

# UNIVERSITÄT TRIER

## Mathematik / Informatik

Forschungsbericht Nr. 01–05

*43. Workshop für  
Komplexitätstheorie, Datenstrukturen  
und  
effiziente Algorithmen  
(Theorietag)*

*Universität Trier*

Martin Mundhenk (Hrsg.)

### Electronic copies of technical reports are available:

Via FTP: URL <ftp://ftp.informatik.uni-trier.de/pub/Users-Root/reports>  
Via WWW: URL <http://www.informatik.uni-trier.de/Reports/Current.html>  
Via email: Send a mail to [ftpmail@ftp.informatik.uni-trier.de](mailto:ftpmail@ftp.informatik.uni-trier.de), subject  
'HELP', for detailed instructions

### Printed copies:

Trierer Forschungsberichte  
Fachbereich IV -  
Mathematik / Informatik  
Universität Trier  
D-54286 Trier

ISSN 0944-0488

# Programm

- 9:15 Begrüßung
- 9:30 – 10:00 Norbert Th. Müller (Trier):  
*Imperative exakte reelle Arithmetik*
- 10:00 – 10:30 Markus E. Nebel (Frankfurt):  
*Der Registerbedarf von arithmetischen Ausdrücken und die Ordnung einsträngiger RNA-Moleküle*
- 10:30 – 11:00 Rustam Mubarakzjanov (Trier):  
*Bounded-Width Probabilistic OBDDs and Read-Once Branching Programs are Incomparable*
- 11:30 – 12:00 Katarzyna Paluch (Wrocław):  
*New Approximation Algorithm for the RTILE Problem*
- 12:00 – 12:30 Andreas Jakoby (Lübeck):  
*The Complexity of some Basic Problems for Dynamic Process Graphs*
- 12:30 – 13:00 Andre Osterloh (Ilmenau):  
*Oblivious Permutation Routing on the Mesh*
- 14:00 – 14:30 Karl-Heinz Niggl (Ilmenau):  
*Loop Languages, Computational Complexity and the Low Grzegorzczuk Classes*
- 14:30 – 15:00 Holger Petersen (Stuttgart):  
*Backing up in Lists with Restricted Assignments*
- 15:00 – 15:30 Jan Reimann (Heidelberg):  
*Hausdorff-Dimension in Exponentialzeit*
- 16:00 – 16:30 Sandra Zilles (Kaiserslautern):  
*Vergleich von Inferenztypen beim Uniformen Lernen*
- 16:30 – 17:00 Oliver Zlotowski (Trier):  
*Software Visualisierung*

# The Complexity of some Basic Problems for Dynamic Process Graphs

Andreas Jakoby and Maciej Liškiewicz

Institut für Theoretische Informatik, Universität zu Lübeck

email: jakoby/liskiewi@informatik.mu-luebeck.de

A fundamental problem in programming multiprocessors is scheduling elementary tasks on the available hardware efficiently. Traditionally one represents tasks and precedence constraints by a *data-flow graph*. This representation requires that the set of tasks to be executed is known beforehand. Such an approach is not appropriate in situations where the set of tasks is not known exactly in advance, for example, when different options how to continue a program may be granted. In this paper *dynamic process graphs* will be used to represent all possible executions of a given program. An important feature of this model is that the encoded executions are directed acyclic graphs having a “regular” structure that is typical of parallel or distributed programs.

With respect to such a graph representation we investigate the computational complexity of some basic graph-theoretic problems like e.g. *What is the depth of a given graph?* or *What is the size of a subgraph induced by a given node  $v$ ?* In the paper the precise complexities of these problems are given. As a consequence we obtain a characterisation of the computation complexity of various variants of scheduling problems.

# Bounded-Width Probabilistic OBDDs and Read-Once Branching Programs are Incomparable

Rustam Mubarakzjanov

Universität Trier

Restricted branching programs are considered by the investigation of relationships between complexity classes of Boolean functions. Read-once ordered branching programs (or *OBDDs*) form the most restricted class of this computation model. Since the problem of proving exponential lower bounds on the complexity for general probabilistic *OBDDs* is open so far, it is interesting to study this problem in a restricted setting. For this reason we deal in this work with probabilistic *OBDDs* whose width is bounded.

We prove in this work that probabilistic *OBDDs* of width bounded by a constant can be more powerful than even non-deterministic read-once branching programs. To do it we present a probabilistic *OBDD* of constant width computing the known function *PERM*. We prove for several known functions that they cannot be computed by probabilistic *OBDDs* of constant width. To show it we present a new method allowing to obtain lower bound  $\Omega(n)$  on the width of corresponding *OBDDs* ( $n$  is the number of variables).

# Imperative exakte reelle Arithmetik

Norbert Th. Müller

Abteilung Informatik — Universität Trier

D-54286 Trier, Germany

mueller@uni-trier.de

<http://www.informatik.uni-trier.de/~mueller/>

Tel: ++49-651-201-2845 / Fax: ++49-651-201-3805

Reale Berechnungen auf den reellen Zahlen werden meist mit Fließkommazahlen durchgeführt, für die sogar IEEE-Standards existieren. Die Grenzen dieser üblicherweise 8-Byte umfassenden Darstellungen lassen sich jedoch schon mit einfachen Beispielen klar aufzeigen. Während die übliche 4-Byte-Darstellung ganzer Zahlen in naheliegender Weise für beliebig große Zahlen erweitert werden kann, ist dies im Bereich der reellen Zahlen nicht so einfach, was im Wesentlichen an deren Überabzählbarkeit liegt.

Im letzten Jahrzehnt sind etliche Ansätze zur Implementierung exakter reeller Arithmetik vorgestellt worden, etwa [BoCa90] (funktionale Sprache mit Dezimal-basierter Darstellung) oder [EdPo97] ('linear fractional transformations'). Allen Ansätzen fehlt jedoch die Möglichkeit, imperativ zu programmieren, was jedoch die Implementierung üblicher numerischer Algorithmen enorm erleichtern würde.

Eine Alternative findet sich in [BrHe95]: Dort haben Brattka und Hertling eine Turing-Maschinen-basierte Simulation einer RAM mit reellen Registern beschrieben. Die Grundidee ist dabei die Iteration einer approximierenden Berechnung mit wachsender Genauigkeit.

In [Mu97] stellte der Autor den Prototyp einer Implementierung vor, die auf der Struktur dieses Simulators aufbaut: die iterative **Real-RAM** (iRRAM). Seitdem wurde die Software stetig weiterentwickelt und war im Jahr 2000 bei einem Wettbewerb zwischen verschiedenen Systemen für exakte Arithmetik beim '4th Workshop on Computability and Complexity in Analysis' in Swansea der klare Gewinner.

Im Vortrag sollen zwei Kernaspekte der Software genauer betrachtet werden: Operatoren zur Berechnung von Limites und die Berechnung mehrwertiger Funktionen. Mit den Limites wird der Bereich algebraischer oder symbolischer Berechnungen verlassen, und Weihrauch's 'Type 2 Theory of Effectivity' [Ko91, We95, We97] ist dann der passende theoretische Unterbau.

Mehrwertige Funktionen erlauben eine Abschwächung der in TTE gültigen 'Regel', daß aus Berechenbarkeit auch Stetigkeit folgen muß: Bei der Berechnung einer solchen Funktion darf ein Wert frei aus einer Vielzahl möglicher Werte gewählt werden, ähnlich wie bei nichtdeterministischen Berechnungen.

# Der Registerbedarf von arithmetischen Ausdrücken und die Ordnung einsträngiger RNA-Moleküle

Markus E. Nebel

Johann Wolfgang Goethe-Universität

Fachbereich Biologie und Informatik, Institut für Informatik

Frankfurt am Main

Germany

Die Horton-Strahler Nummer eines binären Baumes besitzt viele anschauliche Interpretationen in den verschiedensten Wissenschaften. Ein Beispiel aus der Informatik ist der Registerbedarf arithmetischer Ausdrücke aus binären Operatoren. Markiert man den zum Ausdruck gehörenden Syntaxbaum entsprechend der Horton-Strahler Nummer, so gibt die Markierung der Wurzel die Anzahl der Register wieder, die mindestens notwendig sind, um den Ausdruck mittels einer Baumtraversierung zu evaluieren.

In [1] wurde gezeigt, wie durch die Verwendung multivariabler Erzeugendenfunktionen bestimmte mit der Horton-Strahler Nummer verknüpfte Parameter binärer Baumstrukturen uniform, d.h. für viele Baumfamilien gleichzeitig, analysiert werden können. Die dabei hergeleiteten Erzeugendenfunktionen können auch verwendet werden, um ein lange offenes Problem aus der Molekularbiologie zu lösen. Waterman [2] führte 1978 die *Ordnung* eines einsträngigen RNA-Moleküls ein; ein Parameter, der direkt mit dem Energieniveau des Moleküls verknüpft ist. Erst 1985 lösten Vauchassade de Chaumont und Viennot das von Waterman offen gelassene Problem, die Erzeugendenfunktion der RNA-Moleküle der Ordnung  $k$  zu bestimmen. Ergebnisse, wie z.B. das asymptotische Wachstum der Koeffizienten, wurden jedoch nicht abgeleitet. Die gefundene Erzeugendenfunktion jedoch zeigte einen engen Zusammenhang zwischen der Ordnung eines RNA-Moleküls und der Horton-Strahler Nummer eines binären Baumes auf. Diesen Zusammenhang konnten wir zur Herleitung unserer Ergebnisse ausnutzen. Wir werden in diesem Vortrag sehen, wie die Erzeugendenfunktionen aus [1] durch entsprechende Substitutionen benutzt werden können, um die Ordnung einsträngiger RNA-Moleküle zu untersuchen. So zeigen wir, daß die erwartete Ordnung eines Moleküls der Größe  $n$  gegeben ist durch

$$\frac{1}{2} \log_2 \left( \frac{2\pi^2}{\rho^2} n \right) - \frac{\gamma + 2}{2 \ln(2)} + \Delta \left( \log_2 \left( \frac{n}{\rho^2} \right) \right) + \mathcal{O}(n^{-1}), \quad n \rightarrow \infty,$$

wobei  $\rho := \frac{\sqrt{9\sqrt{5}-20}}{5\sqrt{5}-11} = 1.95743\dots$  und  $\Delta$  eine periodische Funktion von kleinem Betrag ( $\leq 0.040597\dots$ ) und einer bekannten Fourier-Reihenentwicklung ist. Die

zugehörige Varianz ist asymptotisch  $0.17939\dots$  plus einer (ebenfalls bekannten) periodischen Funktion von kleinem Betrag. Damit beschreibt der Erwartungswert die "Realität". Außerdem zeigen wir eine Asymptotik für das  $r$ -te Moment und skizzieren, wie die Anzahl der Moleküle aus  $n$  Basen der Ordnung  $k$  bestimmt werden kann. Bisher war diese Anzahl nur für  $k = 1$  bekannt.

## Literatur

- [1] M. E. NEBEL, *A Uniform Approach to the Analysis of Horton-Strahler Parameters of Binary Tree Structures*, submitted.
- [2] M. S. WATERMAN, *Secondary Structures of Single-Stranded Nucleic Acids* in *Studies in Foundations and Combinatorics, Ad. in Maths. Suppl. Studies*, vol. 1, 167-212, 1978.

# Loop languages, computational complexity and the low Grzegorzcyk classes

Lars Kristiansen

Oslo University College

Faculty of Engineering, Norway

e-mail: `larskri@iu.hio.no`

Karl-Heinz Niggl

Technische Universität Ilmenau

Institut für Theoretische Informatik

email: `niggl@theoinf.tu-ilmenau.de`

We consider an imperative programming language  $L$ , a slight modification of a loop language studied intensively in the literature. If a program in  $L$  satisfies certain syntactical criteria, we can guarantee that the program runs in polynomial time (in the unary length of the inputs). In general, it is possible to (automatically) extract a bound on the number of steps in an execution of a program in  $L$  from its syntax. More precisely, we propose a *measure*  $\mu$  which assigns to each program in  $L$  a natural number computable from the syntax of the program.

It is shown for all  $n \geq 0$  that a function  $f$  belongs to Grzegorzcyk class  $\mathcal{E}^{n+2}$  if and only if  $f$  is computable by an  $L$ -program with  $\mu$ -measure  $n$ .

Thus, the measure  $\mu$  separates functions (algorithms) in  $\mathcal{E}^2$  – known to characterise the linear-space computable functions – from functions (algorithms) in  $\mathcal{E}^3$  – known to characterise the Kalmar-elementary functions (iterated exponential growth) –, and so forth.

The hierarchy of Grzegorzcyk classes  $\mathcal{E}^{n+2}$  will be introduced in the talk, in particular no prior knowledge in subrecursion theory is presupposed.



# Oblivious Permutation Routing on the Mesh

Andre Osterloh

Technische Universität Ilmenau

PF 10 05 65, 98684 Ilmenau, Germany

osterloh@theoinf.tu-ilmenau.de

The problem of routing packets through a network is fundamental to the study of parallel computations. The canonical routing problem is that of (*partial*) *permutation* routing. In this case each processor initially and finally contains (at most) one packet. Many different approaches to solve the permutation routing problem have been studied (e.g. adaptive, hot-potato, cut-through, wormhole, fault-tolerant, local, etc. [GHS98]). In [Val82] the concept of *oblivious* routing was introduced. In this routing strategy the path of each packet is determined only by its source and destination position. It depends not on other packets. This makes oblivious routing simple and attractive.

One of the most studied parallel models with a fixed interconnection network is the  $d$ -dimensional mesh-connected processor array. A  $d$ -dimensional mesh  $M_{d,n}$  of side length  $n$  consists of  $N = n^d$  processors, where each processor is identified by a  $d$ -tuple  $(p_0, \dots, p_{d-1})$  in  $[n]^d$ . (For  $x \in \mathbb{N}$  let  $[x]$  be the set  $\{0, \dots, x-1\}$ .) Two processors  $P = (p_0, \dots, p_{d-1})$  and  $Q = (q_0, \dots, q_{d-1})$  of  $M_{d,n}$  are connected by a bidirectional communication link iff  $\sum_{i \in [d]} |p_i - q_i| = 1$ .

We give new bounds for oblivious permutation routing on  $d$ -dimensional meshes ( $d > 1$ ) of side length  $n$ . We present  $O(n^{d/2} \log n)$  algorithm using queue-size  $O(1)$  for the case  $d \geq 2$ . Further we present optimal  $O(n^{d/2})$  algorithms using unlimited queue-size.

## Literatur

- [Val82] L. G. Valiant. A scheme for fast parallel communication. SIAM Journal on Computing, 11(2):350-361, May 1982.
- [GHS98] Miltos D. Grammatikakis and D. Frank Hsu and Jop F. Sibeyn. Packet Routing in Fixed-Connection Networks: A Survey. Journal of Parallel and Distributed Computing, 54(2):77-132, 1998.

# New approximation algorithm for the RTILE problem

Krzysztof Loryś and Katarzyna Paluch

Institute of Computer Science, Wrocław University

The RTILE (Rectangle Tiling) problem we consider belongs to a very wide class of discrete optimization tiling problems. Problems from this class generally require partitioning a multidimensional data set into rectangular partitions so that some optimality criteria are satisfied. In the case of the RTILE problem we are given an array  $A$  of nonnegative numbers, a positive number  $p$  and the task consists in finding the partition of  $A$  into at most  $p$  rectangular tiles so as to minimize the maximal weight of the tile. In the partition the tiles are not to overlap and they are to cover the whole array  $A$ . By the weight of the tile we mean the sum of all the elements that fall within it.

The significance of these problems follows from their practical applicability in many areas of computer science including load balancing in parallel computing environments, data compression, building two-dimensional histograms, and many others.

The problem is not only known to be NP-hard, but it is also NP-hard to approximate it within a factor of  $\frac{5}{4}$ .

We present a  $\frac{9}{4}$ -approximation algorithm .

# Backing up in Lists with Restricted Assignments <sup>1</sup>

Holger Petersen  
Institut für Informatik  
Universität Stuttgart  
Breitwiesenstr. 20–22  
D-70565 Stuttgart

The class LOGSPACE has been characterized in terms of a programming language operating on a linear input list which cannot be modified [2, 3]. This language lacks the ability to allocate memory in addition to the input data and—adopting LISP’s terminology—can be called cons-free.

An interesting problem mentioned in [2] is the question, whether the inability to “back up” in the input leads to a slow-down relative to a computational model that can freely move pointers on the input. In [1] it was shown that the slow-down can be reduced to  $O(n^\varepsilon)$  (where  $n$  is the length of the input) for any  $\varepsilon > 0$ . On the other hand, for a given number of variables there are tasks that require an overhead of  $\Omega(n^\varepsilon)$  for some  $\varepsilon$ . The upper bound is independent of the ability to compare variables.

In this talk we investigate a restriction of the programming language, which excludes general assignments and only allows to reset variables to the first position of the input. In addition variables can be compared. First we show that backing up can be simulated with an overhead of  $O(n/\log n)$  steps. For the task of reversing a list we obtain an  $\Omega(n^2/\log n)$  lower bound, which shows the upper bound to be tight.

## Literatur

- [1] A. M. Ben-Amram and H. Petersen. Backing up in singly linked lists. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC'99)*, pages 780–788, 1999.
- [2] N. D. Jones. *Computability and Complexity — From a Programming Perspective*. MIT Press, Cambridge, Mass., London, England, 1997.
- [3] N. D. Jones. LOGSPACE and PTIME characterized by programming languages. *Theoretical Computer Science*, 228:151–174, 1999.

---

<sup>1</sup>Supported by Deutsche Akademie der Naturforscher Leopoldina, grant BMBF-LPD 9901/8-1 of Bundesministerium für Bildung und Forschung. Part of this work was carried out at the Academic College of Tel-Aviv-Yaffo.

# Hausdorff-Dimension in Exponentialzeit

Jan Reimann

Universität Heidelberg

Die Hausdorff Dimension ist eine Verfeinerung des Lebesgue-Maßes, welche insbesondere in der Geometrischen Masstheorie bei der Untersuchung von fraktalen Strukturen Anwendung findet. Eine Charakterisierung der Hausdorff-Dimension im Cantorraum durch Martingale erlaubt es, effektive und ressourcenbeschränkte Varianten zu betrachten. So ist es z.B. möglich, die *Dimension einer Klasse in E* (der Klasse aller linearer Exponentialzeit berechenbaren Mengen) zu definieren.

Wir bestimmen die Dimension in E von verschiedenen, durch Struktureigenschaften definierten Klassen. So haben die Klasse der immunen Mengen in E und die Klasse der autoreduziblen Mengen in E beide die Dimension 1 in E. Über ein allgemeines Invarianzresultat lässt sich weiter zeigen, dass die Klasse der p-m-vollständigen Mengen Dimension 1 in E hat (jedoch *Maß* 0 in E). Darüber hinaus beweisen wir die Existenz von unteren Kegeln (bzgl. der polynomiellen many-one-Reduzierbarkeit) beliebiger  $\Delta_2^0$ -berechenbarer Dimension in E. In diesem Zusammenhang spielt die binäre Entropiefunktion  $H(p) = -(p \log(p) + (1 - p) \log(1 - p))$  eine wichtige Rolle.

**Zugrundeliegende Arbeit.** Klaus Ambos-Spies, Wolfgang Merkle, Jan Reimann und Frank Stephan, *Hausdorff-Dimension in Exponential Time*, zum Vortrag angenommen auf der *IEEE Conference on Computational Complexity*, Chicago 2001.

**Anschrift.** Jan Reimann, Fakultät für Mathematik und Informatik, Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg.

Email: [reimann@math.uni-heidelberg.de](mailto:reimann@math.uni-heidelberg.de). Telefon: 06221/546282. Telefax: 06221/544465.

# Vergleich von Inferenztypen beim Uniformen Lernen

Sandra Zilles

Fachbereich Informatik

Universität Kaiserslautern

Postfach 3049

D - 67653 Kaiserslautern

`zilles@informatik.uni-kl.de`

Im hier betrachteten Lernmodell soll eine Menge von rekursiven Funktionen anhand einer berechenbaren Strategie im Limes identifiziert werden. Dabei verlangt man von der Lernstrategie, aus sukzessive wachsenden Teilstücken eines Funktionsgraphen aus der gegebenen Menge eine Hypothesenfolge zu generieren, die gegen eine korrekte Hypothese konvergiert. „Korrekt“ heißt hier, daß die ausgegebene Hypothese eine Nummer der zu erlernenden Funktion in einer gegebenen Numerierung ist. Durch verschiedene Anforderungen an die dabei erzeugten Zwischenhypothesen erhält man neue Lernmodelle (Inferenztypen), die in bezug auf ihre Lernpotenz bereits mit dem Lernen im Limes verglichen worden sind.

Beim Uniformen Lernen geht es nun darum, für unendlich viele Mengen rekursiver Funktionen auf uniforme Weise jeweils geeignete Lernstrategien zu entwickeln. Auch hier kann man die Lernpotenz verschiedener Inferenztypen vergleichen. Wählt man eine feste Numerierung, die für alle Funktionenmengen als Hypothesenraum dienen soll, so ergeben sich ähnliche Hierarchien wie im ursprünglichen Fall. Erlaubt man jedoch verschiedene Hypothesenräume für die verschiedenen Mengen von Funktionen, so weichen die Ergebnisse von den bisher bekannten Resultaten ab. Interessant ist, daß beim Uniformen Lernen in den meisten Fällen bereits endliche Funktionenmengen ausreichen, um zwei Inferenztypen voneinander zu trennen.

# Software Visualisierung

Oliver Zlotowski

Universität Trier

Die Leistung moderner Computersysteme hat die Entwicklung rechnergestützter Darstellungen komplizierter Sachverhalte in den letzten Jahren stark beschleunigt. So sind viele Bereiche der Industrie, Wirtschaft und Wissenschaft in hohem Maße von derartigen Visualisierungen abhängig.

In diesem Vortrag werde ich einen kurzen Überblick über das Gebiet der Visualisierung von Software geben und einige Ideen für eine Umsetzung eines solchen SV-System darlegen. Dabei wird die spätere Anwendungsumgebung in Abhängigkeit von der (technischen) Realisierung genauer untersucht.