicht leeren Schnitt mit range$(g)$. Dies läßt sich als eine mehrwertige Funktion von $C([0,1],[0,1]^2)$ nach $[0,1]^2$ formalisieren, wobei mindestens zwei Varianten denkbar sind, wie die Ausgabeinformation gegeben wird:

(1) die Ausgabe ist ein Paar $(x, y)$ mit $(x, y)$ im Schnitt von range$(f)$ und range$(g)$,
(2) die Ausgabe ist ein Paar $(t, s)$ mit $f(t) = g(s)$.

Wenn wir diese Probleme als $\text{PIT}_{\text{range}}$ und PIT bezeichnen, dann ist es klar, dass

$$\text{IVT} \leq_W \text{PIT} \leq_W \text{WKL und } \text{PIT}_{\text{range}} \leq_W \text{PIT und } \text{PIT}_{\text{range}} <_W \text{WKL}.$$

Dabei bezeichnet $\leq_W$ die Weihrauch-Reduzierbarkeit und $<_W$ die echte Version davon (bei der keine Äquivalenz vorliegt), IVT das Intermediate Value Theorem und WKL Weak König's Lemma. Die Echtheit $<_W$ folgt, da Weihrauch bewiesen hat, dass für berechenbare $f, g$ ein berechenbares $(x, y)$ existiert. D. h. man kann zumindest PIT als Verallgemeinerung von IVT sehen und die Frage ist, ob man einen engeren Zusammenhang zu IVT herstellen kann. Genauer kann man fragen:

(1) Gilt $\text{PIT} \leq_W \text{IVT}^{[n]}$ für eine natürliche Zahl $n$?
(2) Eine analoge Frage läßt sich für $\text{PIT}_{\text{range}}$ stellen.
(3) Gilt $\text{PIT} \equiv_W \text{PIT}_{\text{range}}$?

Dabei bezeichnet $f^{[n]}$ das $n$-fache kompositionelle Produkt $*$ von $f$ und $\equiv_W$ die Weihrauch-Äquivalenz. Man kann in den Fragen auch $\text{IVT}^{[n]}$ durch $\text{IVT}^n$ ersetzen, was für das $n$-fache parallele Produkt steht. D. h. (1) und (2) fragen, ob sich die Intuition, dass PIT ein höherdimensionales Analogon von IVT ist, auch algebraisch formalisieren läßt.

**Florian Steinberg**: Structure theory of second-order polynomials

The class of second-order polynomials is the smallest class of functions of type $\mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ such that:

– For any polynomial $p \in \mathbb{N}[X]$ the function $(l, n) \mapsto p(n)$ is included.
– whenever $P$ and $Q$ are included, so are $P + Q$ and $P \cdot Q$.
– Whenever $P$ is included, then so is the function $P^+(l, n) := l(P(l, n))$.

Second-order polynomials play important roles as higher order bounds in second-order complexity theory and proof theory. While regular polynomials allow for normal form theorems, a notion of degree, etc. second-order polynomials are notoriously difficult to handle. For a normal form theorem for second-order polynomials to exist, for instance, it seems necessary to have injectivity of the operation $P \mapsto P^+$ the second-order polynomials were required to be closed under. The problem here is that second-order polynomials are considered functions: $P \neq Q$ may be spelled out to $\exists l, n\colon P(l, n) \neq Q(l, n)$. However, the witness of this existential statement may not proof the inequality of $P^+$ and $Q^+$, as it may map the different values $P(l, n)$ and $Q(l, n)$ to the same value. A stronger statement that would imply the injectivity of the operation $P \mapsto P^+$ would therefore be: If two second-order polynomials differ, then they already differ on an injective function argument. The open question I would like to see settled is: Is the operation $P \mapsto P^+$ injective? I welcome both a counterexample and a proof.

## Ein Limit-Kontrollsatz mit Anwendungen

**Vasco Brattka**

Institut für Theoretische Informatik, Mathematik und Operations Research
Universität der Bundeswehr München

Ich plane über einen Limit-Kontrollsatz zu berichten, der verschiedene Anwendungen in der Berechenbaren Analysis hat. Dabei geht es darum, berechenbare Funktionen zu kontrollieren, die alle gegen einen Punkt konvergierenden Folgen wieder auf konvergente Folgen abbilden. Dabei soll die Kontrolle in der Weise erfolgen, dass die Berechnung auf den Grenzwerten nachvollzogen wird. Drei Anwendungen dieses Satzes sind:

1. Die Klassifikation der bezüglich der naiven Cauchy-Darstellung berechenbaren Funktionen.
2. Die Faktorisierung von König's Lemma in die Limes-Abbildung und das Cohesiveness-Problem (eine Eigenschaft, die ihrerseits Anwendungen in der Untersuchung des Satzes von Ramsey hat).
3. Ein Sprung-Inversionssatz im Weihrauch-Verband.

## Effective Descriptions of Mathematical Objects and the BSS-RAM Model

**Christine Gaßner**

Institut für Mathematik unfd Informatik
Universität Greifswald

We deal with questions arising from the comparison of some machine-oriented models of computation. Models, such as the uniform BSS model and the model based on Type-2 Turing machines, are used in order to analyze the algorithmic complexity of mathematical problems and their degree of unsolvability, respectively. Since, in this connection, various mathematical functions, several types of sequences and their limits, algebraic sets and their measures, and the like play an important role, we also want to consider some further types of Type-2 machines that can work over suitable algebraic structures. For finite alphabets, the random access machines of Type-2 correspond to the Type-2 Turing machines studied in the field of computable analysis. Other machines are generalizations of the analytic machines and infinite Turing machines. We will discuss the possibilities to transfer results from one model to another and suitable representations for the considered mathematical objects within this framework.

## Projection Operators in Computable Analysis

**Guido Gherardi**

Dipartimento di Filosofia e Comunicazione
Università di Bologna, Italy

Projecting an arbitrary point x over a non-empty given subset of the Euclidean space is an example of a mathematical problem deeply grounded in our geometrical intuition of the spatial continuum that has important applications in higher mathematics. The question that I address in my talk is then the following: does the intuitive, even empirical, naturalness of this problem correspond to an algorithmic simplicity of solution? The problem of finding points in a closed set A of minimal distance with respect to a given point x has been investigated already by other authors and proved to be computable for some types of metric spaces (and closed sets representations) in presence of some optimal conditions. But what does it happen when such optimal conditions may fail to hold? It is not very surprising that this problem is then no longer computably solvable, but I will classify the degrees of incomputability in the Weihrauch lattice of the projection operators defined by the different usual types of information for closed sets. Beside the projection operators of exact precision, I will also consider corresponding approximated versions, to determine whether they are really computably simpler, as the intuition suggests. It turns out that such operators are useful to characterize some Weihrauch degrees of fundamental importance. The approximated projection operators with total information for closed sets are even computable. Still, they might be seen of no practical importance, unless concrete examples of applications are shown. I will then analyze the classical Whitney Extension Theorem as a case study. This is a joint work with Alberto Marcone improving some preliminary results presented at CCA in Faro.

## Rank and Randomness

**Rupert Hölzl**

Institut für Theoretische Informatik, Mathematik und Operations Research
Universität der Bundeswehr München

We show that for each computable ordinal $\alpha > 0$ it is possible to find in each random $\Delta_2^0$ degree a sequence $R$ of Cantor-Bendixson rank $\alpha$, while ensuring that the sequences that inductively witness $R$'s rank are all Martin-Löf random with respect to a single countably supported and computable measure. This is a strengthening for random degrees of a recent result of Downey, Wu, and Yang, and can be understood as a randomized version of it. (joint work with Christopher P. Porter)

## Welcher Weihrauch-Grad entspricht $\mathrm{ATR}_0$?

**Arno Pauly**

Université Libre de Bruxelles & University of Birmingham

Marcone hat die Frage aufgeworfen, welcher Weihrauch-Grad dem Axiomsystem $\mathrm{ATR}_0$ der Reverse Mathematics entspricht. Kandidaten sind "Unique Choice on Baire space" und "Closed choice on Baire space". Anhand der Beispieltheoreme "Open determinacy" und "Perfect Tree Theorem" werde ich das für und wider der beiden Kandidaten diskutieren, aufzeigen, dass es mindestens noch einen weiteren Kandidaten gibt, und letztlich schlussfolgern, dass eine einfache elegante Antwort auf Marcone's Frage vermutlicherweise nicht möglich ist. Dies basiert auf gemeinsamer Arbeit mit Takayuki Kihara.

## Unendliche Turingmaschinen und erkennbare Mengen

**Philipp Schlicht**

Mathematisches Institut
Rheinische Friedrich-Wilhelms-Universität Bonn

Wir betrachten unendliche Turingmaschinen mit einem speziellen Zustand, der zu allen Zeitpunkten in einer gegebenen Klasse angenommen wird. Der Vortrag beginnt mit einer Einleitung in diese Maschinen. Wir untersuchen, welche Mengen natürlicher Zahlen von solchen Maschinen erkannt werden können. Dies ist eine gemeinsame Arbeit mit Merlin Carl.

## Hybrid Representations as a Tool for Complexity Theory

**Matthias Schröder**

Fachbereich Mathematik
Technische Universität Darmstadt

We investigate hybrid representations as a new approach to Complexity Theory in Computable Analysis. One motivation is to unify the approach by Kawamura and Cook and the TTE-approach to Complexity Theory. Moreover, we will discuss Co-Polish spaces. They allow the measurement of complexity in a discrete parameter on the input, whereas for general spaces an uncountable parameter is necessary.

## Language Learning from a Computational Viewpoint

**Karen Seidel**

Hasso-Plattner-Institut
Universität Potsdam

This talk introduces Computational Learning Theory, which originates from formally modeling language acquisition of children. I am going to present a summary of the most important results (due to John Case, 2016) in this area, since Gold's seminal paper in 1967, interlaced with recent joint work about Non-U-Shaped Learning with Kötzing and Schirneck.

## Complexity in Analysis and Parameters

**Florian Steinberg**

Mathematik AG1
Technische Universität Darmstadt

In their 2010 paper "Complexity theory for operators in analysis" Kawamura and Cook introduced what is currently the most used framework for complexity in analysis. While it is well accepted by the theoretical side of the computable analysis community, many of the people that produce software based on computable analysis do not fully accept it. We give some examples where there indeed is a discrepancy between the predictions of the framework and what seems to be efficiently computable in practice. We present parameterized spaces as a fix of this. We give some basic results that justify the design choices. Finally, we give a fairly long list of notions showing up in literature that feature very similar ideas.

## On the Computability and Reducibility of Approximable Real Numbers

**Xizhong Zheng**

Department of Computer Science and Mathematics
Arcadia University Glenside, USA

A real number $x$ is called computable if there is a computable sequence $(x_s)$ of rational numbers which converges to $x$ effectively. If the computable sequence is only increasing, then the limit $x$ is called computably enumerable (c.e.). The arithmetical closure of the computable enumerable real numbers is the class of d.c.e. real numbers. Furthermore, the closure of d.c.e. real numbers under the total computable real functions is the class of divergence bounded computable (d.b.c.) real numbers. Finally, the limits of computable sequences of rational numbers without any convergence restriction are computably approximable real numbers. In this talk we are going to explore how the classes of d.c.e. and d.b.c. real numbers are related to different kind of reducibilities. In particular we will show that, a real number is d.c.e. iff it is Solovay reducible to a random c.e. real numbers, and it is d.b.c. iff it is convergence-dominated reducible to a random c.e. real number.

# Concurrent program extraction in computable analysis

Ulrich Berger and Hideki Tsuiki

In constructive logic and mathematics the meaning of a proposition is defined by describing how to prove it, that is, how to construct evidence for it. This is called the Brouwer-Heyting-Kolmogorov interpretation. For example,

- evidence for a conjunction, $A \wedge B$, is a pair $(d, e)$ where $d$ is evidence for $A$ and $e$ is evidence for $B$,
- evidence for a disjunction, $A \vee B$, is a pair $(i, d)$ where $i$ is 0 or 1 such that if $i = 0$ then $d$ is evidence for A and if $i = 1$ then $d$ is evidence for $B$,
- evidence for an implication, $A \to B$, is a computable procedure that transforms evidence for $A$ into evidence for $B$.

Formalising this interpretation of propositions and the corresponding constructive proof rules leads to a method of program extraction from constructive proofs: From every constructive proof of a formula one can extract a program that computes evidence for it. The extracted programs are functional and possibly higher-order and can be conveniently coded in programming languages such as ML, Haskell or Scheme.

If one attempts to develop program extraction into a method of synthesising 'correct-by-construction' software, one realizes that one misses out an indispensable element of modern programming: *Concurrency*, that is, the composition of independently executing computations.

Our work is an attempt to fill this gap. We present an extension of constructive logic by a new formula construct $\mathbf{S}_n(A)$ with the following BHK interpretation:

- Evidence for $\mathbf{S}_n(A)$ is tuple of at most $n$ computations running concurrently, at least one of which terminates, and each of which, if it terminates, computes evidence for $A$.

It turns out that the operator $\mathbf{S}_n$ becomes useful only in conjunction with a strong form of implication, $A \parallel B$, to be read '$A$ restricted to $B$'. The BHK interpretation of restriction is as follows:

- Evidence for $A \parallel B$ is a computation $a$ such that
  - if there is evidence for $B$, then $a$ terminates;
  - if $a$ terminates, then it does so with a result that provides evidence for $A$.

We present proof rules for $\mathbf{S}_n(A)$ and $A \parallel B$ and give examples of proofs that give rise to concurrent extracted programs. Somewhat surprisingly, the two operators validate a concurrent version of the Law of Excluded Middle,

$$\frac{A \parallel B \quad A \parallel \neg B}{\mathbf{S}_2(A)}$$

Indeed, assuming evidence $a$ for $A \| B$ and $b$ for $A \| \neg B$, one obtains evidence for $\mathbf{S}_2(A)$ by executing $a$ and $b$ concurrently.

We look at two examples of proofs with concurrent computational content in the area of computable analysis.

The first example is concerned with infinite Gray code, an extension of the well-known Gray code for integers to a representation of the real numbers, introduced by Tsuiki [3]. One can prove that the (coinductive) predicate characterising this representation implies a concurrent version of the predicate characterising the signed digit representation and extract from this a concurrent program that translates infinite Gray code into signed digit representation. The extracted program is the same as the one given in [3].

The second example is about finding in a non-zero vector of real numbers an entry that is apart from zero. A concurrent program solving this problem can be extracted from a proof in the new logic. This can be further used to prove the invertibility of non-singular quadratic matrices and hence to extract a program for matrix inversion using a concurrent version of Gaussian elimination.

Currently, program extraction in this extended logic is done informally and the extracted programs are implemented in a concurrent extension of Haskell. It is future work to integrate the concurrent proof rules in a suitable interactive proof system (for example, Minlog) and to implement the corresponding program extraction procedure to make it fully automatic.

Prior to this work, a (non-concurrent) program translating an intensional version of infinite Gray code into signed digit representation has been extracted from a proof implemented in the Minlog system [1]. A precursor of our logical system is presented in [2]. It allows for the extraction of non-determinism and concurrent programs, however, without control over the number of threads, that is, processes running concurrently at the same time.

## Acknowledgments

This work was supported by the International Research Staff Exchange Scheme (IRSES) Nr. 612638 CORCON and Nr. 294962 COMPUTAL of the European Commission, the JSPS Core-to-Core Program, A. Advanced Research Networks and JSPS KAKENHI Grant Number 15K00015.

The latest results were obtained while the authors were visiting the University of Canterbury, Christchurch, New Zealand, in April 2017. We are grateful to Douglas Bridges and Hannes Diener and the Mathematics Department of UC for hosting our visits which were part of the CORCON project.

## References

[1] U. Berger, K. Miyamoto, H. Schwichtenberg, and H. Tsuiki. Logic for Gray-code computation. In *Concepts of Proof in Mathematics, Philosophy, and Computer Science*, Ontos Mathematical Logic 6. de Gruyter, 2016.

[2] Ulrich Berger. Extracting Non-Deterministic Concurrent Programs. In *CSL 2016*, volume 62 of *LIPIcs*, pages 26:1–26:21, 2016.

[3] H. Tsuiki. Real Number Computation through Gray Code Embedding. *Theoretical Computer Science*, 284(2):467–485, 2002.

# On the Definition of Computational Complexity in Analysis

Robert Rettinger
Fachhochschule Dortmund
robert.rettinger@fh-dortmund.de

In this talk we will discuss several definitions of complexity in analysis based on the type-2-Turing-machine model.

Whereas the definition of computability in analysis based on the type-2-Turing-machine model gives us a robust and widely accepted notion, a similar natural definition of complexity seems to be much harder to find. The basic, naive definition of complexity is probably the most natural definition. Unfortunately, this basic definition is applicable only in very restricted circumstances. On the other side there is a widely applicable definition of polynomial time complexity introduced by Kawamura and Cook based on type-2-polynomials. We will motivate several different definitions of complexity and discuss some arguments in favour of and against these definitions.

We aim to start a broader discussion on the notion of complexity rather than giving a perfect definition. In addition we will briefly discuss further aspects of complexity beyond the so far discussed definitions.