26. Workshop über Komplexitätstheorie, Datenstrukturen und effiziente Algorithmen TU-Berlin

20. Juni 1995

26. Workshop über Komplexitätstheorie, Datenstrukturen und effiziente Algorithmen

TU Berlin, 20. Juni 1995

Programm

ab 9.15	IMBISS
9.50	BEGRÜSSUNG
10.00	Martin Dietzfelbinger (Dortmund) *Universal hashing and k-wise independent random variables via integer arithmetic without primes*
10.25	Claudia Bertram, Thomas Hofmeister (Dortmund) *Multiples Produkt modulo beliebiger Zahlen*
10.50	PAUSE
11.10	Bernd Borchert (Heidelberg) *Describing Boolean Functions*
11.35	Rolf Niedermeier (Tübingen), Peter Rossmanith (München) *PRAM's Towards Realistic Parallelism: BRAM's*
12.00	Andreas Birkendorf, Hans Ulrich Simon (Dortmund) *Gute Lösungen schwerer Optimierungsprobleme lassen sich „lernen"!*
12.25	MITTAGSPAUSE
14.15	Daniel Hammer (Berlin) *Kolmogorov Complexity and Polymatroids*
14.40	Wolfgang Merkle, Yongge Wang (Heidelberg) *Separations by Random Oracles and Almost Classes for Generalized Reducibilities*
15.05	PAUSE
15.20	Lane A. Hemaspaandra (Rochester), Jörg Rothe (Jena), Gerd Wechsung (Jena) *Easy Sets and Hard Certificates*
15.45	PROBLEM-SESSION
16.30	PAUSE
16.50	Lane A. Hemaspaandra (Rochester), Zhigen Jiang (Beijing), Jörg Rothe (Jena), Osamu Watanabe (Tokio) *Multi-Selectivity and Complexity-Lowering Joins*
17.15	Susanne Kaufmann (Karlsruhe) *Effektive Suchprobleme bei breitenbeschränkten Bäumen*
17.40	Hans-Jörg Burtschick (Berlin) *1NL ≠ 1UL*
18.05	ENDE

Universal hashing and k-wise independent random variables via integer arithmetic without primes

Martin Dietzfelbinger
Fachbereich Informatik, Universität Dortmund, Germany
email: dietzf@ls2.informatik.uni-dortmund.de

Let $u, m \geq 1$ be arbitrary integers and let $r \geq um$ be any multiple of m. The main result of this talk is that the multiset $\mathcal{H} = \{h_{a,b} \mid 0 \leq a, b < r\}$ of functions from $U = \{0, \ldots, u-1\}$ to $M = \{0, \ldots, m-1\}$, where

$$h_{a,b}(x) = ((ax + b) \bmod r) \operatorname{div} (r/m), \text{ for } x \in U,$$

is a $(c, 2)$-universal class of hash functions from U to M in the sense of Carter and Wegman (1979), with $c = 5/4$. More precisely, we show that for h chosen from $\mathcal{H}$ at random we have

$$\left| \mathbf{Prob}(h(x_1) = i_1 \wedge h(x_2) = i_2) - 1/m^2 \right| \leq (u/2r)^2 \leq 1/4m^2,$$

for distinct $x_1, x_2 \in M$ and arbitrary $i_1, i_2 \in M$. Among the many known constructions of $(c, 2)$-universal classes there was none that would get by with pure integer arithmetic without the assumption that a prime number of size the order of $|U|$ or at least $|M|$ was given.

Varying this result, we obtain: (a) two-independent (or almost two-independent) sequences of random variables; (b) universal hash classes of higher degree ("(c, k)-universal" classes) and k-wise independent random variables, for $k \geq 2$; (c) algorithms for static and dynamic perfect hashing with an optimal number of random bits; all using pure integer arithmetic without the need for providing prime numbers (arbitrary or random) of a certain size. Our results may be helpful both in practical and theoretical applications of hashing or two-wise or k-wise independent sampling. It should be noted that the focus here is not on minimizing the size of the probability space used, as in much of the recent work on "almost k-independent random variables", but on the realization of such variables or hash classes using the most natural and most widely available operations, viz., integer arithmetic. Incidentally, our construction provides universal hash classes with the smallest known circuit complexity and, using a result by Mansour, Nisan, and Tiwari (1993), yields the best known time-space tradeoff for multiplication of integers in binary representation.

Multiples Produkt modulo beliebiger Zahlen

Claudia Bertram, Thomas Hofmeister
Lehrstuhl Informatik II
Universität Dortmund
D-44221 Dortmund
Email: bertram / hofmeist } @ls2.informatik.uni-dortmund.de

In den letzten Jahren wurden Thresholdschaltkreise konstanter Tiefe eingehend untersucht. Trotz der scheinbar beschränkten Leistungsfähigkeit der Thresholdgatter, die nur entscheiden können, ob eine bestimmte Mindestanzahl von Einsen im Input vorliegt, scheint der Beweis einer superpolynomiellen unteren Schranke selbst für Schaltkreise der Tiefe 3 sehr schwierig zu sein. Die erste exponentielle untere Schranke für Thresholdschaltkreise der Tiefe 2 zeigten [HMPST] für das „Innere Produkt modulo 2", definiert durch $IP_n(x_1, y_1, \ldots, x_n, y_n) := x_1y_1 \oplus \cdots \oplus x_ny_n$.

Im Laufe der Zeit stellten sich Thresholdschaltkreise als überraschend mächtig heraus. Selbst für so komplexe Boolesche Funktionen, wie z.B. die Division zweier Binärzahlen, ließen sich polynomiell große Thresholdschaltkreise der Tiefe 3 konstruieren. Dies gelingt, weil der Wert dieser Funktionen modulo einer Primzahl p leicht berechnet werden kann. Mit Hilfe des Chinesischen Restsatzes kann man nämlich eine Zahl Z auf einfache Art „rekonstruieren", wenn man ihren Wert modulo einiger Primzahlen $p_1, \ldots, p_T$ kennt.

Eine der wenigen Funktionen, für die wir einen Thresholdschaltkreis der Tiefe 4, aber keinen der Tiefe 3 kennen, ist das „Multiple Produkt". Die Boolesche Funktion „Multiples Produkt" MP_n berechnet in Binärdarstellung den Wert des Produkts von n Zahlen der Länge n. Warum funktioniert die Zerlegung mittels Chinesischem Restsatz hier nicht?

Wir zeigen, daß die Schwierigkeit, Schaltkreise kleinerer Tiefe für MP_n zu konstruieren, daher rührt, daß schon für alle Zahlen $m \notin \{2, 3, 4, 6, 8, 12, 24\}$ die Berechnung von MP_n modulo m nicht in Tiefe 2 bei polynomieller Größe möglich ist. Dieses Ergebnis gilt auch noch, wenn m exponentiell in n wächst.

In [K1] wurde bereits für *spezielle* Thresholdschaltkreise der Tiefe 2 gezeigt, daß für Zahlen m mit $O(\log n)$ Bits, die einen Primfaktor größer als 3 haben, die Berechnung des multiplen Produkts modulo m exponentielle Größe benötigt.

Wir untersuchen ferner Moduli, die die Form 2^i3^j haben und zeigen, daß es polynomiell große Thresholdschaltkreise der Tiefe 2 für die Berechnung von MP_n modulo m gibt, für $m \in \{2, 4, 8\}$, und daß es keinen solchen Schaltkreis gibt, wenn m durch 9 oder 16 teilbar ist.

Literatur

[HMPST] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, G. Turán, *Threshold circuits of bounded depth,* Proceedings of 28th FOCS, 1987, 99-110.

[K1] M. Krause, *On realizing iterated multiplication by small depth threshold circuits,* Proc. of 12th STACS (1995), 83-94.

Describing Boolean Functions

Bernd Borchert[1]

A Boolean function is a function $\{0,1\}^n \to \{0,1\}$ for some n. There are many ways of describing Boolean functions, for example circuits, formulas, branching programs, decision trees, or simply full truth-tables. Additionally, there are all kinds of syntactical variations of these, for example DNF, KNF, restricted $\wedge$-$\vee$-depth, restricted fanout, operator sets different from $\{\wedge, \vee, \neg\}$, bounded width, and many more variations, most of them restrictions. So there are many different description systems for one single set of objects, namely Boolean functions, and it should be natural to compare them in terms of translatability. For example, a decision tree can easily be translated into a formula by replacing each node labeled with x by $(x \vee \ldots) \wedge (\neg x \vee \ldots)$. As another example, formulas are trivially translatable into circuits because every formula is basically already a circuit.

This idea of comparing different description systems for the same set of objects is well-known: in Recursion Theory description systems are called *numberings*, and names are natural numbers. Two numberings are compared by the existence of a computable function which translates one into the other, see for example K. Weihrauch, *Computability*, Springer Verlag, 1987. In order to transfer the concepts to Complexity Theory, the words on the alphabet $\Sigma = \{0,1\}$ will serve as names, and a *description system* (or *notation*) for a set S is just a surjective partial function ν from Σ^* to S for which the set $\{w | \nu(w)$ is not defined$\}$ is polynomial-time computable. For two description systems ν, μ for a set S say that ν is *p-translatable* into μ, in short $\nu \leq_m^p \mu$, if there is a total polynomial time computable function $f : \Sigma^* \to \Sigma^*$ such that $\nu(w) = \mu(f(w))$ for all w in the domain of ν. The relation is obviously reflexive and transitive.

Choose your favorite natural encodings of circuits, formulas, branching programs, decision trees, and full truth-tables into words of Σ^*, the inverse mappings will induce description systems $\nu_{\mathrm{cir}}, \nu_{\mathrm{for}}, \nu_{\mathrm{bra}}, \nu_{\mathrm{dec}}, \nu_{\mathrm{tru}}$ for Boolean functions. It was indicated above that for example $\nu_{\mathrm{dec}} \leq_m^p \nu_{\mathrm{for}}$ and $\nu_{\mathrm{for}} \leq_m^p \nu_{\mathrm{cir}}$. Some other relations do provably not hold, for example, it is witnessed by the parity function that formulas and branching programs are not p-translatable into decision trees. Some relations are not known to the author, for example whether circuits are p-translatable into formulas or into branching programs.

Each of the five description systems for Boolean functions from above has the following property: given a description w for a Boolean function f and an assignment a (given as a sequence of bits) for the variables of f, the Boolean value $f(a)$ can be computed in polynomial time in $|w| + |a|$. Call description systems which have this natural property *p-evaluatable*. The main result of this talk is the following: a description system for Boolean functions is p-evaluatable if and only if it is p-translatable into ν_{cir}. This shows a kind of universality (or completeness) of the description system ν_{cir} which the other description systems may possibly lack.

[1] Address: Im Neuenheimer Feld 294, 69120 Heidelberg, bb@math.uni-heidelberg.de

PRAM's Towards Realistic Parallelism: BRAM's

Rolf Niedermeier* Peter Rossmanith†

The obvious dilemma between theory (Parallel Random Access Machines) and practice (asynchronous, distributed memory machines) of parallel computation is the seemingly large gap between ideal model and real machine. There roughly are two main ways to bridge the gap. The first is to try to preserve the full PRAM on the level of algorithm design and to show up ways how to implement PRAM's on bounded degree networks. The second one is to abandon the intact world of PRAM's and to use models of computation closer to existing parallel machines. Whereas the first approach suffers at least from large constant factors for PRAM simulations on existing machines, the second approach destroys the conceptual simplicity of the underlying model and makes it harder or even impossible to create a structural complexity theory upon it. So we subsequently try to find a middle-way between both the extremes, introducing the BRAM.

The BRAM (Block-RAM) model evolves by putting together the concept of *data-independent* computations with the *owner read, owner write* restriction for PRAM's. In addition, we assume that *each* of the p processors of a BRAM holds n data-items of an input of total size pn in its *local memory* and that the number of global memory cells may range from p to p^2. Note that, for example, BRAM's with global memory size $O(p)$ in a sense model bounded degree networks of processors and that BRAM's with global memory size p^2 in fact model completely connected parallel machines.

As to the algorithmic problems studied in this work, let us only mention that the sorting problem can be solved work-optimally with a speedup factor between $p/2$ and $p/3$ compared to the best sequential algorithm. Often we are able to *prove* the *work-optimality* of our algorithms. Algorithms designed for the BRAM due to their static communication pattern known at compile time, due to their owner read, owner write protocol inter alia avoiding contention effects, and due to their in general small number of blocked communications enabling the bypassing of latency effects and requiring only simple synchronization mechanisms and a strong use of *locality*, should allow an easy and efficient mapping to existing distributed memory machines.

*Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Sand 13, D-72076 Tübingen, niedermr@informatik.uni-tuebingen.de

†Fakultät für Informatik, Technische Universität München, Arcisstr. 21, D-80290 München, Fed. Rep. of Germany, rossmani@informatik.tu-muenchen.de

Gute Lösungen schwerer Optimimierungsprobleme lassen sich „lernen“ !

Andreas Birkendorf * und Hans Ulrich Simon †
FB Informatik, LS II, Univ. Dortmund, 44221 Dortmund, Germany

In unserem Vortrag wollen wir eine neue Idee vorstellen, wie man „Maschinelles Lernen“ als Strategie einsetzen kann, um schwere kombinatorische Optimierungsprobleme zu attackieren. Soweit wir wissen, ist dies das erste Mal, daß die beiden Gebiete „Maschinelles Lernen“ und „Kombinatorische Optimierung“ in dieser Art miteinander verknüpft werden.

Das von uns betrachtete Optimierungsproblem besteht darin, minimale Repräsentationen von partiell definierten Funktionen zu finden:

Formal haben wir es mit einer beliebigen Funktionenklasse $F = \{f \mid f : X \to Y\}$ und einer Repräsentationssprache $R(F)$ für F zu tun. Dabei ist $R(F)$ eine Menge von Strings (Repräsentationen). Über eine Semantik-Abbildung ist jedem $f \in F$ eine Untermenge von Repräsentationen $R(f)$ mit $\emptyset \neq R(f) \subset R(F)$ zugeordnet. Bekannte Repräsentationssprachen im Falle binärer Funktionen ($X = \{0,1\}^n$, $Y = \{0,1\}$) sind *Endliche Automaten*, *Neuronale Netze* oder die in den letzten Jahren populär gewordenen *Ordered Binary Decision Diagrams* (OBDDs).

Sind eine Funktion $f \in F$ und irgendeine Repräsention $r \in R(f)$ gegeben, dann lassen sich minimale Repräsentationen für f in vielen Repräsentationssprachen in polynomieller Zeit bezogen auf die Länge von r finden. Stichworte sind hier z. B. Automatenminimierung und Reduktionsregeln für OBDDs.

Erweitert man die Klasse jedoch auf partielle Funktionen $F^D = \{g \mid g : D \to Y,\ D \subset X\}$, so stellt sich dasselbe Problem oft als sehr viel schwieriger dar. Unter einer (minimalen) Repräsentation für ein $g \in F^D$ verstehen wir dabei eine (minimale) Repräsentation aus der Menge $R(F_g^D)$ von Repräsentationen der Funktionen $F_g^D = \{f \in F \mid f(x) = g(x)\ \forall x \in D\}$. In diesem Sinne ist z.B. das Problem der Suche nach minimalen partiellen endlichen Automaten oder OBDDs NP-hart.

Akzeptiert man P$\neq$ NP, sind für diesen Problemtyp also kluge Heuristiken notwendig, um in Polynomialzeit wenn schon nicht optimale so doch zumindest fast optimale Lösungen zu finden.

Unsere Idee ist es, einen *On-Line*-Lernalgorithmus A als Heuristik einzusetzen:

Dieser befragt ein Orakel O nach der unbekannten Zielfunktion $g \in F^D$ in Form sogenannter *Membership*- und *Equivalence*-Fragen. In unserer Anwendung kann das Orakel durch Verwendung einer voroptimierten Repräsentation $r_f \in R(f)$, wobei $f \in F_g^D$ ist, simuliert werden.

Während des Lernens ergibt sich eine Menge $M \subset X$ von Beispielen x, deren Funktionswert $f(x)$ der Lernalgorithmus A kennt.
Es gibt Lernalgorithmen, denen es gelingt, nach einer polynomiellen Anzahl von Anfragen eine korrekte Hypothesenfunktion $h \in F_g^D$ für g in Form einer Repräsentation $r_h \in R(h)$ zu liefern. Zusätzlich ist r_h minimal in der Repräsentationsmenge $R(F_g^{M \cup D})$, was fast unserem Ziel entspricht, eine minimale Repräsentation in $R(F_g^D)$ zu finden.

Einen solchen Lernalgorithmus mit zugehörigem Orakel wollen wir als Anwendungsbeispiel unserer Idee für OBDDs entwickeln. Mit diesem Werkzeug gelingen uns auf in der Praxis vorkommenden partiellen OBDDs Reduktionsraten von über 30 %, insbesondere durch die Verwendung des „iterierten“ Lernens: Dabei wählen wir r_h als r_f und lernen erneut.

*email: birkendo@ls2.informatik.uni-dortmund.de. Supported by DFG grant Si 498/3-1
†email: simon@ls2.informatik.uni-dortmund.de. Supported in part by BMFT grant 01 IN 102 C/2

Kolmogorov Complexity and Polymatroids

Daniel Hammer

Technische Universität Berlin
hammer@math.tu-berlin.de

Polymatroids are well studied objects in combinatorial geometry and matroid theory [2]. A polymatroid $\mathbb{P}$ is a pair (S, f) where S, a non-empty finite set and f, a non-negative real valued function on the powerset $\mathcal{P}(S)$ of S satisfying

(1) $f(\varnothing) = 0$ (normalization)
(2) $A \subseteq B \subseteq S$ implies $f(A) \leqslant f(B)$ (monotonicity)
(3) $A, B \subseteq S$ implies $f(A \cup B) + f(A \cap B) \leqslant f(A) + f(B)$ (submodularity)

and the vectors $x \in \mathbb{R}^S_{\geqslant 0}$ that satisfy $x(A) \leqslant f(A)$ for all $A \subseteq S$, the independent vectors of $\mathbb{P}$.
We use the theory of polymatroids and their independence polytopes to develop a conception of describing finite sets $\Xi(n) \stackrel{\text{def}}{=} \{x_1, x_2, \ldots, x_{n-1}, x_n\} \subset \{0,1\}^*$, $(n \in \mathbb{N}, n \geqslant 2)$ of binary words by means of the (simple) Kolmogorov complexity K [1] of their subsets. The desired description should work in both directions; given a set $\Xi(n)$ we want to know all true linear inequalities of Kolmogorov complexity of the words $x_1, x_2, \ldots, x_{n-1}, x_n$ itself, all pairs of them, all 3-tuples, ... etc; and given 2^n (not necessary distinct) natural numbers that satisfy such system of inequalities we want to specify a witness-set $\Xi(n)$ such that these numbers correspond to the complexities of respective sets in $\mathcal{P}(\Xi(n))$. An inequality is considered to be true, if it holds up to the additive term $O(\log_2(\mathrm{K}(\Xi(n))))$.
Any polymatroid function, as well as the set of all polymatroid functions, over a given $\mathcal{P}(\Xi(n))$ is defined by a finite set of inequalities — corresponding to submodularity and monotonicity. We prove that the measure K is monotone, quasi-normalized and log-submodular, and, therefore, it can be considered as special type of polymatroid function. The induced inequalities are the same as for "usual" polymatroids, except of their sharpness. It turns out that this difference does not influence the geometrical and combinatorial properties of the generated cone. The polymatroid cone generated by polymatroid functions over $\mathcal{P}(\Xi(n))$ is identical with the complexity cone C_n that is generated by true all linear inequalities of Kolmogorov complexity of binary representations of the subsets of $\Xi(n)$. Presenting a witness-set in $\{0,1\}^*$ for the extremal rays of the complexity cone C_n is of rising difficulty with growing cardinality n of the set $\Xi(n)$ and leads to interesting results.

References

[1] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications.* Springer Verlag, New York, 1993.

[2] D.J.A. Welsh. *Matroid Theory.* Academic Press, London, second printing, 1986.

Separations by Random Oracles and Almost Classes for Generalized Reducibilities

Wolfgang Merkle Yongge Wang

Mathematisches Institut
Universität Heidelberg
Im Neuenheimer Feld 294
69120 Heidelberg
Germany
e-mail:{merkle | wang }.math.uni-heidelberg.de

May 1995

Abstract

Given two binary relations $\leq_r$ and $\leq_s$ on 2^ω which are closed under finite variation (of their set arguments) and a set X chosen randomly by independent tosses of a fair coin, one might ask for the probability that the lower cones $\leq_r (X)$ and $\leq_s (X)$ w.r.t. $\leq_r$ and $\leq_s$ are different. By closure under finite variation, the Kolmogorov 0-1 Law yields immediately that this probability is either 0 or 1; in the case it is 1, the relations are said to be separable by random oracles. Again by closure under finite variation, the probability that a randomly chosen set X is in the upper cone of a fixed set A w.r.t. $\leq_r$ is either 0 or 1. So let Almost_r contain all sets for which the upper cone w.r.t. $\leq_r$ has measure 1.

In the following, separations by random oracles and Almost classes are considered in the context of generalized reducibilities, that is, binary relations on 2^ω which can be defined by a countable set of total continuous functionals on 2^ω in the same way as the usual resource bounded reducibilities are defined by an enumeration of appropriate oracle Turing machines. The concept generalized reducibility comprises all natural resource bounded reducibilities, but is more general. In particular it does not involve any kind of specific machine model or even effectivity.

It is shown that results about separations by random oracles and about Almost classes can be obtained for generalized reducibilities under rather general additional assumptions. From these results, one then obtains as corollaries several results that have previously been shown in the context of time or space bounded oracle Turing machine computations.

Easy Sets and Hard Certificate Schemes

Lane A. Hemaspaandra
Dept. of Computer Science
University of Rochester
Rochester, NY 14627, USA

Jörg Rothe *
Institut für Informatik
Friedrich-Schiller-Universität Jena
07743 Jena, Germany

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

Abstract

Can easy sets only have easy certificate schemes? In this paper, we study the class of sets that, for all NP certificate schemes (i.e., NP machines), always have easy acceptance certificates (i.e., accepting paths that can be computed in polynomial time). We also study the class of sets that, for all NP certificate schemes, infinitely often have easy acceptance certificates. We give structural conditions—regarding immunity and class collapses—that control the size of these classes. We also provide negative results showing that some of our positive claims are optimal with regard to being relativizable. Our negative results are proven using a novel observation: we show that the classic "widespacing" oracle construction technique yields instant non-bi-immunity results. In particular, we establish a result that improves upon Baker, Gill, and Solovay's classic result that $\mathrm{NP} \neq \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}$ holds in some relativized world, and that in addition links their result with the theorem of Borodin and Demers.

*Email: rothe@mipool.uni-jena.de.

Multi-Selectivity and Complexity-Lowering Joins

Lane A. Hemaspaandra
Dept. of Computer Science
University of Rochester
Rochester, NY 14627, USA

Zhigen Jiang
Institute of Software
Chinese Academy of Sciences
Beijing 100080, China

Jörg Rothe *
Institut für Informatik
Friedrich-Schiller-Universität Jena
07743 Jena, Germany

Osamu Watanabe
Dept. of Computer Science
Tokyo Institute of Technology
Tokyo 152, Japan

Abstract

This paper introduces a flexible notion of selectivity that generalizes Selman's P-selectivity by allowing the selector to operate on multiple input strings and to output an arbitrary subset of the inputs if a certain promise is not met. Depending on parameters that quantify the "amount of promise," we obtain a selectivity hierarchy, denoted by SH, which we prove does not collapse. We study the internal structure and the properties of SH and completely establish, in terms of incomparability and strict inclusion, the relations between our classes and Ogihara's classes of polynomial-time membership comparable (P-mc, for short) sets. Although SH is a strict hierarchy, we prove that the core results holding for the P-selective sets, and proving them structurally simply, also hold for SH. In particular, all sets in SH have small circuits; the NP sets in SH are in Low_2, the second level of the low hierarchy within NP; and SAT cannot be in SH unless P = NP. Though the P-selective sets are in EL_2, the second level of the extended low hierarchy, we prove that not all sparse sets in SH are in EL_2. This is the strongest known EL_2 lower bound, strengthening the result that P/poly, and indeed SPARSE, is not contained in EL_2. Relatedly, we prove that the join of sets may actually be simpler than the sets themselves: there exist sets that are not in EL_2, yet their join *is* in EL_2. That is, *in terms of extended lowness, the join operator can lower complexity*. We also prove that EL_2 is not closed under union or intersection. Finally, it is known that the P-selective sets are not closed under union or intersection. We introduce an extended selectivity hierarchy that is flexible enough to capture those closures, and yet, in contrast with Ogihara's P-mc hierarchy, is refined enough to distinguish them.

*Email: rothe@mipool.uni-jena.de.

Effektive Suchprobleme bei breitenbeschränkten Bäumen

Susanne Kaufmann

Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe,
D-76128 Karlsruhe, Germany. srogina@ira.uka.de

Wir betrachten Probleme, unendliche Pfade in Binärbäumen zu berechnen. Hierbei ist ein Baum T durch ein Programm gegeben, das die Menge der Knoten von T aufzählt. Das effektive Suchproblem für eine Klasse von Bäumen $\mathcal{A}$ besteht darin, zu jedem $T \in \mathcal{A}$ einen unendlichen Pfad von T auszugeben. Da dieses Problem im allgemeinen nicht algorithmisch lösbar ist, betrachten wir folgende Abschwächung:

Das k-Suchproblem für $\mathcal{A}$
Eingabe: Ein Programm für einen Baum $T \in \mathcal{A}$.
Ausgabe: Eine Liste von k Programmen $p_1, \ldots, p_k$, so daß eines davon bis auf endlich viele Fehler einen unendlichen Pfad ausgibt.

Eine interessante Frage ist hierbei, wie klein k gewählt werden kann; das kleinst mögliche k wird mit $sel(\mathcal{A})$ bezeichnet. Im Vortrag wird $sel(\mathcal{A})$ für verschiedene Klassen breitenbeschränkter Bäume bestimmt. Hierbei heißt ein Baum breitenbeschränkt, wenn die Zahl der Knoten in jeder Tiefe durch eine Konstante beschränkt ist.

1UL $\neq$ 1NL

Hans-Jörg Burtschick

TU-Berlin;
Fachbereich Informatik, Sekr.: FR 6-2;
Franklinstr. 28/29;
10587 Berlin

Eine Sprache ist genau dann in UP enthalten, wenn sie von einer polynomiell zeitbeschränkten NTM M akzeptiert wird und M für alle Eingaben höchstens einen akzeptierenden Pfad besitzt. Die Untersuchung der Klasse UP ist für die Beantwortung kryptologischer Fragestellungen und im Zusammenhang mit Countingklassen wie span-P und #P interessant. Analog zu UP sind die Klassen UL und 1UL für nichtdeterministische logspace- und nichtdeterministische one-way logspace Turingmaschinen definiert. One-way logspace Turingmaschinen bewegen den Eingabekopf nur in eine Richtung.

Ebenso wie für NP und UP ist es auch für NL und UL nicht bekannt, ob die Klassen verschieden sind. In diesem Vortrag wird gezeigt:

$$1\mathrm{UL} \subsetneq 1\mathrm{NL}.$$

Dieses Ergebnis ist im Zusammenhang mit one-way logspace Countingklassen interessant. Alvarez und Jenner zeigten [AJ93]:

$$1\mathrm{UL} = 1\mathrm{NL} \quad \Leftrightarrow \quad \#1\mathrm{L} = \text{span-}1\mathrm{L} \quad \Leftrightarrow \quad \#1\mathrm{L} \subseteq \text{opt-}1\mathrm{L}$$

(Die Klassen #1L, span-1L und opt-1L sind analog zu #L, span-L und opt-L für one-way logspace Turingmaschinen definiert.) Da aus 1UL $\neq$ 1NL auch F1L $\neq$ #1L folgt und span-1L $\neq$ #P ist, gilt also:

$$\mathrm{F1L} \subsetneq \#1\mathrm{L} \subsetneq \text{span-}1\mathrm{L} \subsetneq \#\mathrm{P}.$$

Literatur

[AJ93] C. Àlvarez and B. Jenner. A Very Hard Log Space Counting Class. *Theoretical Computer Science*, 107:3–30, 1993.