COMPLEXITY ABSTRACTS 1998. Vol VIII

Abstract

This is a collection of one page abstracts of recent results of interest to the Complexity community. The purpose of this document is to spread this information, not to judge the truth or interest of the results therein.

Titles of Abstracts

Lower Bounds for Approx. by Low Degree Polys Over $\mathsf{Z_m}$
On Counting AC^0 Circuits with Negative Constants
Sparse Sets, Approximable Sets, and Parallel Queries
Stronger separation for RSRs, rounds and advice
Gaps in Bounded Query Hierarchies
Circuit Lower Bounds Collapse Relativized Complexity Classes
Optimal Prop. Proof Systems and Oracle-Relativized Logic
Delayed Binary Search, or Playing 20 Qs with a Procrastinator
One-sided Versus Two-sided Randomness
Quantum NP is Hard for PH
Justification of Rissanen's Approx. Formula for Prior Prob.
Relativized Worlds with an Infinite Hierarchy
The Communication Complexity of Enumeration
Discontinuities in Recurrent Neural Networks
Tally NP Sets and Easy Census Functions
An Introduction to Query Order
Downward Collapse from a Weaker Hypothesis
What's Up with Downward Collapse: Easy-Hard Tech., BH and PH
Optimal Separations for Parallel vs Sequential Self-Checking
Linear-Nondet, Linear-Sized, Karp-Lipton Advice for P-Sel Sets
2nd Step Towards Circuit Complexity Analogs of Rice's Theorem
Creating Strong Total Comm. Assoc. 1-Way f's from Any 1-Way f's
On Characterizations of the Basic Feasible Functionals- Part I
Improved Time and Space Hierarchies of One-Tape Off-Line TMs
Monotone Real and Non-Monotone Bool. Circuits Are Incomparable
Linear Codes Hard for Obliv. Read-Once Parity BP's
On Branching Programs With Bounded Uncertainty
A Note on the Shortest Lattice Vector Problem
Computational Foundation of Quantum Randomness Hierarchy
On the complexity of moving vertices in a graph
Some Results on Commutative Oracles
The Comp. of Computing Opt. Assignments of Gen. Prop. Fmls.
BPP equals BPLINTIME under an Oracle
Immunity and Simplicity for Counting Classes
On an optimal deterministic algorithm for SAT
Ramsey Theory and the Polynomial Hierarchy
Choosing a Physical Model: Why Symmetries?
Extractors for the real world

Lower Bounds for Approximations by Low Degree Polynomials Over Z_m

Noga Alon, Dept. of Mathematics, Tel Aviv University, 69978 Tel Aviv, ISRAEL (noga@math.tau.ac.il)
Richard Beigel, Dept. of Electrical Engineering and Computer Science, University of Illinois at Chicago, Rm. 1116, SEO Building, M/C 154, 851 S. Morgan Street, Chicago, IL 60607, USA (beigel@eecs.uic.edu)

Abstract Number 98-1

Smolensky (STOC 87) used a dimension argument to prove that a degree-$o(\sqrt{n})$ polynomial over Z_p (p an odd prime) must differ from the parity function on at least a constant fraction of all points in the Boolean n-cube. He later (FOCS 93) used Hilbert functions to improve that result to a $1/2-o(1)$ fraction. Goldmann (IPL 95) proved that a linear function over Z_m (m any odd number) must differ from the parity function on at least a $1/2 - 1/\text{exponential}$ fraction of all points.
We provide the first lower bounds for approximations over Z_m by nonlinear polynomials:

- A degree-2 polynomial over Z_m (m odd) must differ from the parity function on at least a $1/2 - 1/n^{\Omega(1)}$ fraction of all points in the Boolean n-cube.
- A degree-$O(1)$ polynomial over Z_m (m odd) must differ from the parity function on at least a $1/2 - o(1)$ fraction of all points in the Boolean n-cube.

A full paper is not yet available.

On Counting AC^0 Circuits with Negative Constants

Andris Ambainis, Computer Science Division, University of California, Berkeley, (ambainis@cs.berkeley.edu)
David Mix Barrington, Computer Science Department, University of Massachusetts, (barring@cs.umass.edu)
Huong LêThanh, Laboratoire de Recherche en Informatique, Université de Paris-Sud, (huong@lri.fr)

Abstract Number 98-2

Continuing the study of the relationship between TC^0, AC^0 and arithmetic circuits, started by Agrawal et al., we answer a few questions left open in this paper. Our main result is that the classes DiffAC^0 and GapAC^0 coincide, under poly-time, log-space, and log-time uniformity. From that we can derive that under logspace uniformity, the following equalities hold:

$$C_{=}AC^0 = PAC^0 = TC^0.$$

A full paper will be published in the procceding of MFCS'98.

Sparse Sets, Approximable Sets, and Parallel Queries to NP

V. Arvind, Institute of Mathematical Sciences, C. I. T Campus, Chennai 600 113, INDIA, (arvind@imsc.ernet.in)

Jacobo Torán, Abteilung Theoretische Informatik, Universität Ulm, D-89069 Ulm, GERMANY. (toran@informatik.uni-ulm.de)

Abstract Number 98-3

We relate the existence of disjunctive hard sets for NP to other well studied hypotheses in complexity theory showing that if an NP-complete set or a coNP-complete set is polynomial-time disjunctively reducible to a sparse set then $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$. Using similar arguments, and building on a result from [Si98] we also prove that if SAT is $O(\log n)$-approximable then $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$. Since it is already known that $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ implies that SAT is $O(\log n)$-approximable [BFT97], it follows as a consequence of our result that the two hypotheses $\mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}}[\log]$ and SAT is $O(\log n)$-approximable are equivalent, thus solving an open question from [BFT97]. We show also as a consequence of our first result that if an NP-complete set or a coNP-complete set is disjunctively reducible to a sparse set of polylogarithmic density then P = NP. Furthermore, we show that if $\mathrm{Mod}_k\mathrm{P}$ is disjunctively reducible to a sparse set then RP = NP.

References

[BFT97] H. Buhrman, L. Fortnow, and L. Torenvliet. Six hypotheses in search of a theorem. In *Proc. 12th Annual IEEE Conference on Computational Complexity,* 2–12, IEEE Computer Society Press, 1997.

[Si98] D. Sivakumar. On membership comparable sets. To appear in the *13th IEEE Computational Complexity Conference 1998.*

A full paper is available online as ECCC Technical Report No. TR98-027.

Stronger separation for RSRs, rounds and advice

László Babai, Dept. of Computer Science, University of Chicago, 1100 E. 58th Street, Chicago IL 60637, USA. (laci@cs.uchicago.edu)
Sophie Laplante, LRI, Université Paris-Sud, Bâtiment 490, 91405 Orsay, France. (laplante@lri.fr)

Abstract Number 98-4

A function f is random-self-reducible (RSR) if it can be computed in probabilistic polynomial time, with the help of queries to an oracle that computes f. The distribution of each individual query must depend only on the length of the instance that gave rise to it, so that the oracle cannot gain any information about the instance (except perhaps its length) based on the distribution of an individual query. A function f is coherent if it can be computed in probabilistic polynomial time, with access to an oracle for f; the only restriction is that the instance itself may not be queried. Any RSR function is coherent given polynomial length advice.

A few separations between interactive and non-interactive self-reductions are known. Feigenbaum, Fortnow, Lund and Spielman (*Comp. Compl.,* 4:158–174, 1994) show that if $NEEE \not\subseteq BPEEE$ then there is a language in $NP \setminus P$ which is adaptively RSR but not nonadaptively RSR. Hemaspaandra, Naik, Ogihara and Selman (*JCSS,* 53(2):194–209, 1996) show that if $NP \not\subseteq BPE$ then there is such a language in $NP \setminus BPP$, which is also not self-reducible (a deterministic version of coherence.) However, all of these languages are in $P/poly$. Using Kolmogorov complexity, Feigenbaum, Fortnow, Laplante and Naik (*Comp. Compl.,* 7(2), *to appear*) exhibit a set which is adaptively coherent but not nonadaptively coherent, even with polynomial advice.

We show that for random-self-reducible functions, adaptiveness cannot be traded off for advice and indiscretion. More precisely, what we show is that there is a function which is random-self-reducible with 2 rounds of queries, but which is not *coherent*, even if polynomial advice is allowed, when the queries must be made in a single round.

A full paper will be available shortly by email to laplante@lri.fr, and on the Web at http://www.lri.fr/~laplante/.

Gaps in Bounded Query Hierarchies

Richard Beigel, Dept. of Electrical Engineering and Computer Science, University of Illinois at Chicago, Rm. 1116, SEO Building, M/C 154, 851 S. Morgan Street, Chicago, IL 60607, USA (beigel@eecs.uic.edu)

Abstract Number 98-5

Prior results show that most bounded query hierarchies cannot contain finite gaps. For example, it is known that

$$\mathrm{P}^{\mathrm{SAT}}_{(m+1)\text{-tt}} = \mathrm{P}^{\mathrm{SAT}}_{m\text{-tt}} \Rightarrow \mathrm{P}^{\mathrm{SAT}}_{\mathrm{btt}} = \mathrm{P}^{\mathrm{SAT}}_{m\text{-tt}}$$

and for all sets A

- $\mathrm{FP}^{A}_{(m+1)\text{-tt}} = \mathrm{FP}^{A}_{m\text{-tt}} \Rightarrow \mathrm{FP}^{A}_{\mathrm{btt}} = \mathrm{FP}^{A}_{m\text{-tt}}$
- $\mathrm{P}^{A}_{(m+1)\text{-T}} = \mathrm{P}^{A}_{m\text{-T}} \Rightarrow \mathrm{P}^{A}_{\mathrm{bT}} = \mathrm{P}^{A}_{m\text{-T}}$
- $\mathrm{FP}^{A}_{(m+1)\text{-T}} = \mathrm{FP}^{A}_{m\text{-T}} \Rightarrow \mathrm{FP}^{A}_{\mathrm{bT}} = \mathrm{FP}^{A}_{m\text{-T}}$

where $\mathrm{P}^{A}_{m\text{-tt}}$ is the set of languages computable by polynomial-time Turing machines that make m nonadaptive queries to A; $\mathrm{P}^{A}_{\mathrm{btt}} = \bigcup_m \mathrm{P}^{A}_{m\text{-tt}}$; $\mathrm{P}^{A}_{m\text{-T}}$ and $\mathrm{P}^{A}_{\mathrm{bT}}$ are the analogous adaptive queries classes; and $\mathrm{FP}^{A}_{m\text{-tt}}$, $\mathrm{FP}^{A}_{\mathrm{btt}}$, $\mathrm{FP}^{A}_{m\text{-T}}$, and $\mathrm{FP}^{A}_{\mathrm{bT}}$ in turn are the analogous function classes.

It was widely expected that these general results would extend to the remaining case — languages computed with nonadaptive queries — yet results remained elusive. The best known was that

$$\mathrm{P}^{A}_{2m\text{-tt}} = \mathrm{P}^{A}_{m\text{-tt}} \Rightarrow \mathrm{P}^{A}_{\mathrm{btt}} = \mathrm{P}^{A}_{m\text{-tt}}.$$

We disprove the conjecture. In fact,

$$\mathrm{P}^{A}_{\lfloor \frac{4}{3}m \rfloor\text{-tt}} = \mathrm{P}^{A}_{m\text{-tt}} \not\Rightarrow \mathrm{P}^{A}_{(\lfloor \frac{4}{3}m \rfloor+1)\text{-tt}} = \mathrm{P}^{A}_{\lfloor \frac{4}{3}m \rfloor\text{-tt}}.$$

Thus there is a $\mathrm{P}^{A}_{m\text{-tt}}$ hierarchy that contains a finite gap.

We also make progress on the 3-tt vs. 2-tt case:

$$\mathrm{P}^{A}_{3\text{-tt}} = \mathrm{P}^{A}_{2\text{-tt}} \Rightarrow \mathrm{P}^{A}_{\mathrm{btt}} \subseteq \mathrm{P}^{A}_{2\text{-tt}}/\mathrm{poly}.$$

A full paper is available from ftp://ftp.eccc.uni-trier.de/pub/eccc/reports/1998/TR98-026/index.html

Circuit Lower Bounds Collapse Relativized Complexity Classes

Richard Beigel, Dept. of Electrical Engineering and Computer Science, University of Illinois at Chicago, Rm. 1116, SEO Building, M/C 154, 851 S. Morgan St., Chicago, IL 60607-7053, USA (beigel@eecs.uic.edu)
Alexis Maciel, Dept. of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699-5815, USA (alexis@sun.mcs.clarkson.edu)

Abstract Number 98-6

Beigel, Buhrman and Fortnow (STOC 98) constructed an oracle A such that

$$\mathrm{P}^A = \oplus\mathrm{P}^A \text{ and } \mathrm{NP}^A = \mathrm{EXP}^A.$$

We construct an oracle B such that

$$\mathrm{P}^B = \mathrm{NP}^B \text{ and } \oplus\mathrm{P}^B = \mathrm{EXP}^B.$$

An interesting aspect of our result is that we obtain our relativized collapses as a consequence of a circuit *lower bound*. Unlike previous papers, which used lower bounds on the size of small-depth AND–OR circuits that compute parity in order to separate the complexity class $\oplus\mathrm{P}^B$ from PH^B, we use those lower bounds in order to collapse two pairs of complexity classes. (Of course, the now well-known oracle separation follows from our pair of collapses, by the time hierarchy theorem.)

Fortnow has pointed out that ours is the first oracle relative to which NP has small circuits but $\oplus$P does not, answering a question of Homer's.

A full paper is not yet available.

Optimal Propositional Proof Systems and Oracle-Relativized Logic

Shai Ben-David, Computer Science Dept., Technion, Haifa, Israel (shai@cs.technion.ac.il)
Anna Gringauze, IBM Haifa Research Laboratory, MATAM, Haifa, Israel, (vanna@cs.technion.ac.il)

Abstract Number 98-7

We introduce a purely combinatorial property of complexity classes - the notions of *slim* vs. *fat* classes. These notions partition the collection of all previously studied time-complexity classes into two complementary sets. We prove a 'dichotomy theorem' for complexity classes along the lines of this partition; Considering the issue of existence of optimal propositional proof systems, we show that for every slim class $\mathcal{F}$, $TallyCoN\mathcal{F} \subseteq N\mathcal{F}$ imply the existence of an optimal propositional proof system. On the other hand, we introduce a notion of a propositional proof system *relative to an oracle* and show that for every fat class there exists an oracle relative to which even the entailment '$N\mathcal{F} = \mathcal{F}$' $\Longrightarrow$ '*optimal propositional proof systems exist*' fails.

As the classes $\mathcal{P}$ (polynomial functions), $\mathcal{E}$ ($2^{O(n)}$ functions) and $\mathcal{EE}$ ($2^{O(2^n)}$ functions) are slim, this result includes all the previously known sufficiency conditions for the existence of optimal propositional proof systems.

On the other hand, the classes $\mathcal{EXP}$, $\mathcal{QP}$ (the class of quasi-polynomial functions) and $\mathcal{EEE}$ ($2^{O(2^{2^n})}$ functions), as well as any other natural time-complexity class which is not covered by our sufficiency result, are fat classes.

As the proofs of all the known sufficiency conditions for the existence of optimal propositional proof systems carry over to the corresponding oracle-relativized notions, our oracle result shows that no extension of our sufficiency condition to non-slim classes can be obtained by the type of reasoning used so far in proofs on these issues.

A full paper is available by email to shai@syseng.anu.edu.au

Delayed Binary Search, or Playing Twenty Questions with a Procrastinator[1]

Abstract Number 98-8

Abstract

We investigate the classic monotone search problem with various amounts of delay between query and answer. For delay 1, for example, an optimal algorithm takes $\log_\varphi(n)$ time steps, where φ is the Golden Mean, the real positive root of x^2-1-1. We also describe and analyze optimal algorithms for delays 2 and higher, using the technique of "off-line dynamic programming", and compare the present problem with previous work on searching for the maximum of a unimodal function.

[1] with apologies to Dagat, Gacs, & Winkler

One-sided Versus Two-sided Randomness

Harry Buhrman, CWI. PO Box 94079, 1090 GB Amsterdam, The Netherlands. E-mail: buhrman@cwi.nl

Lance Fortnow, Department of Computer Science, University of Chicago, 1100 E. 58th St., Chicago, IL 60637. Email: fortnow@cs.uchicago.edu.

Abstract Number 98-9

Andreev, Clementi and Rolim show how given access to a quick hitting set generator, one can approximate the size of easily describable sets. As an immediate consequence one gets that if quick hitting set generators exist then P = BPP. Andreev, Clementi, Rolim and Trevisan simplify the proof and apply the result to simulating BPP with weak random sources.

Much earlier, Lautemann gave a proof that BPP is in Σ_2^p, simplifying work of Gács and Sipser. Lautemann's proof uses two simple applications of the probabilistic method to get the existence results needed. As often with the case of the probabilistic method, the proof actually shows that the overwhelming number of possibilities fulfill the needed requirements. With this observation, we show that Lautemann's proof puts BPP in the class $\mathrm{RP}^{PromiseRP[1]}$. Since quick hitting set generators derandomize PromiseRP problems, we get the existence of quick hitting set generators implies P = BPP. This greatly simplifies the proofs of Andreev, Clementi and Rolim and Andreev, Clementi, Rolim and Trevisan.

The difference between RP and PromiseRP is subtle but important. In the class RP we require the probabilistic Turing machine to either reject or accept with probability at least one-half for all inputs. In PromiseRP we only need to solve instances where the machine rejects or accepts with probability at least one-half.

A survey paper by Clementi, Rolim and Trevisan asks whether we can remove the promise in our result, i.e., whether BPP is in RP^{RP}. We give a relativized counterexample to this conjecture by exhibiting and oracle A such that $\mathrm{P}^A = \mathrm{RP}^A$ but $\mathrm{P}^A \neq \mathrm{BPP}^A$. Since virtually all the techniques used in derandomization relativize, this means that new techniques will be required to collapse BPP in this way.

An extended abstract is available from http://www.cs.uchicago.edu/~fortnow/papers.

Quantum NP is Hard for PH

Stephen Fenner, Computer Science Department, University of Southern Maine, Portland, ME 04103, (fenner@cs.usm.maine.edu)

Fred Green, Department of Mathematics and Computer Science, Clark University, Worcester, MA 01610, (fgreen@black.clarku.edu)

Steve Homer, Computer Science Department, Boston University, Boston, MA 02215, (homer@cs.bu.edu)

Randy Pruim, Department of Mathematics, Calvin College, Grand Rapids, MI 49546, (rpruim@calvin.edu)

Abstract Number 98-10

In analogy with NP, a language L is in the class NQP if the strings in L are exactly those accepted by a polynomial-time quantum machine with non-zero probability. Previously, Adleman, Demarrais and Huang showed that NQP $\subseteq$PP. The sharper upper bound NQP $\subseteq$ co-$C_=$P is implicit in Adleman et al. and results of Fortnow and Rogers. Here we prove that in fact NQP and co-$C_=$P are the same class, by showing that determining whether a quantum computation has a non-zero probability of accepting is hard for co-$C_=$P. This implies that NQP is hard for the polynomial-time hierarchy. The hardness result also applies to determining whether a given quantum basis state appears with nonzero amplitude in a superposition, or whether a given quantum bit has positive expectation value at the end of a quantum computation.

A full paper is available by email to homer@cs.bu.edu.

Justification of Rissanen's Approximate Formula for Prior Probability

Francisco Fernandez, Vladik Kreinovich, and Luc Longpré, Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968, USA, (ffernand@cs.utep.edu, vladik@cs.utep.edu, longpre@cs.utep.edu)

Abstract Number 98-11

At any given moment of time, several models are consistent with all available data. A statistically correct way to estimate the probabilities of different models is to fix some *prior* probability distribution $p_0(n)$, and to use Bayes formula to transform the prior probability distribution to the desired (*posterior*) distribution.

Ideally, as the amount of data increases, the posterior probability should tend to the actual probability distribution. In this ideal case, in the large-time scale, the choice of the prior probability is not very important. However, if we choose a prior probability in which the probability of a certain event is 0, this probability will remain 0 no matter what we observe, and so if the actual probability of this event is not 0, we will never be able to approximate this actual probability. So, it is desirable to select a prior probability distribution which has as few measure-0 sets as possible. In other words, we would like to choose a prior distribution which is *universal* in some reasonable sense (e.g., if an event has a 0 probability in this prior distribution, it has to have 0 probability in any other reasonable distribution as well). It is well known that although there exists no computable universal probability measure, there does exist a universal *semi*-computable (semi-)measure. Since this universal distribution is not computable, for practical applications, we need a computable approximation. Rissanen has shown that reasonably good applications stem from the choice of $p_0(n) = \text{const}/(n \cdot \log^2(n))$. We explain why.

In decision making, every event E is characterized by its probability $p(E)$, and every outcome a by its *utility* $u(a)$, so that the expected utility of each decision is equal to $p(E_1) \cdot u(a_1) + \ldots + u(E_n) \cdot u(a_n)$. If we select a level u_0 below which outcomes can be neglected, then we can characterize each probability p by the smallest utility $u = u_0/p$ for which this outcome is not negligible. In these terms, we must explain the utility $u(n) = \text{const} \cdot n \cdot \log^2(n)$. Since the utility value depends on u_0, we must select not a single function, but a family $\{C \cdot u(n)\}$. If we assume that this family is scale-invariant, then we get $u(n) = n^\alpha$. Out of families with $\sum p_0(n) = 1$, we must choose the one which is the slowest to get to 0, and there are none (the closer α to 1, the slower). However, if we consider 2D families $\{C_1 \cdot u_1(n) + C_2 \cdot u_2(n)\}$, then the slowest family exists, and the corresponding $p_0(n)$ is described exactly by Rissanen's formula.

This work was supported in part by NASA under cooperative agreement NCC5-209. A full paper will be available shortly.

Relativized Worlds with an Infinite Hierarchy

Lance Fortnow, Department of Computer Science, University of Chicago, 1100 E. 58th St., Chicago, IL 60637. Email: fortnow@cs.uchicago.edu.

Abstract Number 98-12

We introduce the "Book Trick" as a method for using results about random oracles to create relativized worlds with an infinite polynomial-time hierarchy. We use it to show, for example, a single relativized world where

- One can find satisfying assignments to satisfiable formulae with only nonadaptive queries to an NP oracle,
- the counting class SPP strictly contains the polynomial-time hierarchy,
- NP does not have measure zero in EXP and
- the polynomial-time hierarchy is infinite.

An preliminary paper is available at http://www.cs.uchicago.edu/~fortnow/papers.

The Communication Complexity of Enumeration

William I. Gasarch, Department of Computer Science, University of Maryland, College Park, MD 20742, (gasarch@cs.umd.edu)

Abstract Number 98-13

Let $g : \{0,1\}^n \times \{0,1\}^n \to Z$. Communication complexity asks how many bits need to be exchanged between Alice (who holds x) and Bob (who holds y) to compute $g(x,y)$. We look at the number of bits needed to be exchanged to compute *a set of possiblities* for $g(x,y)$, one of which is correct.

Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and $f^k(x_1, x_2 \dots x_k, y_1, y_2 \dots y_k) = f(x_1,y_1) \dots f(x_k,y_k)$ (all the x_i and y_i are of length n). Alice is given $x_1, \dots, x_k$ and Bob is given $y_1, \dots, y_k$. Note that there are 2^k possiblities for $f^k(x,y)$.

f^k is (b,e)-*comm-enumerable* if there is a b-bit protocol that, on input (x,y), determines a set of e strings such that one of them is $f^k(x,y)$.

1. $EQ(x,y)$ is 1 if $x = y$ and 0 otherwise.
2. The function $IP : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined by

$$IP(x_1 \dots x_n, y_1 \dots y_n) = \sum_{i=1}^{n} x_i y_i \pmod 2.$$

3. We can view $x \in \{0,1\}^n$ as a subset of $\{1, \dots, n\}$ represented by a bit vector. With this in mind the function $DISJ : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined by

$$DISJ(x,y) = \begin{cases} 1 & \text{if } x \cap y = \emptyset; \\ 0 & \text{if } x \cap y \neq \emptyset. \end{cases}$$

We have shown the following:

1. EQ^k is (clearly) $(n+1, 2^{k-1})$-enumerable, but is not $(n, 2^k - 1)$-enumerable.
2. For $f = IP$ or $f = DISJ$, for $k << n$, f^k is (clearly) $(n+1, 2^{k-1})$-enumerable, however there exists a constant c such that f^k is not $(cn, 2^k - 1)$-enumerable.

A full paper is available by email to gasarch@cs.umd.edu

Discontinuities in Recurrent Neural Networks

Ricard Gavaldà, Departament L.S.I., Universitat Politècnica de Catalunya, Jordi Girona 1–3, 08034 Barcelona, SPAIN, (gavalda@lsi.upc.es)
Hava T. Siegelmann, Faculty of Industrial Engineering and Management, Technion, Haifa 32000, ISRAEL. (iehava@ie.technion.il)

Abstract Number 98-14

This paper studies the computational power of various discontinuous real computational models that are based on the classical analog recurrent neural network (ARNN). This ARNN consists of finite number of neurons; each neuron computes a polynomial net function and a sigmoid-like continuous activation function.
We introduce "arithmetic networks" as ARNN augmented with a few simple discontinuous (e.g., threshold or zero test) neurons. We argue that even with weights restricted to polynomial-time computable reals, arithmetic networks are able to compute arbitrarily complex recursive functions. We identify many types of neural networks that are at least as powerful as arithmetic nets, some of which are not in fact discontinuous but they boost other arithmetic operations in the net function, e.g. neurons that can use divisions and polynomial net functions inside sigmoid-like continuous activation functions. These arithmetic networks are equivalent to the Blum-Shub-Smale (BSS) model, when the latter is restricted to a bounded number of registers.
With respect to implementation on digital computers, we show that arithmetic networks with rational weights can be simulated with exponential precision; but even with polynomial-time computable real weights arithmetic networks are not subject to any fixed precision bounds. This is in contrast with the ARNN that are known to demand only precision that is linear in the computation time.
When nontrivial periodic functions (e.g. fractional part, sine, tangent) are added to arithmetic networks, the resulting networks are computationally equivalent to a massively parallel machine. Thus, these highly discontinuous networks can solve the presumably intractable class of PSPACE-complete problems in polynomial time.

A full paper is available from http://www.lsi.upc.es/~gavalda/papers.html and by email to gavalda@lsi.upc.es

Tally NP Sets and Easy Census Functions

Judy Goldsmith, Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA (goldsmit@cs.engr.uky.edu)
Mitsunori Ogihara, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA (ogihara@cs.rochester.edu)
Jörg Rothe, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (rothe@informatik.uni-jena.de)

Abstract Number 98-15

We study the question of whether every P set has an easy (i.e., polynomial-time computable) census function. We characterize this question in terms of unlikely collapses of language and function classes such as $\#\mathrm{P}_1 \subseteq \mathrm{FP}$, where $\#\mathrm{P}_1$ is the class of functions that count the witnesses for tally NP sets. We prove that every $\#\mathrm{P}_1^{\mathrm{PH}}$ function can be computed in $\mathrm{FP}^{\#\mathrm{P}_1^{\#\mathrm{P}_1}}$. Consequently, every P set has an easy census function if and only if every set in the polynomial hierarchy does. We show that the assumption $\#\mathrm{P}_1 \subseteq \mathrm{FP}$ implies $\mathrm{P} = \mathrm{BPP}$ and $\mathrm{PH} \subseteq \mathrm{MOD}_k\mathrm{P}$ for each $k \geq 2$, which provides further evidence that not all sets in P have an easy census function. We also relate a set's property of having an easy census function to other well-studied properties of sets, such as rankability and scalability (the closure of the rankable sets under P-isomorphisms). Finally, we prove that it is no more likely that the census function of any set in P can be approximated (more precisely, can be n^α-enumerated in time n^β for fixed α and β) than that it can be precisely computed in polynomial time.

A full paper is available as UR-DCS-TR-98-684 at http://www.cs.rochester.edu/trs.

An Introduction to Query Order

Edith Hemaspaandra, Department of Mathematics, Le Moyne College, Syracuse, NY 13214, USA (edith@bamboo.lemoyne.edu)
Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA (lane@cs.rochester.edu)
Harald Hempel, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (hempel@informatik.uni-jena.de)

Abstract Number 98-16

Hemaspaandra, Hempel, and Wechsung ("Query Order," *SIAM Journal on Computing,* to appear; see also UR-CS-TR-95-596) raised the following questions in 1995: If one is allowed one question to each of two different information sources, does the order in which one asks the questions affect the class of problems that one can solve with the given access? If so, which order yields the greater computational power?
The answers to these questions have been learned—inasfar as they can be learned without resolving whether or not the polynomial hierarchy collapses—for both the polynomial hierarchy and the boolean hierarchy. In the polynomial hierarchy, query order never matters. In the boolean hierarchy, query order sometimes does not matter and, unless the polynomial hierarchy collapses, sometimes does matter. Furthermore, the study of query order has yielded dividends in seemingly unrelated areas, such as bottleneck computations and downward translation of equality.
In this article, we present some of the central results on query order. The article is written in such a way as to encourage the reader to try his or her own hand at proving some of these results. We also give literature pointers to the quickly growing set of related results and applications.

A full version appears in Eric Allender's Structural Complexity Column in the October 1997 issue of *BEATCS*, and can also be obtained online, at http://www.cs.rochester.edu/trs, as UR-CS-TR-97-665.

Downward Collapse from a Weaker Hypothesis

Edith Hemaspaandra, Department of Mathematics, Le Moyne College, Syracuse, NY 13214, USA (edith@bamboo.lemoyne.edu)
Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA (lane@cs.rochester.edu)
Harald Hempel, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (hempel@informatik.uni-jena.de)

Abstract Number 98-17

Hemaspaandra, Hemaspaandra, and Hempel ("Translating Equality Downwards," UR-CS-TR-98-657) proved that, for $m > 0$ and $0 < i < k - 1$: if $\Sigma_i^p \mathbf{\Delta} \mathrm{DIFF}_m(\Sigma_k^p)$ is closed under complementation, then $\mathrm{DIFF}_m(\Sigma_k^p) = \mathrm{coDIFF}_m(\Sigma_k^p)$. This sharply asymmetric result fails to apply to the case in which the hypothesis is weakened by allowing the Σ_i^p to be replaced by any class in its difference hierarchy. We so extend the result by proving that, for $s, m > 0$ and $0 < i < k - 1$: if $\mathrm{DIFF}_s(\Sigma_i^p) \mathbf{\Delta} \mathrm{DIFF}_m(\Sigma_k^p)$ is closed under complementation, then $\mathrm{DIFF}_m(\Sigma_k^p) = \mathrm{coDIFF}_m(\Sigma_k^p)$.

A full paper is available, at http://www.cs.rochester.edu/trs, as UR-CS-TR-98-681.

What's Up with Downward Collapse: Using the Easy-Hard Technique to Link Boolean and Polynomial Hierarchy Collapses

Edith Hemaspaandra, Department of Mathematics, Le Moyne College, Syracuse, NY 13214, USA (edith@bamboo.lemoyne.edu)
Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA (lane@cs.rochester.edu)
Harald Hempel, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (hempel@informatik.uni-jena.de)

Abstract Number 98-18

During the past decade, nine papers have obtained increasingly strong consequences from the assumption that boolean or bounded-query hierarchies collapse. The final four papers of this nine-paper progression actually achieve downward collapse—that is, they show that high-level collapses induce collapses at (what before-the-fact seemed to be) lower complexity levels. For example, for each $k > 1$ it is now known that if $\mathrm{P}^{\Sigma_k^p[1]} = \mathrm{P}^{\Sigma_k^p[2]}$ then $\mathrm{PH} = \Sigma_k^p$. This article surveys the history, the results, and the method—the so-called easy-hard technique—of this nine-paper progression.

1. J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. *SIAM Journal on Computing,* 17(6):1263-1282, 1988.
2. K. Wagner. Number-of-query hierarchies. Technical Report 158, Institut für Mathematik, Universität Augsburg, Augsburg, Germany, October 1987.
3. K. Wagner. Number-of-query hierarchies. Technical Report 4, Institut für Informatik, Universität Würzburg, Würzburg, Germany, February 1989.
4. R. Chang and J. Kadin. The boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing,* 25(2):340-354, 1996.
5. R. Beigel, R. Chang, and M. Ogiwara. A relationship between difference hierarchies and relativized polynomial hierarchies. *Math. Systems Theory,* 26(3):293-310, 1993.
6. E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. An upward separation in the polynomial hierarchy. TR-Math/Inf/96/15, FSU Jena, Germany, June 1996.
7. E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. A downward collapse within the polynomial hierarchy. *SIAM Journal on Computing,* to appear.
8. H. Buhrman and L. Fortnow. Two queries. In *Proc. of the 13th Annual IEEE Conference on Computational Complexity.* IEEE Computer Society Press, June 1998, to appear.
9. E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. Translating equality downwards. TR-657, Dept. of Comp. Sci., Univ. of Rochester, Rochester, NY, April 1997.

A full paper is available, at http://www.cs.rochester.edu/trs, as UR-CS-TR-98-682.

Optimal Separations for Parallel versus Sequential Self-Checking: Parallelism Can Exponentially Increase Self-Checking Cost

Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA (lane@cs.rochester.edu)
Harald Hempel, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (hempel@informatik.uni-jena.de)
Jörg Vogel, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (vogel@informatik.uni-jena.de)

Abstract Number 98-19

We provide optimal inclusions and separations between parallel and sequential self-checking, i.e., regarding the parallel and sequential reduction relationships between functions and their graphs. In particular, we show that there are functions for which parallel self-checking is exponentially more expensive than sequential self-checking. Prior to this work, it had not been established that parallel self-checking ever needed to be even one query more expensive than sequential self-checking.

A full paper is available, at http://www.cs.rochester.edu/trs, as UR-CS-TR-98-691.

Linear-Nondeterminism, Linear-Sized, Karp-Lipton Advice for the P-Selective Sets

Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA. Email: `lane@cs.rochester.edu`.
Christopher Nasipak, University of Rochester, Rochester, NY 14627, USA. Email: `cn004f@uhura.cc.rochester.edu`.
Keith Parkins, University of Rochester, Rochester, NY 14627, USA. Email: `kp004f@uhura.cc.rochester.edu`.

Abstract Number 98-20

Hemaspaandra and Torenvliet showed that each P-selective set can be accepted by a polynomial-time nondeterministic machine using linear advice and quasilinear nondeterminism. We show that each P-selective set can be accepted by a polynomial-time nondeterministic machine using linear advice and linear nondeterminism.

A full paper is available as UR-DCS-TR-97-667 at `http://www.cs.rochester.edu/trs`.

A Second Step Towards Circuit Complexity-Theoretic Analogs of Rice's Theorem

Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA (lane@cs.rochester.edu)
Jörg Rothe, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (rothe@informatik.uni-jena.de)

Abstract Number 98-21

Rice's Theorem states that every nontrivial language property of the recursively enumerable sets is undecidable. Borchert and Stephan initiated the search for circuit complexity-theoretic analogs of Rice's Theorem. In particular, they proved that every nontrivial counting property of circuits is UP-hard, and that a number of closely related problems are SPP-hard.

The present paper studies whether their UP-hardness result itself can be improved to SPP-hardness. We show that their UP-hardness result cannot be strengthened to SPP-hardness unless unlikely complexity class containments hold. Nonetheless, we prove that every P-constructibly bi-infinite counting property of circuits is SPP-hard. We also raise their general lower bound from unambiguous nondeterminism to constant-ambiguity nondeterminism.

A full paper is available as UR-DCS-TR-98-662 at http://www.cs.rochester.edu/trs.

Creating Strong Total Commutative Associative One-Way Functions from Any One-Way Function

Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA (lane@cs.rochester.edu)
Jörg Rothe, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (rothe@informatik.uni-jena.de)

Abstract Number 98-22

Rabi and Sherman presented novel digital signature and unauthenticated secret-key agreement protocols, developed by themselves and by Rivest and Sherman. These protocols use "strong," total, commutative (in the case of multi-party secret-key agreement), associative one-way functions as their key building blocks. Though Rabi and Sherman did prove that associative one-way functions exist if P $\neq$ NP, they left as an open question whether any natural complexity-theoretic assumption is sufficient to ensure the existence of "strong," total, commutative, associative one-way functions. In this paper, we prove that if P $\neq$ NP then "strong," total, commutative, associative one-way functions exist.

A full paper is available as UR-DCS-TR-98-688 at http://www.cs.rochester.edu/trs.

On Characterizations of the Basic Feasible Functionals, Part I

Robert J. Irwin, Dept. of Elec. Eng. and Computer Science, Syracuse University, Syracuse, NY 13244 USA, (rjirwin@top.cis.syr.edu)
Bruce M. Kapron, Dept. of Computer Science, University of Victoria, Victoria, BC V8W 3P6 CANADA, (bmkapron@maclure.csc.uvic.ca)
James S. Royer, Dept. of Elec. Eng. and Computer Science, Syracuse University, Syracuse, NY 13244 USA, (royer@top.cis.syr.edu)

Abstract Number 98-23

The type-2 Basic Feasible Functionals (abbreviated, BFF_2) have proven to be the most robust of the various proposals for a natural type-2 analog of the polynomial-time computable functions. One problem with BFF_2 is that all of characterizations of it seem to have *ad hoc features. In this paper we introduce a typed programming formalism, type-2 inflationary tiered loop programs or* ITLP_2*, that characterizes* BFF_2*.* ITLP_2 *is based on the type-theoretic characterizations of polynomial-time of Bellantoni and Cook and of Leivant. While the Bellantoni-Cook and Leivant systems are strictly predicative, it seems necessary to partially break predicativity under our general approach. Also, in contrast to the prior programming formalisms characterizing* BFF_2*, our formalism is very close in spirit to Kapron and Cook's machine-based characterization of* BFF_2*.*

A full paper is available via: ftp://top.cis.syr.edu/users/royer/cbff1.ps

Improved Time and Space Hierarchies of One-Tape Off-Line TMs

Kazuo Iwama, Kyoto University, Kyoto 606-8501, Japan (iwama@kuis.kyoto-u.ac.jp)
Chuzo Iwamoto, Hiroshima University, Kagamiyama, Higashi-Hiroshima 739-8527, Japan (iwamoto@ke.sys.hiroshima-u.ac.jp)

Abstract Number 98-24

We present improved time and space hierarchies of one-tape off-line Turing Machines (TMs), which have a single worktape and a two-way input tape: (i) For any time-constructible functions $t_1(n)$ and $t_2(n)$ such that $\inf_{n\to\infty} \frac{t_1(n)\log\log t_1(n)}{t_2(n)} = 0$ and $t_1(n) = n^{O(1)}$, there is a language which can be accepted by a $t_2(n)$-time TM, but not by any $t_1(n)$-time TM. This result substantially improves Hartmanis and Stearns' $(\log t_1(n))$-gap which survived more than 30 years. (ii) For any space-constructible function $s(n)$ and positive constant ϵ, there is a language which can be accepted in space $s(n)+\log s(n)+(2+\epsilon)\log\log s(n)$ by a TM with two worktape-symbols, but not in space $s(n)$ by any TM with the same worktape-symbols. Thus the *additive* gap is enough to separate the space-complexity in the case of the same worktape-symbols, which also improves the previous result that needs the *multiplicative* $(1+\epsilon)$-gap for the separation.

An extended abstract will appear in the proceedings of MFCS'98. The paper is available by email to iwamoto@ke.sys.hiroshima-u.ac.jp.

Monotone Real and Non-Monotone Boolean Circuits Are Incomparable

Stasys Jukna, Universität Trier, Fachbereich Informatik, D-54286 Trier, Germany & Institute of Mathematics and Informatics, LT-2600 Vilnius, Lithuania (jukna@ti.uni-trier.de)

Abstract Number 98-25

The famous Razborov's $n^{\Omega(\log n)}$ lower bound on the size of any monotone Boolean circuit for the perfect matching problem established a super-polynomial gap between monotone and non-monotone complexity in the case of *Boolean* circuits. Tardos (1987) observed that for some other problems the gap is truly exponential. A graph function $\varphi(G)$ is *clique-like* if $\omega(G) \leq \varphi(G) \leq \chi(G)$, where $\omega(G)$ is the clique number and $\chi(G)$ is the chromatic number. For $2 \leq k < m$, let $T_\varphi(m,k)$ denote the monotone Boolean function on $n = \binom{m}{2}$ boolean variables encoding the edges of a graph on m vertices, which outputs 1 iff $\varphi(G) \geq k$. This function is monotone if the underlying graph function φ is such. Using Lovász-capacity of graphs, Tardos defined a graph function φ which is monotone and belongs to P; hence, the corresponding monotone Boolean function $T_\varphi(m,k)$ can be computed by a circuit over $\{\wedge,\vee,\neg\}$ of polynomial size. On the other hand, an improvement of Razborov's lower bound for Clique function given by Alon and Boppana (1987) implies that for *every* monotone clique-like function φ, the function $T_\varphi(m,k)$ requires monotone circuits over $\{\wedge,\vee\}$ of size exponential in $\Omega(\sqrt{k})$.

What happens if we allow non-decreasing *real-valued* functions $\phi : R^2 \to R$ as gates? It appears that the picture here is quite different. Rosenbloom [IPL, 61(3), 1997] observed that *every* slice function (this is a monotone Boolean function which is non-trivial only on one slice of the cube) can be computed by a monotone real circuit of linear size. On the other hand, an easy counting shows that *some* slice functions require non-monotone circuits over $\{\wedge,\vee,\neg\}$ of exponential size. Thus, *monotone real* circuits may be exponentially more powerful than non-monotone (!) Boolean circuits. We prove that for some functions, the same holds also in the opposite direction: though Tardos function $T_\varphi(m,k)$ has a non-monotone circuit over $\{\wedge,\vee,\neg\}$ of polynomial size, it cannot be computable by monotone circuits of size smaller than $\exp\left(\Omega(\sqrt{k}/d)\right)$ even if we allow *arbitrary* non-decreasing *real-valued* functions $\phi : R^d \to R$, as gates. Thus, unlike in the Boolean case, the power of monotone real and non-monotone Boolean circuits is *incomparable*.

This result is a part of a revised version of a bigger paper submitted to *Combinatorica* and will available shortly at http://www.informatik.uni-trier.de/~jukna/papers

Linear Codes Are Hard for Oblivious Read-Once Parity Branching Programs

Stasys Jukna, Universität Trier, Fachbereich Informatik, D-54286 Trier, Germany & Institute of Mathematics and Informatics, LT-2600 Vilnius, Lithuania. (jukna@ti.uni-trier.de)

Abstract Number 98-26

Interesting aspect of linear codes is that their characteristic functions appear to be hard for both known "reading-restricted" models of branching programs (b.p.): for syntactic deterministic (Okolnishnikova, 1991) and non-deterministic (Jukna, 1992) read-k times branching programs, where in every path every variable appears at most k times, and for $(1,+s)$-branching programs (Jukna & Razborov, 1996), where along every consistent path at most s variables are tested more than once. On the other hand, looking at parity-check matrix of C we see that each such function f_C is just an And of $m \leq n$ (negations of) parity functions $\bigoplus_{j\in S_i} x_j$ for particular subsets $S_1,\ldots,S_m \subseteq \{1,\ldots,n\}$. Because of their intimate relation to Parity, it is natural to ask if linear code functions can be computed more efficiently using the *parity accepting mode*. So far, this problem is open even for read-once parity branching programs (1-p.b.p.); such a program is a usual non-deterministic read-once b.p. which accepts an input iff the number of accepting paths is odd.

We answer this question negatively under additional restriction that branching programs are *oblivious*. Practical interested in this oblivious model (known in CAD community as a Parity-OBDD) is that it (just like its deterministic counterpart - OBDD) can be used for verification of chips. We prove the following theorem:

> Let $C \subseteq \{0,1\}^n$ be a linear code of minimal distance d_1, and let its dual $C^\perp$ has minimal distance d_2. If $d_1 \geq d_2$ then any oblivious 1-p.b.p. computing the characteristic function f_C of C has size at least 2^{d_2-1}.

The proof is very simple, and is based on the following property of linear codes: If C be a linear code with the minimal distance $d+1$ then its dual $C^\perp$ is d-universal, i.e. the projection of $C^\perp$ onto any set of d coordinates gives the whole d-cube. When applied to Reed-Muller code, the theorem immediately yields the lower bound $2^{\Omega(\sqrt{n})}$.

Draft version of this note is available by email to the author.

On Branching Programs With Bounded Uncertainty

Stasys Jukna, Universität Trier, Fachbereich Informatik, D-54286 Trier, Germany & Institute of Mathematics and Informatics, LT-2600 Vilnius, Lithuania (jukna@ti.uni-trier.de)
Stanislav Žák, Institute of Computer Science, Academy of Sciences, 182 00 Prague 8, Czech Republic (stan@uivt.cas.cz)

Abstract Number 98-27

We propose an information-theoretic approach to proving lower bounds on the size of branching programs (b.p.). The argument is based on Kraft-McMillan type inequalities for the average amount of uncertainty about (or entropy of) a given input during various stages of the computation.

The approach. In the program P computing a given Boolean function f we stop each computation in an appropriate node. For some nodes in P we obtain classes of inputs, the computations on which are stopped there. Using the properties of f we then try to show that, for each such class A, the average entropy $\mathcal{E}(A) = \frac{1}{|A|}\sum_{a\in A}\mathcal{E}(a)$ cannot be large. Kraft-McMillan type inequalities imply that $\log|A| \leq \mathcal{E}(A)$, and hence, any class A with a small average entropy must be also small. Hence, we must have many classes A, and therefore, we need many nodes in P.

The results. We first show that for read-once b.p. finding non-trivial upper bounds for the average entropy $\mathcal{E}(A)$ is an easy task. Looking for larger classes of b.p. where this task is still tractable, we define one general property of branching programs – the 'gentleness'. Roughly, a program P is gentle if at some of its nodes some large set of inputs is classified in a 'regular' manner, where the regularity requires that the uncertainty about the individual inputs at these nodes has some special *form*. We then prove the following:

1. Read-once branching programs are gentle.

2. Explicit functions, which are hard for *all* previously considered restricted models of b.p.(such as the characteristic functions of linear codes), can be easily computed by small gentle b.p.. This fact is not very surprising – it just indicates that 'gentleness' is a new type of restriction.

3. We isolate a new combinatorial property of Boolean functions – the 'strong stability', and (using the bounds on the average entropy $\mathcal{E}(A)$), prove that any such function requires gentle b.p. of exponential size. This criterion implies that some explicit Boolean functions – the Clique function and a particular Pointer function (which belongs to AC^0) – cannot be computed by gentle programs of polynomial size.

A full paper available at http://www.informatik.uni-trier.de/~jukna/papers
Extended abstract appears in Proceedings of ICALP'98

A Note on the Shortest Lattice Vector Problem

S. Ravi Kumar (ravi@almaden.ibm.com)
IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, USA.
D. Sivakumar (siva@cs.uh.edu)
Department of Computer Science, University of Houston, Houston, TX 77204, USA.

Abstract Number 98-28

In a recent breakthrough, Ajtai (STOC '98) showed that the problem of finding shortest vectors in rational lattices is NP-hard under randomized reductions. Ajtai further showed that the corresponding decision problem of whether a given lattice L has a non-zero vector of length less than a given value r is NP-complete under randomized reductions.

A remarkable feature of Ajtai's reduction from NP to (the decision version of) the shortest lattice vector problem is that it is not *parsimonious*, that is, it does not preserve the number of solutions to the search problems associated with the decision problems. Therefore it is not clear whether the structural complexity of (the decision version of) the shortest lattice vector problem is similar to that of standard NP-complete problems like *SAT*. Specifically, the following two questions arise:

(1) Does the the problem of deciding whether a given rational lattice L has a non-zero vector of length less than a given value r remain NP-complete under the *promise* that L has exactly zero or one non-zero vector of length less than r?
(2) Is the problem of counting the number of vectors in a given lattice L of length at most a given value r complete for the counting class #P?

In this note, we answer the first question raised above in the affirmative. That is, notwithstanding the fact that Ajtai's reduction is not parsimonious (and the possibility that a parsimonious reduction may not exist at all), the promise problem mentioned in (1) is still NP-complete under randomized reductions. Our proof is an encoding of the reduction of Valiant and Vazirani (from *SAT* to the promise version of *SAT*) into instances of the shortest lattice vector problem.

The paper is available at www.cs.uh.edu/~siva/papers.html, and also by email to the authors.

Computational Foundation of Quantum Randomness Hierarchy

Luc Longpré and Vladik Kreinovich, Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968, USA, (longpre@cs.utep.edu, vladik@cs.utep.edu)

Abstract Number 98-29

Quantum physics is one of the main areas of application for the complexity-based Kolmogorov-Martin-Löf formalizations of the notions of a random sequence and of a random element. The main reason for this application is that quantum physics only predicts the wave function $\psi(x)$ and hence the *probabilities* of different measurement results, and the only information about the actual sequence of measurement results is that this sequence must be random relative to the corresponding probability measure μ.

This brief description does not capture all the nuances of quantum physics. Namely, this description corresponds to the *first* (particle) quantization, when the wave function is described by the Schroedinger equation $\dot{\psi}(x,t) = H(\psi(x,t))$, where $\dot{\psi}$ denotes time derivative, and H is a known operator (e.g., $H = \Delta^2 + V(x)$, where $V(x)$ is a potential energy function). For a known (and computable) operator H, and with a computable $\psi(x,0)$, ψ is computable and hence, μ is computable. A more adequate description of quantum phenomena requires *second* quantization, in which, instead of assuming that we know exactly how the potential energy $V(x)$ depends on the particle locations, we assume that fields like $V(x)$ can have quantum fluctuations as well, i.e., that $V(x)$ is not a deterministically computable function, but a function which is random relative to some (computable) probability measure μ_1. Theoretical physicists discuss the possibility of *third* quantization, in which μ_1 is also not computable, but random relative to some other measure μ_2, etc.

We show that this hierarchy can be theoretically justified within the Kolmogorov-Martin-Löf definition of randomness. Indeed, according to one of the equivalent versions of this definition, if we are given a probability measure μ on the set of all binary sequences, then we can define a random sequence as a sequence which does not belong to any constructive set of μ-measure 0. We start with computable measures, i.e., measures for which the function $x_1 \ldots x_n \to \mu(x_1 \ldots x_n)$ describing the probability of a random sequence to start with $x_1 \ldots x_n$ is computable; sequences which are random relative to such measures will be assigned to level 1. Now, we can take measures for which the sequence $\mu(x_1 \ldots x_n)$ is a level 1 sequence, and assign sequences which are random relative to such measures to level 2, etc.

Our main result is that this hierarchy does not collapse, i.e., that on every level, there is a sequence which does not belong to the previous levels.

A full paper will be available shortly.

On the complexity of moving vertices in a graph

Antoni Lozano, Dept. L.S.I., Universitat Politècnica de Catalunya, Jordi Girona 1–3, Mòdul C6, 08034-Barcelona, Catalonia, EU. (antoni@lsi.upc.es)
Vijay Raghavan, CS Dept., Vanderbilt University, Nashville, TN 37235, USA. (raghavan@vuse.vanderbilt.edu)

Abstract Number 98-30

We consider the problem of deciding whether a given graph G has an automorphism which moves at least k vertices (where k is a function of $|V(G)|$), a question originally posed by Lubiw (1981). Here we show that this problem is equivalent to the one of deciding whether a graph has a nontrivial automorphism, for $k \in O(\log n / \log \log n)$.
It is commonly believed that deciding isomorphism between two graphs is strictly harder than deciding whether a graph has a nontrivial automorphism. Indeed, we show that an isomorphism oracle would improve the above result slightly—using such an oracle, one can decide whether there is an automorphism which moves at least k' vertices, where $k' \in O(\log n)$.
If P $\neq$ NP and Graph Isomorphism is not NP-complete, the above results are fairly tight, since it is known that deciding if there is an automorphism which moves at least n^ϵ vertices, for any fixed $\epsilon \in (0, 1)$, is NP-complete. In other words, a substantial improvement of our result would settle some fundamental open problems about Graph Isomorphism.

A full paper is available by email to antoni@lsi.upc.es

Some Results on Commutative Oracles

Timothy H. McNicholl, Department of Mathematics and Computer Science, Fisk University Nashville, Tennessee 37208, USA, (tmcnicho@dubois.fisk.edu)

Abstract Number 98-31

We study the following classes:

- $Q^*(r_1A_1, \ldots, r_kA_k)$ = the collection of all sets that can be computed by a Turing machine that on any input makes a total of r_i queries to A_i for all $i \in \{1, \ldots, k\}$.
- $Q(r_1A_1, \ldots, r_kA_k)$ which is defined like $Q^*(r_1A_1, \ldots, r_kA_k)$ except that queries to A_i must be made before queries to A_{i+1} for all $i \in \{1, \ldots, k-1\}$.
- $QC(r_1A_1, \ldots, r_kA_k)$ which is defined like $Q(r_1A_1, \ldots, r_kA_k)$ except that the Turing machine must halt even if given incorrect answers to some of its queries.

We show that if for each $i \in \{1, \ldots, k\}$ there exists j such that A_i is Σ_j-complete, then all three of these classes are identical and are not changed if we replace $(r_1A_1, \ldots, r_kA_k)$ with $(r_{\sigma(1)}A_{\sigma(1)}, \ldots, r_{\sigma(k)}A_{\sigma(k)})$ where σ is any permutation of $\{1, \ldots, k\}$. For this reason, we say that the Σ-complete sets *commute* with each other. We give general sufficient conditions for commutativity, and provide examples to show that these conditions can not be weakened. These conditions allow us to obtain examples of commutative sets that are not Σ-complete or Π-complete. We also explore commutativity in the hyperarithmetical hierarchy.

A full paper is available by email to tmcnicho@dubois.fisk.edu.

The Complexity of Computing Optimal Assignments of Generalized Propositional Formulae

Steffen Reith and Heribert Vollmer, Lehrstuhl für Theoretische Informatik, Universität Würzburg, Am Exerzierplatz 3, D-97072 Würzburg, GERMANY ({streit, vollmer}@informatik.uni-wuerzburg.de)

Abstract Number 98-32

We consider the problems of finding the lexicographically minimal (or maximal) satisfying assignment of propositional formulae for different restricted formula classes. We obtain two dichotomy theorems in the style of Schaefer: For all formula classes from our framework, the above problem is either polynomial time solvable or complete for OptP under metric reductions. We also consider the problem of deciding if in the optimal assignment the largest variable gets value 1. We show that this problem is either in P or P^{NP} complete under many-one reductions.

A full paper is available. Send e-mail to the authors.

BPP equals BPLINTIME under an Oracle

Robert Rettinger, (robert.rettinger@fernuni-hagen.de)
Rutger Verbeek, (verbeek@fernuni-hagen.de)
Department of computer science, FernUniversität Hagen
Feithstraße 140, D-58084 Hagen

Abstract Number 98-33

The problem of separating complexity classes of probabilistic machines with small error probability (Monte Carlo machines) came to our attention, when we unsuccessfully tried to derive separations for Monte Carlo space complexity classes from the existence of very slowly increasing Monte Carlo space constructible bounds [Karpinski and Verbeek 87] (cf. also [Freivalds 81]). The reason for the difficulties is the fact that the Monte Carlo and Las Vegas (zero error) classes are promise classes, i.e. no recursive enumerable representation of the appropriate machines is known. Therefore no hard sets of low complexity are known for complexity classes and the standard separation techniques (diagonalisation or recursive padding) are not applicable; the best known separation is obtained via deterministic simulation and padding ([Allender, Beigel, Hertrampf and Homer 93]).
For promise classes there are some relativization results, e.g. the existence of universal sets is oracle-dependent (c.f. [Sipser 82], [Hartmanis and Hemachandra 86], [Hemachandra 88], [Bovet, Crescenzi and Silvestri 92] and [Hemaspaandra, Jain and Vereshchagin 93]).
In 1989 Fortnow and Sipser claimed the existence of an oracle, under which Las Vegas linear time equals Monte Carlo polynomial time. We unsuccessfully tried to fix the proof (in an extensive email discussion with L. Fortnow between 1989 and 1992); in 1997 the paper was retracted by the authors, even the weaker version for truth-table oracle machines.
In our paper we re-establish a slightly weaker version of the claim of Fortnow and Sipser. While the original proof for truth-table oracles can be fixed with some smaller modifications, we can verify in the general case only $\mathrm{BPP}^B = \mathrm{BPTIME}^B(n)$ (not $\mathrm{BPP}^B = \mathrm{ZPTIME}^B(n)$). Even this construction requires new and very involved techniques. Moreover we extend the collapse to its theoretical limit: if f is a time constructible increasing function with $f^k(n) \in O(2^n)$, then $\mathrm{BPTIME}^B(n) = \mathrm{BPTIME}^B(f)$ under some recursive oracle B.

A full paper is available by email from the authors.

Immunity and Simplicity for Exact Counting and Other Counting Classes

Jörg Rothe, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany (rothe@informatik.uni-jena.de)

Abstract Number 98-34

Ko and Bruschi independently showed that, in some relativized world, PSPACE (in fact, $\oplus$P) contains a set that is immune to the polynomial hierarchy (PH). In this paper, we study and settle the question of (relativized) separations with immunity for PH and the counting classes PP, $C_=P$, and $\oplus$P in all possible pairwise combinations. Our main result is that there is an oracle A relative to which $C_=P$ contains a set that is immune to $BPP^{\oplus P}$. In particular, this $C_=P^A$ set is immune to PH^A and to $\oplus P^A$. Strengthening results of Torán and Green, we also show that, in suitable relativizations, NP contains a $C_=P$-immune set, and $\oplus$P contains a PP^{PH}-immune set. This implies the existence of a $C_=P^B$-simple set for some oracle B, which extends results of Balcázar et al., and provides the first example of a simple set in a class not known to be contained in PH. Our proof technique requires a circuit lower bound for "exact counting" that is derived from Razborov's circuit lower bound for majority.

A full paper is available as UR-DCS-TR-98-679 at http://www.cs.rochester.edu/trs.

On an optimal deterministic algorithm for SAT

Zenon Sadowski, Institute of Mathematics, University at Białystok, ul. Akademicka 2, 15-257 Białystok, POLAND, (sadowski@math.uw.bialystok.pl)

Abstract Number 98-35

A deterministic algorithm recognizing SAT is optimal if no other algorithm recognizing SAT has more than a polynomial speed-up over its running time. Two versions of optimality appear in Computational Complexity: Levin's optimality and Krajíček - Pudlák's optimality. In this paper we are mainly concerned with an optimal algorithm possessing Krajíček - Pudlák's optimality property. If the optimality property is stated only for any input string x which belongs to SAT and nothing is claimed for other x's, we name such an algorithm as an almost optimal deterministic algorithm for SAT.
J. Krajíček and P. Pudlák proved that an almost optimal deterministic algorithm for $TAUT$ exists if and only if there exists a p-optimal proof system for $TAUT$. In this paper we prove that an almost optimal deterministic algorithm for SAT exists if and only if there exists a p-optimal proof system for SAT. Combining Krajíček and Pudlák's result with our result we show that an optimal deterministic algorithm for SAT exists if and only if both p-optimal proof systems for $TAUT$ and for SAT exist.

Submitted to CSL'98. A full paper is available by email to sadowski@math.uw.bialystok.pl

Abstract Number 98-36

Graph Ramsey Theory and the Polynomial Hierarchy

by

Marcus Schaefer
Department of Computer Science
University of Chicago
1100 East 58th Street
Chicago, Illinois 60637, USA
schaefer@cs.uchicago.edu

Abstract

In the Ramsey Theory of graphs $F \to (G, H)$ means that for every coloring of F with colors red and blue there is either a red subgraph G or a blue subgraph H. The problem $\to$ of deciding whether $F \to (G, H)$ lies in coNP^{NP} and was shown to be coNP-hard by Burr [1]. As we prove in this paper $\to$ is actually coNP^{NP}-complete, simultaneously settling a conjecture of Burr and providing a rare natural example of a problem complete for a higher level of the polynomial hierarchy.

References

[1] Stefan A. Burr. On the Computational Complexity of Ramsey-Type Problems. In: Jaroslav Nešetřil, Vojtěch Rödl. *Mathematics of Ramsey Theory*, Springer, 1990.

Choosing a Physical Model: Why Symmetries?

Raúl A. Trejo, Vladik Kreinovich, and Luc Longpré, Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968, USA,
(rtrejo@cs.utep.edu, vladik@cs.utep.edu, longpre@cs.utep.edu)

Abstract Number 98-37

One of the main applications of Kolmogorov complexity ideas to data processing is via the Minimum Description Length principle. According to this principle, if several different models (or theories) are consistent with the same observations, then we should choose a model with the shortest description, i.e., crudely speaking, a model with the smallest value of Kolmogorov complexity. This principle is in perfect agreement with the Occam principle (it actually formalizes Occam's principle), and it has been successfully applied to various problems.

In particular, it has been successfully used in physics, where Occam's principle originated and where it has been successfully used. In modern fundamental physics, however, symmetry groups play such an important role that often, physicists choose not the simplest model, but the model which corresponds to the simplest symmetry group. At first glance, this restriction to *symmetry*-defined models seem to prevent us from considering possible simple *non-group* models, and thus, make this symmetry-group version of Occam principle worse than the unrestricted one. However, the success of this direction in theoretical physics seems to indicate that this restriction does not bring any disadvantage at all. Our analysis shows that this restriction is indeed non-essential.

To formalize the physicists' idea, we define a *symmetry* as a program which transforms strings into strings and which is a bijection (1-1 and onto). By a *complexity* of a symmetry s, we mean the length $\mathrm{len}(s)$ of this program. We say that a symmetry s *defines a string* x *uniquely* if $s(x) = x$ and $s(y) \neq y$ for all $y \neq x$. Now, for every string x, we can define its *group-symmetric complexity* $K_{\mathrm{sym}}(x)$ as the smallest complexity of a symmetry which determines x uniquely. Our main result is that $K(x) \preceq K_{\mathrm{sym}}(x)$ (where $\preceq$, as usual, means equality modulo an additive constant).

Thus, K_{sym} is asymptotically equivalent to the usual Kolmogorov complexity and hence, choosing a model for which $K_{\mathrm{sym}}(x)$ is the smallest is asymptotically equivalent to choosing the simplest model.

This work was supported in part by NASA under cooperative agreement NCC5-209. A full paper will be available shortly.

Extractors for the real world

Kun Xue and *Marius Zimand*,
School of Computer and Applied Sciences, Georgia Southwestern State University.
(kx@canes.gsw.peachnet.edu, zimand@gswrs6k1.gsw.peachnet.edu)

Abstract Number 98-38

An extractor is a combinatorial object that is used to improve the quality of a source of randomness. More precisely, an (n, k, d, m, ϵ) extractor is a bipartite regular multigraph $G = (V_{left}, V_{right}, E)$, where V_{left} has 2^n nodes identified with the strings of length n, V_{right} has the 2^m nodes identified with the strings of length m, and with the degree of each node in V_{left} equal to 2^d; it has the property that if x is chosen randomly in V_{left} according to a distribution D with min-entropy at least k, and if y is chosen uniformly at random among the 2^d edges outgoing from x, the distribution of the string $(E(x, y), y)$ is ϵ-close to the uniform distribution. The best results have been established by Zuckerman and Ta-Shma. In the case in which $k = \Omega(n)$, Zuckerman has constructed an extractor with $d = O(\log(n) + log(\epsilon^{-1}))$ and $m = \Omega(n)$. Ta-Shma's extractor works for any k (of course, $k \leq n$) and has $d = poly(log(n) + \log(\epsilon^{-1}))$ and $m = k$. These results are impressive and they have lead to breaktrough results in some of the applications of extractors. On the other hand, these extractors are the result of a sophisticated amalgamation of some more basic extractors and of some other combinatorial objects and, as a consequence, the hidden constants appear to be very large. We investigate here constructions of extractors that are simple, efficient, easy to program, and whose parameters are good in a realistic setting (this meaning for reasonable values of n). We consider randomized and deterministic constructions. For example, we build a new family H of of hash funtions $h : \{0,1\}^n \to \{0,1\}^m$ having $\approx 2^{-m}$ collisions and $|h| = 2m \cdot (log(n/(2m)) + 1/2)$. By a variant of the leftover hash lemma, H can be viewed as an $(n, k, |h|, m, \approx \sqrt{2^{-m} + 2^{-(k-m)}})$ extractor. For realistic values of n, our construction needs fewer random bits than an extractor of Srinivasan and Zuckerman, which has been used as a starting point in both Ta-Shma's and Zuckerman's constructions. Our method is easy to program and outperforms by far the method of Srinivasan and Zuckerman. Its description fits into a few lines: (i) in a first step, the input a of length n is viewed as a polynomial p_a of $s = \log(n/(2m))$ variables $y_1, y_2, \ldots, y_s$ and we calculate $p_a(y_1, \ldots, y_s)$ in the finite field $GF(2^{2m})$; (ii) in the second step, the output of the first step is viewed as a linear function $l_{c,d}(z) = cz + d$ in $GF(2^m)$ and the hash function provides the value of z. Thus $h(a) = l_{p_a(y_1,\ldots,y_s)}(z)$, where $h = (y_1, \ldots, y_s, z)$. The core of the paper has an experimental nature and highlights the practicality of our method. A full paper will soon be available at http://gswrs6k1.gsw.peachnet.edu/~zimand/home.html