TUM

INSTITUT FÜR INFORMATIK

27. Workshop über Komplexitätstheorie, Datenstrukturen und Effiziente Algorithmen

Stefan Bischof Thomas Erlebach
Volker Heun Ulla Koppenhagen Klaus Kühnle
Ernst W. Mayr Hans Stadtherr

TUM-I9530
November 1995

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-11-1995-I9530-150/1.-FI

Typescript: ---

Druck: Mathematisches Institut und
Institut für Informatik der
Technischen Universität München

Programm des 27. Workshops über Komplexitätstheorie, Datenstrukturen und Effiziente Algorithmen

09:30	*Martin Löbbing, Ingo Wegener (Dortmund)* Zählen mit Binary Decision Diagrams: Die Anzahl der Springertouren beträgt 33.439.123.484.294
09:50	*Carsten Damm (Trier), Stasys Jukna (Vilnius), Jiří Sgall (Prag)* Some Bounds on Multiparty Communication Complexity of Pointer Jumping
10:10	*Sven Schuierer (Freiburg)* Optimal Robot Localization in Trees
10:30	*Oliver Kullmann (Frankfurt)* Worst case-Analyse und untere Schranken für effizientere SAT-Algorithmen
10:50	PAUSE
11:20	*Gerhard Weissenfels (Mainz)* Über die Approximierbarkeit von Färbungsproblemen für geometrische Graphklassen
11:40	*Thomas Hofmeister, Hanno Lefmann (Dortmund)* A Combinatorial Design Approach to MAXCUT
12:00	*Jürgen Dedorath (München), Jordan Gergov, Torben Hagerup (Saarbrücken)* More Efficient Parallel Flow Algorithms
12:20	*Elias Dahlhaus (Sydney)* Ein einfacher effizienter paralleler Algorithmus für die Erkennung von Intervallgraphen
12:40	MITTAGSPAUSE
13:30	Mitgliederversammlung der GI-FG 0.1.3: "Parallele und verteilte Algorithmen" *(im Raum S2229)*
14:15	*Henning Fernau, Klaus-Jörn Lange, Klaus Reinhardt (Tübingen)* Ein Plädoyer für den Eigentümer
14:35	*Harald Hempel (Jena)* Ein allgemeiner Optimierungsoperator
14:55	*Steve Fenner (Portland), Fred Green (Worcester), Steve Homer (Boston), Alan Selman (Buffalo), Thomas Thierauf (Ulm), Heribert Vollmer (Würzburg)* On the Complements of Multivalued Functions
15:15	PAUSE
15:40	*Hans-Jörg Burtschick (Berlin), Wolfgang Lindner (Ulm), Arfst Nickelsen (Berlin)* Über p-selektive E-bi-immune Mengen
16:00	*Thomas Schwentick (Mainz)* Teilklassen von Binärem NP
16:20	*Hans-Jörg Burtschick (Berlin), Heribert Vollmer (Würzburg)* Charakterisierungen von Komplexitätsklassen durch Blattsprachen und endliche Modelltheorie
16:40	PAUSE
17:05	*Bernd Borchert (Heidelberg)* Circuit Kolmogorov Complexity
17:25	*Martin Kummer (Karlsruhe)* Die Komplexität von Zufallsstrings
17:45	*Werner Stein (Kaiserslautern)* Learning Environments leading to Inefficient Learnability
18:05	ENDE DES WORKSHOPS

Zählen mit Binary Decision Diagrams: Die Anzahl der Springertouren beträgt 33.439.123.484.294

Martin Löbbing *Ingo Wegener*

Fachbereich Informatik Lehrstuhl II
Universität Dortmund
D-44221 Dortmund, Germany
{loebbing|wegener}@ls2.informatik.uni-dortmund.de

Zu einer wachsenden Zahl von Erkenntnissen in der Graphentheorie, der Kombinatorik oder der theoretischen Informatik gelangt man heute mit Hilfe des Computers. Der Beweis des Vier-Farben-Theorems oder die Berechnung bestimmter Ramsey-Zahlen sind typische Beispiele. Der Grund dafür ist häufig, daß die Anzahl der zu untersuchenden, atomaren Fälle sehr groß ist.

Es gibt verschiedene Darstellungsformen für Boolesche Funktionen, darunter sind die Binary Decision Diagrams (BDDs), auch Branching Programme genannt. Seit Bryant 1986 OBDDs eingeführt hat, erfreuen sie sich insbesondere auf den Gebieten des Hardwaredesigns und der Verifikation wachsender Popularität.

Je nach Anwendungsgebiet werden verschiedene Ansprüche an die gewählte Datenstruktur gestellt. So sollen Funktionen kompakt dargestellt werden können, sie sollen leicht ausgewertet werden können, die Synthese zweier Funktionen sollte leicht möglich sein, zwei Funktionen sollen auf Äquivalenz getestet werden können, etc.

Verschiedene Varianten von BDDs erfüllen diese Anforderungen mehr oder weniger gut, so ist bei vielen Varianten die Darstellung einer Funktion eindeutig, so daß ein Äquivalenztest einfach ist. Eine weitere Eigenschaft, die einige Varianten besitzen, ist, daß die Anzahl der erfüllenden Belegungen der Funktion effizient bestimmt werden kann. Diese kann dazu benutzt werden, kombinatorische Anzahlprobleme zu lösen. Besonders günstig ist außerdem, daß isomorphe Teilprobleme automatisch erkannt werden.

Um die Anwendbarkeit dieses Ansatzes zu zeigen, haben wir die Anzahl der Springertouren auf einem 8×8 Schachbrett bestimmt. Hierbei besucht ein Springer jedes Feld des Schachbretts genau einmal, um am Ende wieder das Ausgangsfeld zu erreichen. Es handelt sich um die Anzahl der ungerichteten Hamilton-Kreise auf dem Springergraphen, die Knoten entsprechen den Feldern des Schachbretts, jeder legale Zug entspricht einer Kante.

Neben Standardtechniken wie Divide-and-Conquer und Backtracking beschreiben wir die Anforderungen, die an die Darstellungsform für die Booleschen Funktionen gestellt werden, und welche BDD-Varianten diese am besten erfüllen. Damit ist es dann zum ersten Mal möglich, die Zahl der Kreisüberdeckungen als obere Schranke und die exakte Zahl der Springertouren zu bestimmen.

Some Bounds on Multiparty Communication Complexity of Pointer Jumping

Carsten Damm
Fachbereich IV — Informatik
Universität Trier
D-54286 Trier
Germany
damm@uni-trier.de

Stasys Jukna[1]
Institute of Mathematics
and Informatics
Akademijos 4
LT-2600 Vilnius
Lithuania

Jiří Sgall[2]
Mathematical Institute
AV ČR, Žitná 25
115 67 Praha 1
Czech Republic
sgallj@earn.cvut.cz

We introduce the model of *conservative one-way* multiparty complexity and prove lower and upper bounds on the complexity of *pointer jumping*.

The pointer jumping function takes as its input a directed layered graph with a starting node and k layers of n nodes, and a single edge from each node to one node from the next layer. The output is the node reached by following k edges from the starting node.

In a conservative protocol Player i can see only the node reached by following the first $i-1$ edges and the edges on the jth layer for each $j > i$ (compared to the general model where he sees edges of all layers except for the ith one). In a one-way protocol, each player communicates only once: first Player 1 writes a message on the blackboard, then Player 2, etc., until the last player gives the answer. The cost is the total number of bits written on the blackboard.

Our main results are the following bounds on k-party conservative one-way communication complexity of pointer jumping with k layers:

(1) A lower bound of $\Omega(n/k^2)$ for any $k = O(n^{1/3-\varepsilon})$. This is the first lower bound on multiparty communication complexity that works for more than $\log n$ players.

(2) Matching upper and lower bounds of $\Theta(n \log^{(k-1)} n)$ for $k \leq \log^* n$. No better one-way protocols are known, even if we consider non-conservative ones.

Keywords: multiparty communication complexity, one-way communication, pointer jumping

[1] This work was done at Fachbereich IV — Informatik, Universität Trier, Germany, supported by DFG grant Me 1077/5-2.

[2] This work was done at Institute of Computer Science, Hebrew University, Jerusalem, Israel, supported in part by Golda Meir Postdoctoral Fellowship.

Optimal Robot Localization in Trees

Sven Schuierer

Institut für Informatik
Albert-Ludwigs-Universität Freiburg
Am Flughafen 17
D-79110 Freiburg
schuiere@informatik.uni-freiburg.de

The problem of *localization*, i.e. of a robot finding its own position on a map, is an important task for autonomous mobile robots which has applications in numerous areas of robotics ranging from aerial photography to autonomous vehicle exploration. In this paper we present a new strategy for a robot to find its position on a map, where the map is represented as a geometric tree. Our strategy exploits to a high degree the self-similarities that occur in the environment.

We use the framework of competitive analysis to analyze the performance of our strategy. In particular, we show that the distance traveled by the robot is at most $O(\sqrt{n})$ times longer than the shortest possible route to localize the robot, where n is the number of vertices of the tree. This is a significant improvement over the best known previous bound of $O(n^{2/3})$. Moreover, since there is a lower bound of $\Omega(\sqrt{n})$, our strategy is optimal up to a constant factor.

Using the same approach we can also show that the problem of searching for a target in a geometric tree, where the robot is given a map of the tree and the location of the target, can be solved by a strategy with a competitive ratio of $O(\sqrt{n})$.

Worst case-Analyse und untere Schranken für effizientere SAT-Algorithmen

Oliver Kullmann

Fachbereich Mathematik
Johann Wolfgang Goethe-Universität
D-60054 Frankfurt, Germany
kullmann@mi.informatik.uni-frankfurt.de

In meinem Vortrag möchte ich, ausgehend von den Ergebnissen und Methoden in [KuLu95], [Ku95a] und [Ku95b], diskutieren (und auch spekulieren), wie erst durch das Zusammenspiel von worst case-Analyse, algorithmischen Betrachtungen und Reflektion der Methoden durch untere Schranken, neue Methoden zur SAT-Entscheidung entstehen.

Die worst case-Analyse ermöglicht hierbei, als eine Art Kompaß, durch ihre numerische Bewertung der Methoden eine Richtung zu finden und einzuschlagen, in der auch tatsächlich ein Fortschritt liegt, während ohne diese Richtschnur die Suche tendenziell eher ziellos wirkt.

Untere Schranken von Algorithmenklassen aufzuzeigen durch Simulation von Algorithmenläufen auf unerfüllbaren Eingaben mittels Beweissystemen, für die eine untere Schranke bekannt ist, lenkt auf der anderen Seite den Blick auf die prinzipielle Beschränktheit der verwendeten Methoden, und vermag, durch Vergleich mit stärkeren Beweissystemen, Hinweise auf neue Methoden zu geben, die einen qualitativen Sprung mit sich bringen könnten.

Literatur

[Ku95a] KULLMANN, O.: A systematical approach to 3-SAT-decision, yielding 3-SAT-decision in less than 1.5045^n steps. Submitted to: Theoretical Computer Science.

[Ku95b] KULLMANN, O.: A note on a generalization of Extended Resolution. In preparation.

[KuLu95] KULLMANN, O. and LUCKHARDT, H.: Various upper bounds on the complexity of algorithms for deciding propositional tautologies. Submitted to: Information and Computation.

Über die Approximierbarkeit von Färbungsproblemen für geometrische Graphklassen

Gerhard Weissenfels

Institut für Informatik
Fachbereich 17
Johannes Gutenberg-Universität Mainz
Staudingerweg 9
D-55099 Mainz
wf@muwiinfa.geschichte.uni-mainz.de

Eines der schwierigsten Probleme bei der Konzeption von Funknetzen ist die Kanalzuweisung, da einerseits die Anzahl der zur Verfügung stehenden Kanäle beschränkt ist und andererseits starke Interferenzstörungen zwischen Sendern, die den gleichen Kanal benutzen, auftreten.

Wir modellieren ein Sendernetz durch einen Graphen, dessen Knotenmenge die Sender repräsentiert. Eine Kante gibt es in diesem Graphen genau dann, wenn die den Knoten entsprechenden Sender bei Benutzung des gleichen Kanals einen vorgegebenen Störwert überschreiten. Das Kanalzuweisungsproblem in seiner einfachsten Form entspricht in diesem Graphen dem klassischen Färbungsproblem. Da dieses Problem nicht nur NP-vollständig, sondern sogar mit keinem konstanten Faktor approximierbar ist, betrachten wir dem Problem angepaßte spezielle Klassen von Graphen. Da die Störwerte zwischen den Sendern im wesentlichen von deren geographischem Abstand abhängen, bieten sich dafür geometrische Graphklassen an. Als sehr einfache, aber für theoretische Untersuchungen gut geeignete Graphklasse betrachten wir zunächst die Schnittgraphen von Einheitskreisen in der euklidischen Ebene, die sogenannten *Unit-Disk*-Graphen (UD-Graphen). Dann werden Verallgemeinerungen dieser Klasse diskutiert. Ferner wird eine Verallgemeinerung des klassischen Färbungsproblems eingeführt, das eine realistischere Modellierung des Kanalzuweisungsproblems erlaubt.

Das Färbungsproblem für diese Graphklassen haben wir auf Komplexität und Approximierbarkeit untersucht. Neben einigen einfachen sequentiellen Färbungstechniken wird ein ungewöhnlicher Färbungsalgorithmus, der sogenannte Streifenalgorithmus vorgestellt. Dieser zerlegt den Graphen in perfekte Teilgraphen, die in polynomieller Zeit optimal gefärbt werden können und kombiniert diese Teilfärbungen.

Für diese Algorithmen werden obere und untere Qualitätsschranken angegeben, sowohl für das klassische Färbungsproblem als auch für einen Spezialfall der angesprochenen verallgemeinerten Variante.

A Combinatorial Design Approach to MAXCUT

Thomas Hofmeister *Hanno Lefmann*

Lehrstuhl Informatik II
Universität Dortmund
D-44221 Dortmund, Germany
{hofmeist|lefmann}@Ls2.informatik.uni-dortmund.de

The k-MAXCUT problem for undirected graphs $G = (V, E)$ consists of finding a partition $V = V_1 \cup \cdots \cup V_k$ such that the number of edges with endpoints in two different sets V_i is maximized. We offer a new approach to this problem by showing that the combinatorial notion of block designs can be used to algorithmically obtain partitions which achieve lower bounds for which until now only existence proofs were known.

In the case of $k = 2$, we show that already known approaches can be improved by giving a simpler linear-time algorithm which also yields better bounds. In particular, we give a linear-time algorithm which achieves a bound of Edwards which was previously proved by intricate methods.

For general k and graphs with m edges, we are able to compute partitions of size $m \cdot (k-1)/k \cdot (1 + 1/\Delta)$ if the maximum degree Δ of G is odd. The algorithms can also be applied to weighted graphs.

More Efficient Parallel Flow Algorithms

Jürgen Dedorath[1]
Bayerische Hypotheken- und Wechsel-Bank AG
O/EDV-AE3-W1
D–81925 München, Germany
dedorath@Hypo.DE.

Jordan Gergov[2] *Torben Hagerup*[2]
Max-Planck-Institut für Informatik
D–66123 Saarbrücken, Germany
{gergov|torben}@mpi-sb.mpg.de

One of the best sequential algorithms for computing maximum flows is the dynamic-trees algorithm of Goldberg and Tarjan (1988), which runs on networks with n vertices and m edges in $O(nm\log(n^2/m))$ time, while the *maximum-distance discharge* (MDD) algorithm of the same authors was shown by Cheriyan and Maheshwari (1989) to have a worst-case running time of $\Theta(n^2\sqrt{m})$. If the edge capacities are integral and bounded by $U \geq 2$, the excess scaling maximum-flow algorithm of Ahuja and Orlin (1989) achieves a running time of $O(nm + n^2 \log U)$. A parallel implementation of the MDD algorithm that runs in $O(n^2 \log(2m/n + p)(\sqrt{m}/p))$ time on a CREW PRAM with $p \leq \sqrt{m}$ processors was given by Goldberg (1991). Observing that Goldberg's parallel algorithm is slower than the dynamic-trees algorithm unless $2m/n+p = \Omega(n^{1/4})$, we may as well replace $\log(2m/n+p)$ by $\log n$ in his time bound. For $m/n \geq 2$, a parallel version of the algorithm of Ahuja and Orlin can be executed in $O(n^2 \log p \log U)$ time on an EREW PRAM with $p = \lceil m/n \rceil$ processors.

A problem closely related to the computation of a maximum flow is that of computing a blocking flow. As demonstrated by Dinic (1970), a blocking flow in an n-vertex acyclic network can be computed in $O(n^2)$ time. A parallel algorithm for this problem that runs in $O(n \log n)$ time using $O(n^2 \log n)$ operations on an EREW PRAM was described by Shiloach and Vishkin (1982, 1992). This algorithm and both parallel maximum-flow algorithms mentioned above can be seen to fall short, by a factor of $\Theta(\log n)$ (Goldberg and Shiloach/Vishkin) and between $\Theta(\log(m/n))$ and $\Theta(\log(m/n)\log U)$ (Ahuja/Orlin), of being implementations with optimal speedup of the corresponding sequential algorithms, i.e., of executing the same number of operations, up to a constant factor. We show how to reduce this logarithmic factor or to eliminate it altogether. While this factor may appear insignificant compared to the overall operation bound, it should be noted that the factor by which a parallel algorithm deviates from optimal speedup is precisely the minimum number of processors needed for a meaningful execution of the algorithm (if fewer processors are available, the sequential algorithm will be faster).

[1] Part of the research was carried out while this author was with the Max-Planck-Institut für Informatik.

[2] Supported by the ESPRIT Basic Research Actions Program of the EU under contract No. 7141 (project ALCOM II).

Ein einfacher effizienter paralleler Algorithmus für die Erkennung von Intervallgraphen

Elias Dahlhaus

Basser Department of Computer Science
Madsen Building F09
University of Sydney
NSW 2006
Australia
dahlhaus@cs.su.oz.au

Es wird ein paralleler Algorithmus zur Erkennung von Intervallgraphen vorgestellt. Die Laufzeit auf einer CRCW-PRAM ist logarithmisch und die Prozessorenzahl ist linear. Dieser Algorithmus benutzt keine P-Q-Bäume und keine Eliminationsprozeduren als Unterprozeduren. Man kann mit denselben Methoden auch Hypergraphen oder Mengensysteme intervallartig anordnen, falls dies möglich ist, d.h. die Grundmenge oder Knotenmenge kann derart angeordnet werden, daß die Knoten jeder Hyperkante bzw. die Elemente jeder Menge des Mengensystems aufeinanderfolgend auftreten.

Ein Plädoyer für den Eigentümer

Henning Fernau *Klaus-Jörn Lange* *Klaus Reinhardt*

Wilhelm-Schickard-Institut für Informatik
Universität Tübingen
Sand 13
D-72076 Tbingen,Germany
{fernau|lange|reinhard}@informatik.uni-tuebingen.de

Dymond & Ruzzo (ICALP'86) sowie Monien, Schapers & Rytter (TCS) entwickeln schwierige Algorithmen zum Nachweis von *DAuxpdaTimeSpace*(Pol,log) = *CROW*(log,Pol). Wir stellen einen einfacheren Algorithmus zum Nachweis einer verallgemeinerten Beziehung auf. Dazu betrachten wir CROW PRAM's mit einer Registerwortlänge von $O(\log(g))$ bei g Prozessoren.

Satz: *DAuxpdaTimeSpace*$(f^{O(1)}, \log(g))$ =*CROW*$(O(log(f)), g^{O(1)})$.

Schwierig ist die Richtung $\subseteq$. Der augmentierte Kellerautomat kann nur $g^{O(1)}$ Oberflächenkonfigurationen annehmen. Daher kann auch die Kellerhöhe maximal $g^{O(1)}$ werden (Sonst existiert eine ∞-Schleife beim Einkellern). Daher kann auch die Menge K der erweiterten Oberflächenkonfigurationen, die außer dem Zustand, dem Inhalt des Arbeitsbandes und dem obersten Kellerzeichen auch die gegenwärtige Kellerhöhe enthält, nur $g^{O(1)}$ Elemente besitzen.

CROW-PRAM-Algorithmus:
for each $x, y \in K$ **do in parallel**
  begin
    $N_0(x) :=$ Nachfolger von x, falls definiert, sonst x
    $G_0(x) := N_0(x)$, falls gleiche Kellerhöhe wie x, sonst x
    $D_0(x, y) :=$ Nachfolger von y, falls y das Zeichen auskellert, das x einkellerte, sonst y
  end
for $k := 0$ **to** $\log(n)$ **do**
  for each $x, y \in K$ **do in parallel**
    begin
      $N_{k+1}(x) := N_k(D_k(x, N_k(x)))$
      if $D_k(x, N_k(G_k(x))) \equiv_h x$
        then $G_{k+1}(x) := G_k(D_k(x, N_k(G_k(x))))$
        else $G_{k+1}(x) := G_k(x)$
      if $y \leq_h D_k(x, N_k(x))$
        then $D'_k(x, y) := D_k(x, y)$
        else $D'_k(x, y) := D_k(D_k(x, N_k(x)), y)$
      $D_{k+1} := D'_k(x, G_{k+1}(D'_k(x, G_{k+1}(D'_k(x, y)))))$
    end
end

Dabei: $\equiv_h, \leq_h$ Kellerhöhenvergleiche, x, y Oberflächenkonfigurationen (einschl. Kellerhöhe).

Folgerungen (beispielsweise):
(1) *DAuxpdaTimeSpace*(Pol,log) = *CROW*(log,Pol) = LOGDCFL
(2) *DAuxpdaTimeSpace*(Pol,Polylog) = *CROW*(log,c^{Polylog}) = SC

Ein allgemeiner Optimierungsoperator

Harald Hempel

Friedrich-Schiller-Universität Jena
07743 Jena
hempel@pdec01.mipool.uni-jena.de

In seiner Arbeit "The Complexity of Optimization Problems" (1986) definiert Krentel die Klassen max-P und min-P. Seine Definition ist stark an die Arbeitsweise einer nichtdeterministischen Turingmaschine angelehnt. Wir definieren einen allgemeinen Maximierungsoperator für Komplexitätsklassen.

Sei K eine Komplexitätsklasse, dann ist

$$f \in max \cdot K \iff \bigvee_{A \in K} \bigvee_{Pol\ p} \bigwedge_{x \in \Sigma^*} f(x) = max\{y : 0 \leq y \leq 2^{p(|x|)} \wedge \langle x, y \rangle \in A\}$$

und falls diese Menge leer ist, sei $f(x) = 0$.

Man überzeugt sich leicht, daß $max \cdot \mathrm{NP} =$ max-P. Für die Klassen $max \cdot \mathrm{P}$ und $max \cdot \mathrm{coNP}$ gibt es keine Übereinstimmung mit bereits bekannten Funktionenklassen. Was läßt sich nun zumindest bezüglich der Inklusionsbeziehungen zu den bereits wohl untersuchten Funktionenklassen $max \cdot \mathrm{NP}$, $\#\mathrm{P}$, span-P, $\#\mathrm{coNP}$ über die "neuen" Klassen $max \cdot \mathrm{P}$, $min \cdot \mathrm{P}$, $max \cdot \mathrm{coNP}$ und $min \cdot \mathrm{coNP}$ aussagen?

Wir untersuchen die Inklusionsstruktur dieser Klassen, geben Charakterisierungen an und weisen nach, daß sie unter allgemein akzeptierten komplexitäts-theoretischen Annahmen mit keiner der bekannten Funktionenklassen zusammenfallen.

Nebenbei ergeben sich eine Vielzahl von Charakterisierungen zentraler Problemklassen der Komplexitätstheorie.

On the Complements of Multivalued Functions

Steve Fenner

Dept. of Computer Science
University of Southern Maine
96 Falmouth st.
Portland, ME 04103, U.S.A.
`fenner@usm.maine.edu`

Fred Green

Dept. of Mathematics and Computer Science
Clark University
Worcester, MA 01610, U.S.A.
`fgreen@black.clarku.edu`

Steve Homer

Dept. of Computer Science
Boston University
Boston, MA 01003, U.S.A.
`homer@cs.bu.edu`

Alan Selman

Dept. of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260-2000, U.S.A.
`selman@cs.buffalo.edu`

Thomas Thierauf

Abt. Theoretische Informatik
Universität Ulm
Oberer Eselsberg
D-89069 Ulm
`thierauf@informatik.uni-ulm.de`

Heribert Vollmer

Institut für Informatik
Universität Würzburg
Am Exerzierplatz 3
D-97072 Würzburg
`vollmer@informatik.uni-wuerzburg.de`

NPMV is the class of (multivalued) functions that are computed by nondeterministic polynomial-time Turing transducers. NPMV is precisely the class of functions that computes witnesses for NP sets in the following sense. For any NP set L there exist a set $A \in$ P and a polynomial p such that for all x, we have

$$x \in L \iff \exists y \in \Sigma^{p(|x|)} :\ (x, y) \in A.$$

A y such that $(x, y) \in A$ is called a *witness for* x (with respect to A). The function that outputs all witnesses for x is a NPMV function. On the other hand, any NPMV function defines an NP set.

We extend the notion of a witness to Σ_2^p. For any Σ_2^p set L there exist a set $B \in$ coNP and a polynomial p such that for all x, we have

$$x \in L \iff \exists y \in \Sigma^{p(|x|)} :\ (x, y) \in B.$$

A y such that $(x, y) \in B$ is called a *witness for* x (with respect to B). What function class captures the computation of witnesses for Σ_2^p sets?

Since Σ_2^p = NP$^{\mathrm{NP}}$, one might expect that the answer is NPMV$^{\mathrm{NP}}$. However, we will show that NPMV$^{\mathrm{NP}}$ is too powerful for this task, unless the Polynomial Hierarchy collapses.

We define *coNPMV* as the class of (multivalued) functions that can be expressed as the complements of NPMV functions, $\Sigma^{p(|x|)} - f(x)$, for an NPMV function f and a polynomial p. We show that coNPMV is precisely the class of functions that computes witnesses for Σ_2^p sets.

Having motivated to consider coNPMV, we study further properties of this class.

Über p-selektive E-bi-immune Mengen

Hans-Jörg Burtschick *Arfst Nickelsen*

Fachbereich 13 Informatik
TU Berlin
Sekr.: FR 6-2, Franklinstr. 28/29
D-10587 Berlin
{hjoerg|nicke}@cs.tu-berlin.de

Wolfgang Lindner

Abt. Theoretische Informatik
Universität Ulm
Oberer Eselsberg
D-89069 Ulm
lindner@informatik.uni-ulm.de

In der Rekursionstheorie wurden verschiedene Begriffe entwickelt, die ausdrücken, ob eine Menge strukturell schwierig oder strukturell einfach ist (unabhängig davon, ob sie überhaupt entscheidbar ist).

So ist eine *bi-immune* Menge A in gewisser Weise strukturell schwierig, da weder A noch das Komplement $\overline{A}$ eine unendliche entscheidbare Teilmenge hat.

Eine *semirekursive* Menge B ist in gewisser Weise strukturell einfach, da es eine berechenbare Selektorfunktion f gibt, die für zwei Eingabewörter dasjenige ausgibt, welches "wahrscheinlicher" in B enthalten ist. Für alle x und y gilt also: $f(x,y) \in \{x,y\}$ und $(x \in B \vee y \in B) \Rightarrow f(x,y) \in B$.

Diese Begriffe schließen sich nicht gegenseitig aus, denn Jokusch konnte 1967 zeigen, daß es semirekursive bi-immune Mengen gibt.

In dem Vortrag untersuchen wir die entsprechende komplexitätstheoretische Fragestellung:

Eine Menge ist *p-selektiv*, wenn sie semirekursiv ist und die Selektorfunktion f polynomiell zeitbeschränkt berechenbar ist. Eine Menge ist *$\mathcal{C}$-bi-immun*, wenn weder sie noch ihr Komplement eine unendliche Teilmenge aus der Klasse $\mathcal{C}$ besitzt.

- Der Beweis von Jokusch kann für den komplexitätstheoretischen Fall übertragen werden, d. h. es existieren p-selektive P-bi-immune Mengen.
- Durch die Kombination dieser beiden Begriffe ist die Selektorfunktion (als Ordnung betrachtet) in gewisser Weise festgelegt. Daraus lassen sich einige Eigenschaften von p-selektiven E-bi-immunen Mengen ableiten.

Außerdem interessiert uns die Frage, in welcher Komplexitätsklasse sich p-selektive P-bi-immune Mengen finden lassen (obere und untere Schranken).

Teilklassen von Binärem NP

Thomas Schwentick

Institut für Informatik
Johannes Gutenberg-Universität Mainz
D-55099 Mainz
tick@informatik.mathematik.uni-mainz.de

Fagin (74) zeigte, daß NP gerade diejenigen Mengen endlicher Strukturen umfaßt, die durch eine existentielle Second-Order Formel (i.e. eine Formel vom Typ $\exists X_1, \ldots, X_l \varphi$ mit Relationenvariablen X_i und einer first-order Formel φ) charakterisiert werden können. Um NP$\neq$co-NP zu nachzuweisen, würde es also genügen, zu zeigen, daß existentielle und universelle Second-Order Logik auf endlichen Strukturen von unterschiedlicher Ausdrucksstärke sind.

Monadic NP sei die Klasse aller Mengen endlicher Strukturen, die durch existentielle second-order Formeln charakterisiert werden, in denen nur unäre Relationen (i.e. Mengen) quantifiziert werden. Fagin (75) zeigte bereits, daß Graph-Zusammenhang nicht in Monadic NP und damit diese Klasse nicht komplementabgeschlossen ist.

Binary NP sei analog zu Monadic NP definiert mit Quantifizierung von höchstens binären Relationen. Ajtai (83) zeigte, daß Binary NP (und alle anderen k-ären NPs) nicht komplementabgeschlossen ist. Dazu benutzte er allerdings Strukturen mit 3-ären Relationen. Zu zeigen, daß eine NP-Menge von Graphen nicht in Binary NP ist, ist bisher nicht gelungen. Deshalb liegt es nahe, Teilklassen von Binary NP zu untersuchen.

Durand, Lautemann, Schwentick (95) untersuchten solche Teilklassen, die durch semantische Einschränkung der zu quantifizierenden Relationen entstehen. Es stellte sich heraus, daß alle untersuchten Teilklassen in eine 5-stufige Hierarchie eingeordnet werden können:

1. Binary NP, partielle Ordnungen
2. Unäre Funktionen, lineare Ordnungen, Äquivalenzrelationen
3. Nachfolgerrelationen
4. Bijektionen
5. Monadic NP, Mengen.

Es wurde gezeigt, daß die Inklusionen zwischen 1. und 2. und zwischen 4. und 5. echt sind. Zwischen 2., 3. und 4. blieb die Echtheit offen.

Wir zeigen hier, daß 3. und 4. zusammenfallen, während die Inklusion zwischen 2. und 4. (3.) echt ist.

Für das erste Resultat zeigen wir, daß Graph-Zusammenhang in 4. liegt. Das zweite Resultat zeigt, daß die Eigenschaft "G hat genauso viele Knoten wie Kanten" nicht in 4. ausdrückbar ist. Der Beweis verbindet ein Ergebnis von Ajtai (83) mit Techniken zum Nachweis von Gewinnstrategien in Ehrenfeucht-Spielen.

Charakterisierungen von Komplexitätsklassen durch Blattsprachen und endliche Modelltheorie

Hans-Jörg Burtschick
Fachbereich 13 Informatik
TU Berlin
Sekr.: FR 6-2, Franklinstr. 28/29
D-10587 Berlin
hjoerg@cs.tu-berlin.de

Heribert Vollmer
Institut für Informatik
Universität Würzburg
Am Exerzierplatz 3
D-97072 Würzburg
vollmer@informatik.uni-wuerzburg.de

In der Komplexitätstheorie versucht man unter anderem, bekannte Klassen mit unterschiedlichen Formalismen zu charakterisieren, um so zu strukturellen Aussagen über die untersuchten Klassen zu gelangen.

Zum Beispiel ermöglicht die Charakterisierung von Komplexitätsklassen mittels sogenannter Blattsprachen eine Reihe von bemerkenswerten Normalformen für Klassen wie PP, #P, PSPACE, etc.

Ebenso gibt es für viele Komplexitätsklassen jeweils logische Charakterisierungen. Dabei korrespondiert die Resourcenbeschränkung für das Akzeptieren einer Sprache mit der Komplexität der Formel, die diese Sprache (als Klasse endlicher Modelle aufgefaßt) definiert.

In diesem Vortrag kombinieren wir beide Beschreibungsformalismen:

- Es ist bekannt, daß Blattsprachen aus AC^0 die Sprachen der polynomiellen Hierarchie charakterisieren. Außerdem lassen sich die Sprachen aus AC^0 durch first-order Formeln und die Sprachen aus der polynomiellen Hierarchie durch second-order Formeln definieren.

  Wir zeigen Charakterisierungen durch Blattsprachen auch für die einzelnen Stufen der polynomiellen Hierarchie. Die Klasse der Sprachen, die (als Modellklasse betrachtet) mittels Σ_k-*second-order* Formeln definierbar sind, wird durch Blattsprachen charakterisiert, die durch Σ_k-*first-order* Formeln definiert werden können.

- Ist die Blattsprache ein generalisiertes Spektrum im Sinne von Fagin, so ist die durch sie charakterisierte Komplexitätsklasse ein (klassisches) Spektrum.

Circuit Kolmogorov Complexity

Bernd Borchert

Universität Heidelberg
Im Neuenheimer Feld 294
D-69120 Heidelberg
bb@math.uni-heidelberg.de

Let a Boolean circuit $c(x_1, \ldots, x_n)$ with n occuring variables $x_1, \ldots, x_n$ be given. The circuit $c = c(x_1, \ldots, x_n)$ describes in a natural way a word $\mathbf{w}(c)$ of length 2^n: its ith letter is the value of the ith assignment, where assignments are ordered lexicographically (it is assumed that there is a linear order on variables). In other words, $\mathbf{w}(c)$ is the result column of the truth table representation of c.

Examples: For $c = c(x_1, x_2) = x_1 \wedge x_2$ the word $\mathbf{w}(c)$ equals 0001, for $d = d(x_1, x_2, x_3)$ $= (x_2 \wedge \neg x_3) \vee \neg x_1$ the word $\mathbf{w}(d)$ equals 11110010.

In order to let circuits also describe words with a length not a power of 2, take any pair (c, m) of a circuit $c = c(x_1, \ldots, x_n)$ and a natural number m. Let the *word described by* c *and* m, formally $\mathbf{w}(c, m)$, be the length-m prefix of $\mathbf{w}(c)$. Note that this implies $\mathbf{w}(c) = \mathbf{w}(c, 2^n)$.

Examples: For the circuit $c = c(x_1, x_2) = x_1 \wedge x_2$ from above, $\mathbf{w}(c, 0) = \epsilon$, $\mathbf{w}(c, 1) = 0$, $\mathbf{w}(c, 2) = 00$, $\mathbf{w}(c, 3) = 000$, and $\mathbf{w}(c, m) = 0001$ for any $m \geq 4$.

The notion $\mathbf{w}(c, m)$ gives another kind of resource-bounded Kolmogorov complexity: For a word x let its *circuit Kolmogorov complexity* be the length of the shortest word $\langle c, m \rangle$ such that $\mathbf{w}(c, m) = x$, where $\langle \ldots \rangle$ is a usual pairing function, the circuit c is encoded in a usual way, and the number m is encoded in binary.

Several notions from Kolmogorov Complexity Theory can be transfered to this modified concept. For example, a *circuit random word* (or *circuit random string*) is a word whose circuit Kolmogorov complexity is not smaller than its length. The well-known Shannon counting argument shows that most words are circuit random. A word which is not circuit random is called *circuit compressible*. A trivial example of a circuit compressible word is the all-0-word $0 \ldots 0$ of length n (for $n \geq 20$ or so) which equals $\mathbf{w}(c_1 \wedge \neg c_1 \wedge \ldots \wedge c_m \wedge \neg c_m, n)$ where $m \geq \log_2(n)$. It should be clear that by the circuit description words can be compressed at most logarithmically.

This talk will compare circuit Kolmogorov complexity with the other known notions of resource-bounded Kolmogorov complexity.

Die Komplexität von Zufallsstrings

Martin Kummer

Institut für Logik
Komplexität und Deduktionssysteme
Universität Karlsruhe
76128 Karlsruhe
kummer@ira.uka.de

Ein String x der Länge n heißt *zufällig*, im Sinne der Kolmogorov-Komplexität, wenn jedes Programm, das x ausdruckt, mindestens die Länge n hat. Zufällige Strings spielen in fast allen Anwendungen der Kolmogorov-Komplexität eine wichtige Rolle.

Sei R die Menge aller zufälligen Strings. Ein klassisches Resultat besagt, daß R Turing-vollständig ist, d.h. es gibt einen Algorithmus, der das Halteproblem K mit Hilfe von Anfragen an R löst. Jedoch ist die Laufzeit dieses Algorithmus durch keine berechenbare Funktion beschränkt. Wir geben einen neuen Algorithmus an, der diese Schwäche vermeidet: für jede Eingabe x berechnet er eine endliche Menge S_x, so daß $x \in K$ gdw. S_x keinen zufälligen String enthält. Damit folgt insbesondere, daß R bereits Truth-table-vollständig ist.

Der Beweis benutzt die Tatsache, daß die Kolmogorov-Komplexität bezüglich sogenannter linear-optimaler Gödelnumerierungen von Programmen definiert ist. Dies ist tatsächlich notwendig, denn wir konstruieren eine polynomial-optimale Gödelnumerierung, bezüglich der die Menge der zufälligen Strings *nicht* Truth-table-vollständig ist (sie ist aber immer noch Turing-vollständig).

Learning Environments leading to Inefficient Learnability

Werner Stein

Center for **L**earning **S**ystems and **A**pplications
Algorithmic **L**earning Group
University Kaiserslautern
stein@informatik.uni-kl.de http://pfah.informatik.uni-kl.de

In the pioneering papers of Solomonoff (1964) and of Gold (1965) the idea of what we call learning (*identifying in the limit*) is born. Given the graph $f(0), f(1), \ldots$ of an unknown object f, an algorithm S is identifying f in the limit if S produces a sequence of guesses and after some finite point in the sequence all the guesses are correct and the same.

In the following years research has been concentrated in finding (natural) learning conditions and showing bounds of learnability. The findings of these studies gave us a first view inside the problem of hard learnability, since one could show that some problems can be learned under one condition, but not under another. The question was "*can* something be learned or not". Only few scientists investigate the question "can something be learned *fast*". Additionally most of the few results concerning fast learnability are restricted to special learning environments. For example: 10 years after Solomonoff's introducing paper, Barzdin (1974) showed that the claim for guesses being consistent with the graph seen so far is a condition restricting the power of learning in the limit. Lange and Wiehagen (1991) proved the existence of a problem that can be learned fast but not in a consistent way, as long as P≠NP. They also provide a general technology how to learn fast in "almost all" (other then consistent) learning classes. But still 1994 Wiehagen and Zeugmann asked for conditions yielding that a given learning problem has no fast consistent solution, but a fast inconsistent.

Therefore an answer for this question gives us a first reason why some problems only can be learned inefficiently. In addition to our last talk (1994) we present a characterization of *consistent learnability with polynomial update times*.

We also asked for conditions other then consistency having non-polynomial worst case bounds. Now we answer this question, showing that learning with a polynomial bounded point of convergence in Gödelnumberings is such a condition. The importants of this result is emphasized, giving a hypotheses boosting algorithm such that polynomial learning is possible in "most" learning classes. A characterization of *learnability with a polynomial bounded point of convergence in Gödelnumberings* is still an open problem.

BARZDIN, J. (1974), Inductive inference of automata, functions and programs, *in* "Proceedings International Congress of Mathematicians", pp. 455 - 460.

LANGE, S. AND WIEHAGEN, R. (1991), Polynomial–time inference of arbitrary pattern languages, *New Generation Computing*, 8:361 - 370.

SOLOMONOFF, R. (1964), A formal theory of inductive inference, *Information and Control*, 7:1 - 22, 234 - 254.

STEIN, W. (1994), Exploring the Sources of Computational Ineffiency in Inductive Inference, *23. Workshop über Komplexitätstheorie, Datenstrukturen und effiziente Algorithmen*, Wanka, R., Hrsg., Bericht: tr-ri-94-147.

WIEHAGEN, R., AND ZEUGMANN, T. (1994), Ignoring data may be the only way to learn efficiently, *Journal of Experimental and Theoretical Artificial Intelligence* **6**, 131 - 144.