LNCS 5635

Klaus Ambos-Spies
Benedikt Löwe
Wolfgang Merkle (Eds.)

Mathematical Theory and Computational Practice

5th Conference on Computability in Europe, CiE 2009
Heidelberg, Germany, July 2009
Proceedings

Springer

Lecture Notes in Computer Science 5635

Klaus Ambos-Spies Benedikt Löwe
Wolfgang Merkle (Eds.)

Mathematical Theory and Computational Practice

5th Conference on Computability in Europe, CiE 2009
Heidelberg, Germany, July 19-24, 2009
Proceedings

Springer

Volume Editors

Klaus Ambos-Spies
Wolfgang Merkle
Ruprecht-Karls-Universität Heidelberg
Institut für Informatik
Im Neuenheimer Feld 294, 69120, Heidelberg, Germany
E-mail: {ambos,merkle}@math.uni-heidelberg.de

Benedikt Löwe
Universiteit van Amsterdam
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
E-mail: bloewe@science.uva.nl

Library of Congress Control Number: 2009930216

CR Subject Classification (1998): F.1, F.2.1-2, F.4.1, G.1.0, I.2.6, J.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-03072-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-03072-7 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12718504 06/3180 5 4 3 2 1 0

Preface

CiE 2009: Mathematical Theory and Computational Practice
Heidelberg, Germany, July 19–24, 2009

After several years of research activity, the informal cooperation "Computability in Europe" decided to take a more formal status at their meeting in Athens in June 2008: the *Association for Computability in Europe* was founded to promote the development, particularly in Europe, of computability-related science, ranging over mathematics, computer science, and applications in various natural and engineering sciences such as physics and biology, including the promotion of the study of philosophy and history of computing as it relates to questions of computability. As mentioned, this association builds on the informal network of European scientists working on computability theory that had been supporting the conference series CiE-CS over the years, and now became its new home.

The aims of the conference series remain unchanged: to advance our theoretical understanding of what can and cannot be computed, by *any* means of computation. Its scientific vision is broad: computations may be performed with discrete or continuous data by all kinds of algorithms, programs, and machines. Computations may be made by experimenting with any sort of physical system obeying the laws of a physical theory such as Newtonian mechanics, quantum theory or relativity. Computations may be very general, depending on the foundations of set theory; or very specific, using the combinatorics of finite structures. CiE also works on subjects intimately related to computation, especially theories of data and information, and methods for formal reasoning about computations. The sources of new ideas and methods include practical developments in areas such as neural networks, quantum computation, natural computation, molecular computation, and computational learning. Applications are everywhere, especially, in algebra, analysis and geometry, or data types and programming. Within CiE there is general recognition of the underlying relevance of computability to physics and a broad range of other sciences, providing as it does a basic analysis of the causal structure of dynamical systems.

This volume, *Mathematical Theory and Computational Practice*, comprises the proceedings of the fifth in a series of conferences of CiE, that was held at the Ruprecht-Karls-Universität Heidelberg, Germany.

The first four meetings of CiE were at the University of Amsterdam in 2005, at the University of Wales Swansea in 2006, at the University of Siena in 2007, and at the University of Athens in 2008. Their proceedings, edited in 2005 by S. Barry Cooper, Benedikt Löwe and Leen Torenvliet, in 2006 by Arnold Beckmann, Ulrich Berger, Benedikt Löwe and John V. Tucker, in 2007 by S. Barry Cooper, Benedikt Löwe and Andrea Sorbi, and in 2008 by Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe were published as *Springer Lecture Notes in Computer Science*, volumes 3526, 3988, 4497 and 5028, respectively.

CiE and its conferences have changed our perceptions of computability and its interface with other areas of knowledge. The large number of mathematicians and computer scientists attending those conferences had their view of computability theory enlarged and transformed: they discovered that its foundations were deeper and more mysterious, its technical development more vigorous, its applications wider and more challenging than they had known. The annual CiE conference has become a major event, and is the largest international meeting focused on computability theoretic issues. Future meetings in Ponta Delgada, Açores (2010, Portugal), Sofia (2011, Bulgaria), and Cambridge (2012, UK) are in planning. The series is coordinated by the CiE Conference Series Steering Committee consisting of Arnold Beckmann (Swansea), Paola Bonizzoni (Milan), S. Barry Cooper (Leeds), Benedikt Löwe (Amsterdam, Chair), Elvira Mayordomo (Zaragoza), Dag Normann (Oslo), and Peter van Emde Boas (Amsterdam).

The conference was based on invited tutorials and lectures, and a set of special sessions on a range of subjects; there were also many contributed papers and informal presentations. This volume contains 17 of the invited lectures and 34% of the submitted contributed papers, all of which have been refereed. There will be a number of post-conference publications, including special issues of *Annals of Pure and Applied Logic*, *Journal of Logic and Computation*, and *Theory of Computing Systems*.

The tutorial speakers were Pavel Pudlák (Prague) and Luca Trevisan (Berkeley).

The following invited speakers gave talks: Manindra Agrawal (Kanpur), Jeremy Avigad (Pittsburgh), Mike Edmunds (Cardiff, Opening Lecture), Peter Koepke (Bonn), Phokion Kolaitis (San Jose), Andrea Sorbi (Siena), Rafael D. Sorkin (Syracuse), Vijay Vazirani (Atlanta).

Six special Sessions were held:

Algorithmic Randomness. *Organizers:* Elvira Mayordomo (Zaragoza) and Wolfgang Merkle (Heidelberg).
Speakers: Laurent Bienvenu, Bjørn Kjos-Hanssen, Jack Lutz, Nikolay Vereshchagin.

Computational Model Theory. *Organizers:* Julia F. Knight (Notre Dame) and Andrei Morozov (Novosibirsk).

Speakers: Ekaterina Fokina, Sergey Goncharov, Russell Miller, Antonio Montalbán.

Computation in Biological Systems — Theory and Practice.
Organizers: Alessandra Carbone (Paris) and Erzsébet Csuhaj-Varjú (Budapest).
Speakers: Ion Petre, Alberto Policriti, Francisco J. Romero-Campero, David Westhead.

Optimization and Approximation. *Organizers:* Magnús M. Halldórsson (Reykjavik) and Gerhard Reinelt (Heidelberg).
Speakers: Jean Cardinal, Friedrich Eisenbrand, Harald Räcke, Marc Uetz.

Philosophical and Mathematical Aspects of Hypercomputation.
Organizers: James Ladyman (Bristol) and Philip Welch (Bristol).
Speakers: Tim Button, Samuel Coskey, Mark Hogarth, Oron Shagrir.

Relative Computability. *Organizers:* Rod Downey (Wellington) and Alexandra A. Soskova (Sofia)
Speakers: George Barmpalias, Hristo Ganchev, Keng Meng Ng, Richard Shore.

The conference CiE 2009 was organized by Klaus Ambos-Spies (Heidelberg), Timur Bakibayev (Heidelberg), Arnold Beckmann (Swansea), Laurent Bienvenu (Heidelberg), Barry Cooper (Leeds), Felicitas Hirsch (Heidelberg), Rupert Hölzl (Heidelberg), Thorsten Kräling (Heidelberg), Benedikt Löwe (Amsterdam), Gunther Mainhardt (Heidelberg), and Wolfgang Merkle (Heidelberg).

The Program Committee was chaired by Klaus Ambos-Spies and Wolfgang Merkle:

Klaus Ambos-Spies (Heidelberg, Chair)
Giorgio Ausiello (Rome)
Andrej Bauer (Ljubljana)
Arnold Beckmann (Swansea)
Olivier Bournez (Palaiseau)
Vasco Brattka (Cape Town)
Barry Cooper (Leeds)
Anuj Dawar (Cambridge)
Jacques Duparc (Lausanne)
Pascal Hitzler (Karlsruhe)
Rosalie Iemhoff (Utrecht)
Margarita Korovina (Novosibirsk)
Hannes Leitgeb (Bristol)
Daniel Leivant (Bloomington)
Benedikt Löwe (Amsterdam)
Giancarlo Mauri (Milan)
Elvira Mayordomo (Zaragoza)
Wolfgang Merkle (Heidelberg, Chair)
Andrei Morozov (Novosibirsk)
Dag Normann (Oslo)
Isabel Oitavem (Lisbon)
Luke Ong (Oxford)
Martin Otto (Darmstadt)
Prakash Panangaden (Montréal)
Ivan Soskov (Sofia)
Viggo Stoltenberg-Hansen (Uppsala)
Peter van Emde Boas (Amsterdam)
Jan van Leeuwen (Utrecht)
Philip Welch (Bristol)
Richard Zach (Calgary)

We are delighted to acknowledge and thank the following for their essential financial support: Deutsche Forschungsgemeinschaft (German Research Foundation), Ruprecht-Karls-Universität Heidelberg (Bioquant and Department of Mathematics and Computer Science), The Elsevier Foundation.

We were proud to offer the program "Women in Computability" funded by the Elsevier Foundation as part of CiE 2009. The Steering Committee of the conference series CiE-CS is concerned with the representation of female researchers in the field of computability. The series CiE-CS has actively tried to increase female participation at all levels in the past years. Starting in 2008, our efforts are being funded by a grant of the Elsevier Foundation under the title "*Increasing representation of female researchers in the computability community.*" As part of this program, we had another workshop, a grant scheme for female researchers, a mentorship program, and free childcare.

The high scientific quality of the conference was possible through the conscientious work of the Program Committee, the special session organizers, and the referees. We are grateful to all members of the Program Committee for their efficient evaluations and extensive debates, which established the final program. We also thank the following referees:

Pavel Alaev
Eric Allender
Luis Antunes
Argimiro Arratia
Matthias Aschenbrenner
George Barmpalias
Verónica Becher
Ulrich Berger
Daniela Besozzi
Laurent Bienvenu
Stephen Binns
Achim Blumensath
Volker Bosserhoff
Patricia Bouyer
Thomas Brihaye
Dan Browne
Marius Bujorianu
Andrés Caicedo
Wesley Calvert
Riccardo Camerlo
Alessandra Carbone
Lorenzo Carlucci
Douglas Cenzer
Peter Cholak
Robin Cockett
Alessandro Colombo
Erzsébet Csuhaj-Varjú
Ali Dashti
Jan de Gier
Gianluca Della Vedova
Marina De Vos
Pietro Di Gianantonio
Kenny Easwaran
Leah Epstein
Javier Esparza
Peter Fejer
Fernando Ferreira
Ekaterina B. Fokina
Lance Fortnow
Willem L. Fouché
Pierre Fraigniaud
Peter Gács
Nicola Galesi
Hristo Ganchev
Christine Gaßner
Lucas Gerin
Guido Gherardi
Ilaria Giordani
Rob Goldstone
Erich Grädel
Daniel Graça
Noam Greenberg
Marek Gutowski
Amit Hagar
Magnús Halldórsson
Valentina Harizanov
Monika Heiner
Peter Hertling
Stefan Hetzl
Denis Hirschfeldt

John Hitchcock
Mark Hogarth
Mathieu Hoyrup
Rupert Hölzl
Tseren-Onolt Ishdorj
Emil Jeřábek
Herman Jervell
Jan Johannsen
Iskander Kalimullin
Anna Kasprzik
Klaus Keimel
Viv Kendon
Gabriele Kern-Isberner
Bjørn Kjos-Hanssen
Leszek Kołodziejczyk
Thorsten Kräling
Oleg V. Kudinov
Oliver Kullmann
Geoffrey LaForte
Branimir Lambov
Dominique Larchey-Wendling
Dominique Lecomte
Gyesik Lee
Alexander Leitsch
Steffen Lempp
Alberto Leporati
Davorin Lesnik
Andrew Lewis
Angsheng Li
María López
Gunther Mainhardt
Pierre McKenzie
Carlo Mereghetti
Joseph S. Miller
Russell Miller
Malika More
Markus Müller
Norbert Müller
Thomas Müller
André Nies
Stela Nikolova
István Németi
Paulo Oliva
Antonio Porreca
Valeria de Paiva
Jun Pang
Arno Pauly
Gheorghe Paun
Thanases Pheidas
Thomas Piecha
Sergei Podzorov
Lucia Pomello
Petrus Potgieter
R. Ramanujam
Ramyaa Ramyaa
Rolf Rannacher
Sidney Redner
Jan Reimann
Gerhard Reinelt
Klaus Reinhardt
Fred Richman
Beatrice Riviere
Vladimir Rybakov
Dov Samet
Matthias Schröder
Peter Schuster
Monika Seisenberger
Pavel Semukhin
Stephen G. Simpson
Alla Sirokofskich
Andrea Sorbi
Alexandra Soskova
Dieter Spreen
Frank Stephan
Thomas Strahm
Alexey Stukachev
Zhuldyz Talasbaeva
Paul Taylor
Sebastiaan Terwijn
Tristan Tomala
Hideki Tsuiki
John Tucker
Nikolas Vaporis
Panayot S. Vassilevski
Giuseppe Vizzari
Paul Voda
Heribert Vollmer
Nicolai Vorobjov
Peter B. M. Vranas
Andreas Weiermann

Thomas Wilke
Jon Williamson
Liang Yu
Karim Zahidi
Claudio Zandron
Martin Ziegler

We thank Andrej Voronkov for his EasyChair system, which facilitated the work of the Program Committee and the editors considerably.

May 2009

Klaus Ambos-Spies
Benedikt Löwe
Wolfgang Merkle

Table of Contents

Computability of Analytic Functions with Analytic Machines

Tobias Gärtner[1] and Günter Hotz[2]

[1] Max Planck Institut für Informatik, Saarbrücken
[2] Fachbereich für Informatik, Universität des Saarlandes, Saarbrücken

Abstract. We present results concerning analytic machines, a model of real computation introduced by Hotz which extends the well known Blum, Shub and Smale machines by infinite converging computations. We use the machine model to define computability of complex analytic (i.e. holomorphic) functions and examine in particular the class of analytic functions which have analytically computable power series expansions. We show that this class is closed under the basic analytic operations composition, local inversion and analytic continuation.

1 Introduction

In order to define computable functions on the real and complex numbers, several different proposals have been made. One prominent approach, introduced by Blum, Shub and Smale [1] defines a machine model that deals with real numbers as atoms. A computation on such a *BSS machine* is a finite sequence of arithmetic operations and conditional branches. Blum, Shub and Smale and others have developed an elaborate complexity theory for these machines, and e.g. posed the $P \neq NP$ problem over the real numbers. In [2], a comprehensive description of the theory is presented. A main characteristic of the functions computable within this model is that those function are, in a sense, piecewise defined rational functions. Elementary transcendental functions like the exponential function or trigonometric functions are not computable in this model.

Another successful approach, *recursive analysis*, defines computability over the real and complex numbers by allowing infinite computations on Turing machines, for instance by Grzegorczyk [3]. In the *type two theory of effectivity (TTE)*, continuous structures like the real and complex numbers are mapped to the tape of the machine by means of representations or naming systems, and the Turing machines perform infinite converging computations. An overview over the theory can be found in Weihrauch's book [4]. A main characteristic of the functions defined in this framework is that they are automatically continuous, and therefore discontinuous functions like the Heaviside function are not computable. A comprehensive overview over many different approaches to computability over the real and complex numbers can be found in [5].

There are also approaches which try to connect the different approaches, like Brattka and Hertling's feasible real RAM. In this paper, we continue the work

K. Ambos-Spies, B. Löwe, and W. Merkle (Eds.): CiE 2009, LNCS 5635, pp. 250–259, 2009.

of Hotz, Chadzelek and Vierke [6,7,8] and use the approach of the *analytic machines*. These machines are a kind of synthesis of BSS machines and recursive analysis, since they allow exact arithmetic operations with real numbers as atoms on the one hand and infinite converging computations on the other. The model can be restricted to perform rational arithmetic on rounded real inputs. A certain class, the 'quasi-strongly δ-$\mathbb{Q}$ analytic functions' subsumes both BSS computable and type 2 computable functions. One of the main results of the theory is Chadzelek's hierarchy theorem [7], which in particular states that analytic machines are not closed under composition.

The focus of this paper is the well known class of holomorphic or complex analytic functions. Complex function theory has been one of the most successful parts of mathematics and is, by many, also perceived as one of the most beautiful. The fact that each analytic function can be expressed by a power series also suggests that these functions have interesting computability properties, as a power series actually can be regarded as a specification of computation. We use the model of analytic machines in order to define computable analytic functions. We define the class of (analytically) computable analytic function and the class of coefficient computable functions, which are those functions whose power series expansion is computable by an analytic machine. We show that coefficient computable analytic functions are also computable. Furthermore, we show that the class of coefficient computable analytic functions is closed under composition, local inversion and analytic continuation. In view of the fact that analytic machines are not closed under composition, the closure properties of coefficient computable functions were not entirely expected.

The computability of analytic functions has already been studied in the framework of TTE, for example Ker-I Ko [9], Müller [10] have studied polynomial time computable analytic functions, and Müller [11] has studied constructive aspects of analytic functions. In the last work, Müller shows that in the framework of TTE, that the power series expansion of type 2 computable analytic functions is computable, and also the converse, that a function is computable if its power series is computable. He goes further and examines which information is necessary for these results to be constructive. Since in contrast to type 2 machines, the convergence speed of analytic machines is not known, his methods are not applicable to our machine model.

2 Definition and Properties of Analytic Machines

The machine model underlying this work is a register machine model, which is similar to the model of Blum, Shub and Smale (BSS machines). For finite computations, the models have the same computational power. In a sense, the analytic machines extend the BSS machines by infinite converging computations. In the following section, we will briefly present the model of analytic machines. The definition has been adapted and shortened from [7]. For a more detailed description we refer the reader to this work.

Machine Model. First, we give the definition of our mathematical machine that allows us to reason formally about computations and the functions computed by a machine. A mathematical machine over an alphabet A (not necessarily finite) is a tuple $\mathcal{M} = (K, K_{\mathrm{i}}, K_{\mathrm{f}}, K_{\mathrm{t}}, \Delta, A, \mathrm{in}, \mathrm{out})$. The set K is the *set of configurations* of $\mathcal{M}$, $K_{\mathrm{i}}, K_{\mathrm{f}}, K_{\mathrm{t}} \subset K$ are *initial*, *final* and *target configurations*. The function $\Delta : K \to K$ with $\Delta|_{K_{\mathrm{f}}} = id_{K_{\mathrm{f}}}$ is the *transition function*, and the functions $\mathrm{in} : A^* \to K_{\mathrm{i}}$ and $\mathrm{out} : K \to A^*$ are *input* and *output functions* over A.

A sequence $b = (k_i)_{i\in\mathbb{N}} \in K^{\mathbb{N}}$ with $k_0 = \mathrm{in}(x)$ and $k_{i+1} = \Delta(k_i)$ is called *computation of* $\mathcal{M}$ applied to x. The computation is called *finite* or *halting* if there is an $n \in \mathbb{N}$ such that $k_n \in K_{\mathrm{f}}$, and the minimal n with this property is called the *length* of the computation and $\mathrm{out}(k_n)$ its *result*.

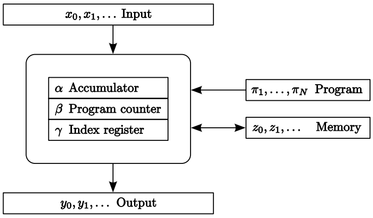
Assuming a metric on A^*, we can also consider infinite converging computations. If for a computation $b = (k_i)_{i\in\mathbb{N}}$ a target configuration is assumed infinitely often, and the limit $\lim_{n\to\infty} \mathrm{out}(k_{i_n})$ for the subsequence $(k_{i_n})_{n\in\mathbb{N}} \subset K_{\mathrm{t}}$ of target configurations exists, we call the computation *analytic*. In this case, this limit is the *result of the analytic computation* b, and $\mathrm{out}(k_{i_n})$ is called the *n-th approximation*. If for an input $x \in A$ the resulting computation is neither finite nor analytic, then we say that $\mathcal{M}$ *does not converge* on this input, otherwise we say that the *machine converges* on this input.

The *domain* $\mathcal{D}_M$ of a machine $\mathcal{M}$ is the set of all $x \in A^*$ such that the computation of $\mathcal{M}$ with input x is analytic or finite, and its *halting set* $\mathcal{H}_M$ is the set of all $x \in A^*$ such that its computation with input x is finite.

On its domain, the machine $\mathcal{M}$ now gives rise to the definition of a function $\Phi_M : \mathcal{D}_M \to A^*$, the *function defined by* $\mathcal{M}$. For inputs x such that the computation of $\mathcal{M}$ with input x is finite or analytic with result $y \in A^*$, we define $\Phi_M(x) = y$.

The mathematical machine will now be specialized to register machines over the ring $\mathcal{R}$. These machines are random access machines that operate with elements of $\mathcal{R}$ as atoms. For generality, the definition will be made for any ring $\mathcal{R}$ containing the integers (endowed with a metric in the case of analytic machines). The machines will, however, only be used for the fields of rationa, real an complex numbers $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$. The basic notion of the machines is depicted in Fig. 1. A machine consists of a control unit with an accumulator α, a program counter β and an index register γ. There is an infinite input tape $x = (x_0, x_1, \ldots)$ that can be read, an infinite output tape $y = (y_0, y_1, \ldots)$ that can be written on, and an infinite memory $z = (z_0, z_1, \ldots)$. Furthermore, each machine has a finite program $\pi = (\pi_1, \ldots, \pi_N)$ with instructions from the instruction set $\Omega = \Omega_{\mathcal{R}}$ shown in Fig. 1. The set of configurations of a machine with program π is given by the contents of its registers and memory cells:
$K = \left\{k = (\alpha, \beta, \gamma, \pi, x, y, z) \,\middle|\, \alpha \in \mathcal{R},\ \beta, \gamma \in \mathbb{N}, x, y, z \in \mathcal{R}^{\mathbb{N}}\right\}$. Initial, final and target configurations are given by $K_{\mathrm{i}} = \{k \in K \,|\, \alpha{=}\gamma{=}0, \beta{=}1, \forall j : y_j, z_j{=}0\}$, $K_{\mathrm{f}} = \{k \in K \,|\, \pi_\beta = \texttt{end}\}$ and $K_{\mathrm{t}} = \{k \in K \,|\, \pi_\beta = \texttt{print}\}$.

The instruction `print` is used to identify target configurations, and `exception` stops the machine in a non-halting configuration.



1. assignments
   - $\alpha := x_i$, $\alpha := z_i$, $y_i := \alpha$, $z_i := \alpha$, $\quad i \in \mathbb{N} \cup \{\gamma\}$
   - $\alpha := c$, $c \in \mathcal{R}$
   - $\gamma := 0$
2. arithmetics
   - $\alpha := \alpha + z_i$, $\alpha := \alpha \cdot z_i$
   - $\alpha := -\alpha$, $\alpha := \frac{1}{\alpha}$
   - $\gamma := \gamma + 1$, $\gamma := \gamma \dot{-} 1$
3. branches
   `goto` m
   `if` $\alpha > 0$ `then goto` m `else goto` n ($\mathcal{R}$ ordered)
   `if` $\alpha \neq 0$ `then goto` m `else goto` n ($\mathcal{R}$ not ordered)
   `if` $|\alpha| > |z_i|$ `then goto` m `else goto` n ($\mathcal{R} = \mathbb{C}$)
4. special instructions
   `end`, `print`, `exception`

Fig. 1. Register machine and set of instructions

With the definition of our machine model, we now define computable functions.

Definition 2.1. *Let $D \subset \mathcal{R}^*$. We call a function $f : D \to \mathcal{R}^*$* analytically $\mathcal{R}$-computable *if there exists an $\mathcal{R}$-machine $\mathcal{M}$ such that $D \subset \mathcal{D}_{\mathcal{M}}$ and $f = \Phi_{\mathcal{M}}|_D$. We call f* $\mathcal{R}$-computable *if there exists an $\mathcal{R}$-machine $\mathcal{M}$ such that $D \subset \mathcal{H}_{\mathcal{M}}$ and $f = \Phi_{\mathcal{M}}|_D$.*

A set $A \subset \mathcal{R}^*$ is called (analytically) decidable *if its* characteristic function *χ_A is (analytically) computable.*

Properties of Computable Functions. For the finitely computable functions, there is Blum, Shub and Smale's well-known representation theorem for $\mathbb{R}$-computable functions (see, e.g. [2]).

Theorem 2.2. *Suppose $D \subset \mathbb{R}^n$ (or $\mathbb{C}^n$), and let f be an $\mathbb{R}$-computable (or $\mathbb{C}$-computable) function on D. Then D is the union of countably many semi-algebraic sets, and on each of those sets f is a rational function.*

Undecidable Problems. For Turing machines, the well known halting problem is undecidable: There is no Turing machine, that accepts as inputs the description of a Turing machines and decides if that given machine halts on a given input or not. This problem is, however, easily seen to be decidable by an analytic $\mathbb{R}$-machine.

The analogon of the halting problem for analytic machines is the *convergence problem*. That is, decide (analytically) whether a given analytic machine converges on its input or not. By diagonalization, it is easy to see that this function is not analytically computable [7,6]. If we compose several analytic machines, however, the convergence problem becomes decidable. Intuitively, the term *composition* in the context of analytic machines means that an analytic machine uses the result of another analytic machine as input. Formally it is based on the computed functions, we say that a function f is *computable by i composed analytic machines*, if there are i $\mathbb{R}$-analytically computable functions $f_1, \ldots, f_i$ such that $f = f_i \circ \cdots \circ f_1$.

Chadzelek and Hotz [7] have shown that the convergence problem for analytic machines is computable by the composition of *three* analytic machines. Chadzelek has pointed out that it is possible to decide the problem with two analytic machines, but only if infinite intermediate results are used. The following theorem improves this result and shows that *two* machines suffice to decide the convergence problem, thus answering the open question of Chadzelek [8].

Theorem 2.3. *The convergence problem for analytic machines over the real numbers is decidable by two composed analytic machines.*

Proof. We give the description of two machines $\mathcal{M}_1$ and $\mathcal{M}_2$ such that the composition of their computed functions decides the convergence problem.
Let $(a_n)_{n\in\mathbb{N}}$ be the sequence of outputs of target configurations of the input machine $\mathcal{M}$. The goal is to check whether (a_n) is a Cauchy sequence. Let $b_k := \sup_{n>m\geq k}|a_n - a_m|$. Then, the sequence (a_n) is convergent iff $\lim_{k\to\infty} b_k = 0$. In order to decide convergence of a sequence to zero, it is sufficient to consider the sequence formed of the numbers rounded to the next higher power of two. Therefore we define $r(x,k) := \max\{j \leq k : 2^{-j} \geq x\}$ for $x < 1$ and $r(x,k) := 0$ for $x > 1$, $k \in \mathbb{N}$ and set $\tilde{b}_k := r(b_k, k)$.

Then we have $\lim_{k\to\infty} b_k = 0$ iff $\lim_{k\to\infty} \tilde{b}_k = \infty$ (note that this also covers sequences that are ultimately constantly zero). The machine $\mathcal{M}_1$ now computes approximations $\tilde{b}_{k,j}$ of $\tilde{b}_k$. Let $b_{k,j} := \max\{|a_n - a_m| \,|\, j > n > m \geq k\}$ and $\tilde{b}_{k,j} = r(b_{k,j}, k)$. Then $\tilde{b}_{k,j}$ is non-increasing with j and therefore for each k there is a j such that $\tilde{b}_{k,j} = \tilde{b}_k$, i.e. the sequence $(\tilde{b}_{k,j})_j)$ becomes stationary.
By parallel computation in rounds of k, $\mathcal{M}_1$ computes the approximation $\tilde{b}_{k,j}$ of $\tilde{b}_k$ for larger and larger j, and for each k, this approximation becomes stationary for sufficiently large j. The machine $\mathcal{M}_1$ stores the information about the $\tilde{b}_{k,j}$ in a single real number by a unary encoding. For $n \in \mathbb{N}$ let $\mathrm{u}(n)$ be the unary encoding of n. Then, in each round $\mathcal{M}_1$ outputs the real number $\mathcal{M}_1^{(j)} = 0.0\mathrm{u}(\tilde{b}_{1,j})0\mathrm{u}(\tilde{b}_{2,j})0\mathrm{u}(\tilde{b}_{3,j})0\ldots0\mathrm{u}(\tilde{b}_{j,j})$. Since for each k the computation of $\tilde{b}_{1,j}$ will become stationary, this sequence of outputs converges to a real number $\mathcal{M}_1(\mathcal{M}) = 0.0\mathrm{u}(\tilde{b}_1)0\mathrm{u}(\tilde{b}_2)0\mathrm{u}(\tilde{b}_3)0\ldots$

Now, it follows that the original sequence (a_n) converges iff $\mathcal{M}_1(\mathcal{M})$ is an irrational number. Because if the original sequence converges, then $\tilde{b}_k$ is unbounded and the number computed by $\mathcal{M}_1$ has no periodic dual expansion. If, on the other hand, the original sequence is not convergent, then $\tilde{b}_k$ will become stationary (since it is nondecreasing with k and bounded) and the number computed by $\mathcal{M}_1$ has a periodic dual expansion.

The second machine $\mathcal{M}_2$ just has to decide whether its input is rational or not, which an analytic machine simply does by enumerating all rational numbers. □

The undecidability of the convergence problem for analytic machines and the decidability of this problem by two composed analytic machines imply that those functions cannot be closed under composition (this follows in fact already from Chadzelek's result that three composed analytic machines decide convergence):

Corollary 2.4. *The set of analytically computable functions is not closed under composition.*

3 Computability of Analytic Functions

Now we come to the main part of this paper, the application of the machine model of analytic machines to define the computability of analytic functions. By an *analytic function* on a region of the complex plane $\mathbb{C}$ we mean a function that has a power series expansion in each point of the region, or equivalently, which is complex-differentiable in every point of D. Those functions are also called holomorphic functions, but in our context we prefer the term analytic because it stresses the power series aspect of the function.

We will define two classes of computable analytic functions, first those functions which are analytic and also analytically computable (i.e. computable by an analytic machine) and then those functions which have a power series expansion that is computable by an analytic machine, the functions which we call coefficient computable. We will then show properties of the functions in the respective classes and examine the relationship between those two classes. In particular, we will show that every coefficient computable analytic function is also computable, and that the class of coefficient computable analytic functions is closed under composition, local inversion and analytic continuation.

Notation. In this section, if not mentioned otherwise, D denotes a *region* in $\mathbb{C}$, i.e. a nonempty open connected subset of $\mathbb{C}$. For a complex number z_0 and $r > 0$, $U_r(z_0)$ is the open disc with center at z_0 and radius r and $\overline{U}_r(z_0)$ its closure.

Computable Analytic Functions. First, we show that the case of finitely computable functions or, equivalently, BSS-computable functions is not very interesting in the context of analytic functions:

Lemma 3.1. *Let D be a region in $\mathbb{C}$ and $f : D \to \mathbb{C}$ be analytic and (finitely) $\mathbb{C}$-computable. Then f is a rational function on D.*

Proof. This follows by the representation theorem 2.2 and the identity theorem for analytic functions.

In light of Lemma 3.1, we will focus our attention on analytic functions which are not necessarily finitely computable. Therefore, in the following, we will freely omit the term 'analytic' when we talk about computability, i.e. instead of 'analytically computable analytic functions' we will just say 'computable analytic functions'.

Definition 3.2. *Let $D \subset \mathbb{C}$ be a region in the complex plane, and $f : D \to \mathbb{C}$ be a function. We call f a*

– computable analytic function *if f is holomorphic and analytically computable on D. We call f*

– coefficient computable in $z_0 \in D$ *if f is holomorphic on D and the power series expansion of f in z_0 is an analytically computable sequence, and we call f*
– coefficient computable on D *if f is holomorphic on D and coefficient computable in each point $z_0 \in D$ and*
– uniformly coefficient computable on D *if the mapping* $\mathbb{N} \times D \to \mathbb{C}, (n, z_0) \mapsto$ $a_k(z_0)$ *is analytically computable, where $a_k(z_0)$ is the k-th coefficient of the power series of f in z_0.*

Now, we show that coefficient computability is the stronger notion, i.e. that each coefficient computable function is also computable.

Theorem 3.3. *Suppose $f : D \to \mathbb{C}$ is an analytic function on D with power series expansion $f(z) = \sum_{k=0}^{\infty} a_k(z-z_0)^k$ at z_0, such that the sequence $(a_k)_{k\in\mathbb{N}}$ is analytically computable, i.e. f is coefficient computable in z_0. Let be $R > 0$ such that $\overline{U}_R(z_0) \subseteq D$. Then f is analytically computable in the open disc $U_R(z_0)$.*

Proof. Let $\mathcal{K}$ be the machine that (analytically) computes the coefficients of the power series of f in z_0. We give the description of a machine $\mathcal{M}$ which analytically computes f in $U_R(z_0)$. We denote the n-th approximation of $\mathcal{K}$'s computation of the coefficient a_k with $a_k^{(n)}$. Let $z \in U_R(z_0)$ be the input for $\mathcal{M}$. The basic idea would now be to approximate the value $f(z)$ by a diagonal approximation $\sum_{k=0}^{n} a_k^{(n)}(z - z_0)^k$. The problem is that there is no information about the speed of convergence of $a_k^{(n)}$ with n, which could also be different for each k. The trick is to use upper bounds on the coefficients given by basic theorems of function theory. We recall that by Cauchy's inequalities (cf. eg. [12]) that for f there is a constant $M > 0$ such that $|a_k| \leq \frac{M}{R^k}$ for all $k \in \mathbb{N}$. The machine $\mathcal{M}$ now has information about M and R stored as constants (w.l.o.g rational bounds can be chosen), and if a current approximation $a_k^{(n)}$ exceeds the bound $\frac{M}{R^k}$, it is replaced by this bound. Summarizing, $\mathcal{M}$ computes (as n-th approximation)

$$\mathcal{M}^{(n)}(z) = \sum_{k=0}^{n} b_k^{(n)}(z - z_0)^k \quad \text{with} \quad b_k^{(n)} = \begin{cases} a_k^{(n)} & : \ |a_k^{(n)}| \leq \frac{M}{R^k} \\ \frac{M}{R^k} & : \ \text{else} \end{cases}$$

We assert that the sequence of outputs $\mathcal{M}^{(n)}(z)$ converges to $f(z)$. In order to see this, the difference $\left|f(z) - \mathcal{M}^{(n)}(z)\right| = \left|f(z) - \sum_{k=0}^{n} b_k^{(n)}(z - z_0)^k\right|$ is split into three parts, each of which becomes arbitrarily small for large n. The first part is $\sum_{k=0}^{n_0} |(a_k - b_k^{(n)})(z - z_0)^k|$ for a fixed n_0, and the second and third parts are $\sum_{k=n_0}^{n} |(a_k - b_k^{(n)})(z - z_0)^k|$ and $\sum_{k=n}^{\infty} |a_k(z - z_0)^k|$, respectively. By choosing n_0 large enough, the second and third parts can be made arbitrarily small. For n_0 then fixed, the first part can also be made arbitrarily small, since for large enough n the fixed number n_0 of the analytic computations of $a_0^{(n)}, \ldots, a_{n_0}^{(n)}$ can be made arbitrarily precise, and hence their finite sum. □

A refinement of the proof of Thm. 2.2 will lead us now to an even stronger result, namely that not only a function is computable if it is coefficient computable, but also all its derivatives. For the following, recall the notation $[k]_d := \frac{k!}{(k-d)!} = k \cdot (k-1) \cdots (k-d+1)$. We observe that if the sequence (a_k) is analytically computable, then so are all sequences $(a_{(d,k)})_k = ([k+d]_d a_{k+d})_k$. Furthermore, they are *uniformly* computable, i.e. the mapping $(d,k) \mapsto a_{(d,k)}$ is analytically computable. Now, we can show that

Theorem 3.4. *Suppose $f : D \to \mathbb{C}$ is an analytic function on D which is coefficient computable in $z_0 \in D$. Let further be $R > 0$ such that $\overline{U}_R(z_0) \subseteq D$. Then each derivative $f^{(d)}$ is analytically computable in the open disc $U_R(z_0)$, and furthermore, all derivatives are* uniformly *computable, i.e. the mapping $(d, z) \mapsto f^{(d)}(z)$ is analytically computable on $\mathbb{N} \times U_R(z_0)$.*

Proof. The proof is similar to the proof of Thm. 2.2. Using Cauchy's inequalities for the derivatives we get the estimate $|a_{d,k}| = (k+d)_d a_{k+d} \leq \frac{M}{R^{k+d}}(k+d)_d$. The approximation of $f(z)$ given by the machine is again split into three parts, and with a little more technical effort, each part can be shown to become arbitrarily small. □

Corollary 3.5. *Suppose $f : D \to \mathbb{C}$ is an analytic function on D which is coefficient computable in $z_0 \in D$. Let further be $R > 0$ such that $\overline{U}_R(z_0) \subseteq D$. Then f is uniformly coefficient computable on $U_R(z_0)$.*

This shows that coefficient computability in a point is the basic property for the computability of analytic functions in our sense, since it implies the computability and even uniform coefficient computability in a disc around this point.

Closure Properties of Coefficient Computable Functions. We now turn our attention to closure properties of the class of coefficient computable functions. We show that this class is closed under the basic analytic operations composition, local inversion and analytic continuation.

Composition. Since the class of generally analytically computable functions is not closed under composition (cf. Cor. 2.4), one could expect that this is also the case for the class of coefficient computable functions, which are defined by means of analytic machines. But with the theorems of the last section we can show that this class actually is closed under composition. The standard approach here would be to use a recursive description of the coefficients of the composition to compute these; since the precision of the approximations are, however, unknown, this approach is not successful. Instead, we directly express the coefficients in terms of the derivatives and apply Thm. 3.4.

Theorem 3.6. *Suppose $g : D \to \mathbb{C}$ and $f : g(D) \to \mathbb{C}$ are coefficient computable functions. Then the composition $f \circ g : D \to \mathbb{C}$ is also coefficient computable.*

Proof. The general form of the n-th derivative of the composition $f \circ g$ at z_0 is given by *Faa di Bruno's Formula*:

$$(f \circ g)^{(n)}(z_0) = \sum_{(k_1,\ldots,k_n) \in P_n} \frac{n!}{k_1! \cdot \cdots \cdot k_n!} \left(f^{(k_1+\cdots+k_n)}\right)(z_1) \prod_{m=1}^{n} \left(\frac{g^{(m)}(z_0)}{m!}\right)^{k_m}$$

Here, the set $P_n = \{(k_1, \dots, k_n) \,|\, k_1 + 2k_2 + \cdots + nk_n = n\}$ denotes the set of partitions of n.
In order to compute the coefficients of $f \circ g$ at z_0 observe that the coefficients of each derivative of f at z_1 and g at z_0 are uniformly computable, and that the n-th derivative of $f \circ g$ is a finite polynomial in those. □

Local Inversion. Similarly to case of composition, we can apply Thm. 3.4 to show that the local inversion of a coefficient computable function is again coefficient computable:

Theorem 3.7. *Suppose $f : D \to \mathbb{C}$ is coefficient computable and $f'(z_0) \neq 0$. Then the local inverse f^{-1} of f is coefficient computable in a neighborhood of $f(z_0)$.*

Proof. Similarly to Thm. 3.6 one can show by induction, the coefficients of f^{-1} at $z_1 = f(z_0)$ can be expressed as rational functions in the derivatives of f at z_0 of order less or equal than n and therefore in the coefficients of the power series of f in z_0. □

Analytic Continuation. A very important notion in the realm of analytic functions is the notion of *analytic continuation*. We briefly recall the basic concept. Let $f : D \to \mathbb{C}$ be an analytic function, $z_0 \in D$ and $f(z) = \sum_{k=0}^{\infty} a_k (z - z_0)^k$ the power series expansion of f with convergence radius $r > 0$. Let $U_r(z_0)$ be the disk with center z_0 and radius r. Then it can happen that the range of convergence $U := U_r(z_0)$ of the power series exceeds the boundary of D. If we assume further that $D \cap U$ is connected, a well-defined analytic function g is defined on $D \cup U$ by $g(z) = f(z)$ if $z \in D$ and $\sum_{k=0}^{\infty} a_k (z - z_0)^k$ if $z \in U$. It is well-defined because $D \cap U$ is connected, since then f and the function defined by the power series have to be equal on all of $D \cap U$ because of the identity theorem (note that connectedness is sufficient for the application of this theorem; simple connectedness is not necessary). The function g is called *immediate analytic continuation* of f. This process can be iterated, and an extension of f on a much larger region G may be defined. If the requirement of the connectedness of $D \cap U$ is omitted, the resulting 'function' can have multiple values. Consequently developed, this leads to the theory of *Riemann surfaces*, but this is not the subject of this article. We confine ourselves to regions that are subsets of the complex plane.
A very interesting result about coefficient computable functions is that they are closed under analytic continuation:

Theorem 3.8. *Suppose that $f : D \to \mathbb{C}$ is coefficient computable and that f has an analytic continuation $g : G \to \mathbb{C}$ on a region $G \supset D$. Then g is coefficient computable on G.*

Proof. This is a consequence of Cor. 3.5. In order to show that f is coefficient computable in $w \in G$, fix $z_0 \in D$. Inductively, we obtain a sequence of points $z_1, z_2, \dots, z_n = w$ and functions $f = f_0, f_1, f_2, \dots, f_n = w$ such that f_{i+1} is an analytic continuation of f_i, z_{i+1} is in the disc of convergence of f_i at z_i and f_i is coefficient computable at z_i. □

This result is interesting because it relates an important notion of complex function theory to computability: In function theory, there is the notion of a *function germ*, which conveys that a holomorphic function is completely determined by the information of its power series expansion in a single point. Theorem 3.8 shows that a similar notion holds for coefficient computability: Coefficient computability of a function in a single point implies its computability in every point of its domain.

4 Conclusion and Outlook

In this paper, we have used the model of analytic machines to define two notions of computability for complex analytic functions. We have shown coefficient computability to be the stronger of the two notions. We have further shown that, in contrast to the general analytically computable functions, the class of coefficient computable functions is closed under composition, and also under local inversion and analytic continuation.

Those results have in a sense, however, not been constructive, since in order to show computability, constants containing information about the regarded functions have been assumed in the constructed machines. Open questions remain whether any of these results can, at least in part, be gained constructively. Another question that remains open is whether or to which extent analytically computable analytic functions are also coefficient computable, i.e. the converse of Thm. 3.3.

It could also be interesting to compare these results to the corresponding ones for type 2 Turing machines from the viewpoint of relativization [13].

References

1. Blum, L., Shub, M., Smale, S.: On a theory of computation and complexity over the real numbers. Bull. of the AMS 21, 1–46 (1989)
2. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer, New York (1998)
3. Grzegorczyk, A.: On the definition of computable real continuous functions. Fundamenta Mathematicae 44, 61–71 (1957)
4. Weihrauch, K.: Computable Analysis. Springer, Berlin (2000)
5. Ziegler, M.: Real computability and hypercomputation. TR C-07013, KIAS (2007)
6. Hotz, G., Vierke, G., Schieffer, B.: Analytic machines. ECCC 2, 25 (1995)
7. Chadzelek, T., Hotz, G.: Analytic machines. TCS 219, 151–167 (1999)
8. Chadzelek, T.: Analytische Maschinen. Dissertation, Saarland University (1998)
9. Ko, K.I.: Complexity Theory of Real Functions. Birkhäuser, Basel (1991)
10. Müller, N.T.: Uniform computational complexity of Taylor series. In: Ottmann, T. (ed.) ICALP 1987. LNCS, vol. 267, pp. 435–444. Springer, Heidelberg (1987)
11. Müller, N.T.: Constructive aspects of analytic functions. Informatik Berichte FernUniversität Hagen 190, 105–114 (1995)
12. Rudin, W.: Real and Complex Analysis. McGraw-Hill, New York (1987)
13. Ziegler, M.: Computability and continuity on the real arithmetic hierarchy and the power of type-2 nondeterminism. In: Cooper, S.B., Löwe, B., Torenvliet, L. (eds.) CiE 2005. LNCS, vol. 3526, pp. 562–571. Springer, Heidelberg (2005)

Author Index