Structures Abstracts 1992. Vol II

Abstract

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

List of Titles of Abstracts

This is a list of the titles of the abstracts in the order they appear. The order alphabetical by the author.

A Simplified Approach to Prob. Poly-Time Complexity Classes
Quantity vs. Quality: Power of Oracle Access in Rel. Poly Time Hier.
Multiplication and Division can be computed in Depth 3
Robust Algorithms with Bounded Query
Computing Aribitrary Symmetric Functions
Completeness and Limited Nondeterminism
New Classes of Counting Functions
On the Computational Complexity of Inferring Evolutionary Trees

Queries are Easier Than You Thought (Probably)

Serge Abiteboul, INRIA, BP 105, 78153 Le Chesnay CEDEX, France
abitebou@inria.inria.fr
Kevin Compton, EECS Dept., University of Michigan, Ann Arbor, MI 48109-2122, USA
kjc@eecs.umich.edu
Victor Vianu, CSE C-0114, U.C. San Diego, La Jolla, CA 92093-0114, USA
vianu@cs.ucsd.edu

The optimization of a large class of queries is explored, using a powerful normal form recently proven. The queries include the *fixpoint* and *while* queries, and an extension of *while* with arithmetic. The optimization method is evaluated using a probabilistic analysis. In particular, the average complexity of *fixpoint* and *while* is considered and some surprising results are obtained. They suggest that the worst-case complexity is sometimes overly pessimistic for such queries, whose average complexity is often much more reasonable than the provably rare worst case. Some computational properties of queries are also investigated. A probabilistic notion of *boundedness* is defined, and it is shown that all programs in the class considered are bounded almost everywhere. An effective way of using this fact is provided.

This paper appeared in PODS 92. A full paper is not yet available.

Computing With Infinitary Logic

Serge Abiteboul, INRIA, BP 105, 78153 Le Chesnay CEDEX, France
abitebou@inria.inria.fr
Moshe Vardi, IBM Almaden Research Center, San Jose, CA 95120-6099, USA
vardi@almaden.ibm.com
Victor Vianu, CSE C-0114, U.C. San Diego, La Jolla, CA 92093-0114, USA
vianu@cs.ucsd.edu

Most recursive extensions of relational calculus converge around two central classes of queries: *fixpoint* and *while*. Infinitary logic (with finitely many variables) is a very powerful extension of these languages which provides an elegant unifying formalism for a wide variety of query languages. However, neither the syntax nor the semantics of infinitary logic are effective, and its connection to practical query languages has been largely unexplored. We relate infinitary logic to another powerful extension of *fixpoint* and *while*, called *relational machine*, which highlights the computational style of these languages. Relational machines capture the kind of computation occurring when a query language is embedded in a host programming language, as in C+SQL. The main result of this paper is that relational machines correspond to the natural effective fragment of infinitary logic. Other well-known query languages are related to infinitary logic using syntactic restrictions formulated in language-theoretic terms. For example, it is shown that *while* corresponds to infinitary logic formulas which can be described by a regular language. As a side effect to these results, we obtain interesting normal forms for infinitary logic formulas.

This paper is to appear in *Proc. Int'l. Conf. On Database Theory*, Berlin, 1992. A full paper is not yet available.

Generic Computation and Its Complexity

Serge Abiteboul, INRIA, BP 105, 78153 Le Chesnay CEDEX, France
abitebou@inria.inria.fr
Victor Vianu, CSE C-0114, U.C. San Diego, La Jolla, CA 92093-0114, USA
vianu@cs.ucsd.edu

There is a fundamental mismatch between the hardness of database queries and their Turing complexity. For example, the *even* query on a set has low Turing complexity but is by all accounts a hard query. The mismatch is due to the abstract, *generic* nature of database computation: data items which are indistinguishable by logical properties are treated uniformly. The issues specific to generic computation are obscured in the Turing model. Two models of generic computation are proposed. They are extensions of Turing Machines with a *relational store*. The machines differ in the interaction between the Turing component and the relational store. The first, simpler machine, allows limited communication with the relational store. However, it is not complete. Nonetheless, it subsumes many query languages which have emerged as central, including the *fixpoint* and *while* queries. We prove a normal form for the machine, which provides a key technical tool. The normal form specialized to the *while* queries allows resolving the open problem of the relation of *fixpoint* and *while*: they are equivalent iff PTIME = PSPACE. The second machine allows extended communication between the relational store and the Turing component, and is complete. The machine involves parallelism. Based on it, we define complexity measures for generic mappings. We focus on notions of polynomial time and space complexity for generic mappings, and show their robustness. In agreement with our intuition, we show that *even* is a hard query with respect to the new complexity measures. The results point to a trade-off between complexity and computing with an abstract interface.

This paper appeared in STOC 91. A full paper is to appear in the *JCSS* special issue on STOC 91 under the title *Computing with First-Order Logic.*

A uniform circuit lower bound for the permanent

Eric Allender Department of Computer Science, Rutgers University, New Brunswick, NJ, 08903.
Vivek Gore, Department of Computer Science, Rutgers University, New Brunswick, NJ, 08903.

We show that uniform families of ACC circuits of subexponential size cannot compute the permanent function. This also implies similar lower bounds for certain sets in PP. This is one of the very few examples of a lower bound in circuit complexity where the uniformity condition is essential; it is still unknown if there is any set in Ntime $(2^{n^{O(1)}})$ that does not have nonuniform ACC circuits.

A full paper is available.

Reductions to Sets of Low Information Content

V. Arvind, Department of Computer Science and Engineering, Indian Institute of Technology–Delhi, New Delhi 110016, India.
Y. Han, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA.
L. Hemachandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA.
J. Köbler, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany.
A. Lozano, Department of Software (L.S.I.), Universitat Politècnica de Catalunya, Pau Gargallo 5, E-08028 Barcelona, Spain.
M. Mundhenk, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany.
M. Ogiwara, Department of Computer Science and Information Mathematics, University of Electro-Communications, 1-5-1, Chofugaoka, Chofu-shi, Tokyo 182, Japan.
U. Schöning, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany.
R. Silvestri, Dipartimento di Scienze dell'Informazione, Università degli Studi di Roma "La Sapienza," 00198 Rome, Italy.
T. Thierauf, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany.

This paper studies the complexity of sets that reduce to sparse sets (and tally sets), and the complexity of the simplest sparse sets to which such sets reduce. We show even with respect to very flexible reductions that NP cannot have sparse hard sets unless P = NP. We also show that any set A that reduces to some sparse set (via various types of reductions) in fact reduces by the same type of reduction to a sparse set that is simple relative to A. We give a complete characterization of the sets of low instance complexity in terms of reductions to tally sets; it follows that if P $\neq$ NP, then no set of low instance complexity can be complete for NP with respect to disjunctive reductions or conjunctive reductions. Finally, we provide a refutation of hopes that the exciting recent work of Ogiwara and Watanabe can be extended to show robustly that sparse $\omega(\log n)$-tt complete sets for NP imply that the boolean hierarchy collapses.
A extended abstract will appear in ICALP '92 and a full version version is available. If you'd like a copy of it, or would like to have your name added to the author list, please send your address—and, in the case of becoming a coauthor, your check or money order—to lane@cs.rochester.edu.

Bounded truth-table and conjunctive reductions to sparse and tally sets

V. Arvind, Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi 110016, India,
J. Köbler and M. Mundhenk, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany. (email: mundhenk@informatik.uni-ulm.de)

In this paper we study the consequences of the existence of sparse hard sets for different complexity classes under certain types of deterministic, randomized and nondeterministic reductions. For example, we prove that if NP $\subseteq R^p_{btt}(R^p_{ctt}(\text{SPARSE}))$ then P = NP.
In the case of randomized reductions to sparse sets we show that if NP $\subseteq R^p_{btt}(R^{co\text{-}rp}_m(R^p_{ctt}(\text{SPARSE})))$ then RP = NP. Relatedly, we show that if some solution of the promise problem (1SAT,SAT) is in $R^p_{btt}(R^p_{ctt}(\text{SPARSE}))$ then there is a solution of (1SAT,SAT) in P. We also prove that the conclusion RP = NP can be derived from the assumption that some solution of the promise problem (1SAT,SAT) is in $R^p_{btt}(R^{co\text{-}rp}_m(R^p_{ctt}(\text{SPARSE})))$.
Finally, we consider nondeterministic reductions and show that if a coNP-complete set can be reduced via a nondeterministic many-one reduction to a co-sparse set then the polynomial hierarchy collapses to Θ^p_2. We also prove that if $\Sigma^p_2 \subseteq R^p_{btt}(R^{np}_m(\text{coSPARSE}))$ then the polynomial hierarchy collapses to Δ^p_2. On the other hand we show that nondeterministic many-one reductions to sparse sets surprisingly turn out to be as powerful as nondeterministic Turing reductions to sparse sets.

A full version is available as Technical Report "Ulmer Informatik-Berichte, 92-01" from Universität Ulm.

Sparseness, Simplicity, and Lowness

V. Arvind, Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi 110016, India,
J. Köbler and M. Mundhenk, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany.(email: mundhenk@informatik.uni-ulm.de)

This paper mainly deals with locating several reduction classes to sparse and tally sets in the extended low hierarchy. The lowness proofs follow a certain approach: for a set A that reduces to a sparse set via some type of reduction, we first show that A in fact reduces via the same type of reduction to a sparse set S whose complexity is relatively simple compared with A. Using the simplicity result and the sparseness of S we carry out an oracle replacement similar to Mahaney's enumeration technique or Kadin's census technique to derive the lowness result.
As main results we locate $R^p_{hd}(R^p_{ctt}(\text{SPARSE}))$ and $R^p_{dtt}(\text{SPARSE})$ in EL^{Θ}_3, where $R^p_{hd}(\mathcal{C})$ denotes the closure of $\mathcal{C}$ under Hausdorff reductions. Since $R^p_{hd}(R^p_{ctt}(\text{SPARSE})) \supseteq R^p_{btt}(R^p_{ctt}(\text{SPARSE}))$, the open problem of locating the closure of sparse sets under bounded truth-table reductions optimally in the extended low hierarchy is solved.
A summary of the results is given in the table below. The first row for example states that for every set $A \in R^p_{hd}(R^p_{ctt}(\text{SPARSE}))$ there is a sparse set $S \in \text{NP}^A$ such that $A \in R^p_{hd}(R^p_{ctt}(S))$, and, as a consequence, that $A \in EL^{\Theta}_3$. The third row means that for every set $A \in R^p_{ctt}(\text{TALLY}) \cap R^p_{dtt}(\text{TALLY})$ (which equals IC[log,poly]) there is a tally set $T \in \text{P}^{\text{SAT}\oplus A}$ such that $A \in R^p_{ctt}(T) \cap R^p_{dtt}(T)$, implying that $A \in EL^{\Sigma}_1$.

reduction class	simplicity	lowness
$A \in R^p_{hd}(R^p_{ctt}(\text{SPARSE}))$	NP^A	EL^{Θ}_3
$A \in R^p_{dtt}(\text{SPARSE})$	P^{NP^A}-printable	EL^{Θ}_3
$A \in R^p_{ctt}(\text{TALLY}) \cap R^p_{dtt}(\text{TALLY})$	$\text{P}^{\text{SAT}\oplus A}$	EL^{Σ}_1
$A \in R^{np}_{ctt}(\text{TALLY}) \cap R^{co\text{-}np}_{dtt}(\text{TALLY})$	$\text{P}^{\text{SAT}\oplus A}$	EL^{Σ}_1
$A \in R^{co\text{-}np}_m(\text{SPARSE}) \cap \text{NP}^{\text{NP}\cap\text{SPARSE}}$	$R^{np}_m(A)$	$\Theta^p_2(A) \subseteq \Theta^p_2$
$A \in R^p_{hd}(R^{co\text{-}np}_m(\text{SPARSE}))$	$\text{NP}^{\text{SAT}\oplus A}$	$\Theta^p_3(A) \subseteq \Theta^p_2(\Sigma^p_2 \oplus A)$

A full version is available as Technical Report "Ulmer Informatik-Bericht, 92-04 " from Universität Ulm.

On conjunctive, randomized and nondeterministic reductions to sets of high information content

V. Arvind, Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi 110016, India,
J. Köbler and M. Mundhenk, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany.(email: mundhenk@informatik.uni-ulm.de)

We show that if a set A in ESPACE reduces to a set of high information content (in the sense of Book and Lutz, see this Structures proceedings) via a polynomial time conjunctive [disjunctive, randomized many-one, or nondeterministic many-one, respectively] reduction then A in fact reduces via the same type of reduction to some sparse set (in some cases to a co-sparse set). As a consequence, the existence of a hard set (under conjunctive or disjunctive polynomial time reductions) of high information content for several complexity classes contained in ESPACE (like NP, $\oplus$P, or PP) implies that the class is contained in P. We derive weaker conclusions (e.g. NP=RP or PH= Θ_2^p) in the case of randomized and nondeterministic reductions.

A full paper is under preparation.

Probabilistic ACC Circuits with an Exact-Threshold Gate*

Richard Beigel † Jun Tarui ‡ Seinosuke Toda §

Building on Yao's work [2], Beigel and Tarui [1] showed that every Boolean function family in ACC can be computed by depth-2 circuits with a symmetric gate at the root and $n^{\log^{O(1)} n}$ AND gates of fan-in $\log^{O(1)} n$ at the next level. In [1], the class of Boolean function families computable by such circuits was called SYMMC. It is not hard to see that every Boolean function family in SYMMC can be computed by depth-3 size-$n^{\log^{O(1)} n}$ threshold circuits of a simple form: OR gate at the root, exact-threshold gates at the next level, and AND gates of fan-in $\log^{O(1)} n$ at the third level. (An exact-threshold gate outputs 1 if exactly k of its inputs are 1, where k is a parameter; it outputs 0 otherwise.)

At present, proving that some "natural" function is outside SYMMC is an important open problem in circuit complexity. Towards this end, it is interesting to consider augmenting the power of ACC circuits by allowing randomness or a less restricted type of gate at the root. Beigel and Tarui [1] proved that even if we allow a symmetric gate at the root, but do not allow randomness, that kind of circuit computes only Boolean functions in SYMMC. In this paper, we allow an exact-threshold gate at the root *and we allow randomness*; we prove that this kind of augmented ACC circuit still computes only Boolean functions in SYMMC.

Our proofs use new techniques concerning low-degree polynomials, and also yield the following new results for counting classes:

$$\mathrm{BP} \cdot \mathrm{C}_{=} \cdot \mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}[1]} \quad (\text{thus } \mathrm{C}_{=}\mathrm{P}^{\mathrm{PH}} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}); \quad \text{and} \quad \mathrm{BP}\#\mathrm{PH} \subseteq \mathrm{FP}^{\#\mathrm{P}}.$$

References.

[1] R. Beigel and J. Tarui. On ACC. In *Proc. 32nd FOCS*, pages 783–792. 1991.

[2] A. Yao. On ACC and threshold circuits. In *Proc. 31st FOCS*, pages 619–627. 1990.

*An extended abstract is available from the authors.

†Dept. of Computer Science, Yale University, 51 Prospect St., P.O. Box 2158, Yale Station, New Haven, CT 06520-2158, USA. Supported in part by NSF grant CCR-8958528. beigel-richard@cs.yale.edu

‡Dept. of Computer Science, University of Warwick, Coventry, CV4 7AL, United Kingdom. jun@dcs.warwick.ac.uk, Partially supported by the ESPRIT II BRA Programme of the EC under contract # 7141 (ALCOM II). Part of the work was done while the author was a student at University of Rochester.

§Dept. of Computer Science and Information Mathematics, University of Electro-Communications, 1-5-1 Chofu-gaoka, Chofu-shi, Tokyo, 182, Japan. toda@pspace.cs.uec.ac.jp

SPARSE reduces conjunctively to TALLY

Harry Buhrman, University of Amsterdam, Computer Science Department, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands. buhrman@fwi.uva.nl,
Luc Longpré, Northeastern University, College of Computer Science, 161CN, Boston, MA 02215, luc@corwin.ccs.northeastern.edu,
Edith Spaan, University of Amsterdam, Computer Science Department, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands. edith@fwi.uva.nl

We show how to conjunctively reduce any sparse set to a tally set. This is in contrast with disjunctive reductions, where there are sparse sets that are not disjunctively reducible to any tally set.
Let R_r(SPARSE) and R_r(TALLY) be the classes of sets that are $\leq_r$-reducible to SPARSE and TALLY sets, respectively. Then, the result stated above is equivalent to the following:

Result 1 $R_{ctt}(SPARSE) \subseteq R_{ctt}(TALLY)$.

Using this result we are able to refute one of Ko's conjectures and prove:

Result 2 $R_{bdtt}(SPARSE) \subseteq R_{ctt}(SPARSE)$.

Extending a technique from Gavaldà and Watanabe we prove:

Result 3 $R_{f(n)\text{-}ctt}(TALLY) \not\subseteq R_{dtt}(SPARSE)$.

Using again the first result we get the following:

Corollary 1 *For all polynomial time computable unbounded functions* f, $R_{f(n)\text{-}dtt}(TALLY) \not\subseteq R_{ctt}(SPARSE)$.

These theorems enable us to further understand the structure of the classes of sets reducible to Sparse sets. We show for example that several of them are not closed under complementation. A full discussion can be found in the paper.
A Technical Report is available on request.

Promise Problems and Fault-Tolerant Access to Unambiguous Computation

Jin-yi Cai, Dept. of Computer Science, Princeton University, Princeton, NJ, 08544, USA, *Lane A. Hemachandra* and *Jozef Vyskŏc,* Dept. of Computer Science, University of Rochester, Rochester, NY, 14627, USA.

This paper studies the power of three types of access to unambiguous computation: nonadaptive access, fault-tolerant access, and guarded access. (1) Though for NP it is known that nonadaptive access has exponentially terse adaptive simulations, we show that UP has no such relativizable simulations: there are worlds in which $(k+1)$-truth-table access to UP is not subsumed by k-Turing access to UP. (2) Though fault-tolerant access (i.e., "1-helping" access) to NP is known to be no more powerful than NP itself, we give both structural and relativized evidence that fault tolerant access to UP suffices to recognize even sets beyond UP. Furthermore, we completely characterize, in terms of locally positive reductions, the sets that fault-tolerantly reduce to UP. (3) In guarded access, Grollmann and Selman's natural notion of access to unambiguous computation, a deterministic polynomial-time Turing machine asks questions to a nondeterministic polynomial-time Turing machine in such a way that the nondeterministic machine never accepts ambiguously. In contrast to guarded access, the standard notion of access to unambiguous computation is that of access to a set that is *uniformly* unambiguous—even for queries that it never will be asked by its questioner, it must be unambiguous. We show that these notions, though the same for nonadaptive reductions, differ for Turing and strong nondeterministic reductions.
An extended abstract will appear in MFCS '92; a full paper is available from the authors (please send requests to lane@cs.rochester.edu).

An Alternate proof of Toda's theorem

Suresh Chari and *Pankaj Rohatgi,*
Department of Computer Science, Cornell University, Ithaca, NY, 14853, USA.

One of the seminal results in structural complexity theory is Toda's theorem which states that every language in the Polynomial Hierarchy randomly reduces to a language in $\oplus$P (ParityP). A crucial requirement of Toda's proof is an efficient randomized procedure for isolating an odd sized subset from the set of all possible witnesses for an NP computation (and the successful existential moves at higher levels of the Hierarchy). All known proofs use the reduction of Valiant and Vazirani, which isolates a unique element from a given initial set of witnesses, to accomplish this. We show that this reduction and simple generalizations of it inherently isolate a unique witness *i.e.*, for certain inputs, these reductions isolate an odd number of witnesses only by isolating a single witness. We give the following simple and intuitive procedure to directly isolate an odd number of witnesses:

> Let S be the set of satisfying assignments of a boolean formula F(or the succesful existenial moves at higher levels of the Hierarchy). Let h be a randomly chosen hash function and r a randomly chosen point in $\{0,1\}^n$. Then the set
>
> $$S' = \{x \in S \mid h(x) < r\}$$
>
> is odd sized with high probability.

The hash functions are randomly chosen from a $\frac{n}{\log(n)}$−universal family of hash functions.
A full paper is available by email to `chari@cs.cornell.edu`.

Random Walks in Colored Graphs

Anne Condon, Computer Science Department, University of Wisconsin, Madison, WI 53706,
Diane Hernek, Computer Science Division, University of California, Berkeley, CA 94720.

We introduce the notion of a random walk in an undirected graph with colored edges, and analyze the expected cover time of such walks. An infinite sequence $C = C_1C_2C_3\ldots$ of colors defines a *random walk* on a colored graph from a fixed start node s, where at the i-th step, an edge, chosen randomly and uniformly from the edges of color C_i at the current node, is followed.

We say that G can be *covered from* s if, on every infinite sequence C of colors, a random walk on C starting at s visits every node with probability one. The *expected cover time* of G is defined to be the supremum, over all infinite sequences of colors C, and nodes s, of the expected time to visit all nodes of G on C, starting at s.

In this paper we consider colored graphs that can be covered from every node and obtain the following results about their expected cover time.

1. Exponential upper and lower bounds on the expected time to cover undirected graphs with two colors.
2. Double-exponential upper and lower bounds on the expected time to cover undirected graphs with three or more colors.
3. Polynomial bounds on the expected cover time of colored graphs when the underlying graphs are aperiodic and have the same stationary distribution.
4. Polynomial bounds on the expected time to cover colored graphs on sequences of the form $(C_1C_2\ldots C_l)^\omega$ when the corresponding matrix product $C_1C_2\cdots C_l$ is irreducible and all entries of its stationary distribution are at least $1/\text{poly}(n)$.

We also characterize the expected cover time of a graph on a random color sequence. We describe applications to understanding the eigenvectors of products and weighted averages of matrices. Our results also introduce a complexity theory perspective to problems on time inhomogeneous Markov chains.

A draft of the full paper is available from the authors.

Some New Aspects of Parameterized Complexity

Rod Downey, Victoria University, P.O. Box 600, Wellington, New Zealand, and Cornell University, Ithaca, NY 14853
Mike Fellows, University of Victoria, Victoria, British Columbia, Canada.

With several other coauthors, we have explored various aspects of complexity theory through parameterized eyes in the setting introduced in for instance [DF1]. In [ADF], Karl Abrahamson and the authors extended the arena to parameterized $PSPACE$. The appropriate notion seems to be games and their winning strategies. The key class we call AW^* and is an analogue of $QBFSAT$. We show that while some problems such as parameterized alternating hitting set are fixed parameter tractable the analogue of geography (is there a winning strategy in k moves) is AW^* complete. In [ADF] several other concrete completeness results can be found for $W[P]$. In [FK], the second author and N. Koblitz showed that parameterization can yield nice insights into crypto. In particular, assuming a widely held conjecture on the distribution of smooth numbers, they show that the question which, when given n, asks if it has a factor $\leq k\log n$, is randomized f.p. tractable. This has implications for the design of chips for the practical inplementation of crypto based on discrete factorization. Together with Patrica Evans, the authors in [DEF] have applied the ideas to Angluin style exact learning. The observation is that while classical reductions do not preserve learnability with a little modification, the reductions we use in our earlier papers do preserve learnability. This allows for a technique of 'concurrent' learning where for instance one learns a t-CNF formula in strips by weight. Using this technique we can show that all weft one formulae are polynomially learnable, and that if if there is a protocol to learn monotone CNF using only equivalence queries then all CNF is learnable. Finally in [DF2], and [CD], the authors and Peter Cholak explore structural aspects of the degree structures generated by the reducibilities. In [DF3], basic structural aspects are analysed and for instance density akin to Ladner's theorem, is proven for the most uniform reducibility. This uses a nontrivial priority argument using ideas from the so-called tree of strategies method from classical recursion theory. In [CD], very complex $\mathbf{0}''$, $\mathbf{0}'''$, and $\mathbf{0}^{(4)}$ priority techniques combined with speedup type ideas are used to establish the undecidability of certain of the m-degree structures. Such complexity is needed since the reductions are intrinsically more complex than even those used in clasical recursion theory.

Various papers are available from Downey (downey@math.cornell.edu) or Fellows
(mfellows@csr.uvic.ca).

[ADF] Abrahamson, K., R. Downey, and M. Fellows, *Fixed Parameter Intractability*

II, to appear
[CD] Cholak, P. and R. Downey, *Undecidability and Definability for Parameterized Polynomial Time m-Reducibilities,* to appear.
[DEF] Downey, R., P. Evans, and M. Fellows, *in preparation.*
[DF1] Downey, R. and M. Fellows, *Fixed Parameter Intractability*, These Proceedings
[DF2] Downey, R. and M. Fellows, *Fixed Parameter Tractability and Completeness III: Some Structural Aspects of the W-Hierarchy,* to appear.
[FK] Fellows, M. and N. Koblitz, *in preparation.*

A Simple Proof that Connectivity Separates Monadic NP from Monadic co-NP

Ronald Fagin
Larry Stockmeyer
Moshe Vardi
IBM Research Division, Almaden Research Center, 650 Harry Rd., San Jose, CA 95120

The *computational complexity* of a problem is the amount of resources, such as time or space, required by a machine that solves the problem. Complexity theory traditionally has focused on the computational complexity of problems. A more recent branch of complexity theory focuses on the *descriptive complexity* of problems, which is the complexity of describing problems in some logical formalism. An ongoing development is the discovery of intimate connections between computational and descriptive complexity. In particular, in the early 1970's, Fagin showed that the complexity class NP coincides with the class of properties of finite structures expressible in existential second-order logic (where we are allowed to existentially quantify over not just points, as in first-order logic, but also over relations). A consequence of this result is that NP = co-NP if and only if existential and universal second-order logic have the same expressive power over finite structures. This equivalence of questions in computational and descriptive complexity holds the promise that techniques from one domain could be brought to bear on questions in the other domain.

One way of attacking these difficult questions is to restrict the classes under consideration, such as considering the monadic restriction of these logics, i.e., the restriction obtained by allowing second-order quantification only over sets (as opposed to quantification over, say, binary relations). This restriction gives us the classes "monadic NP" and "monadic co-NP". This line of attack was pursued by Fagin in early papers, where he separated monadic NP from monadic co-NP. Specifically, he showed that connectivity of finite graphs is not in monadic NP (although it is easy to see that it is in monadic co-NP). Fagin's proof involved showing that a player has a winning strategy in certain 2-person games on graphs. The strategy, however, is very complicated, which made the proof very difficult.

The new work gives a simple proof of this result. It is accomplished using two tools: a recent new approach to second-order Ehrenfeucht-Fraïssé games by Ajtai and Fagin, and an old but relatively unknown technique of Hanf for showing that two (infinite) structures agree on all first-order sentences, under certain conditions. In particular, we give a version of Hanf's result which is better suited for use as a tool in inexpressibility proofs for classes of finite structures. Possibly these techniques will be useful in other applications.

A full paper is available.

An Oracle Relative to which the Isomorphism Conjecture Holds

Steve Fenner, University of Southern Maine, 96 Falmouth St., Portland, ME 04103.
Lance Fortnow, University of Chicago, 1100 E. 58th St., Chicago, IL 60637.
Stuart Kurtz, University of Chicago, 1100 E. 58th St., Chicago, IL 60637.

We introduce symmetric perfect generic sets. These sets vary from the usual generic sets by allowing limited infinite encoding into the oracle. We then show that the Berman-Hartmanis isomorphism conjecture holds relative to any sp-generic oracle, i.e., for any symmetric perfect generic set A, all $\mathbf{NP}^A$-complete sets are polynomial-time isomorphic relative to A. Prior to this work there were no known oracles relative to which the isomorphism conjecture held.

As part of our proof that the isomorphism conjecture holds relative to symmetric perfect generic sets we also show that $\mathbf{P}^A = \mathbf{FewP}^A$ for any symmetric perfect generic A.

A full paper is available from Lance Fortnow (fortnow@cs.uchicago.edu).

Gap-Definability as a Closure Property

Stephen Fenner, Department of Computer Science, University of Southern Maine, Portland, ME, 04103, USA,
Lance Fortnow, Department of Computer Science, University of Chicago, Chicago, IL, 60637, USA.
Lide Li, Department of Computer Science, University of Chicago, Chicago, IL, 60637, USA.

We are interested in studying certain closure properties of counting classes. The notion of gap-definability is useful in studying properties common to most of these classes. Fenner, Fortnow, and Kurtz showed that every countable class $\mathcal{C}$ of languages is contained in a unique minimum gap-definable class $\mathrm{GapCl}(\mathcal{C})$. We provide a simplified definition of the GapCl operator, and show that in most cases, gap-definability is a simple and very inclusive condition intimately related to closure under union and intersection with languages in the class **SPP**.

Theorem 1 *Let $\mathcal{C}$ be a countable class of languages containing $\emptyset$. The following are equivalent:*

1. *$\mathcal{C}$ is gap-definable.*
2. *For all $L_1, L_2 \in \mathcal{C}$ and disjoint $S_1, S_2 \in \mathbf{SPP}$, we have $(L_1 \cap S_1) \cup (L_2 \cap S_2) \in \mathcal{C}$.*

Theorem 1 implies that if $\{\emptyset, \Sigma^*\} \subseteq \mathcal{C}$ and $\mathcal{C}$ is closed under union and intersection, then $\mathcal{C}$ is gap-definable iff $\mathbf{SPP} \subseteq \mathcal{C}$. We also characterize gap-definability when $\mathcal{C}$ is closed under the $\mathbf{R}\cdot$ operator.

Proposition 2 *If $\mathcal{C}$ is gap-definable, then $\mathbf{BP}\cdot\mathcal{C}$ and $\mathbf{NP}\cdot\mathcal{C}$ are gap-definable. If $\mathcal{C}$ is also closed under join and $\{\emptyset, \Sigma^*\} \subseteq \mathcal{C}$, then $\mathbf{P}^{\mathcal{C}}$ and $\mathbf{NP}^{\mathcal{C}}$ are gap-definable.*

We now know that the following classes are gap-definable: $\mathbf{P}^{\#\mathbf{P}}$, $\mathbf{P}^{\#\mathbf{P}[1]}$, $\mathbf{BP}\cdot\oplus\mathbf{P}$, **MP**, **PSPACE**, **EXP**, and **NEXP**.
The GapCl operator can be characterized more simply as follows:

Theorem 3 *Let $\mathcal{C}$ be a countable class and L an arbitrary language. $L \in \mathrm{GapCl}(\mathcal{C})$ iff there exist $L_1, \ldots, L_k \in \mathcal{C}$ and $S_1, \ldots, S_k \in \mathbf{SPP}$ such that*

1. *$S_i \cap S_j = \emptyset$ for all $i \neq j$,*
2. *$\bigcup_{i=1}^{k} S_i = \Sigma^*$, and*
3. *$L = (L_1 \cap S_1) \cup \cdots \cup (L_k \cap S_k)$.*

We also show that the GapCl operator preserves closure under union and intersection, which together with Theorem 1 implies that if $\mathcal{C}$ is closed under union and intersection, then $\mathrm{GapCl}(\mathcal{C})$ is the closure under union and intersection of $\mathcal{C} \cup \mathbf{SPP}$. This then gives a simple characterization of $\mathrm{GapCl}(\mathbf{NP})$, $\mathrm{GapCl}(\mathbf{BPP})$, and many other classes. We show that the conditions we impose on $\mathcal{C}$, although weak, are nontrivial: there are gap-definable classes not closed under union, intersection, or complement. Finally, we show that if $\mathbf{SPP} \neq \mathbf{PP}$, then there is a maximal gap-definable proper subclass of **PP** which is closed under m-reductions. A full paper is not yet available.

Using Functions as Oracles

Stephen Fenner, Department of Computer Science, University of Southern Maine, Portland, ME, 04103, USA,
Steven Homer, Department of Computer Science, Boston University, Boston, MA, 02215, USA.

We are interested in classifying the complexity of functions and in particular in the difficulty of finding a witness to an **NP** computation. To this end we define a functional analog of the Boolean hierarchy. The levels of the hierarchy are defined by allowing Turing machines access to an oracle which is an **NPSV** function f. When a query x to f is made, the result is either the (unique) value $f(x)$ or a special symbol indicating x is not in the domain of f. We can now define our function hierarchy:

Definition: Let $r(n)$ be an integer function. $\mathbf{PF}^{\mathbf{NPSV}[r]}$ is the class of functions obtained by a polynomial time oracle Turing machine making at most r queries to an **NPSV** oracle. (As usual n denotes the length of the input to the computation.)
When $r(n) = k$, k fixed, we denote this class by $\mathbf{PF}^{\mathbf{NPSV}[k]}$.

Our first results relate our hierarchy to that which allows only **NP** sets as oracles.

Theorem 1:

- For all k, $\mathbf{PF}^{\mathbf{NPSV}[k]}$ is contained in $\mathbf{PF}^{\mathbf{NPSV}[\log]} \subseteq \mathbf{PF}_{tt}^{\mathbf{NP}}$.
- $\mathbf{PF}^{\mathbf{NPSV}} = \mathbf{PF}_{tt}^{\mathbf{NPSV}}$ iff $\mathbf{PF}^{\mathbf{NP}} = \mathbf{PF}_{tt}^{\mathbf{NP}}$ iff $\mathbf{P}^{\mathbf{NP}} = \mathbf{P}_{tt}^{\mathbf{NP}}$.

As is the case with the Boolean hierarchy, we next show that the collapse of our function hierarchy at finite levels implies the collapse of the **PH**.

Theorem 2: If $\mathbf{PF}_{tt}^{\mathbf{NPSV}[k]} = \mathbf{PF}_{tt}^{\mathbf{NPSV}[k+1]}$ then the Boolean hierarchy collapses to the $(k+1)^{st}$ level (and so the polynomial hierarchy collapses to the third level).

The next theorem relates classes which use adaptive and non-adaptive queries to **NPSV**. It implies a similar collapse as in Theorem 2 for classes using adaptive queries.

Theorem 3: For all k, $\mathbf{PF}^{\mathbf{NPSV}[k]} \subseteq \mathbf{PF}_{tt}^{\mathbf{NPSV}[2^k-1]} \subseteq \mathbf{PF}^{\mathbf{NPSV}[k+1]}$.

A consequence of these results is that if the (lexicographically) greatest satisfying assignment to a propositional formula can be found within any finite level of the function hierarchy then the hierarchy collapses to that level and so the **PH** collapses as well. More formally, let **maxsat** be the problem of computing the greatest satisfying assignment. Then, as **maxsat** is complete for $\mathbf{PF}^{\mathbf{NP}}$,

Theorem 4: If **maxsat** is in $\mathbf{PF}^{\mathbf{NPSV}[k]}$ then $\mathbf{PF}^{\mathbf{NPSV}[k]} = \mathbf{PF}^{\mathbf{NP}}$ (and so, by Theorem 3, the Boolean hierarchy collapses).

This work builds on the results of Beigel, Buss, Hay, Krentel, Selman and others. In the full paper we also show that decision problems using an **NPSV** oracle are (almost always) no stronger than using an **NP** oracle. Most of our results hold for computations using **NPMV** oracles as well. A full paper is not yet available.

Dense Properties and Generic Witnesses

James A. Foster, Department of Computer Science, University of Idaho, Moscow, ID, 83855, USA,
email: `foster@cs.uidaho.edu`

I have defined a framework for building generic sets, which are "typical" with respect to properties which can be enforced by a particular type of construction (delayed diagonalization arguments), of various strengths. Using this framework, I have begun unifying the current approaches to building generic sets into a single theory. I have applied this theory to disprove the generic oracle hypothesis, which says that generic sets will not mislead a computation into making erroneous conclusions when added to that computation—even for computationally very restricted types of generic sets.

I have also used this theory to develop a hierarchy of classes of generic sets whose properties reflect those of the polynomial hierarchy. The polynomial hierarchy is a mathematical structure which sits between problems which are feasible with respect to time and those which are feasible with respect to space. We know almost nothing about the structure of this hierarchy. For example, we do not know if there are problems which are feasible with respect to time, but which require an unacceptable amount of space. My generic hierarchy offers a new approach to solving this problem.

Full papers are available.

Alternating Time Versus Deterministic Time: A Separation

Sanjay Gupta, Department of Computer and Information Sciences, The Ohio State University, Columbus, OH 43210, USA

We show that for each time-constructible function $t(n)$, DTIME$(t(n)) \subset \Sigma_2(t(n))$; that is, the set of languages accepted by multi-tape deterministic Turing machines is *strictly* contained in the set of languages accepted by alternating Turing machines which make at most two alternations. This substantially refines the previously known separation, DTIME$(t(n)) \subset$ DSPACE$(t(n))$ proved by Hopcroft, Paul and Valiant.

An important corollary of our result is that one (or both) of the following statements have to be true: NTIME$(n) \neq$ co $-$ NTIME(n) or, for each time-constructible function $t(n)$, DTIME$(t(n)) \subset$ NTIME$(t(n))$. This shows that at least one of the two immediate generalizations of the result DTIME$(n) \subset$ NTIME(n) (proved by Paul, Pippenger, Szemeredi and Trotter) has to be true.

A full paper will be available soon as a technical report.

Banishing Robust Turing Completeness

Lane A. Hemachandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA.
Sanjay Jain, Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA.
Nikolai K. Vereshchagin, Department of Mathematical Logic, Moscow State University, Moscow, Russia 119899.

Complete languages have long been a useful tool in complexity theory. Much of our knowledge about NP comes from studying the NP-complete set SAT. Most common complexity classes—NP, coNP, PSPACE, etc.—have many-one complete sets that help us study them. Sipser noted, however, that some classes may lack complete sets. His paper sparked much research into which classes robustly (i.e., with respect to all oracles) possess complete languages, and what strengths of completeness results (e.g., many-one or Turing) can be obtained.

Sipser showed that R and NP$\cap$coNP do not robustly possess many-one complete languages. Hartmanis and Hemachandra showed that UP—unambiguous polynomial time—does not robustly possess many-one complete languages. Gurevich showed that NP$\cap$coNP has many-one complete languages if and only if it has Turing complete languages. Ambos-Spies's elegant generalization of this states that for any class $\mathcal{C}$ closed under Turing reductions, $\mathcal{C}$ has Turing complete sets if and only if $\mathcal{C}$ has many-one complete sets.

This paper proves that "promise classes" are so fragilely structured that they do not robustly possess Turing-hard sets even in classes far larger than themselves. In particular, this paper shows that FewP does not robustly possess Turing-hard sets for UP $\cap$ coUP and IP $\cap$ coIP does not robustly possess Turing-hard sets for ZPP. It follows that ZPP, R, coR, UP$\cap$coUP, UP, FewP$\cap$coFewP, FewP, and IP$\cap$coIP do not robustly possess Turing complete sets. This both resolves open questions of whether promise classes lacking robust downward closure under Turing reductions (e.g., R, UP, FewP) might robustly have Turing complete sets, and extends the range of classes known not to robustly contain many-one complete sets.

A Preliminary version of the paper is to appear in the *Proceedings of the Second Symposium on Logical Foundations of Computer Science, Tver, Russia, 1992.* A full paper is available from the authors.

Defying Upward and Downward Separation

Lane A. Hemachandra and *Sudhir K. Jha,* Dept. of Computer Science, University of Rochester, Rochester, NY, 14627, USA.

Upward and downward separation results link the collapse of small and large classes, and are a standard tool in complexity theory. We study the limitations of upward and downward separation.

We show that the exponential-time limited nondeterminism hierarchy does not robustly possess downward separation. We show that probabilistic classes do not robustly possess Hartmanis-Immerman-Sewelson [HIS85] upward separation. Though NP is known [HIS85] to robustly possess Hartmanis-Immerman-Sewelson upward separation, we show that NP does not robustly possess Hartmanis-Immerman-Sewelson upward separation with respect to strong (immunity) separation. On the other hand, we provide a structural sufficient condition for upward separation.

A full paper is available from the authors (please send requests to lane@cs.rochester.edu).

Easily Checked Self-Reducibility

Lane A. Hemachandra, Department of Computer Science, University of Rochester, Rochester, NY, 14627, USA,
Riccardo Silvestri, Dipartimento di Scienze dell'Informazione, Università degli Studi di Roma "La Sapienza," 00198 Rome, ITALY.

This paper explores two generalizations, within NP, of self-reducibility: kernel constructibility [AB83,AB89] and committability [BK91]. Informally stated, kernel constructible sets have self-reductions that are easy to check (though perhaps hard to compute), and committable sets are those sets for which the potential correctness of a partial proof of set membership can be checked via a query to the same set (that is, via a self-reduction). We study these two notions of self-reducibility on non-dense sets. We show that sparse kernel constructible sets are of low complexity, we extend previous results showing that sparse committable sets are of low complexity, and we provide structural evidence—of interest in its own right—namely that if all sparse disjunctively self-reducible sets are in P then FewP $\bigcap$ coFewP is not P-bi-immune—that our extension is unlikely to be further extended. We obtain density-based sufficient conditions for kernel-constructibility: sets whose complements are captured by non-dense sets are perforce kernel constructible. Using sparse languages and Kolmogorov complexity theory as tools, we argue that kernel constructibility is orthogonal to standard notions of complexity.
A full paper is available; please send requests to lane@cs.rochester.edu.

Taking it to the Limit: On Infinite Variants of NP-Complete Problems

Tirza Hirst and David Harel, Dept. of Applied Math. & Computer Science
The Weizmann Institute of Science, Rehovot 76100, Israel.

A significant amount of work has been carried out in recent years regarding the complexity of problems on infinite recursive graphs. The present work was motivated by trying to understand what makes some NP-complete problems highly undecidable, *viz,* Σ_1^1-complete, in the infinite case (the first example of such a problem seems to be Hamiltonicity; see Harel, STOC '91), while others (e.g., 3-colorability; Beigel and Gasarch, private communication) remain on low levels of the arithmetic hierarchy.

We first set up a general way to obtain infinite versions of NP maximization and minimization problems. Thus, if P is the problem that asks for a maximum by:

$$\max_S |\{w: A \models \phi(w, S)\}|,$$

where ϕ is first-order and A is a finite structure (say, a graph), we define P^∞ as the problem that asks whether there is an S such that the set $\{w: A^\infty \models \phi(w, S)\}$ is infinite, where A^∞ is an infinite recursive structure. Thus, for example, MAX CLIQUE becomes the question of whether a recursive graph contains an infinite clique.

We have been able to prove two results. One enables using knowledge about the finite case to yield implications to the infinite case, and the other, somewhat suprisingly, enables implications in the other direction. Moreover, taken together, the two results provide a method for proving (finitary) problems to be outside MAX NP, and hence outside MAX SNP too. This seems to be of some significance, given the renewed interest in these sets, that follows recent developments to the effect that, whereas all problems in MAX SNP are approximable in polynomial-time to within *some* constant, the ones that are hard for MAX SNP have no polynomial-time approximation scheme.

For the first result, we define a special kind of monotonic transformation between NP optimization problems, which we call an *M-reduction*. The details are omitted here, but the idea is, essentially, that (in a maximization problem) enriching the structure in one problem enriches it in the other, as well as making the objective functions grow. We prove that M-reductions between conventional finitary problems become Σ_1^1 reductions when "lifted up" to the infinite case. We have used this result to prove the Σ_1^1-completeness of many additional problems. Here is a partial list of problems whose infinite versions we now know to be Σ_1^1-complete: MAX CLIQUE, MAX INDP SET, MAX HAM PATH, MAX SUBGRAPH, MAX COMMON SUBSEQUENCE, MAX COLOR, MAX EXACT COVER BY PAIRS, MAX TILING.

The second result is that if $P \in$ MAX NP then $P^\infty \in \Pi_2^0$. This implies that the finitary version of any problem whose infinite version is higher than Π_2^0 must be outside MAX NP, and hence outside MAX SNP too. In particular, the problems listed above are not in these sets, since their infinite versions are outside the arithmetic hierarchy.

A full paper is in preparation.

On Malign Input Distributions for Algorithms

Kojiro Kobayashi, Department of Information Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, JAPAN.

A measure μ (that is, a function μ from $\{0,1\}^*$ to real numbers such that $\mu(x) \geq 0$ and $\mu(\{0,1\}^*) < \infty$) is said to be *malign* if it satisfies the condition: if an input x from $\{0,1\}^*$ is given to algorithms with the probability $\mu(x)/\mu(\{0,1\}^n)$ then the worst-case computation time $t_A^{\mathrm{wo}}(n)$ and the average-case computation time $t_A^{\mathrm{av},\mu}(n)$ of an algorithm A for inputs of length n are of the same order (that is, there is a constant $c(> 0)$ such that $t_A^{\mathrm{wo}}(n) \leq c\ t_A^{\mathrm{av},\mu}(n)$ for any n) for any algorithm A. Li and Vitányi found that measures that are known as *a priori measures* are malign.

We define a measure μ to be *strongly malign* if for any algorithms A, B there exists a constant $c(> 0)$ such that $\mu(\phi_A(x)) \geq c\ \mu(\phi_B^{-1}(x))$ for any x $(\in \{0,1\}^*)$such that $\phi_A(x)$ is defined. Here $\phi_A(x)$ denotes the partial recursive function from $\{0,1\}^*$ to $\{0,1\}^*$ that is computed by an algorithm A. We show that the following sequence of four classes of measures is strictly increasing with respect to class inclusion:

(1) the class of a priori measures,

(2) the class of semicomputable strongly malign measures,

(3) the class of strongly malign measures,

(4) the class of malign measures.

This result shows that "a priori"-ness and malignness are different in one strong sense.

We also show two results concerning a priori measures and Kolmogorov complexity. For a rational number p such that $0 < p < 1$, let $P_p(x)$ denote the probability that a universal prefix-free algorithm $\phi_{A_{\mathrm{U}}}$ outputs x $(\in \{0,1\}^*)$ when the input u $(\in \{0,1\}^*)$ to $\phi_{A_{\mathrm{U}}}$ is randomly selected with p as the probability that the symbol 1 is selected. It is known that $P(x) = P_{1/2}(x)$ is an a priori measure. We show that $P_p(x)$ is an a priori measure for each p. Let $H_p(x)$ denote the smallest of $|u|_p$ for u $(\in \{0,1\}^*)$ such that $\phi_{A_{\mathrm{U}}}$ outputs x when u is given to $\phi_{A_{\mathrm{U}}}$ as an input, where $|u|_p$ denotes the value $(-\log_2(1-p))$(number of 0's in u) + $(-\log_2 p)$(number of 1's in u). The value $H(x) = H_{1/2}(x)$ is known as the (prefix-free algorithm based) Kolmogorov complexity of x. We show that for each p there exists a constant c such that $|H(x) - H_p(x)| \leq c$ for any x.

A full paper is available.

P/*poly* is contained in EL_3^{Θ}

J. Köbler, Abteilung für Theoretische Informatik, Universität Ulm, Oberer Eselsberg, D-W-7900 Ulm, Germany.

We show that for every set $A \in \mathrm{P}/poly$ correct advice can be computed in $\mathrm{FP}(\mathrm{NP}(A) \oplus \Sigma_2^p)$. The proof builds on ideas of R. Gavalda who showed that correct advice is computable in $\mathrm{FP}(\mathrm{NP}(A) \oplus \Sigma_3^p)$. Furthermore, we locate $\mathrm{P}/poly$ in the EL_3^{Θ}-level of the extended low hierarchy. This is optimal since Allender and Hemachandra have shown that there exists a sparse set not contained in EL_2^{Σ}.

A full paper is under preparation.

Computational and Statistical Indistinguishabilities

Kaoru Kurosawa, Dept. of Electrical & Electronic Eng., Tokyo Institute of Technology, Tokyo 152, JAPAN,
Osamu Watanabe, Dept. of Computer Science, Tokyo Institute of Technology, Tokyo 152, JAPAN (watanabe@cs.titech.ac.jp).

It is proved that a pair of polynomially samplable distributions are statistically indistinguishable if and only if no polynomial size circuits relative to NP sets (nonuniform P^{NP}-distinguishers) can tell them apart. As one application of this observation, we classify "zero-knowledge" notions that are used for interactive proof systems.

An extended abstract, which was submitted to the third Int. Sympos. Algorithms and Computation (ISAAC'92), is available.

On Symmetry of Information and Polynomial Time Invertibility

Luc Longpré, College of Computer Science, Northeastern University, Boston, MA 02115, U.S.A. (luc@corwin.ccs.northeastern.edu),
Osamu Watanabe, Dept. of Computer Science, Tokyo Institute of Technology, Tokyo 152, JAPAN (watanabe@cs.titech.ac.jp).

Symmetry of information states that for two strings x and y, $\mathrm{K}(xy) = \mathrm{K}(x) + \mathrm{K}(y \mid x) \pm O(\log |xy|)$. We consider the statement of whether symmetry of information holds in a polynomial time bounded environment. Intuitively, this problem is related the complexity of inverting a polynomial time computable function. We give some evidence supporting this intuition, by proving the following relations: (1) If the polynomial time symmetry of information holds, then there is a polynomial time algorithm that computes the shortest description of a string for "almost all" strings. (2) If the polynomial time symmetry of information holds, then every polynomial time computable function is probabilistic polynomial time invertible for "almost all" strings in its domain. (3) If P = NP (i.e., every polynomial time computable function is polynomial time invertible), then the polynomial time symmetry of information holds.

An extended abstract, which was submitted to the third Int. Sympos. Algorithms and Computation (ISAAC'92), is available.

Abstract Degree Structures

William Mueller, Department of Mathematics and Computer Science, Mount Holyoke College, South Hadley, MA 01075, USA

We introduce an abstract notion of deterministic reducibility as a class of register machine programs, and then, on an arbitrary domain of functions, disscuss formation of the associated degree structure. We produce axioms on the reducibility and the domain which lead to each of the following structural properties in the degrees: existence of a least degree, existence of least upper bounds, existence of incomparable degrees, non-existence of minimal degrees, density, and the existence of minimal pairs. These axioms generalize the requirements of some of the better known techniques used in complexity theory and the degrees of unsolvability for demonstrating comparable structure.

A full paper is available.

Extensions on Set Bit Enumeration

J. Ramachandran, Department of Computer and Information Science, Ohio State University, Columbus, Ohio, 43210, USA.

An m-set-bit-enumerator for a function f is a source of information that, in response the query: "How many set bits ('1's) are there in the binary representation of $f(x)$?", computes a list of m numbers, one of which is the correct value. Our paper "Set Bit Enumeration is Hard", in the 1992 Structure in Complexity Theory conference proceedings, contains the proof of the following theorem about the class OptP.

> If polynomial time computable 2-set-bit-enumerators exist for all OptP functions, then OptP = PF.

This actually forms the base case for the following more general theorem, which weakens the above hypothesis, and extends the conclusion to all the levels, Σ_k^{MM}, of Krentel's polynomial time function hierarchy [note: n is the length of the input to a Σ_k^{MM} function].

> If polynomial time computable $\sqrt[3]{\log n}$-set-bit-enumerators exist for all Σ_k^{MM} functions, then Σ_k^{MM} = PF.

A tech report will be available soon.

Primality Testing is in $\bigoplus$P

Desh Ranjan, Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

PRIMES, the set of prime numbers, has fascinated mathematicians for centuries. In theoretical computer science it is widely believed that primality testing, although not quite polynomial time, is much simpler than the NP-complete problems. We provide further evidence for this by showing that PRIMES $\in \bigoplus$P. This is not known to be the case for any of the NP-complete problems. Although, by Toda's result that PH $\subseteq$ BP.$\bigoplus$P, $\bigoplus$P is quite powerful when allowed randomness, this result shows that it may be able to do some things without the power of randomness too.

A full paper is not yet available.

On the Power of One Bit of a #P Function

Kenneth W. Regan, Dept. of Computer Science, SUNY/Buffalo, Buffalo, NY, 14260, USA.
Thomas Schwentick, Fachbereich Informatik, Universität Mainz, Germany.

We introduce the class MP of languages L which can be solved in polynomial time with an oracle that returns one bit of a #P function value $f(x)$. That one bit suffices for any L in the polynomial hierarchy follows from the proof of S. Toda's theorem [FOCS 1989, SIAM J.C. 1991] that PH $\subseteq$ P$^{\#\mathrm{P}}$. Hence we have: PH $\subseteq$ BP[$\oplus$P] $\subseteq$ C$\oplus$P $\subseteq$ MP $\subseteq$ P$^{\#\mathrm{P}[1]}$. We show that the middle bit of $f(x)$ is as powerful as any other bit, and that a wide range of bits around the middle have the same power. By contrast, the $O(\log n)$-many least significant bits are equivalent to $\oplus$P [Beigel, Gill, and Hertrampf, STACS 1990], while we show that the $O(\log n)$-many most significant bits are equivalent to PP. Thus the bits at either end are probably weaker.

MP is interesting because it is a natural complexity class with complete problems at a level where most of our current knowledge of counting classes and corresponding circuit-theory problems runs out. We study the subclass *Amp*MP of languages whose MP representations can be "amplified." We show that BP[$\oplus$P] $\subseteq$ *Amp*MP, and that for reasonably well-behaved subclasses $\mathcal{C}$ of *Amp*MP, MP$^{\mathcal{C}}$ = MP. Hence BP[$\oplus$P] is low for MP, and if C$_=$P $\subseteq$ *Amp*MP, then the counting hierarchy collapses to MP. Finally, our work leads to a purely mathematical question about the size of integer-valued polynomials $p(x, y)$ which satisfy certain congruence relations.

A full paper is available. See also the paper on this subject by F. Green, J. Köbler, and J. Torán in the proceedings of this conference.

Linear Time and Memory Efficient Computation

Kenneth W. Regan, Dept. of Computer Science, SUNY/Buffalo, Buffalo, NY, 14260, USA.

B. Alpern, A. Aggarwal, A. Chandra, and M. Snir [STOC 1987] introduced a model for machines with *hierarchical memory*, meaning informally that access to higher-numbered memory registers takes more time than access to "low memory." Their model, called HMM, has a *memory cost function* μ as a parameter; the time charge for reading cell m is $\mu(m)$. The standard unit-cost RAM and Turing machine models of computation have $\mu(m) = 1$ for all m; whereas the *log-cost criterion* $\mu(m) = \log m$ and the "d-dimensional layout functions" $\mu_d(m) = m^{1/d}$ $(1 \leq d \leq 3)$ better reflect time for actual implementations on today's computers. A program is *memory-efficient* [Alpern, L. Carter, and E. Feig, FOCS 1990] if its running time under μ cost is within a constant factor of its time reckoned at unit cost. Intuitively, such a program makes good use of a high-speed memory cache.

Aggarwal, Chandra, and Snir [FOCS 1987] added a "block transfer" capability to the HMM, which can move the content of registers $[a_1 \ldots a_2]$ to locations $[b_1 \ldots b_2]$ and incur the μ-charge only for addressing the endpoints of the blocks, with unit cost for each item in the block. They give tight nonlinear bounds for sorting, matrix transpose, FFT graph computation, and other natural functions. However, this model still has the theoretical drawback that nonlinear time is already required just to read all of the input. Our *Block Machine* (BM) model allows the data in a block move to be run through any finite-state machine (figuratively, a "filter"). This small change makes a wide class of interesting functions computable in linear time, even under the stringent 1-D layout function $\mu_1(m) = m$, which yields a subclass of TM linear time that we call TLIN.

This paper shows that several formulations of the BM and variants of other models in the literature are all equivalent up to constant factors in μ-time (for $\mu = \mu_d$), stamping the BM as a fairly robust and natural model of computation. By a linear speed-up theorem for the BM, these become "real-time" simulations. We also show that like the RAM but unlike what is known for the multitape TM, the BM has a tight deterministic time hierarchy; the proof exploits a uniform way for a BM to solve the word problem for finite monoids. We show that several list-processing and proof-checking functions belong to TLIN. The BM seems to lie "just above" machine and circuit models for which nonlinear lower bounds have been proven; we show that several functions witnessing these results also belong to TLIN. For matrix transpose and sorting, the BM runs against what Aggarwal and J. Vitter call a "challenging open problem" [CACM 1988, end]. We discuss these problems, and the new criteria for analyzing algorithms which the BM raises, at the end of the paper.

This work was supported by NSF RIA Grant CCR-9011248. A full draft paper is available.

Complexity Models for Incremental Computation

S. Sairam, Jeffrey Scott Vitter, and *Roberto Tamassia,* Department of Computer Science, Brown University, Providence, RI, 02912-1910, USA,

We present a new complexity theoretic approach to incremental computation. We define complexity classes that capture the intuitive notion of incremental efficiency and study their relation to existing complexity classes. We show that all common LOGSPACE-complete problems for P are *incr*-POLYLOGTIME-complete for P. This suggests that non-redundant problems that seem inherently unparallelizable also seem hard to dynamize. We show that a form of transitive closure is complete under incremental reduction for NLOGSPACE and give similar problems which are incrementally complete for the classes LOGSPACE and non-uniform NC^1. We show that under certain restrictions problems which have efficient dynamic solutions also have efficient parallel solutions.

We also look at the time complexity of circuit value and network stability problems restricted to comparator gates. We show that the dynamic version of the comparator-circuit value problem and "Lex-First Maximal Matching" problem is in LOGSPACE and that of the comparator-network stability problem and "Man-Optimal Stable Marriage Problem" is in LOGSPACE given an oracle which operates in NLOGSPACE. This shows that the dynamic versions of these problems are solvable quickly in parallel even though there are no known NC algorithms to solve them from scratch.

A full version of the paper is available upon request.

Structural Average Case Complexity

Rainer Schuler, Abteilung Theoretische Informatik, Universität Ulm, Oberer, Eselsberg, 7900 Ulm, GERMANY,
Tomoyuki Yamakami, Department of Computer Science, Gunma University, Kiryu, Gunma 376, JAPAN.

We consider structural properties of average-case complexity classes and give a general framework of the average-case analysis initiated by Levin.

Let t be a function from non-negative integers to non-negative real numbers and let μ be a probabilistic density function. A function g from strings to non-negative real numbers is called *t on μ-average* if $Prob_\mu[g(x) > t(|x| \cdot m)] < \frac{1}{m}$ for any positive integer m. A Turing machine is called *t time bounded on μ-average* if its running time is t on μ-average. For simplicity, we assume that every computation path terminates at the same length. For a set $\mathcal{F}$ of density functions, $\mathrm{Aver}\langle \mathrm{Dtime}(t), \mathcal{F}\rangle$ ($\mathrm{Aver}\langle \mathrm{Ntime}(t), \mathcal{F}\rangle$) denotes the collection of all randomized decision problems (D, μ) such that $\mu \in \mathcal{F}$ and there exists a deterministic (nondeterministic) Turing machine computing D which is t time bounded on μ-average. The sets $\mathrm{Aver}\langle \mathrm{P}, * \rangle$ and $\mathrm{Aver}\langle \mathrm{NP}, * \rangle$ respectively coincide with Aver-P and Aver-NP, where $*$ means the set of all density functions.

A problem (D, μ) is called *polynomial-time Turing reducible to* (E, ν), denoted by $(D, \mu) \leq^{\mathrm{p}}_{\mathrm{T}} (E, \nu)$, if there exist a polynomial q and a deterministic polynomial-time oracle Turing machine M such that $L(M, E) = D$ and $\nu(y) \geq \sum_{x:y \in Q(M,E,x)} \frac{\mu(x)}{q(|x|)}$ for all y, where $Q(M, E, x)$ denotes the set of all strings y such that on input x, M makes a query y to the oracle E. Clearly $\mathrm{Aver}\langle \mathrm{P}, \mathcal{F}\rangle$ is closed under polynomial-time Turing reducibility if $\mathcal{F}$ is closed under polynomial domination. For every recursive decision problem D not in P, there exist polynomial-time computable density functions μ and ν such that $(D, \mu) \not\leq^{\mathrm{p}}_{\mathrm{T}} (D, \nu)$ and $(D, \nu) \not\leq^{\mathrm{p}}_{\mathrm{T}} (D, \mu)$. This is shown by a slow diagonalization technique.

The notion of self-reducibility can be naturally introduced and then we show that for some polynomial-time computable density function μ, (SAT, μ) is self-reducible.

We further discuss relativizations of average-case complexity classes. Let (E, ν) be a randomized decision problem. A randomized problem (D, μ) is in $\mathrm{Aver}\langle \mathrm{Dtime}(t), \mathcal{F}\rangle^{(E,\nu)}$ ($\mathrm{Aver}\langle \mathrm{Ntime}(t), \mathcal{F}\rangle^{(E,\nu)}$) if $\mu \in \mathcal{F}$ and there exists a deterministic (nondeterministic) Turing machine M such that $L(M, E) = D$ and M^E is t time bounded on μ-average and for all y, $\nu(y) \geq \sum_{x:y \in Q(M,E,x)} \frac{\mu(x)}{t(|x|)}$. For each $k > 1$, we then define $\mathrm{Aver}\langle \Sigma^{\mathrm{p}}_k, \mathcal{F}\rangle$ to be the union of all $\mathrm{Aver}\langle \mathrm{NP}, \mathcal{F}\rangle^{(E,\nu)}$ for every $(E, \nu) \in \mathrm{Aver}\langle \Sigma^{\mathrm{p}}_{k-1}, \mathcal{F}\rangle$. Similarly $\mathrm{Aver}\langle \Delta^{\mathrm{p}}_k, \mathcal{F}\rangle$ is defined. It is not hard to see that in some relativized world $\mathrm{Aver}\langle \mathrm{P}, \text{P-comp}\rangle \neq \mathrm{Aver}\langle \mathrm{NP}, \text{P-comp}\rangle$, where *P-comp* denotes the set of all density functions whose distribution functions are polynomial-time computable.

For any time-bounded complexity class $\mathcal{C}$, '$\mathcal{C}$ (or *real-$\mathcal{C}$*) means the class of sets D such that for every recursive density function μ, $(D, \mu) \in \mathrm{Aver}\langle \mathcal{C}, * \rangle$. We prove that $\text{'}\Sigma^{\mathrm{p}}_k = \Sigma^{\mathrm{p}}_k$ for all $k > 0$ by using results on proper hard cores.

A full paper is not yet available.

A Simplified Approach to Probabilistic Polynomial-Time Complexity Classes

Alan T. Sherman, Computer Science Department, University of Maryland Baltimore County, Baltimore, Maryland 21228, USA; and Institute for Advanced Computer Studies, University of Maryland College Park, College Park, Maryland 20742, USA

A *probabilistic Turing machine* is an otherwise deterministic Turing machine equipped with a true random number generator. I present a simplified approach for understanding the practical limits and capabilities of these machines.

First, I show how the probabilistic polynomial-time complexity classes ZPP, R, CoR, BPP, PP, as well as the more familiar polynomial-time classes P, NP, $CoNP$, $\Delta = NP \cap CoNP$, can be defined solely in terms of three simple and intuitive error rates (false-positive, false-negative, and indecision). This approach simplifies previous work by Gill and Zachos.

Second, I prove that any combination of these three error rates defines one of the aforementioned nine complexity classes, or the trivial class of all languages. This proof helps confirm that the definitions of these nine classes are robust and that all interesting polynomial-time complexity classes based on asymptotic upper bounds have been identified.

A full paper is in preparation.

Quantity vs. Quality: The Power of Oracle Access in Relativized Polynomial-Time Hierarchies

Ming-Jye Sheu Department of Computer and Information Science, The Ohio State University, Columbus, OH, 43210, USA.

In a recent result, Sheu and Long showed that there is a relativized polynomial-time hierarchy that is an infinite hierarchy and with Θ_k^P properly contained in Δ_k^P for $k \geq 2$. In this paper we consider classes in relativized polynomial-time hierarchies with restricted access to oracles. Using circuit lower bound techniques, we give *optimal* separations with Δ-levels of relativized polynomial-time hierarchies by proving that for any polynomial-time computable, polynomial-bounded function f, there is an oracle set relative to which $\mathrm{P}^{\Sigma_{k-1}^P[f]}$ is properly contained in $\mathrm{P}^{\Sigma_{k-1}^P[f+1]}$, for all $k \geq 2$. Moreover, we show that the Σ_{k-1} oracle set needed for such separations does not need to be a complete set for Σ_{k-1}. Specifically, we construct an oracle set A such that there is a set B in $L_k^{P,\Sigma}(A)$, level k of the low hierarchy in $\mathrm{NP}(A)$, with $\mathrm{P}^{\Sigma_{k-2}^P(B)[f+1]}$ not contained in $\mathrm{P}^{\Sigma_{k-1}^P(A)[f]}$ for all $k \geq 2$. In this case, a complete set for $\mathrm{NP}(A)$ is replaced with a weaker set B, a low set in $\mathrm{NP}(A)$. Yet, one additional query in $\mathrm{P}^{\Sigma_{k-2}^P(B)[f+1]}$ computation allows recognition of languages not in $\mathrm{P}^{\Sigma_{k-2}^P(\mathrm{NP}(A))[f]}$.

A full paper is not yet available.

Multiplication and Division can be computed in Depth-3 Threshold Circuits*

Kai-Yeung Siu, Dept. of Electrical & Computer Engineering, University of California, Irvine, CA 92717, Email: siu@balboa.eng.uci.edu.
Vwani Roychowdhury, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907, Email: vwani@drum.ecn.purdue.edu

Using a recent result of Goldmann, Håstad and Razborov [1], we show that iterated addition and multiplication can be computed in depth-2 and depth-3 threshold circuits with polynomial size and polynomially bounded integer weights, respectively. These results answer two of the open questions in [1]. Moreover, it follows from the lower bound result of Hajnal et al. [2] that these threshold circuits are optimal in circuit depth. We also indicate that these techniques can be applied to construct polynomial-size depth-3 threshold circuits for division and powering, and depth-4 threshold circuit for multiple product.

[1] M. Goldmann, J. Håstad, and A. Razborov. *Majority Gates vs. General Weighted Threshold Gates*, Seventh Annual Conference on Structure in Complexity Theory.
[2] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan. *Threshold circuits of bounded depth*, IEEE Symp. Found. Comp. Sci., 28:99–110, 1987.

*Preliminary version of this paper appeared as a technical report No. ECE-92-05 of University of California, Irvine. Complete version of the paper may be obtained by sending requests to the authors by e-mail.

Robust Algorithms with Bounded Query

Jan Sochor, Department of Computer Science, Charles University, Prague, Czechoslovakia

In this paper we investigate robust Turing machines. Especially, we investigate the number of questions to oracle and the length of these questions.
The robust Turing machines were defined by Uwe Schöning as machines for which the accepted set does not depend on oracle but some oracle helps the machine to work in polynomial time. Let us repeat the motivation for this concept: The computer solves some hard problem too slow. A man helps it to speed up the computation. But the computer does not know whether the man is really expert or whether the man tries to confuse it. It was proved that such a computer can in polynomial time accept only sets from $NP \cap co{-}NP$. In later papers Uwe Schöning and Ker-I Ko investigated especailly whether some set can help the computer to solve some problems. This is one view on the question how hard problems must been solved by the man who helps the computer. In this paper we will measure this hardness by a different way. We will measure the number and the length of queries which the man must answer.
In this paper we prove several theorems which bound the length of query for robust Turing machines. We show that for any set accepted by some robust Turing machine with the number of queries bounded by some function f there exists another machine which accepts the set with the length of query bounded by $\log(f(n)) + c \cdot \log(n)$ or even bounded by $\log(f(n))$. Generally, we prove that no questions longer than $n + c \cdot \log(n)$ are necessary. We relativize the problem whether classes of sets accepted by robust algorithms with query length (or number of queries) bounded by two different functions f and g are different. We prove that this problem can be relativized with both positive and negative results. We show that the class of all sets accepted by robust algorithms with logarithmical length of query probably differs both from P and from $NP \cap co{-}NP$. (More precisely, $NEXPTIME \cap co{-}NEXPTIME \neq EXPTIME \Longrightarrow P_{help}^{L\log} \neq P$ and $NP \cap co{-}NP \not\subseteq P/Poly \Longrightarrow NP \cap co{-}NP \neq P_{help}^{L\log}$.) We prove that bounded number of queries is the same as bounded nondeterminism.

Ask author for the full paper.

Author's correspondence address: Jan Sochor
Charles University, Dept. of Computer Science
Malostranské nám. 25
118 00 Praha 1
Czechoslovakia

E-mail address: sochor@cspguk11.bitnet

May 1992

Computing Aribtrary Symmetric Functions [1]

Daniel A. Spielman[2] Department of Computer Science, Yale College.

We begin by surveying some results concerning the gate complexity of computing arbitrary boolean and symmetric functions in threshold circuits. Paturi and Saks [3] proved that $\Omega\left(n/\log^2 n\right)$ gates are necessary to compute parity by a depth-2 threshold circuit with bounded weights. By a theorem of Winder [5], the same bound holds for most symmetric functions without any restrictions on weights. Siu, Roychowdhury and Kailath [4] present a construction that computes parity in depth $d+1$ using $O\left(dn^{1/d}\right)$ threshold gates. In contrast, it is implicit in the work of Lupanov [2] that $2(\frac{n}{\log n})^{1/2}$ gates are necessary to compute most symmetric functions, even without bounds on weight or depth. Combining constructions of [4] and [2], we present circuits that achieve this bound for all symmetric functions. Comparing results of [5] and [4], we prove that, for most sizes G, more functions can be computed by depth-3 threshold circuits of size $(\sqrt{2}+\epsilon)G$ than by unbounded-depth circuits of size G, for any $\epsilon > 0$.

It is implicit in [5] that $\frac{2^n}{n^2}(1-o(1))$ threshold gates are needed to compute arbitrary boolean functions in depth 2. Using a novel application of error-correcting codes, we show that arbitrary boolean functions can be computed by depth-2 threshold circuits with $3 \cdot 2^{n-1}/n$ gates. In the last section, we construct wire-efficient circuits that compute arbitrary symmetric functions. Our construction simplifies the construction of Beame, Brisson and Ladner [1] and improves its constant factors.

References.

[1] P. Beame, E. Brisson, and R. Ladner. The complexity of computing symmetric functions using threshold circuits. TR 90-01-01, Univ. of Wash., Dept. of Comp. Sci. and Eng., 1990.

[2] O. B. Lupanov. Circuits using threshold elements. *Soviet Physics Doklady*, 17(2):91–93, 1971.

[3] R. Paturi and M. E. Saks. On threshold circuits for parity. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pp. 397–404, 1990.

[4] K.-Y. Siu, V. Roychowdhury, and T. Kailath. Depth-size tradeoffs for neural computation. *IEEE Trans. on Computers*, 40(12):1402–1412, 1991.

[5] R. O. Winder. Bounds on threshold gate realizability. *IEEE Trans. on Computers*, C-12:561–564, 1963.

---

[1] Paper is TR 906. Write to Yale Univesity, Dept. of Comp. Sci., 51 Prospect Street, New Haven, CT 06520-2158.

[2] Until September 1992, may be reached at 2037 Pine St., Philadelphia, PA 19103. After September 1992, may be reached at MIT Department of Applied Mathematics. Supported in part by NSF grant CCR-8958528 under an REU supplement.

Completeness and Limited Nondeterminism

Robert Szelepcsényi,
Department of Computer Science,
University of Rochester, Rochester, NY, 14627

Kintala and Fischer [SIAM JCOMP, 1980] defined the limited nondeterminism hierarchy within NP, the so-called β hierarchy. [Diaz and Toran MST, 1990] state, and attempt to justify, the statement: "by restricting the amount of nondeterminism in NP-complete problems, we do not seem to obtain complete problems for β_k." We note that their justification confuses nondeterminism with number of variables, and we prove the opposite of their claim; via the development of limited-nondeterminism-preserving reductions, we obtain complete problems for β_k by restricting the amount of nondeterminism in NP-complete problems. We also discuss the connections between β hierarchy completeness and greedy algorithms.
A full version of this paper is not yet available.

New Classes of Counting Functions

Vinodchandran N.V. and Meena Mahajan, Dept. of Computer Science and Engineering, Indian Institute of Technology, Madras 600 036, India

Several natural classes of functions have been defined by considering operators acting on the computation trees of non-deterministic polynomial-time Turing machines. For instance, $\#P$ counts the number of accepting computations in such computation trees [Val79], *SpanP* counts the number of distinct outputs produced in such a tree when the machine is also acting as a transducer [Val79], and so on.

Since $\#P$ is not closed under subtraction, the gap analog *GapP*, which is the closure of $\#P$ under subtraction and also possesses several interesting properties by itself, was introduced in [FFK91].

Along similar lines, we define, in this paper, the gap analog of *SpanP* functions as the class *GspanP*. Functions in this class count the difference between the number of distinct witnesses a counting machine can produce for and against a particular input. This function class contains *SpanP* as well as *GapP*.

A restricted version, *GspanP1*, which counts the number of distinct outputs that a counting machine produces for an input x but never produces against x, is also defined, and is shown to lie between *SpanP* and *GspanP*. Additionally, it is contained in $\#NP$.

Interestingly, though we cannot show an equivalence between *SpanP* and *GspanP1*, we show that their closures under subtraction are both equal, and are exactly equal to *GspanP*.

We then study some language classes defined using these functions. These are the gap-span analogs of the classes *PP*, *SPP* and *UP*. We examine lowness of these languages for *GspanP*, and also study the implications of *GspanP* being no more powerful than *GapP*. We give a simple proof to show that $C.D^P$ exactly equals the class *PP(NP)*, by using a characterization of $\#NP$ proved in [KST89].

References.

[FFK91] S.A. Fanner, L.J. Fortnow and S.A. Kurtz, Gap-definable counting classes, in: *Proceedings of Sixth Annual Conference on Structure in Complexity Theory* (1991) 30–42.

[KST89] J. Kobler, U. Schoning, and J. Toran, On counting and approximation, *Acta Informatica*, 26 (1989), 363–379.

[Val79] L.G. Valiant, The complexity of computing the permanent, *Theoretical Computer Science*, 8 (1979), 189–201.

A full version will shortly be available as a Technical Report.

On the Computational Complexity of Inferring Evolutionary Trees

Todd Wareham Department of Computer Science, Memorial University of Newfoundland, St. John's, NF, Canada, A1C 5S7,

Reconstructing evolutionary trees can be viewed formally as an optimization problem. Since 1982, the decision problems associated with the most commonly used approaches to reconstructing such trees have been shown to be NP-complete. Using the work of Chen, Gasarch, Krentel, Köbler, Rappoport, Schöning, Selman, Toda, and Torán, these results have been extended to derive bounds on the complexity of several functions associated with each of these decision problems, namely

- *evaluation functions*, which return the cost of the optimal tree(s).
- *solution functions*, which return an optimal tree.
- *spanning functions*, which return the number of optimal trees.
- *enumeration functions*, which systematically enumerate all optimal trees.

Where applicable, bounds are also presented for the versions of these functions that operate on trees of a given cost or of cost less than than a given limit. Based in part on these results and theorems from the work of Krentel and Arora et al., bounds are also given on how closely polynomial-time algorithms can approximate optimal trees.

This research is available in the author's M.Sc. thesis.