T U M

I N S T I T U T F Ü R I N F O R M A T I K

50. Workshop über Komplexitätstheorie, Datenstrukturen und Effiziente Algorithmen

Ernst W. Mayr Sven Kosub (Hrsg.)

TUM-I0403
März 2004

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-03-I0403-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

50. Workshop über Komplexitätstheorie, Datenstrukturen und Effiziente Algorithmen

9. März 2004

Technische Universität München, Fakultät für Informatik
Campus Garching, FMI-Gebäude

Raum MI 00.08.038 (Multimedia-Hörsaal)

Programm

9:45-10:00	Begrüßung
10:00-10:30	Alexander Offtermatt-Souza, Angelika Steger (ETH Zürich) *The Expected Competitive Ratio for Weighted Completion Time Scheduling*
10:30-11:00	Arfst Nickelsen, Birgit Schelm (TU Berlin) $HP \neq AvgP$
11:00-11:30	Pause
11:30-12:00	Henning Fernau (U Newcastle, Callaghan & U Tübingen) *Parametric Duality: Kernel Size & Algorithmics*
12:00-12:30	Jiong Guo, Rolf Niedermeier (U Tübingen) *Exact Algorithms for Tree-like Weighted Set Cover*
12:30-13:00	Sebastian Wernicke (TU München), Jochen Alber, Jens Gramm, Jiong Guo, Rolf Niedermeier (U Tübingen) *Vermeidung verbotener Teilmatrizen durch Zeilenlöschungen*
13:00-14:00	Mittagspause
14:00-14:30	Elmar Böhler, Christian Glaßer, Bernhard Schwarz, Klaus W. Wagner (U Würzburg) *Generierungsprobleme*
14:30-15:00	Daniel Meister (U Würzburg) *Auswertung von rekurrenten Systemen über den natürlichen Zahlen*
15:00-15:30	Stephen Travers (U Würzburg) *Die Komplexität von Auswertungsproblemen für Schaltkreise über Mengen von ganzen Zahlen*
15:30-16:00	Pause
16:00-16:30	Edith Hemaspaandra (RIT), Holger Spakowski (U Düsseldorf), Mayur Thakur (U Rochester) *Complexity of Cycle Length Modularity Problems in Graphs*
16:30-17:00	Christian Glaßer (U Würzburg) *Counting with Counterfree Automata*
17:00-17:30	Holger Petersen (U Stuttgart) *Das Eindeutigkeitsproblem auf Turingmaschinen*

The Expected Competitive Ratio for Weighted Completion Time Scheduling

Alexander Offtermatt-Souza *Angelika Steger*
Eidgenössische Technische Hochschule Zürich

A set of n independent jobs is to be scheduled without preemption on one machine. The processing time P_j of each job j is drawn from a certain probability distribution. The scheduler is given the expectation $\mu_j = \mathbb{E}[P_j]$, but the actual duration is not known in advance. A positive weight w_j is associated with each job j and all jobs are ready for execution at time zero. The objective is to minimise the expected competitive ratio $\mathbb{E}\left[\frac{\sum_j w_j C_j}{\text{OPT}}\right]$, where C_j denotes the completion time of job j and OPT the offline optimum value. A so called online list scheduling algorithm is given weight w_j and mean μ_j for all jobs and based on that information, at time zero determines a list of the jobs. This list is then scheduled non-preemptively.

We show a general bound on the expected competitive ratio for online list scheduling algorithms, which holds for a class of refined new-better-than-used processing time distributions. This bound depends on the probability of any pair of jobs being in the wrong order in the list of an arbitrary online list algorithm, compared to an optimum list. As a special case, we show that the so-called WSEPT algorithm achieves $\mathbb{E}\left[\frac{\text{WSEPT}}{\text{OPT}}\right] \leq 2$ for exponential distributed processing times.

The analysis transfers to the case of scheduling m independent identical machines and yields a slightly weaker bound.

HP $\neq$ AvgP

Arfst Nickelsen *Birgit Schelm*
Technische Universität Berlin

Die Klasse AvgP wurde 1984 von Levin eingeführt und besteht aus den randomisierten Problemen, die im Schnitt effizient lösbar sind. Ein randomisiertes Problem ist dabei ein Paar (A, μ) bestehend aus einer Sprache $A \subseteq \Sigma^+$ und einer Eingabeverteilung μ auf Σ^+.

Definition [AvgP] Ein randomisiertes Problem (A, μ) ist in AvgP falls eine DTM M und natürliche Zahlen k and c existieren, so dass

$$L(M) = A \quad \text{und} \quad \sum_{|x| \geq 1} \frac{(t_M(x))^{1/k}}{|x|} \cdot \mu(x) \leq c\,.$$

Die Klasse HP (für *heuristic polynomial time*) wurde 1995 von Impagliazzo eingeführt, erschien aber, in etwas eingeschränkter Form, unter dem Namen ApproxP bereits 1990 in einem Artikel von Schapire.

Definition [HP] Ein randomisiertes Problem (A, μ) ist in HP, falls eine polynomiell zeitbeschränkte DTM M existiert, so dass für alle $m \in \mathbb{N}$ für alle Eingaben der Form$\langle x, 1^m \rangle$, $x \in \Sigma^+$,

$$\text{Prob}_\mu[M(\langle x, 1^m \rangle) \neq \chi_A(x)] < \frac{1}{m}\,.$$

Eine solche DTM wird von Impagliazzo auch als *Algorithmenschema* bezeichnet. Ein Algorithmenschema zeichnet sich dadurch aus, dass der Anteil der fehlerhaften Berechnungen beliebig weit gedrückt werden kann, allerdings auf Kosten der Laufzeit. Schränkt man HP auf p-berechenbare Verteilungen ein, so erhält man die Klasse ApproxP.

Dass AvgP $\subseteq$ HP gilt, ist relativ leicht zu sehen wenn man die Charakterisierung von AvgP durch *gutartige (benign) Algorithmenschemata* betrachtet. Diese sind ähnlich wie die Algorithmenschemata definiert, dürfen statt einer fehlerhaften Ausgaben jedoch nur „? “ausgegeben. Die Wahrscheinlichkeit für eine solche Ausgabe ist ebenfalls durch m^{-1} beschränkt, wobei m Teil der Eingabe ist. Impagliazzo hat gezeigt, dass ein randomisiertes Problem genau dann in AvgP liegt, wenn es ein gutartiges Algorithmenschema besitzt.

Schapire hat vermutet, dass die Einschränkung von AvgP auf randomisierten Probleme mit p-berechenbaren Verteilungen eine echte Teilmenge von ApproxP ist. Wir können diese Vermutung bestätigen, woraus eine entsprechende Beziehung auch für die Klassen AvgP und HP folgt.

Wir konstruieren (zu einer p-berechenbaren Standardverteilung μ) per Diagonalisierung eine Sprache A, so dass $(A, \mu) \in$ HP, aber nicht in AvgP. Wir beweisen also:

Satz: HP $\not\subseteq$ AvgP.

Damit können wir zeigen, dass gutartige Algorithmenschemata eine echte Einschränkung von Algorithmenschemata sind.

Parametric Duality: Kernel Sizes & Algorithmics

Henning Fernau
The University of Newcastle, Callaghan & Eberhard-Karls-Universität Tübingen

We derive certain properties for problems which are parameterized tractable both in their "primal" and in their "dual" parameterization. In particular, we derive the first ever lower bounds result for kernel sizes of parameterized problems. We discuss various consequences of this result.
More precisely, assuming $\mathcal{P}$ is not equal to $\mathcal{NP}$, we can conclude:

1. For any $\epsilon > 0$, there is no $(4/3 - \epsilon)k$ kernel for PLANAR k-VERTEX COVER.
2. Conversely, for any $\epsilon > 0$, there is no $(2 - \epsilon)k_d$ kernel for PLANAR k_d-INDEPENDENT SET.

Moreover, we explain how to get improved non-parameterized algorithms from known parameterized algorithms by a "two-side attack."

Exact Algorithms for Tree-Like Weighted Set Cover

Jiong Guo   *Rolf Niedermeier*
Eberhard-Karls-Universität Tübingen

We study a natural NP-complete special case of the WEIGHTED SET COVER problem and show its fixed-parameter tractability with respect to maximum subset size. More precisely, in this practically relevant variant (which also appears in computational biology in phylogenetics) we restrict the problem instances of SET COVER to be tree-like, which is equivalent to the problem of finding a minimum edge cover in an edge-weighted acyclic hypergraph. Acyclic hypergraphs e.g. arise in the study of relational data bases. Our main result also implies the first fixed-parameter tractability result for the NP-complete MULTICUT IN TREES problem complementing previous approximation results. In addition, we present a polynomial-time dynamic programming algorithm for instances that are only path-like instead of tree-like.

Keywords: NP-hard problems, (Weighted) Set Cover, Multicut in Trees, Minimum Weighted Edge Cover on acyclic hypergraphs, exact algorithms, dynamic programming, fixed-parameter tractability.

[1]Supported by the Deutsche Forschungsgemeinschaft (DFG), Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

Vermeidung verbotener Teilmatrizen durch Zeilenlöschungen

Sebastian Wernicke
Technische Universität München

Jochen Alber *Jens Gramm* *Jiong Guo* *Rolf Niedermeier*
Eberhard-Karls-Universität Tübingen

Es wird eine systematische Untersuchung des Problems Row Deletion(B) auf Matrizen vorgestellt: Gegeben eine (fixe) „verbotene Teilmatrix“ B, und eine Matrix A als Eingabe (die Einträge von sowohl A als auch B entstammen einem Alphabet endlicher Größe), welches ist die minimale Anzahl von Zeilen in A, die gelöscht werden muß, damit A keine Teilmatrix besitzt, die äquivalent zu einer Zeilen- oder Spaltenpermutation von B ist? Eine Anwendung dieser Fragestellung findet sich unter anderem in der Erstellung perfekter Phylogenien. Durch das Aufzeigen einer engen Verbindung zum NP-vollständigen Hitting Set Problem wird gezeigt, daß Row Deletion(B) für viele B NP-vollständig ist. Ein positiver Aspekt dieser Verbindung ist die Existenz von Approximationsalgorithmen mit konstanter Güte sowie Fixparameteralgorithmen für das Row Deletion(B) Problem.

[1]Supported by the Deutsche Forschungsgemeinschaft (DFG), project PEAL (parameterized complexity and exact algorithms), NI 369/1; project OPAL (optimal solutions for hard problems in computational biology), NI 369/2; junior research group „PIAF“ (fixed-parameter algorithms), NI 369/4.

Generierungsprobleme

Elmar Böhler *Christian Glaßer* *Bernhard Schwarz* *Klaus W. Wagner*
Julius-Maximilians-Universität Würzburg

Wir untersuchen die von zweistelligen Polynomen erzeugten Hüllen. Die Erzeugung einer solchen Hülle funktioniert wie folgt. Zunächst einigen wir uns auf ein festes Polynom (z.B. $x^2 + 2y$). Starten wir nun mit einer gegebenen endlichen Menge B von natürlichen Zahlen, so erhalten wir neue Zahlen z, indem wir x und y aus B wählen und (entsprechend dem vereinbarten Polynom) $z = x^2 + 2y$ setzen. Diese generierte Zahl z nehmen wir in B auf und führen den Generierungsschritt erneut aus. Die so erhaltene (im Allgemeinen undendliche) Menge nennen wir die Menge der aus B generierbaren Zahlen. Wir definieren $\mathrm{GEN}(x^2 + 2y)$ als die Menge aller Paare (B, d), so dass d aus B generiert werden kann.

Wir untersuchen $\mathrm{GEN}(p)$ für verschiedene zweistellige Polynome p. Dabei stellt sich heraus, dass $\mathrm{GEN}(p)$ oftmals in NP liegt. Es gibt jedoch auch Fälle (wie unser Beispiel $x^2 + 2y$) bei denen es bisher nicht gelang, die Zugehörigkeit zu NP zu beweisen. Allerdings wissen wir schon, dass viele dieser Probleme NP-hart, also mindestens so schwierig wie NP sind. Anders als bei vielen NP-vollständigen Problemen scheint hier nicht die Härte, sondern die Zugehörigkeit zu NP der schwierigere Teil zu sein.

Auswertung von rekurrenten Systemen über den natürlichen Zahlen

Daniel Meister
Julius-Maximilians-Universität Würzburg

Es bezeichne N die Menge der natürlichen Zahlen. Es seien $\cup$, $\cap$, $^-$, $+$ und $*$ Operationen auf der Potenzmenge der natürlichen Zahlen, die wie folgt definiert sind. Seien A und B Mengen natürlicher Zahlen, so gilt:

1. $A \cup B$ und $A \cap B$ sind die Vereinigung bzw. der Durchschnitt von A und B,
2. $\overline{A} =_{\mathrm{def}} \mathrm{N} \setminus A$,
3. $A + B =_{\mathrm{def}} \{a + b : a \in A \text{ und } b \in B\}$,
4. $A * B =_{\mathrm{def}} \{a \cdot b : a \in A \text{ und } b \in B\}$.

In kanonischer Weise können wir Funktionen unter Verwendung der Operationen $\cup$, $\cap$, $^-$, $+$, $*$ auf der Potenzmenge von N definieren. Falls eine solche Funktion nur Operationen aus $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, *\}$ verwendet, nennen wir sie eine *$\mathcal{O}$-Funktion*. Sei $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, *\}$, und seien $f_1, \ldots, f_n$ n-stellige $\mathcal{O}$-Funktionen. Wir setzen $\mathcal{F} =_{\mathrm{def}} \langle f_1, \ldots, f_n \rangle$. Weiterhin sei $A =_{\mathrm{def}} (a_1, \ldots, a_n)$ mit $a_i \in \mathrm{N}$, $i \in \{1, \ldots, n\}$. Wir nennen $S =_{\mathrm{def}} (\mathcal{F}, A)$ ein *rekurrentes $\mathcal{O}$-System*. Wir definieren weiter:

1. $f_i(0) =_{\mathrm{def}} \{a_i\}$,
2. $f_i(t{+}1) =_{\mathrm{def}} f_i(f_1(t), \ldots, f_n(t))$,
3. $S(t) =_{\mathrm{def}} f_n(t)$.

Rekurrente Systeme sind getaktete Systeme; sie beschreiben Folgen von Mengen natürlicher Zahlen. Bekannte Rekurrenzen lassen sich auf diese Weise formulieren. So beschreibt zum Beispiel das einfache rekurrente $\{+\}$-System $S =_{\mathrm{def}} (\langle f_1(x_1, x_2) =_{\mathrm{def}} x_2, f_2(x_1, x_2) =_{\mathrm{def}} x_1 + x_2 \rangle, (0, 1))$ die Folge der Fibonacci-Zahlen.

Sei wieder $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, *\}$. Wir definieren ein Entscheidungsproblem wie folgt. $\mathrm{M}_{ex}(\mathcal{O})$ ist die Menge aller Paare (S, b) mit S ein rekurrentes $\mathcal{O}$-System und $b \in \mathrm{N}$ so, daß es ein $t \geq 0$ gibt mit $b \in S(t)$. Wir wollen untersuchen, wie sich die Schwierigkeit, $\mathrm{M}_{ex}(\mathcal{O})$ zu entscheiden, in Abhängigkeit von $\mathcal{O}$ verhält. Die eingeschränkten Probleme für $t = 0$ (die Frage nach $b \in S(0)$) wurden eingehend in einer Arbeit von McKenzie und Wagner behandelt. Varianten für $t = 0$ über verschiedenen Grundmengen wurden ebenfalls behandelt (Breunig und Travers). Es zeigt sich in unserem Falle, daß das allgemeine Problem $\mathrm{M}_{ex}(\cup, \cap, ^-, +, *)$ nicht entscheidbar ist; sogar $\mathrm{M}_{ex}(\cup, \cap, +, *)$ und $\mathrm{M}_{ex}(^-, +, *)$ sind nicht entscheidbar. Die übrigen Probleme, soweit untersucht, sind vollständig für die verschiedensten Komplexitätsklassen, angefangen von L über NP zu PSPACE und darüber.

Es ist nicht bekannt, ob $\mathrm{M}_{ex}(\cup, \cap, ^-, +, *)$ aufzählbar ist. Falls dem so wäre, gäbe es einen Entscheidungsalgorithmus, der die Goldbachsche Vermutung beweist oder widerlegt. Dies ist eine triviale Folgerung aus einer Beobachtung von Glaßer im Zusammenhang mit den von McKenzie und Wagner untersuchten Auswertungsproblemen.

Die Komplexität von Auswertungsproblemen für Schaltkreise über Mengen von ganzen Zahlen

Stephen Travers
Julius-Maximilians-Universität Würzburg

Wir betrachten Schaltkreise, die Mengen von ganzen Zahlen berechnen. Neben den arithmetischen Gattern $+$ und $\times$, die als Operationen über Mengen komponentenweise definiert sind, lassen wir außerdem die Mengenoperationen $\cup, \cap$ und $^-$ zu. Die Eingangsgatter werden mit ganzen Zahlen beschriftet. Für solche Schaltkreise untersuchen wir Auswertungsprobleme:
Gegeben ein $\{\cup, \cap, ^-, +, \times\}$-Schaltkreis C und eine ganze Zahl b, ist b in der von C berechneten Menge enthalten?
Lassen wir für $\mathcal{O} \subseteq \{\cup, \cap, ^-, +, \times\}$ nur Gatter aus $\mathcal{O}$ in einem Schaltkreis C zu, so nennen wir C einen $\mathcal{O}$-Schaltkreis; die Einschränkung des oben formulierten Auswertungsproblems auf $\mathcal{O}$-Schaltkreise bezeichnen wir mit $\mathrm{MC}_{\mathbb{Z}}(\mathcal{O})$. Schränken wir $\mathrm{MC}_{\mathbb{Z}}(\mathcal{O})$ auf $\mathcal{O}$-Formeln ein, bezeichnen wir das Problem mit $\mathrm{MF}_{\mathbb{Z}}(\mathcal{O})$. Es zeigt sich, dass die Komplexität von $\mathrm{MC}_{\mathbb{Z}}(\mathcal{O})$ und $\mathrm{MF}_{\mathbb{Z}}(\mathcal{O})$ in Abhängigkeit von $\mathcal{O}$ erheblich variiert.

Das Problem $\mathrm{MC}_{\mathbb{Z}}(\cup, \cap, ^-, +, \times)$ und seine diversen Einschränkungen stellen eine natürliche Modifikation der von McKenzie und Wagner (STACS 2003) untersuchten Auswertungsprobleme für Schaltkreise über Mengen von natürlichen Zahlen dar, die wir hier mit $\mathrm{MC}_{\mathbb{N}}(\ldots)$ bzw. $\mathrm{MF}_{\mathbb{N}}(\ldots)$ bezeichnen.
Wir zeigen, dass sich in einigen Fällen die Komplexität der Probleme über den ganzen Zahlen deutlich von der der korrespondierenden Probleme über den natürlichen Zahlen unterscheidet. Einige der interessanten Unterschiede:

Auswertungsproblem	$\mathbb{N}$-Fall ($X = \mathbb{N}$)	$\mathbb{Z}$-Fall ($X = \mathbb{Z}$)
$\mathrm{MC}_X(\cup, +, \times)$	PSPACE-vollst.	NEXPTIME-vollst.
$\mathrm{MC}_X(\times)$	NL-vollst.	NL$\wedge\oplus$L-vollst.
$\mathrm{MC}_X(^-, +)$	PSPACE-vollst.	P-vollst.
$\mathrm{MF}_X(^-, +)$	PSPACE-vollst.	L-vollst.

Überraschenderweise ist es nicht immer möglich, den $\mathbb{N}$-Fall auf den $\mathbb{Z}$-Fall zu reduzieren: Es gilt $\mathrm{MF}_{\mathbb{N}}(^-, +) \not\leq_m^{\log} \mathrm{MF}_{\mathbb{Z}}(^-, +)$ und $\mathrm{MC}_{\mathbb{N}}(^-, +) \leq_m^{\log} \mathrm{MC}_{\mathbb{Z}}(^-, +)$ gdw. $\mathrm{P} = \mathrm{PSPACE}$.

Da die Fälle $\mathrm{MF}_{\mathbb{N}}(^-, +)$ und $\mathrm{MC}_{\mathbb{N}}(^-, +)$ bei McKenzie und Wagner nicht behandelt wurden, beweisen wir deren PSPACE-Vollständigkeit hier ebenfalls. Als Hilfsmittel führen wir das neue Problem *Quantified Sum of Subset* ein und weisen seine PSPACE-Vollständigkeit nach.

Complexity of Cycle Length Modularity Problems in Graphs

Edith Hemaspaandra
Rochester Institute of Technology

Holger Spakowski
Heinrich-Heine-Universität Düsseldorf

Mayur Thakur
University of Rochester

We study the computational complexity of *cycle length modularity problems*. Roughly speaking, in a cycle length modularity problem, given an input (undirected or directed) graph, one has to determine whether the graph has a cycle C of a specific length (or one of several different lengths), modulo a fixed integer. We denote the two families (one for undirected graphs and one for directed graphs) of problems by (S,m)-UC and (S,m)-DC, where $m \in \mathbb{N}$ and $S \subseteq \{0,1,\ldots,m-1\}$. (S,m)-UC (respectively, (S,m)-DC) is defined as follows: Given an undirected (respectively, directed) graph G, is there a cycle in G whose length, modulo m, is a member of S? In this paper, we fully classify (i.e., as either polynomial-time solvable or as NP-complete) each problem (S,m)-UC such that $0 \in S$ and each problem (S,m)-DC such that $0 \notin S$. We also give a sufficient condition on S and m for the following problem to be polynomial-time computable: (S,m)-UC such that $0 \notin S$.

[1]Supported in part by grants NSF-INT-9815095/DAAD-315-PPP-gü-ab, NSF-CCR-0311021, a DAAD grant, and by DFG project RO 1202/9-1. Will be presented at the 6th Latin American Theoretical Informatics conference (LATIN 2004).

Counting with Counterfree Automata

Christian Glaßer
Julius-Maximilians-Universität Würzburg

Viele bekannte Komplexitätsklassen lassen sich mit Hilfe von Blattsprachen beschreiben. Hierzu stellt man sich zunächst eine nichtdeterministische Maschine bei Eingabe x vor, die auf jedem Pfad ein Symbol ausgibt (sagen wir 0 oder 1). Die so ausgegebenen Symbole erzeugen ein Blattwort w. Wir sagen *x wird von der Maschine akzeptiert*, wenn w Element einer bestimmten Sprache L ist.

Zum Beispiel können wir nun die Klasse NP beschreiben, in dem wir $w \in \{0,1\}^*1\{0,1\}^*$ fordern (d.h. es existiert mindestens ein akzeptierender Pfad). Um gängige Komplexitätsklassen zwischen P und PSPACE mit Hilfe von Blattsprachen zu beschreiben, reicht es im Allgemeinen aus, sich auf reguläre Sprachen L zu beschränken. Wenn die nichtdeterministische Maschine balancierte Berechnungsbäume aufspannt, so sprechen wir von einer balancierten Blattsprache; andernfalls von einer unbalancierten Blattsprache.

Im ersten Teil der Arbeit geht es um Automatentheorie, genauer um Verbotsmustercharakterisierungen bestimmter Teilklassen der regulären Sprachen. Im zweiten Teil geht es eher um Komplexitätstheorie. Hier wenden wir die vorher gewonnenen Resultate an. So erhalten wir zum Beispiel Lückensätze der folgenden Art. Wenn man eine Komplexitätsklasse C durch balancierte Blattsprachen beschreiben kann, so gilt: Entweder ist C in NP, oder C enthält coUP. Außerdem sehen wir, dass das unbalancierte Blattsprachenmodell viel stärker als das balancierte ist. Dies gilt überaschenderweise auch, wenn man sich auf sternfreie reguläre Sprachen L beschränkt. (Letztere werden zwar durch sogenannte *counterfree automata* akzeptiert, man kann mit ihnen dennoch in gewisser Weise zählen.)

Das Eindeutigkeitsproblem auf Turingmaschinen

Holger Petersen
Universität Stuttgart

Die Eingabe für das *Eindeutigkeitsproblem* (Element Distinctness Problem, EDP) besteht aus einer Multimenge von m binären Zeichenketten der Länge ℓ. Eine solche Multimenge soll akzeptiert werden, wenn alle Elemente verschieden sind. Es besteht ein enger Zusammenhang zwischen dem EDP und dem Sortierproblem, denn nach dem Sortieren einer Multimenge kann leicht getestet werden, ob es identische Elemente gibt. Verschiedene Arbeiten beschäftigen sich mit der Zeitkomplexität des EDP auf deterministischen und nichtdeterministischen Turingmaschinen mit einem Band [2, 4, 3, 1]. Für diese Modelle konnten asymptotisch optimale Schranken bewiesen werden.

Im Vortrag werden nichtdeterministische Turingmaschinen mit einem Arbeitsband und einem separatem Eingabeband betrachtet. Das EDP kann für $\ell = O(m)$ von solchen Maschinen in Zeit $O(m^{1.5}\sqrt{\ell})$ gelöst werden, während das beste Sortierverfahren die Zeitkomplexität $O(m^{1.5}\ell)$ besitzt [5]. Mit der gleichen Technik kann die Zeitschranke $O(m^2)$ bei Einweg-Eingabe gezeigt werden.

Literatur

[1] Amir M. Ben-Amram, Omer Berkman, and Holger Petersen. Element distinctness on one-tape Turing machines. *Acta Informatica*, 40:81–94, 2003.

[2] Alejandro López-Ortiz. New lower bounds for element distinctness on a one-tape Turing machine. *Information Processing Letters*, 51:311–314, 1994.

[3] Holger Petersen. Bounds for the element distinctness problem on one-tape Turing machines. *Information Processing Letters*, 81:75–79, 2002.

[4] Andrzej Szepietowski. The element distinctness problem on one-tape Turing machines. *Information Processing Letters*, 59:203–206, 1996.

[5] Juraj Wiedermann. Optimal algorithms for sorting on single-tape Turing machines. In J. van Leeuwen, editor, *Algorithms, Software, Architecture, Proceedings of the IFIP 12th World Computer Congress, Madrid, Spain*, volume I, pages 306–314. Elsevier Science Publishers, 1992.