



Mathematisches Institut  
Abteilung für Informatik

Abstracts zum

**39. WORKSHOP ÜBER EFFIZIENTE ALGORITHMEN,  
DATENSTRUKTUREN UND KOMPLEXITÄTSTHEORIE**

Edited by Frank Gurski

Düsseldorf, 30. November 1999

# 39. WORKSHOP ÜBER EFFIZIENTE ALGORITHMEN, DATENSTRUKTUREN UND KOMPLEXITÄTSTHEORIE

Düsseldorf, 30. November 1999

Heinrich-Heine-Universität Düsseldorf, Universitätsstrasse 1

## Programm

9.25 - 9.30	Begrüßung
9.30 - 10.00	Jörg Rothe (Jena) <i>Heuristiken versus Vollständigkeit</i>
10.00 - 10.30	Hans Ulrich Simon (Bochum) <i>Kombinatorische Invarianten zur Bestimmung der Komplexität des Lernens durch Anfragen</i>
10.30 - 11.00	Kaffeepause
11.00 - 11.30	Carsten Damm (Trier), Markus Holzer, Pierre McKenzie (Québec) <i>The Complexity of Tensor Calculus</i>
11.30 - 12.00	Ulrich Hertrampf (Stuttgart), Steffen Reith, Heribert Vollmer (Würzburg) <i>A Note on Closure Properties of Logspace MOD Classes</i>
12.00 - 13.30	Mittagspause
13.30 - 14.00	Till Tantau (Berlin) <i>Kombinatorische Repräsentation der Truth-Table-Reduktionsabschlüsse von selektiven Sprachen</i>
14.00 - 14.30	Stefan Edelkamp (Freiburg) <i>Sorting: Something Old, Something New, ...</i>
14.30 - 15.00	Thomas Schwentick (Mainz) <i>Ausdrucksstarke und effiziente Abfragesprachen für baumartig strukturierte Daten</i>
15.00	Ende des Workshops

# Heuristiken versus Vollständigkeit

Jörg Rothe

Institut für Informatik  
Friedrich-Schiller-Universität Jena  
rothe@informatik.uni-jena.de

Es werden verschiedene Heuristiken für NP-vollständige Probleme wie z.B. das Graphfärbbarkeitsproblem betrachtet. Für eine fixierte Heuristik werden Antworten auf folgende Fragen gegeben:

1. Wie schwer ist es, diejenigen Probleminstanzen zu erkennen, für die die gegebene Heuristik das Problem (optimal) lösen kann?
2. Wie schwer ist es, diejenigen Probleminstanzen zu erkennen, für die die gegebene Heuristik die Problemlösung in einem fixierten konstanten Faktor approximieren kann?

# Kombinatorische Invarianten zur Bestimmung der Komplexität des Lernens durch Anfragen

Hans Ulrich Simon

Fakultät für Mathematik  
Ruhr-Universität Bochum  
simon@lmi.ruhr-uni-bochum.de

Eines der Standardmodelle in der Theorie des algorithmischen Lernens ist Angluins Modell des *Lernens durch Anfragen an ein Orakel (query learning)*. Klassische Anfragetypen sind die sogenannten “equivalence queries” (kurz EQs) und “membership queries” (kurz MQs). Ziel des Lerners ist es, mit möglichst wenig Anfragen ein unbekanntes Zielkonzept aus einer a-priori bekannten Konzeptklasse  $C$  zu identifizieren.  $C$  ist dabei eine Klasse 0,1-wertiger Funktionen auf einem festen Grundbereich  $X$ . Die vom Lerner benötigte *Anzahl von Fragen zum Lernen von  $C$*  ist definiert als die maximale Anzahl von Fragen, die zur Identifikation eines Zielkonzeptes aus  $C$  benötigt werden.

Wir assoziieren drei kombinatorische Invarianten zu einer gegebenen Konzeptklasse  $C$ : die *Konsistenzdimension*, die *starke Konsistenzdimension* und die *Sphärenzahl* von  $C$ . Wir weisen folgendes nach. Erstens: die Konsistenzdimension ist proportional zur Gesamtanzahl von EQs und MQs zum Lernen von  $C$ . Zweitens: die starke Konsistenzdimension ist proportional zur Anzahl von EQs (ohne Verwendung von MQs). Drittens: falls die Hypothesen des Lerners nicht “komplexer” sein dürfen als das unbekannte Zielkonzept, dann sind starke Konsistenzdimension und Sphärenzahl identisch.

Aus den genannten Resultaten lässt sich ferner folgendes Theorem ableiten. Als Einschränkung von  $C$  bezeichnen wir jede Klasse, die aus  $C$  durch Einschränkung des Grundbereichs  $X$  auf einen Unterbereich  $X' \subseteq X$  hervorgeht. Dann gilt: die Gesamtanzahl von EQs und MQs zum Lernen von  $C$  ist proportional zur Anzahl von EQs zum Lernen der schwersten Einschränkung von  $C$ .

# The Complexity of Tensor Calculus

Carsten Damm<sup>1</sup>

Fachbereich IV — Informatik  
Universität Trier  
damm@uni-trier.de

Markus Holzer<sup>2</sup>, Pierre McKenzie

Département d'I.R.O.  
Université de Montréal(Québec)  
{holzer,mckenzie}@iro.umontreal.ca

Let  $\mathcal{S}$  be the Boolean semiring  $B = (\{0, 1\}, \vee, \wedge)$  or a ring. An  $\mathcal{S}$ -**tensor** is a “multidimensional matrix” over  $\mathcal{S}$ . Sum, tensor product, and junction (partial trace) of tensors are defined as usual. We study the expressive power of this calculus by characterizing the complexity of the following problem:

$\text{TEP}_{\mathcal{S}}$ : Decide, whether an  $\mathcal{S}$ -tensor  $t$  given by an expression involving matrices (i.e., two-dimensional tensors) and the operations sum, tensor product, and junction is non-zero.

We obtain the following results:

(1)  $\text{TEP}_B$  and  $\text{TEP}_{\text{GF}(2)}$  are complete for NP and  $\oplus\text{P}$ , resp., under polynomial time reductions.

(2) Restricted to formulas containing only small sub-formulas, these problems are complete for LOGCFL and  $\oplus\text{LOGCFL}$ , resp., under log-space reductions. Restriction to formulas without tensor products leads to complete problems for NL and  $\oplus\text{L}$ , resp. (Analogous results can be proved for tensors over finite fields and over the integers.)

(3) The permanent of a matrix  $A$  can be described by a tensor expression involving only  $A$ , constant matrices, and the operations above. (This resembles Berkowitz’ theorem that gives a similar formula for the determinant of a matrix but without tensor products.)

We apply (1) and (2) to give a unified and intuitive proof for the inclusions

- $\text{NP}/\text{poly} \subseteq \oplus\text{P}/\text{poly}$ ,
- $\text{LOGCFL}/\text{poly} \subseteq \oplus\text{LOGCFL}/\text{poly}$ , and
- $\text{NL}/\text{poly} \subseteq \oplus\text{L}/\text{poly}$ .

The result in (3) shows that evaluating a tensor expression over the semiring  $(\mathbb{N}, +, \times)$  of natural numbers is Turing-complete for  $\#\text{P}$ .

---

<sup>1</sup>Supported by DFG grant Me 1077/14-1

<sup>2</sup>Supported in part by NSERC of Canada

# A Note on Closure Properties of Logspace MOD Classes

Ulrich Hertrampf

Abteilung Theoretische Informatik  
Universität Stuttgart  
Hertrampf@informatik.uni-stuttgart.de

Steffen Reith, Heribert Vollmer

Theoretische Informatik  
Universität Würzburg  
{streit,vollmer}@informatik.uni-wuerzburg.de

We investigate logspace mod-classes, introduced and studied by Buntrock, Damm, Hertrampf, and Meinel (Mathematical Systems Theory 25 (1992), pp. 223–237). We add to their list of closure properties by proving that the classes  $\text{MOD}_q\text{L}$  for prime number  $q$  are closed under Turing reductions. As an immediate consequence we obtain that the logspace  $\text{MOD}_q\text{L}$  hierarchies collapse for prime  $q$ .

# Kombinatorische Repräsentation der Truth-Table-Reduktionsabschlüsse von selektiven Sprachen

Till Tantau

Fachbereich Informatik  
TU Berlin  
tantau@cs.tu-berlin.de

Eine Sprache ist p-selektiv [4], falls es einen Polynomialzeitalgorithmus gibt, der für je zwei Wörter eines auswählt, so dass wenn wenigstens eines der Wörter in der Sprache ist, so auch das ausgewählte. Die Motivation für die Untersuchung der Reduktionsabschlüsse von p-selektiven Sprachen ist, dass zum einen der Turing-Reduktionsabschluss der p-selektiven Sprachen gerade P/poly ist [4, 3], zum anderen SAT nur dann mit sublinear vielen Fragen auf eine p-selektive Sprache tt-reduzierbar ist, wenn  $P = NP$  [1].

In [2] wurde gezeigt, dass die  $k$ -tt-Reduktionsabschlüsse der p-selektiven Sprachen eine echte Hierarchie bilden. Dieses Resultat soll im Vortrag durch ein kombinatorisches Beweisverfahren reproduziert und verallgemeinert werden.

Auswahlalgorithmen für p-selektive Sprachen sind nur Spezialfälle sogenannter Teilinformatinalgorithmen. Diese berechnen für eine Sprache  $L$  bei Eingabe von  $n$  Wörtern  $w_1, \dots, w_n$  eine Menge  $D$  von Bitstrings, auch Pool genannt, mit  $\chi_L(w_1, \dots, w_n) \in D$ . Hierbei ist  $\chi_L$  die auf Tupel fortgesetzte charakteristische Funktion. Für die p-selektiven Sprachen ist beispielsweise  $n = 2$  und die Auswahlalgorithmen geben immer einen der beiden Pools  $\{00, 01, 11\}$  und  $\{00, 10, 11\}$  aus.

Es soll gezeigt werden, dass die kleinsten Pools, die allgemein für Sprachen im  $k$ -tt-Reduktionsabschluss der p-selektiven Sprachen berechnet werden können, genau Walks auf dem  $n$ -Hyperwürfel sind, bei denen jede Bitposition maximal  $k$  mal ihren Wert ändert. Da sich leicht zeigen lässt, dass es für unterschiedliche  $k$  unterschiedlich lange Walks gibt, folgt, dass die  $k$ -tt-Reduktionsabschlüsse eine echte Hierarchie bilden.

Die Repräsentation der Reduktionsabschlüsse der p-selektiven Sprachen durch Walks lässt sich leicht auf andere rekursiv präsentierbare Sprachklassen erweitern. Insbesondere bilden auch alle  $k$ -tt-logspace-Reduktionsabschlüsse der logspace-selektiven Sprachen eine echte Hierarchie.

Eine genauere Untersuchung zeigt, dass die Repräsentation sogar im rekursiven Fall gültig ist. Insbesondere sind alle  $k$ -tt-Reduktionsabschlüsse der semirekursiven Sprachen unterschiedlich.

## Literatur

- [1] R. Beigel, M. Kummer, and F. Stephan. Approximable sets. *Information and Computation*, 120(2):304–314, 1995.
- [2] L. A. Hemaspaandra, A. Hoene, and M. Ogihara. Reducibility classes of P-selective sets. *Theoretical Comput. Sci.*, 155(2):447–457, 1996.

- [3] K. Ko. On self-reducibility and weak P-selectivity. *J. Comput. Syst. Sci.*, 26(2):209–221, 1983.
- [4] A. L. Selman. P-selective sets, tally languages, and the behaviour of polynomial time reducibilities on NP. *Math. System Theory*, 13:55–56, 1979.



# Sorting: Something Old, Something New, . . .

Stefan Edelkamp

Institut für Informatik  
Albert-Ludwigs-Universität Freiburg  
edelkamp@informatik.uni-freiburg.de

*BOTTOM-UP-HEAPSORT* is a variant of *HEAPSORT* with  $1.5n \log n + O(n)$  comparisons in the worst case. *MDR-HEAPSORT* proposed by McDiarmid and Reed improves this to  $n \log n + 1.1n$ . Dutton presents a further *HEAPSORT* variant called *WEAK-HEAPSORT*, which also contains a new data structure for priority queues. The sorting algorithm and the underlying data structure are analyzed showing that *WEAK-HEAPSORT* is the best *HEAPSORT* variant and that it has a lot of nice properties. It is shown that the worst case number of comparisons is  $n \lceil \log n \rceil - 2^{\lceil \log n \rceil} + n - \lceil \log n \rceil \leq n \log n + 0.1n$  and *Weak-Heaps* can be generated with  $n - 1$  comparisons. A double-ended priority queue based on *Weak-Heaps* can be generated in  $n + \lceil n/2 \rceil - 2$  comparisons. Moreover, examples for the worst and the best case of *WEAK-HEAPSORT* are presented, the number of *Weak-Heaps* on  $\{1, \dots, n\}$  is determined, and experiments on the average case are reported.

## Literatur

- [1] R. D. Dutton. Weak-heap Sort. *BIT*, 33:372–381, 1993.
- [2] C. J. H. McDiarmid and B. A. Reed. Building heaps fast. *Journal of Algorithms*, 10:352–365, 1989.
- [3] Stefan Edelkamp. *WEAK-HEAPSORT*, a fast sorting algorithm (in German). Master's thesis, University of Dortmund, 1996.
- [4] Stefan Edelkamp and I. Wegener. On the performance of *WEAK-HEAPSORT*. In *STACS 2000*. To appear

# Ausdrucksstarke und effiziente Abfragesprachen für baumartig strukturierte Daten

Thomas Schwentick

Institut für Informatik  
Johannes Gutenberg-Universität Mainz  
tick@informatik.uni-mainz.de

Einer der Gründe für den Erfolg relationaler Datenbanken ist die Existenz einer robusten Query-Sprache, First-Order-Logik. Zusammen mit ihren nahen Verwandten, SQL und relationale Algebra, ist sie ausdrucksstark, recht effizient auswertbar und erlaubt die einfache Formulierung von Abfragen.

In vielen neueren Datenbank-Anwendungen sind die Daten jedoch nicht so gleichförmig strukturiert wie in einer relationalen Datenbank. Einen wichtigen Fall stellen baumartig strukturierte Daten dar, die zum Beispiel in heterogenen Datenbank-Umgebungen oder als strukturierte Dokumente vorkommen. Aufgrund des komplizierteren Datenmodells wird für solche Daten eine ausdrucksstärkere Abfragesprache benötigt, für die die Auswertung dann aber komplizierter wird.

Es wäre wünschenswert, auch für solche Daten eine Abfragesprache mit den oben genannten Eigenschaften zu haben. Ein natürlicher Kandidat hierfür ist die monadische Logik zweiter Stufe (MSO). Im Kontext von Baumsprachen wurde diese Logik bereits eingehend untersucht. Sie definiert die sogenannten regulären Baumsprachen, eine natürliche Verallgemeinerung der regulären string-Sprachen. Zu jeder MSO-Formel gibt es einen äquivalenten bottom-up-Baumautomaten, dessen Verhalten auf einem gegebenen Baum in linearer Zeit ausgewertet werden kann.

Ist die Abfrage-Formel jedoch selbst Teil der Eingabe wird das Auswertungsproblem PSPACE-vollständig und der natürliche Auswertungsalgorithmus benötigt exponentielle Zeit in der Größe des Baumes.

Der Vortrag stellt eine Teillogik von MSO für einstellige Abfragen vor, die

- alle MSO-Abfragen ausdrücken kann,
- in linearer Zeit in der Größe des Baumes und exponentieller Zeit in der Größe der Formel ausgewertet werden kann, und
- es erlaubt Abfragen in “natürlicher” Weise zu formulieren.