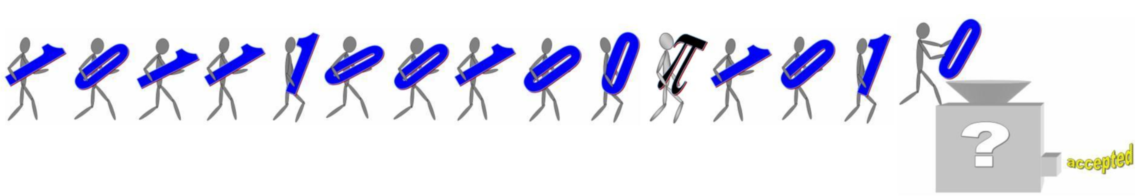
Different models of computation

Greifswald 2011

Hannes Diener
&
Christine Gaßner
&
Paul Grieger
&
Arno Pauly
&
Carsten Rösnick
&
Peter Scheiblechner
&
Philipp Schlicht
&
Isabel Schwende
&
Martin Ziegler



Meeting "Different models of computation"

18 - 22 July 2011, Greifswald

Computable Analysis without Computability

Hannes Diener

Universität Siegen
FB 6: Mathematik, Mathematische Logik und Theoretische Informatik

Alan Turing introduced the idea of axiomatic computability into mathematics when, in 1935, defining the – now canonical – notion of a computable (recursive) function between natural numbers.

Finding a suitable notion of computability for other types, such as functions between real numbers, has been less canonical. Many sensible, interesting, but unfortunately non-equivalent notions have been studied. Turing himself widened the applicability of his idea of computability (already in his seminal paper that introduced computability) to other types by calling a real number x *computable*, if the sequence of digits of one of its decimal expansions is computable. There are some deficiencies to this definition, so nowadays in the Turing-style approach one generally uses some version of the following definition instead: a real number x is computable if there exist two total, computable (recursive) functions $f, g : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$

$$(1) \qquad \left| x - \frac{f(n) - g(n)}{2^n} \right| < 2^{-n}.$$

Similarly one can, for most other notions in analysis such as continuity of functions or metrics, find computable counterparts by defining them in terms of computable (recursive) functions between the natural numbers. Of course, not all theorems of analysis stay valid if one switches to computable notions and interprets them naïvely.

Notice, how one could have made all the "computable counterpart" definitions, not only for the class of computable (recursive) functions, but for an arbitrary class of functions $\mathcal{F} \subseteq \mathbb{N}^{\mathbb{N}}$. The idea of the talk is to do just this. The main question to answer is: Which assumptions do we need to impose on $\mathcal{F}$ in order to prove certain theorems in analysis?

Comparison of models of computation over the reals

Christine Gaßner

Ernst-Moritz-Arndt-Universität Greifswald
Institut für Mathematik und Informatik

Models of computation over the real numbers were developed for several purposes. The real RAM is a model for describing programs of computers in a simple language of a higher level and for providing a first specification of algorithms over the reals (see [Preparata Shamos '85]). The classical real RAM model is the most used model for analyzing the computational complexity of problems. However, it is a non-uniform model of computation and, in general, the program of a RAM works correctly only if the inputs are vectors of a finite dimensional vector space or other suitable vectors. L. Blum, M. Shub, and S. Smale introduced real Turing machines by means of flow charts which are able to process inputs of any length, and they created a uniform complexity theory over the reals (see [Blum et al. '89]). We know that many results of the non-uniform theory could and can be transferred from the classical real RAM model into their BSS model. In later years many similar so-called BSS models are defined. Our BSS model is a special BSS-RAM model (see [Gaßner '97]). It was introduced in order to combine the advantages of both models. The machines of the BSS-RAM model are register machines that allow us to analyze the complexity of uniform algorithms. However the uniform treatment of inputs implies specific features which are typical for Turing machines and independent of the properties of the objects to be processed. We will discuss some computational problems and compare the decidability results and the reducibility of one problem to another problem in order to show that the classical real RAM's and the BSS model really lead to different classes of decidable problems.

[Blum et al. '89] Blum, L., M. Shub, and S. Smale: "On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines"; Bulletin of the Amer. Math. Soc. 21 (1989), 1–46.

[Gaßner '97] Gaßner, C.: "On NP-Completeness for Linear Machines"; Journal of Complexity 13 (1997), 259–271.

[Preparata Shamos '85] Preparata, F. P., and M. I. Shamos: "Computational geometry: An introduction"; Springer (1985).

On algebraic decision trees and the generic path method

Paul Grieger

Ernst-Moritz-Arndt-Universität Greifswald
Institut für Mathematik und Informatik

Algebraic decision trees and the generic path method are a graphical way of proving and their use will be demonstrated on a solution for a P-DNP-Problem, where the classes P and DNP are dedicated to a BSS-RAM, that only does allow linear operations and tests of equation. The solution is based on Meer [Meer '92] and is transferred to the BSS-RAM-model based on [Gaßner '01]. It will be shown that the problem of whether a (real) system of linear equations has a solution consisting of zeroes and ones only is the witness for the negative answer to the P-DNP-Problem. The testing, whether the problem belongs to the class DNP, is quite straight forward, but in proving the non-membership of P, the generic path method will be explained.

[Meer '92] Meer, K.: "A note on a $P \neq NP$-result in a restricted class of real machines"; Journal of Complexity 8 (1992), 451–453.

[Gaßner] Gaßner, C.: "Das Berechnungsmodell http://stubber.math-inf.uni-greifswald.de/biomathematik/gassner/forschung/forschung1.htm".

[Gaßner '01] Gaßner, C.: "The P-DNP problem for infinite abelian groups"; Journal of Complexity 17 (2001), 574–583.

Non-determinism in TTE

Arno Pauly

Computer Laboratory
University of Cambridge, United Kingdom

Non-deterministic Type-2 Machines as suggested by Ziegler [Ziegler '07 B] and studied in more detail by Brattka, de Brecht and P. [Brattka et al.], [Brattka Pauly '10] offer a versatile family of computation models. In addition to the input, a non-deterministic Type-2 machine (NDTM) may guess an element $z \in \mathbf{Z}$ for some represented space $\mathbf{Z}$, and either continues to compute a correct output from the input and an arbitrary name of z, or it eventually rejects the guess z, with the condition that for valid input there has to be an acceptable guess. The power of a NDTM significantly depends on the choice of the advice space $\mathbf{Z}$. The advice space also influences properties such as closure under composition.

Typically it is rather easy to prove that a given multi-valued function is computable by a NDTM - often it is even obvious that guessing and verifying the output works. Such a result provides both an upper bound for the degree of incomputability as well as a recipe how to obtain results about computability. The probably most useful one is the following: A multi-valued function between computable metric spaces computable by a NDTM with a computably compact computable metric space as advice space becomes computable when restricted to the points where the values are unique, this follows from a result by Brattka and Gherardi [Brattka Gherardi '11]. This approach was used (not explicitly!) by Galatolo, Hoyrup and Robas in the realm of computable measure theory [Galatolo et al. '11], and by Rettinger for computable differential geometry [Rettinger '11].

Non-determinism with advice space $\mathbb{N}$ is equivalent to revising computation, and hence is related to BSS-machines [Ziegler '07 A]. It is also possible to conceive of non-deterministic machines as the algorithmic content of certain extensions of intuitionistic logic, e.g. those obtained by including axioms such as LPO and LLPO.

[Brattka et al.] Brattka, V., M. de Brecht & A. Pauly: "Closed Choice and a Uniform Low Basis Theorem"; Annals of Pure and Applied Logic. To appear, available at arXiv:1002.2800.

[Brattka Gherardi '11] Brattka, V. & G. Gherardi: "Weihrauch Degrees, Omniscience Principles and Weak Computability", Bulletin of Symbolic Logic 17, 73 – 117. ArXiv:0905.4679.

[Brattka Pauly '10] Brattka, V. & A. Pauly: "Computation with Advice" (2011). ArXiv:1006.0395.

[Galatolo et al. '11] Galatolo, S., M. Hoyrup & C. Robas: "Dynamics and abstract computability: computing invariant measures"; Discrete and Continuous Dynamical Systems 29 (2011) (1).

[Rettinger '11] Rettinger, R.: "Compactness and the Effectivity of Uniformization". Talk at CCA 2011.

[Ziegler '07 A] Ziegler, M.: "Real Computability and Hypercomputation" (Habilitationsschrift); University of Paderborn (2007).

[Ziegler '07 B] Ziegler, M.: "Real Hypercomputation and Continuity"; Theory of Computing Systems 41 (2007), 177 – 206.

Complexity theory for operators in analysis

Carsten Rösnick

Technische Universität Darmstadt
Fachbereich Mathematik

The computability of numerical operators (functions of signature $F : (\mathbb{R} \to \mathbb{R}) \to (\mathbb{R} \to \mathbb{R})$) and numerical functionals (functions of signature $F : (\mathbb{R} \to \mathbb{R}) \to \mathbb{R}$ like $\textsc{Max}_{[0,1]} : f \mapsto \max_{x\in[0,1]} f(x)$) can be formalized in models like/terms of TTE (Weihrauch) and type-2 oracle machines (Ko/Friedman).

A TTE-machine transforms infinite sequences (of, e. g., rationals) into infinite sequences on a single one-way ouput tape. The restriction for the ouput tape to be one-way ensures that after the machine has performed T steps (for any $T \in \mathbb{N}$), the intermediate result (depending on T) will not be altered in any future iteration of the machine.

Given a natural $n \in \mathbb{N}$, a type-2 oracle machine computing a functional F can ask its oracle questions like "give me an approximation of $f(q)$ for $q \in \mathbb{Q}$ with precision m (=absolute error 2^{-m})" to eventually compute an approximation to $F(f)$ of absolut error 2^{-n}.

However, both models do not lead to a (satisfying) complexity notion for such operators and functionals. In particular for the type-2 oracle approach, the statements are not even effective (i. e., of form "operator O is in class $\mathcal{C}$"). It is the approach by Kawamura and Cook [Kawamura Cook '10] that tries to fill (in particular) this gap, thus giving a complexity notion for operators and functionals. It relies on names of functions being *regular functions*: That is, on functions for which there is a well-defined size function that only depends on the length of the argument n. In contrast to TTE and type-2 oracle machines, this allows defining the complexity of an operator not only in the input (precision n), but also in the *size* of a name for the argument (here: a computable real function).

This talk will range from the basic notions of computable real numbers and computable real functions up to the definition of computable operators. We will see how the definition of Kawamura/Cook complements those by Weihrauch and Ko/Friedman, and see basic examples of proofs of effective statements for numeric functionals, like $\textsc{Max}$ and $\textsc{Int}$.

[Kawamura Cook '10] Kawamura, A. and S. Cook: "Complexity theory for operators in analysis"; In Proceedings of the 42nd ACM symposium on Theory of computing, 495-502. ACM, 2010.

On Lower Bounds for Algebraic Decision Trees

Peter Scheiblechner

Hausdorff-Zentrum für Mathematik
Bonn

We will discuss the use of topological and algebro-geometric methods in proving lower bounds for algebraic decision (or computation) trees solving the membership problem of a given semi-algebraic set in $\mathbb{R}^n$ or algebraic set in $\mathbb{C}^n$. We will present some lower bounds in terms of several topological invariants like the number of connected components [Ben-Or '83], Euler characteristic [Yao '92], and Betti numbers [Yao '94, Scheiblechner '10], and applications thereof, see also the survey [Bürgisser '01]. We will introduce and explain all relevant notions and techniques from topology and algebraic geometry.

[Ben-Or '83] M. Ben-Or, M.: "Lower bounds for algebraic computation trees"; STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing, New York, ACM (1983), 80–86.

[Bürgisser '01] Bürgisser, P.: "Lower bounds and real algebraic geometry"; Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science (2001), 35–54.

[Scheiblechner '10] Scheiblechner, P.: "On lower bounds for algebraic decision trees over the complex numbers"; Poceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Los Alamitos, IEEE Computer Society (2010), 362–365.

[Yao '92] Yao, A.: "Algebraic decision trees and Euler characteristics"; FOCS '92: Proceedings of the 33rd Annual Symposium on Foundations of Computer Science, Washington, IEEE Computer Society (1992), 268–277.

[Yao '94] Yao, A.: "Decision tree complexity and Betti numbers", STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, New York, ACM (1994), 615–624.

Automata on ordinals and linear orders

Philipp Schlicht

Rheinische Friedrich-Wilhelms-Universität Bonn
Mathematisches Institut

I will introduce finite automata with running time a limit ordinal α and consider structures recognizable by such automata whose domain consists of finite words of length α with gaps. The extra ingredient for these automata is a limit rule which maps the set of states which appear cofinally often before the limit to a limit state. Such structures share some properties with structures recognized by finite automata. I will sketch how to determine the suprema of the α-automatic ordinals and the ranks of α-automatic linear orders and show that the power of α-automata increases with every power of ω.

A short review on the history of register machines

Isabel Schwende

Ernst-Moritz-Arndt-Universität Greifswald
Institut für Mathematik unfd Informatik

We want to give a short review on the history of register machines starting with the modified Turing machine by Wang in 1957. The development of real computers beginning in the early 60s justified the importance of register machines and therefore many new models were invented which are based on Wang's work.

We have chosen Kaphengst's from amongst lots of models of register machines published in that time to illustrate the rapprochement of the model and digital computers. Thereafter, to end the diversity of different definitions, Shepherdson and Sturgis gathered some of them and blended them into own models: limited register machine and unlimited register machine. They also have compared and evaluated several previous papers. In conclusion we want to give a short prospect on the following progress in the idea of RAM.

[Wang '57] Hao Wang: "A Variant to Turing's Theory of Computing Machines"; JACM (Journal of the Association for Computing Machinery) 4 (1957), 63–92.

[Kaphengst '59] Heinz Kaphengst: "Eine Abstrakte programmgesteuerte Rechenmaschine"; Zeitschrift für mathematische Logik und Grundlagen der Mathematik: 5 (1959), 366–379.

[Shepherdson Sturgis '63] John C. Shepherdson and H. E. Sturgis: "Computability of Recursive Function"; Journal of the Association of Computing Machinery (JACM) 10(1963), 217–255.

Computational complexity of quantum satisfiability

Martin Ziegler

Technische Universität Darmstadt
Fachbereich Mathematik

Quantum logic generalizes, and in dimension one coincides with, Boolean logic. We show that the satisfiability problem of quantum logic formulas is NP-complete in dimension two as well. For higher higher-dimensional spaces R^d and C^d with $d > 2$ fixed, we establish quantum satisfiability to be polynomial time equivalent to the real feasibility of a multivariate quartic polynomial equation: a problem well-known complete for the counterpart of NP in the Blum-Shub-Smale model of computation lying (probably strictly) between classical NP and PSPACE. We finally investigate the problem over INdefinite finite dimensions and relate it to the real feasibility of quartic NONcommutative *-polynomial equations. (joint work with Christian Herrmann...)