Universität Stuttgart
Fakultät Informatik

Institut für Informatik
Breitwiesenstraße 20-22
D-70565 Stuttgart

32. Workshop über

Komplexitätstheorie, Datenstrukturen und effiziente Algorithmen

Ulrich Hertrampf, Andreas Bergen
(Herausgeber)

Report Nr. 1997/04

17. Juni 1997

Vorwort

Der 32. Workshop über Komplexitätstheorie, Datenstrukturen und effiziente Algorithmen am 17. Juni 1997 ist der erste dieser Reihe, der an der Universität Stuttgart durchgeführt wird. Diese nun schon seit über zehn Jahren regelmäßig dreimal im Jahr stattfindende gemeinsame Veranstaltung der GI-Fachgruppen 0.1.3 (Parallele und verteilte Algorithmen) und 0.1.4 (Komplexität) fand auch in diesem Sommer wieder reges Interesse, so daß wir mit 13 Vorträgen ein volles und über die schon im Titel dokumentierte Vielfalt gestreutes Programm anbieten können.

Ich möchte an dieser Stelle allen Kollegen der Abteilung Theoretische Informatik für ihre tatkräftige Unterstützung danken, ganz besonders Andreas Bergen, der sich um die gesamte Abwicklung des Theorietags gekümmert hat.

Am 28. Mai 1997 verstarb Ronald V. Book im Alter von 60 Jahren. Wir werden ihn als hervorragenden Informatiker, aber auch als angenehmen Kollegen und guten Freund in Erinnerung behalten. Diesen Workshop wollen wir seinem Andenken widmen.

Stuttgart, im Juni 1997 Ulrich Hertrampf

Programm

(Stand 10. Juni 1997)

9:55	Begrüßung
10:00	**Andreas Birkendorf, Andreas Böker, Hans Ulrich Simon** Lernen aus kleinsten Gegenbeispielen
10:25	**Manindra Agrawal, Thomas Thierauf** A Note on Satisfiability Problems
10:50	Kaffeepause
11:15	**Sven Kosub, Heinz Schmitz, Heribert Vollmer** Uniformly Defining Complexity Classes of Functions
11:40	**Paul Fischer, Norbert Klasner, Ingo Wegener** Über den kritischen Punkt bei kombinatorischen Gruppentests
12:05	Mittagspause
13:25	**Carsten Damm** On Spectral Lower Bound Arguments for Decision Trees
13:50	**Stefan Edelkamp** Multi Suffix Tree Dictionary in Optimal Space
14:15	**Jörg Rothe** Two Natural Θ_2^p-Complete Problems
14:40	Kaffeepause
15:05	**Klaus Reinhardt, Eric Allender** NL/poly = UL/poly
15:30	**Jan Behrens, Stephan Waack** Äquivalenztest und Transformation der Ordnung für Parity-OBDDs bezüglich verschiedener Ordnungen
15:55	**Thomas Schwentick** Padding and the Expressive Power of Existential Second-Order Logics
16:20	Kaffeepause
16:45	**Jochen Meßner, Jacobo Torán** Optimale aussagenlogische Beweissysteme und vollständige Sprachen
17:10	**Bernd Borchert, Dietrich Kuske, Frank Stephan** On Existentially First-Order Definable Languages and their Relation to NP
17:35	**Heribert Vollmer, Klaus W. Wagner** Über mögliche logspace-Varianten des IP- und des PCP-Theorems
18:00	Ende

Lernen aus kleinsten Gegenbeispielen

Andreas Birkendorf Andreas Böker Hans Ulrich Simon
Lehrstuhl Informatik II
Universität Dortmund, 44221 Dortmund
{birkendo,simon}@Ls2.informatik.uni-dortmund.de

Es ist bekannt, daß sich deterministische endliche Automaten (DFA's) *nicht* ausschließlich aus (beliebigen) Gegenbeispielen erlernen lassen. In unserem Vortrag wollen wir jedoch einen Algorithmus vorstellen, der DFA's aus *kleinsten* Gegenbeispielen lernt.

Dazu betrachten wir die *kanonische Ordnung* auf Worten $w \in \{0,1\}^*$ und sagen, ein Wort w_1 ist *kanonisch kleiner* als ein Wort w_2, wenn es kürzer ist, oder es ist gleich lang, aber an der ersten unterschiedlichen Position steht bei w_1 eine 0 (und bei w_2 eine 1).

Unser Algorithmus lernt einen unbekannten Ziel-DFA Z mit n Zuständen, indem er einem Orakel Hypothesen-DFA's H vorschlägt. Sind H und Z nicht äquivalent, erhält der Algorithmus als Antwort das kanonisch kleinste Wort aus der symmetrischen Differenz der Sprachen $L(H)$ und $L(Z)$.

Wir zeigen, daß unser Algorithmus nach $O(n^2)$ vielen kleinsten Gegenbeispielen Z erlernt hat.

Eine geschickte Modifikation des Algorithmus ermöglicht es, DFA's für längenkonstante Sprachen (d. h., alle akzeptierten Worte haben dieselbe Länge) aus nur $O(n \log n)$ vielen kleinsten Gegenbeispielen zu erlernen.

Unsere Ergebnisse verbessern eine Arbeit von Ibarra und Jiang, deren Algorithmen $O(n^3)$ bzw. $O(n^2)$ kleinste Gegenbeispiele benötigten.

Zusätzlich ist unser Algorithmus für DFA's längenkonstanter Sprachen optimal, da Betrachtungen der Vapnik-Chervonenkis-Dimension der Klasse der DFA's mit n Zuständen bereits $O(n \log n)$ Gegenbeispiele als untere Schranke liefern.

A Note on Satisfiability Problems

Manindra Agrawal [*]
Dept. of Computer Science
Indian Institute of Technology
Kanpur 208016, India
manindra@iitk.ernet.in

Thomas Thierauf
Abt. Theoretische Informatik
Universität Ulm
89069 Ulm, Germany
thierauf@informatik.uni-ulm.de

We investigate the complexity of certain satisfiability problems. CNF-SAT, which we also denote by $\wedge$-$\vee$-SAT, asks for an assignment for a given formula that satisfies *all* of its clauses. While $\wedge$-$\vee$-SAT is NP-complete, we ask:

- what is the complexity of $\oplus$-$\vee$-SAT, where it is asked for an assignment that satisfies an *odd* number of clauses?

We show that $\oplus$-$\vee$-SAT can be solved by a randomized algorithm in polynomial time. The result is extended to $\oplus$-Th-SAT, where we have a threshold instead of an or at the input level.

[*]Supported in part by an Alexander von Humboldt fellowship.

Uniformly Defining Complexity Classes of Functions

Sven Kosub Heinz Schmitz Heribert Vollmer

Theoretische Informatik
Universität Würzburg
Am Exerzierplatz 3
97072 Würzburg

{kosub, schmitz, vollmer}@informatik.uni-wuerzburg.de

We introduce a general framework for the definition of function classes. Our model, which is based on polynomial time nondeterministic Turing transducers, allows uniform characterizations of deterministic classes (FP), counting classes ($\#\cdot$P, $\#\cdot$NP, $\#\cdot$coNP, GapP, GapNP), optimization classes (max$\cdot$P, min$\cdot$P, max$\cdot$NP, min$\cdot$NP), promise classes ($\#_{\mathrm{few}}\cdot$P, c$\#\cdot$P), and many more. Each such class is defined in our model by a certain family of functions. We introduce a reducibility notion between such families, which allows us to develop a necessary and sufficient criterion for relativizable inclusion between function classes. As it turns out, our criterion is very easily applicable and we get as a consequence e.g. that relative to some oracle, min$\cdot$P $\not\subseteq$ $\#\cdot$NP.

Über den kritischen Punkt bei kombinatorischen Gruppentests

Paul Fischer   Norbert Klasner   Ingo Wegener

FB Informatik, LS II
Universität Dortmund
D-44221 Dortmund

Das (k,n)-Problem ist das Problem, die Disjunktion von k wesentlichen Booleschen Variablen aus n Variablen exakt zu lernen. Eine Frage ist eine Teilmenge S der Variablen, und die Antwort ist die Information, ob S mindestens eine wesentliche Variable enthält.

Eine triviale Lernstrategie besteht darin, nur einelementige Mengen zu fragen. Für diese Strategie sind $n-1$ Fragen notwendig und hinreichend. Wir suchen nun Parameter k, so daß weniger als $n-1$ Fragen ausreichen. Wir stellen dazu einen Algorithmus vor, der für $k < n/3$ mit weniger als $n-1$ Fragen auskommt. Dieser Algorithmus benutzt dabei nur Mengen S der Größe 1 und 2. Im Gegensatz dazu zeigen wir, daß jeder Lernalgorithmus, der nur Fragen der Größe 1 und 2 stellt, für $k \geq n/3$ mindestens $n-1$ Fragen benötigt. In diesem Fall ist also die obige triviale Lernstrategie optimal. Für Fragen der Größe 1 und 2 ist also der kritische Punkt $n/3$ und für beliebig große Fragen ist er mindestens $n/3$. Im Gegensatz dazu ist der kritische Punkt bei statistischen Gruppentests $(3-\sqrt{5})n/2$.

On Spectral Lower Bound Arguments for Decision Trees

Carsten Damm

Universität Trier
Fachbereich IV — Informatik
D-54286 Trier

damm@informatik.uni-trier.de

A decision tree is a rooted binary tree with inner nodes labeled by Boolean variables and edges and leaves labeled by Boolean constants. A decision tree T defines a Boolean function f_T in a natural way: given an input a, start at the root and follow at each inner node the edge according to the value of the variable that labels this node. The label of the terminal node is $f_T(a)$. The *size* of a decision tree is its number of leaves, the *average depth* is the expectation of the depth of the leaf that is reached by a randomly chosen input. Denote by $\mathsf{DT}(f)$ and $\overline{\mathsf{depth}}(f)$ the minimal size and the minimal average depth, respectively, of decision trees that compute a given Boolean function $f : \{0,1\}^n \to \{+1,-1\}$.

We give an new proof for the following lower bounds due to Brandman, Orlitsky, and Hennessy ([1]):

$$\mathsf{DT}(f) \geq 2^n \sum_{S \subseteq [n]} 2^{|S|} \cdot \hat{f}(S)^2, \overline{\mathsf{depth}}(f) \geq \sum_{S \subseteq [n]} |S| \cdot \hat{f}(S)^2.$$

Here the $\hat{f}(S)$ are defined by the decomposition $f = \sum_{S \subseteq [n]} \hat{f}(S) \chi_S$, where χ_S is the ± 1-valued parity of the variables $x_i, i \in S$.

Our argument is extremely easy and it is easily extensible to spectral decompositions with respect to bases different from the standard Fourier basis $\chi_S, S \subseteq [n]$. In case f is symmetric, for instance, it is natural to include symmetric functions as basis functions. Representing symmetric functions as polynomials in a single real variable, we can take orthogonal polynomials as basis functions. An example are the KRAWTCHOUK-polynomials, that have many applications in the theory of error correcting codes (cf. [2]). This leads to modified lower bound formulas involving only a linear number of spectral coefficients.

Literatur

[1] Y. Brandman, A. Orlitsky, and J. Hennessy. A spectral lower bound technique for the size of decision trees and two-level AND/OR circuits. *IEEE Transactions on Computers*, 39:282–287, 1990.

[2] J. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes.* Amsterdam: North-Holland Publishing Co., 1977.

Multi Suffix Tree Dictionary in Optimal Space

Stefan Edelkamp

Institut für Informatik
Albert-Ludwigs-Universität
Am Flughafen 17, D-79110 Freiburg

edelkamp@informatik.uni-freiburg.de

The talk is about a dictionary data structure D for matching multiple pattern. If the input alphabet Σ is bounded and the patterns m in D are considered to be mutually substring free an optimal worst case bound of $O(|m|)$ for both insertion and deletion of a pattern $m \in \Sigma^*$ is achieved. Since the strings in D are not substrings of each other this is a special case of the *dynamic dictionary matching problem* (*DDMP*).

The structure D is realised as a multi suffix tree. Let M be the set of strings stored in the dictionary D resulting from an arbitrary sequence of *Insert* and *Delete* operations. The main result is that the space complexity of the augmented multi suffix tree representing D is $O(d)$ with $d = \sum_{m \in M} |m|$. This bound is optimal if we assume that each pattern in the dictionary uses at least linear space and has not been achieved before.

Further on, I present a new incremental *Search* operation to find one substring of x in time $O(|x|)$. In the general case of the *DDMP* I will show that searching all *tocc* (total number of occurences) substrings in x can be performed in almost $O((|x| + tocc) \log d)$ time without destroying the linear efficiency results of *Insert* and *Delete*.

Last but not least, I will talk about the use of the multi suffix tree dictionary in state space search of huge problem spaces that do not fit in the memory. Thus we may assume that the problem space is described implicitely by a bounded number of transitions and that transition sequences can be regarded as strings. The set of strings M can be used as a set of forbidden words, so-called duplicates, in the search. Duplicates are transition sequences that are more expensive than their counterparts. When a substring of the generated path according to a node in the search tree is encountered the tree is pruned since a shorter path has been or will be examined.

Literatur

[1] A. V. Aho and M. J. Corasick. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340, 1975.

[2] A. Amir, M. Farach, R. M. Idury, J. A. L. Poutré, and A. Schäffer. Improved dynamic dictionary matching. *Information and Computation*, 119(2):258–282, 1995.

[3] E. M. McCreight. A space-economical suffix tree construction algorithm. *Journal of the ACM*, 23(2):262–272, 1976.

[4] L. A. Taylor and R. E. Korf. Pruning duplicate nodes in depth-first search. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 756–761. AAAI Press, 1993.

Two Natural Θ_2^p-Complete Problems

Jörg Rothe

Institut für Informatik
Friedrich-Schiller-Universität Jena
07743 Jena, Germany

rothe@informatik.uni-jena.de

Two natural problems for which previously only NP-hardness or coNP-hardness lower bounds were known are shown to be complete for Θ_2^p (a.k.a. $\mathrm{P}_{||}^{\mathrm{NP}}$ or $\mathrm{P}^{\mathrm{NP[log]}}$), the class of sets solvable via parallel access to NP.

In 1876, Lewis Carroll proposed an election system in which the winner is the candidate who with the fewest changes in voters' preferences becomes a Condorcet winner—a candidate who beats all other candidates in pairwise majority-rule elections. Bartholdi, Tovey, and Trick [BTT89] provided a lower bound—NP-hardness—on the computational complexity of determining the election winner in Carroll's system. We optimally improve their lower bound: Determining the winner in Carroll's system is Θ_2^p-complete. It is by far the most natural complete problem for this class. It follows from our result that determining the winner in Carroll's elections is not NP-complete unless the polynomial hierarchy collapses. This is joint work with Edith and Lane A. Hemaspaandra ([HHR97a], see also the survey [HHR97b]).

Bodlaender, Thilikos, and Yamazaki [BTY97] investigate the computational complexity of the problem of whether the Minimum Degree Greedy Algorithm can approximate the size of a maximum independent set of the input graph within a constant factor of r, for fixed rational $r \geq 1$. They denote this problem by $\mathcal{S}_r$ and prove that for each rational $r \geq 1$, $\mathcal{S}_r$ is coNP-hard. They also provide a P^{NP} upper bound of $\mathcal{S}_r$, leaving open the question of whether this gap between the upper and the lower bound of $\mathcal{S}_r$ can be closed. For the special case of $r = 1$, they show that $\mathcal{S}_1$ even is DP-hard (where DP denotes the class of sets that can be represented as the difference of two NP sets), again leaving open the question of whether $\mathcal{S}_1$ can be shown to be complete for DP or some larger class such as P^{NP}. We completely solve all the questions left open by Bodlaender et al. in [BTY97]. Our main result is that for each rational $r \geq 1$, $\mathcal{S}_r$ is complete for Θ_2^p. This is joint work with Edith Hemaspaandra [HR97].

Literatur

[BTT89] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[BTY97] H. L. Bodlaender, D. Thilikos, and K. Yamazaki. It is hard to know when greedy is good for finding independent sets. *Information Processing Letters*, 61:101–106, 1997.

[HHR97a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *ICALP'97*. To appear.

[HHR97b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Raising NP lower bounds to parallel NP lower bounds. *SIGACT News*, June 1997. To appear.

[HR97] E. Hemaspaandra and J. Rothe. Recognizing when greed can approximate maximum independent sets is complete for parallel access to NP. Technical Report Math/Inf/97/14, Institut für Informatik, Friedrich-Schiller-Universität–Jena, Jena, Germany, May 1997.

NL/poly = UL/poly

Klaus Reinhardt
Wilhelm-Schickard Institut für Informatik
Eberhard-Karls-Universität Tübingen
Sand 13
72076 Tübingen
reinhard@informatik.uni-tuebingen.de

Eric Allender
Department of Computer Science
Rutgers University
P.O. Box 1179, Piscataway
NJ 08855-1179,USA
allender@cs.rutgers.edu

Wir zeigen, daß bei nichtuniformer Komplexität, d.h. wenn für jede Eingabelänge ein Advice-string polynomieller Länge zur Verfügung steht, nichtdeterministische, logarithmisch platzbeschränkte Berechnungen eindeutig gemacht werden können, d.h. es kann nur einen akzeptierenden Pfad geben. Gleiches gilt auch, wenn zusätzlich (bei polynomieller Zeitbeschränkung) ein Keller zur Verfügung steht, d.h. LogCFL/poly = UAuxPDA($\log n, n^{O(1)}$)/poly.

In [Wig94, GW96] beobachtete Wigderson bereits Graphen, in welchen die kürzeste Distanz zwischen jedem Paar von Knoten eindeutig ist. Wir bezeichnen diese Graphen als *min-unique*. Durch die Technik des Isolations Lemmas aus [MVV87] kann mit Hilfe eines Advice strings ein Graph in eine Menge von Graphen umgewandelt werden, in der zumindest einer min-unique ist.

Eine Erweiterung der Zähltechnik aus [Imm88, Sze88], die wir die *Doppelzähltechnik* nennen, weil sie in der k-ten Stufe nicht nur die Anzahl Knoten mit Distanz $\leq k$, sondern auch die Summe dieser Distanzen zählt, ermöglicht es bereits gefundene Knoten auf eindeutige Weise wiederzufinden und somit auf einem eindeutigen Pfad zu entscheiden, ob ein Graph min-unique ist und in diesem Fall die Erreichbarkeit im Graph zu entscheiden.

Literatur

[GW96] A. Gál and A. Wigderson. Boolean vs. arithmetic complexity classes: randomized reductions. *Random Structures and Algorithms*, 9:99–111, 1996.

[Imm88] N. Immerman. Nondeterministic space is closed under complement. *SIAM Journal on Computing*, 17:935–938, 1988.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

[Sze88] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26:279–284, 1988.

[Wig94] A. Wigderson. NL/poly $\subseteq \bigoplus$L/poly. In *Proc. of the 9th IEEE Structure in Complexity Conference*, pages 59–62, 1994.

Äquivalenztest und Transformation der Ordnung für Parity-OBDDs bezüglich verschiedener Ordnungen

Jan Behrens   Stephan Waack

Institut für Numerische
und Angewandte Mathematik
Georg–August–Universität Göttingen
Lotzestr. 16–18, 37083 Göttingen

{jbehrens,waack}@math.uni-goettingen.de

Die von Bryant eingeführten OBDDs haben sich als Datenstruktur für Boolsche Funktionen und zur Schaltkreisverifikation bewährt. Alle „wichtigen“ Operationen (Reduktion, Synthese, ...) lassen sich mit OBDDs in Linearzeit durchführen. OBDDs haben auf Grund ihrer starren Struktur eine begrenzte Darstellungskraft, so daß sich viele Funktionen mit ihnen nicht darstellen lassen.

Eine alternative Datenstruktur bieten die Parity-OBDDs ($\oplus$-OBDDs). Bei ihnen wird die Beschränkung fallengelassen, daß jeder Knoten nur einen 0- und einen 1-Nachfolger haben darf. Das hat zur Folge, daß es zu einer Eingabe möglicherweise mehrere Pfade von der Quelle zu den Senken gibt. Als Funktionswert eines $\oplus$-OBDDs betrachten wir die Anzahl der Wege unter der Eingabe zur 1-Senke modulo 2. Auch für $\oplus$-OBDDs lassen sich die „wichtigen“ Operationen in polynomieller Zeit durchführen ([Waa97]). Durch die allgemeinere Struktur vergrößert sich der Zeitbedarf allerdings ins kubische.

Wie für die OBDDs ist auch für die $\oplus$-OBDDs die Variablenordnung von entscheidender Bedeutung. Betrachten wir nun verschiedene Variablenordnungen, so ergibt sich einerseits das Problem, zwei $\oplus$-OBDDs mit verschiedenen Variablenordnungen auf Äquivalenz zu testen. Andererseits tritt das Problem auf, aus einer gegebenen Darstellung eine Darstellung bezüglich einer anderen Ordnung zu berechnen. (Beide Probleme lassen sich für OBDDs in polynomieller Zeit lösen [FHS78, SW97].)

Wir geben zwei Algorithmen an, die diese beiden Probleme für $\oplus$-OBDDs in polynomieller Zeit lösen. Dabei ist, wie bei der Reduktion von $\oplus$-OBDDs, die algebraische Struktur der $\oplus$-OBDDs von entscheidender Bedeutung.

Literatur

[FHS78] Steven Fortune, John Hopcroft, and Erik Meineche Schmidt. The complexity of equivalence and containment for free single variable program schemes. *Lecture Notes in Computer Science*, 62:227–240, 1978.

[SW97] Petr Savický and Ingo Wegener. Efficient algorithms for the transformation between different types of binary decision diagrams. *Acta Informatica*, 34:245–256, 1997.

[Waa97] Stephan Waack. On the descriptive and algorithmic power of parity ordered binary decision diagrams. *Lecture Notes in Computer Science, STACS 97*, 1200:201–212, 1997.

Padding and the Expressive Power of Existential Second-Order Logics

Thomas Schwentick

Institut für Informatik
Universität Mainz
55099 Mainz

tick@informatik.uni-mainz.de

Padding techniques are well-known from Computational Complexity Theory. Here, an analogous concept is considered in the context of existential second-order logics. Informally, a graph H is a *padded version of a graph* G, if H consists of an isomorphic copy of G and some isolated vertices. A set $\mathcal{A}$ of graphs is called *weakly expressible* by a formula φ *in the presence of padding*, if φ is able to distinguish between (sufficiently) padded versions of graphs from $\mathcal{A}$ and padded versions of graphs that are not in $\mathcal{A}$.

From results of Lynch it can be easily concluded that (essentially) every **NP**-set of graphs is weakly expressible by an existential monadic second-order (mΣ^1_1) formula with polynomial padding and built-in addition.

In particular,

> **NP**$\neq$**coNP** if and only if there is a **coNP**-set of graphs that is not weakly expressible by a mΣ^1_1-formula in the presence of addition, even if polynomial padding is allowed.

In some sense, this implies that mΣ^1_1 is well suited to investigate the **NP** vs. **coNP** question.

In the talk, the following results will be presented:

- In the above statement, addition can be replaced by two unary functions, by built-in relations of degree $O(n^\epsilon)$, for every $\epsilon > 0$, and by built-in relations with $\leq (1+\epsilon)n$ edges, respectively;
- On the other hand, mΣ^1_1 with built-in relations of degree $n^{o(1)}$ or with $n + n^{o(n)}$ edges is weak, in the sense that not every **P**-set of graphs is weakly expressible with polynomial padding in this logic; and
- mΣ^1_1 with a built-in linear order or built-in coloured trees is very weak, in the sense that they are weak and padding does not help at all.
- Corresponding results hold for several sublogics of binary Σ^1_1.

Optimale aussagenlogische Beweissysteme und vollständige Sprachen

Jochen Meßner und Jacobo Torán

Abteilung Theoretische Informatik
Universität Ulm, 89069 Ulm

{messner,toran}@informatik.uni-ulm.de

Eine in polynomieller Zeit berechenbare Funktion $h : \Sigma^* \to \Sigma^*$, die als Bildbereich die aussagenlogischen Tautologien besitzt ($h(\Sigma^*) = \mathrm{TAUT}$) heißt (abstraktes) aussagenlogisches Beweissystem. Ein konkretes Beweissystem wie z.B. die Resolution läßt sich z.B. folgendermaßen darstellen:

$$h(w) = \begin{cases} \varphi & \text{falls } w = \langle \varphi, v \rangle \text{ und } v \text{ ist Resolutionsbeweis für } \varphi, \\ x \vee \overline{x} & \text{sonst.} \end{cases}$$

Um die Stärke verschiedener Beweissysteme vergleichen zu können, wurde von Cook und Reckhow (79) der Begriff der Simulation eingeführt: Ein Beweissystem h simuliert ein Beweissystem g, falls ein Polynom p so existiert, daß für jeden g-Beweis w ein h-Beweis w' mit $g(w) = h(w')$ und $|w'| \leq p(|w|)$ existiert; h p-simuliert g, falls die Abbildung $w \mapsto w'$ in polynomieller Zeit berechnet werden kann. Optimale (bzw. p-optimale) Beweissysteme sind Beweissysteme, die jedes Beweissystem simulieren (bzw. p-simulieren). Der Begriff der Optimalität von Beweissystemen ist in gewissem Sinne eng mit dem Begriff der vollständigen Sprache verwandt, es ist jedoch nicht bekannt ob optimale Beweissysteme existieren. Wir zeigen, daß die Existenz eines p-optimalen Beweissystems many-one vollständige Sprachen für UP bezüglich logspace-Reduktionen impliziert. Unter der schwächeren Voraussetzung optimaler Beweissysteme zeigen wir die Existenz nichtuniform-many-one-vollständiger Probleme für UP und many-one vollständiger Problem für NP $\cap$ SPARSE.

On Existentially First-Order Definable Languages and their Relation to NP

Bernd Borchert, Universität Heidelberg
Dietrich Kuske, TU Dresden
Frank Stephan, Universität Heidelberg

Remember that NP is the set of languages L for which there is a nondeterministic polynomial-time Turing machine (NPTM) M such that a word x is in L iff some computation path of the computation tree of $M(x)$ accepts. It is easy to see that for example the following definition also yields the class NP (note that an NPTM defines in an obvious way an order on the computation paths): NP is the set of languages A for which there is an NPTM M such that a word x is in A iff there is an accepting path p of the computation tree of $M(x)$ and the next path $p+1$ in that computation tree does not accept. Yet another example of a characterization of NP is the following: NP is the set of languages A for which there is an NPTM M such that a word x is in A iff there is a rejecting path p of the computation tree of $M(x)$ and some later path $p' > p$ in that computation tree of $M(x)$ is accepting. The reader will notice that by writing a 1 for acceptance and a 0 for rejection the above three examples of definitions of NP can easily be described by languages: the language corresponding to the first standard definition is $\Sigma^*1\Sigma^*$, the language corresponding to the second example is $\Sigma^*10\Sigma^*$, and the language corresponding to the third is $\Sigma^*0\Sigma^*1\Sigma^*$. This concept is the so-called *leaf language approach* of characterizing complexity classes, more precisely: the polynomial-time unbalanced one.

We had three examples of languages such that the complexity class characterized by it equals NP. Now the obvious question is of course: which are exactly the languages which characterize NP? – at least we would like to know which *regular* languages characterize NP. Because the regular language $1\Sigma^*$ characterizes the complexity class P we would, with an answer to that question, solve the P=NP? question. Therefore, we can not expect a perfect answer. But under the assumption that the Polynomial-Time Hierarchy (PH) does not collapse we are able to give the following answer.

> Assume that PH does not collapse. Then a regular language L characterizes NP as an unbalanced polynomial-time leaf language if and only if L is existentially but not quantifierfree definable in $\mathrm{FO}[<, \min, \max, -1, +1]$.

The difficult part is the statement that (if PH does not collapse) a regular language which is not existentially definable does not characterize NP. The key for this part of the proof will be a characterization of the automata of first-order existentially definable languages which was given 1995 by J.-E. Pin & P. Weil.

Über mögliche logspace-Varianten des IP- und des PCP-Theorems

Heribert Vollmer Klaus W. Wagner

Theoretische Informatik
Universität Würzburg
Am Exerzierplatz 3
97072 Würzburg

{vollmer, wagner}@informatik.uni-wuerzburg.de

Wir formalisieren die Resultate "IP=PSPACE" [Shamir, Lund, Fortnow, Karloff, Nisan] und "NP=PCP($\log n, O(1)$)" [Arora, Lund, Motwani, Sudan, Szegedy] mit Hilfe von Operatoren höherer Ordnung, d. h. Operatoren über Mengen. Wir untersuchen sodann die Gültigkeit dieser Aussagen, wenn man sie in den Kontext logarithmisch-platzbeschränkter Berechnungen überträgt. So gelangen wir zu einer IP-ähnlichen Charakterisierung der Klasse der Probleme, die nichtdeterministisch in polylogarithmischem Raum und polynomieller Zeit gelöst werden können, aber auch zu einer Reihe von offenen Fragestellungen über Klassen zwischen NC^1 und POLYLOGSPACE.