Dagstuhl-Seminar 9717 on

Computability and Complexity in Analysis

Schloß Dagstuhl, April 21–25, 1997

Organizers:

Ker-I Ko (New York),
Anil Nerode (Cornell),
Klaus Weihrauch (Hagen).

Introduction

The seminar "Computability and Complexity in Analysis" was held in Dagstuhl from April 21 to 25, 1997. It was attended by 43 computer scientists, mathematicians and logicians who are interested in the diverse areas of computational analysis. The 33 talks presented here represented many different theories and approaches in this topic.

In the foundational study, they included constructive analysis (of E. Bishop), recursive analysis, type-2 theory of effectivity and the domain-theoretic approach. Classical theories on real-valued functions have been extended to more general functional spaces. Specific problems in computable analysis, such as the Riemann mapping theorem and problems in calculus of variations, have also been presented. In the complexity study, a number of talks were about the modifications and comparisons of three different models: Turing machine-based complexity theory, information-based complexity theory (IBC) and real RAM-based complexity theory (BSS). More practical issues, including interval analysis, automatic result verification, bounded-value problem, on-line computation and geometric and stochastic modelling, were also presented.

Computing the Range of Real Functions Using Interval Arithmetic Tools

Götz Alefeld
Universität Karlsruhe, Germany

It is well known that the range of a real function can be included by simply evaluating the given expression in the interval arithmetic sense. In the talk basic facts concerning the distance of this evaluation to the range of the function are repeated. The so–called centered form and the question of higher order approximation of the range are discussed. Finally some applications of interval arithmetic evaluations are used in showing the existence and nonexistence of zeros of functions.

Computing with Closed Subsets of the Euclidean Space

Vasco Brattka
FernUniversität Hagen, Germany
(joint work with Klaus Weihrauch)

In this talk computability of closed subsets of the Euclidean space is presented in the framework of Type 2 Theory of Effectivity. Two basic representations of the hyperspace, that is the set of closed subsets are introduced and combined and several characterizations are presented. The induced computable objects are the recursive, the recursively enumerable and the co-recursively enumerable closed sets, which appear as natural generalizations of the corresponding discrete sets from recursion theory. The tools of Type 2 Theory of Effectivity enable higly effective versions of classical theorems: as an example an effective Lemma of Urysohn is presented. Furthermore, the computability of several hyperspace operators, such as the border operator and the union operator is studied. The non-computability of the “complement closure operator” leads to stronger representations and to the corresponding notions of bi-recursive and double bi-recursive sets.

Was soll, und was ist wahrlich, die konstruktive Mathematik?

Douglas Bridges
University of Waikato, Hamilton, New Zealand

Constructive mathematics, as expounded by Brouwer, Markov, Bishop, and other pioneers, arose from their algorithmic interpretation of existence. However, as Richman and others have observed, our practice of constructive mathematics (Bishop–style) is really just ordinary mathematics with intuitionistic logic. This interpretation of our work avoids any need to talk of "algorithms", but leaves open the ability to translate our theorems IMMEDIATELY into theorems of intuitionism, recursive mathematics, and classical mathematics.

The second part of the talk introduced axioms for the real line, chosen to reflect the constructive properties obtained from Bishop's Cauchy–sequence development of the reals in constructive analysis.

Do the Zeros of Riemann's Zeta–Function Form a Random Sequence?

Cristian Calude
University of Auckland, New Zealand
(joint work with Peter Hertling and Bakhadyr Khoussainov)

With the identification "zeros of Riemann's zeta–function=energy levels" and "logarithms of primes=lengths of periodic orbits" M. Berry and other physicists have been able to use Riemann's zeta–function to model quantum mechanical chaos in an attempt to bridge the apparently incompatible chaotic and quantum mechanical descriptions of the microscopic world. This is one of the motivations for the question in the title, which will be answered in the *negative*. To achieve this result we define and study the notion of *random sequence of reals*. Open problem: replace "random" by "pseudo–random" in the question in the title; can we get a positive answer?

Π^0_1 Classes and Index Sets in Computable Analysis

DOUGLAS CENZER
University of Florida, USA
(joint work with JEFFREY REMMEL)

The role of Π^0_1 classes in computable analysis is examined. For any computably continuous real function F, the following sets are all Π^0_1 classes:

(1) The Graph of F.

(2) The set of zeroes of F.

(3) The set of maximum/minimum points of F.

(4) The set of fixed points of F.

The Π^0_1 classes are given an effective enumeration P_e and the computably continuous functions are given indices: F_e is the function with index e. The notion of index sets associated with Π^0_1 classes is developed and with computably continuous functions is developed. For example, the set I_F of indices for the computably continuous functions is itself is Π^0_2 complete.

The complexity of various problems is determined by the complexity of the index set. For any property $\mathcal{R}$, let

$I_P(\mathcal{R}) = \{e : P_e \text{ has property } \mathcal{R}\}$ and

$I_F(\mathcal{R}) = \{e \in I_F : F_e \text{ has property } \mathcal{R}\}$

Here is a partial list of results.

Theorem: I_P(measure one) is Π^0_1 complete.

Theorem: I_P(meager) is Π^0_2 complete.

Theorem: I_P(singleton) is Π^0_2-complete.

Theorem: I_P(finite) is Σ^0_3 complete.

Theorem: I_P(countable) is Π^1_1 complete.

Theorem: I_P(finitely many computable members) is Σ^0_4 complete.

Theorem: I_F(unique zero) is Π^0_2-complete.

Theorem: I_F(finitely many minima) is Σ^0_3-complete.

Theorem: I_F(finitely many computable fixed points) is Σ^0_4-complete.

Exact Real Number Computation Using Linear Fractional Transformations

Abbas Edalat
Imperial College, London, United Kingdom
(joint work with Peter John Potts)

We introduce a feasible and incremental framework for exact real number computation which uses linear fractional transformations (lft) with non-negative coefficients. It is based on previous work by Gosper, Vuillemin, Nielson and Kornerup and it leads to the notion of exact floating point which unifies the previous approaches in this subject. Our representation of real numbers is based on the infinite composition of lft's, equivalently the infinite product of matrices, with non-negative coefficients. Any rational interval in the one point compactification of the real line, represented by the unit circle, is expressed as the image under an lft of the non-negative extended real numbers. A sequence of shrinking nested intervals is then represented by an infinite product of matrices with integer coefficients such that the first so-called sign matrix determines an interval on which the real number lies. The subsequent, so-called digit, matrices have non-negative integer coefficients and successively refine that interval. We show that there is a canonical choice of four sign matrices which are generated by rotation of the unit circle by 90 degrees. Furthermore, the ordinary signed digit representation of real numbers in a given base induces a canonical choice of digit matrices. A library of efficient algorithms for elementary functions has been developed in this framework.

Effective Content of the Calculus of Variations

Xiaolin Ge
Anasazi Inc., Phoenix, USA
(joint work with Anil Nerode)

The content of existence theorems in the calculations of variations has been explored and an effective treatment of semi–continuity has been achieved. An algorithm has been developed which captures the natural algorithmic content of the notion of a semi–continuous function and this is used to abtain an

effective version of the "Chattering Lemma" of control theory and ordinary differential equations. An effective version of Ekeland's Principle in control theory has also been investigated.

Algebraic (BSS–like) Computability, Generally and over the Reals

Armin Hemmerling
Universität Greifswald, Germany

We report some basic concepts and results concerning the computability of string functions over the universe of an arbitrary (single–sorted, total) algebraic structure. Our model of computation is an adaptation of the BSS setting, but it is preferably required to be parameter–free; i.e, only the structure's base constant are allowed as direct operands. This approach yields a rich theory including some features of classical recursion theory as well as results relativizing the classical situation. The framework even allows to deal with structures of infinite signature, for which the existence of universal functions and of m–complete sets are discussed.

The satisfiability problem of quantifier–free first–order formulas with parameters over the structure turns out to be NP–complete with respect to nondeterminism by element guessing. Another NP–complete problem is the satisfiability of algebraic circuits over the structure. The binary satisfiability of such circuits is NP–complete with respect to nondeterminism by program branching.

Based on the standard encodings of rational numbers by finite objects, the classical computability leads to a natural concept of computability for functions over the rational numbers. This notion does not imply continuity. It is equivalent to algebraic (i.e., BSS-) computability over the rationals. Applying the same class of programs to real numbers, we obtain the concept of parameter–freely BSS–computable real functions. It is characterized and compared with the class of computable functions in the sense of constructive analysis. The computable real numbers (in the classical sense) are just those reals whose singletons are co-recognizable by parameter–free programs.

The Effective Riemann Mapping Theorem

PETER HERTLING
University of Auckland, New Zealand

The aim of the talk is to present two effective versions of the Riemann mapping theorem. The first, uniform version is based on the constructive proof of the Riemann mapping theorem by Bishop and Bridges [E. Bishop, D. Bridges: Constructive Analysis, Springer-Verlag, 1985, Ch. 5.7] and on the computability framework developed by Weihrauch (see e.g. [K. Weihrauch: Computability, Springer-Verlag, 1987, Part 3]). It states which information precisely one needs about a proper, open, connected, and simply connected subset of the complex plane in order to compute a description of a conformal isomorphism from this set onto the unit disc, and vice versa, which information about the set can be obtained from a description of a conformal isomorphism. The second version, which is derived from the first by considering the sets and the functions with computable descriptions, characterizes the subsets of the complex plane for which there exists a computable conformal isomorphism onto the unit disc. This solves a problem posed by Pour-El and Richards [M. B. Pour-El, J. I. Richards: Computability in Analysis and Physics, Springer-Verlag, 1989, Problem 5]. We also show that this class of sets is strictly larger than a class of sets considered by Zhou [Q. Zhou: Computable real-valued functions on recursive open and closed subsets of Euclidean space, Math. Log. Quart. 42 (1996), 379-409], thus giving a negative answer to the question formulated in his Problem 5.4.

$\mathbb{Q}$–Analytic Machines – Concepts and Problems

GÜNTER HOTZ
Universität des Saarlands, Saarbrücken, Germany

$\mathbb{Q}$–machines in our sense are in great parts $\mathbb{Q}$–machines in the sense of Blum, Shub and Smale. But the input tape has $\mathbb{R}$ as alphabet and is finite. The output tape is infinite and has $\mathbb{Q}$ as alphabet. The machine has additionally a "precision" register δ; it reads dependent from δ a rational number q_δ when q is under the head of the input tape such that $|q - q_\delta| < 2^{-n}$ if $\delta = n$. An accepted computation is an infinite computation passing infinite often through the "final" state. In the final state we have the instruction $\delta := \delta + 1$. The machine "halts" per defintion, if the output tape converges. This is the $\mathbb{Q}$–analytic machine. Functions, which can be computed by this machines, are our $\mathbb{Q}$–analytic functions. *Theorem 1:* f $\mathbb{R}$–computable $\Longrightarrow f$ $\mathbb{Q}$–analytic. The converse is not true. Dynamical systems can be modeled by $\mathbb{Q}$–analytic machines. *Theorem 2:* The stability problem is not generally decidable. The inital value problems for differential equations $y' = f(x, y), x \in \mathbb{R}$, $y \in \mathbb{R}^n$ have solutions for f $\mathbb{R}$–computable which are $\mathbb{Q}$–analytic. ($\mathbb{R}$ is here a ring (without division)). The solutions in general are not unique. One can describe in which way the solutions branch and eventually meet again. It is possible to describe this by a graph dependent from the standard representation $f = \sum_\sigma a_\sigma \cdot f_\sigma$ of f [B.S.S.]. *Motivation:* This theory is thought of as a language to describe hybrid systems in the sense of Nerode; i.e. systems of interaction of digital processors with continuous systems.

A Constructive Criterion for Sequential Continuity for Linear Mappings

Hajime Ishihara
Japan Advanced Institute of Science and Technology, Japan
(joint work with Douglas Bridges)

We show the following generalized version of Hellinger–Toeplitz theorem in Bishop's constructive mathematics. *Theorem:* Let T be a linear mapping of a Banach space X into a normed space Y such that if $a_n \longrightarrow 0$, then $f(Ta_n) \longrightarrow 0$ for all $f \in Y^*$. Then T is sequentially continuous.

Feasibly Continuous Type–Two Functionals

Bruce Kapron
University of Victoria, Canada

We propose a poly–time analog of the result from classical recursion theory (due independently to Nerode and Uspenskii) which equates type–two computability relative to an oracle with continuity on Baire space. We show that the analogous result is true for Mehlhorn's formulation fo type–two poly–time, and discuss the difficulties in proving such a result for the Kapron–Cook formulation.

Control Theory, Modal Logic, and Games

JULIA F. KNIGHT
University of Notre Dame, USA
(joint work with BRIAN LUENSE)

We consider a class of discrete systems, with specifications stated in a certain modal temporal language (chosen for simplicity). We show that if the regulator can in some way guarantee satisfaction of a specification, then it can do so acting as a deterministic finite automaton, and we can effectively find an appropriate automaton, or determine that there is none. Our result is not really new. It (and similar results for more expressive languages) can be obtained easily from a result of Landweber and Büchi on "regular" games, together with the fact that our language gives rise to games of this sort. We use the Forgetful Determinacy Therorem of Gurevich and Harrington to show the existence of appropriate automata. The actual construction is explicit. The set of states is determined through elementary considerations of which partial records might be useful to the regulator.

Polynomial–Time Computability in Analysis

KER–I KO
Department of Computer Science, Stony Brook, USA

We present a survey of complexity theory of real functions in the Turing machine model. Following issues are discussed:

(a) For a notion of computable objects, many formulations which are known to be equivalent in recursive analysis are not necessarily equivalent in polynomial–time complexity theory.

(b) Negative results in recursive analysis can often be carried over to similar negative results in polynomial–time complexity theory.

(c) The proofs for many lower bound results in polynomial–time complexity theory use a different form of reductions that reduce discrete problems to continuous functions.

(d) When real numbers are considered as discrete objects, they provide a rich structure in discrete complexity theory.

Numerics with Automatic Result Verification – from Numerical Mathematics towards Mathematical Numerics

ULRICH KULISCH
Universität Karlsruhe, Germany

Advances in computer technology are now so profound that the arithmetic capability and repertoire of computers can and should be expanded. The quality of the elementary floating–point operations should be extended to the most frequent numerical data types or mathematical spaces of computation (vectors, matrices, complex numbers and intervals over these types). A VLSI co–processor chip with integrated PCI–interface has been developed which provides these operations. The expanded capability is gained at modest hardware cost and does not implicate a performance penalty. Language and programming support (the –XSC languages) are available. There, the chip's functionality is directly coupled to the operator symbols for the corresponding data types. By operator overloading a long real arithmetic (array of reals) and long interval arithmetic as well as automatic differentiation arithmetic become part of the runtime system of the compiler. I.e. derivatives, Taylor–coefficients, gradients, Jacobian and Hessian matrices or enclosures of these are directly computed out of the expression by a simple type change of the operands. Techniques are now available so that with this expanded capability, the computer itself can be used to appraise the quality and the reliability of the computed results over a wide range of applications. Program packages for many standard problems of Numerical Analysis have been developed where the computer itself verifies the correctness of the computed result and proves existence and uniqueness of the solution within the computed bounds.

Many applications require that rigorous mathematics can be done with the computer using floating–point. As an example, this is essential in simulation runs (fusion reactor) or mathematical modeling where the user has to distinguish between computational artifacts and genuine reactions of the model. The model can only be developed systematically if errors entering into the computation can be essentially excluded. Automatic result verification re–integrates digital computing into real mathematics.

Computability and Geometric Modelling

André Lieutier
MATRA Datavision SA, Aix en Provence, France

CAD/CAM software uses boolean operations on BRep (Boundary Representation) algorithms. These algorithms are proved to be correct with the assumption of a real RAM machine. Actual software relying on floating point is not 100% reliable. We use a mapping between Brep to $L^1(\mathbb{R}^3)$ defined by "winding number". The natural ordering and L^1 norm or L^1 induce an ordering and distance on a quotient of BRep, allowing "self intersecting" BRep, i.e. BRep for which the winding number can take any integer value. The set of polyhedrals whose vertices are rational is a countable subset of BRep for which the boolean operations are computable and closed (boolean operation of rational polyhedral is a rational polyhedral). Then we use enclosing sequences of rational polyhedrals as a representation of BRep to show that boolean operations on BRep are computable. This is not a realistic algorithm for industy but it is close to actual algorithms.

Three Metatheorems on Continuity and Computability

Leonid P. Lisovik
Kiev National University, Ukraine

The notion of continuous real function can be represented by R–transducers. These transducers are one–way machines with input and output tapes, where symbols $0, 1, \bar{1}, \bigtriangledown$ are used as input symbols, and among output symbols there are two additional digits 2 and $\bar{2}$. In the general case R–transducer has infinite memory, so the set of its states is infinite. R–transducer is a deterministic machine. It can be compared with nondeterministic R–transducer and R–transducer with oscillation on output tape. The classification of continuous real functions can be connected with classification of R–transducers. Many interesting examples of functions defined by finite R–transducers have been constructed (nowhere differentiable continuous function, Peano's curve and so on). R–transducers can be presented in form of infinite computation labelled trees. Transducers which are working over such labelled trees were called macrotransducers. Functionals and operators in functional spaces can be represented as macrotransducers over computation trees. Classification of macrotransducers generates the classification of continuous and discontinuous functionals. The three following metatheorems can be emphasized. The first one is the algebraic characterization of partially computable real functions. The second one is an implication "continuity implies computability". For example, if function f is continuous and it can be defined by pseudo–algorithmic S–transducer then it will be defined by appropriate algorithmic S–transducer. The third metatheorem is the Kreisel–Lacombe–Shoenfield theorem (KLS) and Ceitin's theorem and its generalizations. Some generalizations of these theorems are connected with R–transucers. Recently we have obtained a simple proof of the Kreisel–Lacombe–Shoenfield theorem.

Stochastic Modelling and Computability — A Case Study

Norbert Müller
Universität Trier, Germany

As an example of the applicability of TTE in queueing theory, we study computational aspects of the well known M/G/1 queueing system.

Using an appropriate and natural representation of equivalence classes of random variables, we are able to show that all computational problems involved are tractable, leading to a result expressing conditions for an effective solution.

Research Issues in Recursive Analysis

Anil Nerode
Cornell University, Ithaca, USA

At present there are several quite different languages for describing recursive or constructive analysis. The most promising are: higher order intuitonistic logics, topos and category theory, untyped and typed λ–calculi. The relations between these subjects, how they are embedded in each other, has been the subject of much research, which every logician should know. Here we are interested in pointing out the importance of recursive realizability and λ–term interpretation. This is because the paradigm of Constable and Martin Löf, that constructive proofs lead algorithmically to programs that compute whatever is asserted to exist from the data provided, can be best understood as a λ–term or recursive realizability interpretation of higher order constructive logics of the kind mentioned above. So, for instance, any of the recursive realizability interpretations applied to Bishop's book, shows that they hold in recursive analysis, and giving a recursive realizability interpretation with φ valid, ψ not valid, shows ψ is not a constructive consequence of φ. Viewed in such a large set–theoretic system of constructive mathematics as intuitionistic Zermelo–Fraenkel, this means that such term interpretations are a tool for both constructive proofs and constructive counter examples, in analogy with independence in ordinary set theory.
We express our conviction that eventually extraction of programs from program specification will be the premier tool for obtaining verified programs, and that constructive systems as mentioned above will play a fundamental role. These systems will have to be refined to finer systems (such as p–time intuitionistic analysis) to be successful.

On the ε–Complexity of BVPs

Erich Novak
Universität Erlangen, Germany

How many arithmetic operations and function evaluations are necessary to compute an ε–approximation (in the H_1–norm) of the solution of $-(ru')'+qu = f$ on $[0,1]$ with $u(0) = u(1) = 0$? For simplicity we assume that

$r, q \in C^\infty$ and $r > 0$ and $q \geq 0$ and $f \in C^m$ with $m \in \mathbb{N}_0$. With a finite element method with quadrature (FEMQ) one can achieve an error $e \leq c \cdot n^{-m}$ where n is the number of arithmetic operations and function evaluations. This is almost optimal since there is a lower bound $e \geq \tilde{c} \cdot n^{-m}$ valid for every method. This result holds in the real number model with exact function values. Special case: for $m = 0$ a small error cannot be achieved by any method – the problem is unsolvable. If each input has an error bounded by δ (now we can use a finite input in the bit number model) then we still get $e \leq c \cdot (n^{-m} + \delta)$ and again this is optimal because we obtain $e \geq \tilde{c} \cdot (n^{-m} + \delta)$ for every method. The multivariate case (BVPs for elliptic PDEs) is much more difficult in a technical sense but Werschulz (1996, 1997) could obtain analogous results. Now the optimal bound $e \asymp n^{-m}$ is replaced by $e \asymp n^{-m/d}$ (d = dimension) and again this can be achieved by stable algorithms.

Computability in Analysis

Marian Boykan Pour-El
University of Minnesota, Minneapolis, USA

The purpose of this talk is two-fold:

- To survey the results in recursive analysis from a somewhat different but fruitful point of view
- To present some new results

The survey begins with a discussion of the following three topics:

- The definition of a recursive real number
- Computability for elements and sequences of elements of $C[a, b]$ – where a and b are recursive reals
- $L^p[a, b]$ – computability, $1 \leq p < \infty$ (a, b, p recursive reals)

Note that $C[a, b]$ and $L^p[a, b]$ are *Banach spaces. This leads to a definition of computability on an arbitrary Banach space.* A major portion of the survey is devoted to the consequences of this definition. They can be summarized in three theorems. Under general conditions which, in practice, are satisfied:

First Main Theorem: Bounded linear operators from one Banach space to another always preserve computability, unbounded linear operators do not.

(The First Main Theorem explains why wave propagation can be noncomputable even though the initial conditions which determine the propagation are computable. Wave propagation is associated with an unbounded operator.)

Second Main Theorem: The eigenvalues of a self adjoint linear operator on Hilbert space are computable reals. However, it may not be possible to arrange them in a computable sequence.

(Actually, the second Main Theorem is considerably broader in scope. In addition, it has some corollaries of independent interest. For further details, see *Computability in Analysis and Physics* by M.B. Pour-El and I. Richards, Springer-Verlag 1989.)

Eigenvector Theorem: Let $H = L^2[0,1]$ with its usual computability structure. There exists an effectively determined compact self adjoint operator T mapping H into H such that 0 is an eigenvalue of T, but none of the eigenvectors corresponding to $\lambda = 0$ is computable.

The new results concern the introduction of the theory of degrees of unsolvability from recursion theory into recursive analysis. They represent work done jointly with Anthony Dunlop.

Online Computations of Differentiable Functions

Matthias Schröder
FernUniversität Hagen, Germany

It is shown that Lipschitz–continuously differentiable functions $f : [-1;1]^k \to \mathbb{R}$ which are computable in regular time $\mathcal{O}(T(n))$ can be computed by an online algorithm in time $\mathcal{O}(T(n) + \mathcal{M}(n) \cdot \log_2(n))$, where $\mathcal{M}(n) = n \cdot \log_2(n) \cdot \log_2 \log_2(n)$ is the Schönhage–Strassen–bound for n–Bit integer multiplication on a Turing machine. The time is measured by the number of steps necessary to produce the first n fractional digits of the result represented by the binary signed digit representation. An algorithm is called online with delay δ, iff for producing the first n digits to the right of the binary point of the result the algorithm only needs the first $n + \delta$ fractional digits of the k input sequences representing the k input reals. The proof bases on an efficient online multiplication algorithm presented on STACS 97.

Sets of Real Numbers Presentable by Turing Machines

Ludwig Staiger
Universität Halle, Germany

The talk considered expansions of real numbers. Sets of real numbers were considered as sets of expansions, that is, as sets of infinite strings (ω–words). In the literature ([Wagner/Staiger 77], [Cohen/Gold 78], [Engelfriet/Hoogeboom 93]) several different conditions for accepting set of ω–words (ω–languages) by Turing machines were introduced. These modes differ mainly in taking into account the behaviour of the accepting machine on the input tape. In the talk the classes of accepted ω–languages were compared and put into the context of recursion theory.

Topological Algebras and Domain Representability

Viggo Stoltenberg–Hansen
Uppsala University, Sweden
(joint work with J.V. Tucker and Jens Blanck)

Some work on computability of topological algebras in terms of approximations is surveyed. The method used is domain representability. A topological algebra A is represented by (D, D^R, φ) where D is a structured algebraic domain, D^R is a substructure of D and $\varphi : D^R \longrightarrow A$ is a quotient epimorphism. The algebraic domain D is obtained as the ideal completion of an approximation structure for A, that is, the representation is determined by the chosen approximations. It is shown that the method of domain representability is equivalent to Weihrauch's method of type two enumerations (TTE) using the Baire space.

A Domain-Theoretic Approach to Computable Analysis

Philipp Sünderhauf
Imperial College, London, United Kingdom
(joint work with Abbas Edalat)

We use basic ingredients of an effective theory of continuous domains as a foundation for computable analysis. Continuous domains provide a convenient framework to capture the notion of partial information.

We investigate computability on the real line and on Banach spaces within our setting. Our approach turns out to be equivalent to the approach of (Pour-El & Richards '88).

Computability Theories for Topological Algebras

J. V. Tucker
University of Wales Swansea, United Kingdom
(joint work with V. Stoltenberg-Hansen and J.I. Zucker)

The lecture surveyed and compared concrete computability theories (such as those based on classical computable analysis and its generalisations) and abstract computabilities (such as those based on programming languages and machine models over many sorted algebras).

New results by the speaker, V Stoltenberg-Hansen (Uppsala) and J I Zucker (McMaster) were reported.

In joint work with V Stoltenberg-Hansen, the axiomatic approach to computability in Banach spaces of Pour-El and Richards has been generalised to many sorted metric algebras. The computability of homomorphisms has been studied and some new characterisation theorems proved. The resulting computability theory is equiavlent to several other notions such as those of algebraic domain representability in the sense of the authors and type 2 computability in the sense of Weihrauch.

In joint work with J I Zucker, the functions on many sorted metric algebras that are approximable by while-array propgrams have been studied. The functions have been shown to be those that are polynomial approximable in many situations. Applications to the problem of connecting concrete and abstract computability theories were presented.

δ-Uniform BSS Machines

SEBASTIANO VIGNA
Universitá di Milano, Italia
(joint work with PAOLO BOLDI)

A δ-uniform BSS machine is a standard BSS machine which does not rely on exact equality tests. We prove that, for any real closed archimedean field R, a set is δ-uniformly semi-decidable iff it is open and semi-decidable by a BSS machine which is locally time bounded. Moreover, we show that the sets semi-decidable by Turing machines are the sets semi-decidable by δ-uniform machines with coefficients in $\mathbf{Q}$ or $\mathbf{T}$, the field of Turing computable numbers. Then, a notion of closure related to Turing machines is defined for archimedean fields, and we show that such fields admit nontrivial δ-uniformly decidable sets iff they are not Turing closed. Finally, the partially ordered set of Turing closed fields is proved isomorphic to the ideal completion of the partially ordered set of unsolvability degrees.

A Foundation for Computable Analysis

KLAUS WEIHRAUCH
FernUniversität Hagen, Germany

A general Type 2 Theory of Effectivity, TTE, which is based on A. Grzegorczyk's definition of computable real functions is presented. TTE intends to characterize and study exactly those functions, operators etc. known from Analysis, which can be realized correctly by digital computers. After a short general introduction to basic principles, the applicabiliy of TTE is shown by several selected examples. Computability on the real numbers, on open sets, compact sets and functions on the set of real numbers is discussed. The problem of zero–finding for continuous functions is considered. Finally computational complexity is introduced and computability in measure theory is sketched.

A Realistic Machine Model for Scientific Computing

Jiří Wiedermann
Institute of Computer Science, AS CR, Prague, Czech Republic
(joint work with Friedhelm Meyer auf der Heide)

Numerical RAM is a parametrized modification of a standard RAM that is suitable to approximate computations over real numbers. The reals are represented by floating point numbers with a fixed number of significant digits only, i.e., with a bounded relative precision that is a parameter of the model. Besides standard RAM operations over integers numerical RAM is equipped with a set of finite precision floating point arithmetic operations and three-way floating point comparisons. The outcome of a floating point comparison of two floating point numbers can be "definitely greater", "definitely smaller", or "approximatively equal". The "tolerance" of such comparisons to relatively bounded perturbations in values of floating point numbers to be compared presents the second parameter of the model. The resulting RAM makes use of unit cost criterion.

It is shown that for any computation of a Real RAM M1 (BSS model) on a given real input data there exists a numerical RAM M2, with specific values of its precision and tolerance parameters such that M2 can in linear time approximate the computation of M1. The accuracy of the approximation depends on the relative precision with which the numerical RAM computes.

Why Does IBC Use the Real Number Model?

Henryk Woźniakowski
Columbia University, New York, USA

We explain why IBC (information–based complexity) uses the real number model. Results in the real number model are essentially the same as in floating point arithmetic with fixed precision modulo the two very important assumptions. These assumptions are: (1) numerical stability, (2) approximation errors are not to small relative to conditioning and the relative precision of floating point arithmetic. We illustrate this by an example of solving nonlinear equations by bisection. We also indicate the possibe tradeoffs between complexity and stability, and the need of using multiple or varying precision for ill–conditioned problems

Programs to Compute Moduli of Continuity on Neighbourhood Spaces

PETER ZAHN
Technische Hochschule Darmstadt, Germany

A system $(X, T, \sigma, \tau, U, \nu)$ is said to be a *computation space* iff X is a set; $\sigma, \tau : \mathbb{Z} \to \mathbb{Z}$ are computable; $T : X \times \mathbb{Z} \to \mathcal{P}(X)$ (the power set of X); for all $x, y, z \in X$ and all $k \in \mathbb{Z}$, we have

$$x \in T(x,k),$$
$$x \in T(y,\sigma_k) \;\wedge\; y \in T(z,\tau_k) \;\Rightarrow\; x \in T(z,k),$$
$$x \neq y \;\Rightarrow\; \exists k \in \mathbb{Z}.\ T(x,k) \cap T(y,k) = \emptyset \text{ (Hausdorff axiom)},$$
$$T(x,k+1) \subseteq T(x,k);$$

moreover, there exists a finite alphabet Σ such that $U \subseteq \Sigma^*$; and $\nu : U \to X$ (called a notation of $\nu(U)$).

Given two computation spaces $(X, T, \sigma, \tau, U, \nu)$ and $(X', T', \sigma', \tau', U', \nu')$. Then the computability of elements of X, X', and of functions of type $\subseteq X \to X'$ can be defined similarly as in the 'Russian approach'. Assume that $A \subseteq X$, and let $P(A)$ be the set of all programs p of type $\mathbb{Z} \to U$ to compute elements $x \in A$, i.e. such that $\forall k \in \mathbb{Z}.\ \nu(p_k) \in T(x,k)$. For $p \in P(A)$ let $\sharp p$ be the element of A that is computed by p. Now we can define: Given $f : A \to X'$, a program Δ of type $\mathbb{Z} \times P(A) \to \mathbb{Z}$ is said to be a *P-modulus of continuity* for f iff for all $(k, p, y) \in \mathbb{Z} \times P(A) \times A$,

$$\sharp p \in T(y, \Delta(k,p)) \;\Rightarrow\; f(\sharp p) \in T'(f(y), k).$$

In some cases in which a computable function $f : A \to X'$, a P-modulus Δ for f, and a not necessarily computable argument $x \in A$ are given, we nevertheless can find an 'approximation of $f(x)$ to a given degree $k \in \mathbb{Z}$ of precision', i.e. an element of $\nu'(U') \cap T'(f(x), k)$. - Due to [1], each effective operator of type $M \to M'$, where M and M' are appropriate metric spaces, is effectively continuous. In [2] there are generalized that and similar results uniformly to proper *topological* spaces. Those results and their proofs, however, do not yet yield a *convenient* method to find moduli of continuity that are useful for concrete numerical applications. However, we can specify an (incomplete) system of rules to produce useful P-moduli for functions between computation spaces. - Computation spaces are just those spaces that can be 'induced' by properly generalized metric spaces.

[1] Ceitin, G.S.: Algorithmic operators in constructive complete separable metric spaces (in Russian), Doklady Akad, Nauk 128, 49-52 (1959).

[2] Spreen, D., and Young, P.: Effective Operators in a Topological Setting. In: Computation and proof theory (M.M. Richter et al. eds.), Springer-Verlag, Berlin, Heidelberg, 1984.

Computability on Continuous, Lower Semi-Continuous and Upper Semi-Continuous Real Functions

Xizhong Zheng
FernUniverstität Hagen, Germany
(joint work with Klaus Weihrauch)

In this paper we extend computability theory to the spaces of continuous, upper semi-continuous and lower semi-continuous real functions. We apply the framework of TTE, Type-2 Theory of Effectivity, where not only computable elements but also computable functions on the spaces can be considered. First some basic facts about TTE are summarized. For each of the function s paces, we introduce several natural representations based on different intiuitive concepts of "effectivity" and prove their equivalence. Computability of several operations on the function spaces is investigated, among others limits, mappings to open sets, images of compact sets and preimages of open sets, maximum and minimum values. The positive results usually show computability in all arguments, negative results usually express non-continuity. Several of the problems have computable but not extensional solutions. Since computable functions map computable elements to computable elements, many previously known results on computability are obtained as simple corollaries.

L^p-Computability

NING ZHONG
University of Connecticut-Hartford, USA

In this paper we investigate conditions for L^p- computability which are in accordance with the classical Grzegorczyk notion of computability for a continuous function. We show that a function $f \in L^P(I)$, where $I \subset R^q$ is a computable rectangle, is L^p-*computable* if and only if (i) it is *sequentially computable* as a linear functional on the continuous function space $C(I)$ and (ii) its associated translation operator $T_f : R^q \to L^p(R^q)$ is *effectively uniformly continuous*, where $T_f(h)(x) = \tilde{f}(x+h)$, for any $h \in R^q$. $\tilde{f} \in L^p(R^q)$ is the zero extension of f, that is,

$$\tilde{f}(x) = \begin{cases} f(x) & x \in I \\ \\ 0 & x \in R^q - I \end{cases}$$

(Reported by VASCO BRATTKA)