Structural Complexity Column for the Bulletin of the European Association for Theoretical Computer Science

I. A Retrospective in Structural Complexity
II. Sparse Complete Sets for NP and the Optimal Collapse of the Polynomial Heirarchy

Juris Hartmanis

87-839
June 1987

Department of Computer Science
Cornell University
Ithaca, New York 14853-7501

The Structural Complexity Column

J. Hartmanis

Department of Computer Science

Cornell University

Ithaca, New York 14853

A Retrospective on Structural Complexity

The systematic study of computational complexity theory was initiated in the early sixties and during the following twenty-five years complexity theory has developed into one of the central and most active research areas of computer science. It has grown into a rich and exciting mathematical theory whose development is motivated and guided by computer science needs and technological advances. At the same time, it is clear that complexity theory, dealing with the quantitative laws of computation and reasoning, is concerned with issues and problems of direct interest to many other disciplines as well. It is quite likely that in the overall impact of computer science on our thinking, complexity theory will be recognized as one of the most influential intellectual contributions.

In particular, complexity theory is of considerable interest to mathematics and some of the key open problems in complexity theory are basic questions about the nature of mathematics. At the same time, it is interesting to recall that initially the mathematical community, with some notable exceptions, was quite reserved about the development of computers and the emergence of computer science. Initially, only a small number of mathematicians participated in (and some strongly influenced) the development of computer science; the mathematical community as a whole showed little awareness or interest. More recently, this situation is changing and, particularly among the younger mathematicians with some early exposure to computer science, interest and participation in computer science is growing and enriching both disciplines. As a matter of fact, we believe that the future development of mathematics will be very strongly influenced by computer science. We expect that in the coming decades, the strongest outside influence on the development of mathematics will come from the extended use of computing and from concepts and problems arising in computer science. This influence on the development of mathematics is likely to be as strong as the earlier influence of natural sciences.

Looking at the development of computational complexity theory, one can observe very definite turning points and descry trends and styles in research. The early work in complexity theory defined the basic computational models, established the standard complexity measures, created the guiding research paradigms, established complexity classes as the dominant structural concept and yielded the initial results about algorithms and complexity classes on which complexity theory is built. Since the early seventies, motivated by Cook's discovery of complete languages in NP and Karp's extension of these concepts to combinatorial problems and other complexity classes, complexity theory has been particularly concerned with the study of the feasible complexity classes below PSPACE. The new concepts of resource bounded reductions and complete languages for natural complexity classes, were modifications of central concepts from recursive function theory, but in the new setting they initiated an unexpectedly rich and exciting area of research. This work also raised some of the most interesting and, apparently some of the hardest open problems in computer science. The best known of these problems is the $P = ?NP$ question.

It is quite surprising and impressive that in the whole awesome mathematical arsenal there seem to be no results or techniques to directly attack these *separation* problems. The questions about the computational power of various computing models and relations between different resource bounded computations have raised new mathematical problems which do not appear to be amenable to any known techniques. This points out dramatically that we still have only a fragmentary understanding of the quantitative aspects of computing and that most likely new proof techniques will be required to gain a deeper understanding. We can expect that major intellectual battles will be fought and that we will have quite a few surprises.

It should also be recalled that some of these *separation* problems are questions about the basic nature of mathematics and thus formal reasoning. In essence, the P = ?NP problem is the problem about the quantitative computational difference between finding a proof for a theorem and checking a given proof for correctness. The NP = ? PSPACE problem is the question about the difference in space needed to "write down a proof and space needed to present the proof". Whereby "presentation" of a proof is meant the process of showing the proof (to a formal proof checker) with the option of erasing proofs of partial results and reusing the space for the rest of the proof [HY]. In essence, these questions ask:

> a) how much faster can problems be solved with inspired guessing than with systematic, deterministic methods which always yield an answer;
>
> b) how much more blackboard space is needed to write down a proof than to present a proof.

In view of these observations, one has to consider the computational complexity separation problems among the hardest and very important open problems in all of mathematical sciences.

It is interesting to note that the formulation and study of these problems, central to a deeper understanding of the nature of mathematics and quantitative aspects of computing, emerged from computer science and not from the mainstream work in logic or recursive function theory, as one could have expected. To appreciate more fully this achievement, it is informative to look at the related, earlier developments of cybernetics in the Soviet Union. A very informative and detailed account of these developments can be found in Tracktenbrot's article in the Annals of the History of Computing [Tr].

During the 1950's, Russian theoretical cyberneticians were concerned with various switching circuit minimization and circuit complexity problems. From this early work emerged a conviction that some of the circuit complexity problems could only be solved by "brute force" or "exhaustive search" methods. In Russian, there is a lovely sounding word for "brute-force", it is *perebor*. Therefore, such problems were referred to as *perebor problems* and the task was to show that perebor could not be eliminated. It is impressive how early the Russian scientists appreciated the possible existence of natural perebor problems and tried to prove that perebor could not be eliminated. (There seem to be no direct parallels to this work in the West before the early sixties). Unfortunately, this appreciation came before the necessary developments in computational complexity theory, and there was no consensus of what constituted a proof that a given problem was a perebor problem. This lack of precise formulation led to at least two published claims that the perebor problem had been solved [Tr].

Only after the development of complexity theory, the definition of P and NP, and the discovery of complete languages in NP do we now have precise and intuitively satisfying formulation of the perebor problem. In the Soviet Union this formulation was obtained independently by Leonid Levin, about a year after Cook's 1971 STOC paper, and published in a painfully loconic form (without any proofs) in 1983. Levin refers to NP-complete problems as "universal perebor problems". A translation of Levin's paper can be found in [Tr].

Though P, NP, and NP complete languages were discovered independently in the West and in the Soviet Union, the following developments are quite different. In the West, particularly the United States, these results were received with great enthusiasm and a veritable gold-rush fever was generated in the search of different NP complete problems. Later came the extensive exploration of relativization techniques, study of different resource bounded reductions, the definition of the polynomial-time hierarchy, and the study of structural relations among the feasible complexity classes. The response in the Soviet Union was much more muted. It would indeed be interesting to carefully compare these divergent developments and to try to understand the reasons for these differences. One can give a few glib explanations that the scientific establishments in the West and the Soviet Union are radically different or that Levin was the "Russian Cook" and that there was no corresponding "Russian Karp". At the same time, these explanations are not convincing and, most likely, the real explanation will be more subtle and more informative.

More recently, one can perceive in the development of complexity theory a growing interest in the structural properties of the feasible computations. This research has a definite flavor and has emerged during the last decade, as a cohesive subfield with a rich set of interlocking results and problems. It is partially characterized by interest in global properties of complexity classes, the relations between complexity classes, logical implications among certain unsolved problems about complexity classes, and the use of relativization to explore possible relationships. We shall refer to this area of work as *structural complexity.* The complimentary research stream in complexity theory is more concerned with specific algorithms and has a more pronounced combinatorial flavor.

This column is dedicated to the structural complexity research stream. We will not try to explicibly define this research stream, but in future columns illustrate it by examples, discuss its developments and trends and try to raise the consciousness about this area of research.

A good representation of recent results in structural complexity can be found in the proceedings of the conference on Structure in Complexity Theory [Se] held last June in Berkeley. In his perceptive review of this conference in this Bulletin, Volume 30, Peter von Emde Boas raises a question about this field and answers it by avoiding to answer it:

> "**Question**: what is the precise scope of the meeting?
>
> **Answer**: Never ask a question you know beforehand to be unanswerable. The experience has shown that there exists a flavour of complexity theory which is more directed to fundamental issues and less directed towards specific algorithmic problems. One might call this direction more "mathematically oriented", but that's definitely not the intention: mathematics as a tool is a necessity; mathematics as a purpose in itself is just mathematics and not computer science. Furthermore workers in this area have felt themselves to be underpresented at other meetings. Also scopes of conferences are determined more by subject lists and Program committee constituencies than by phrases; scopes seem to change faster than we can propose sentences describing these scopes."

The Second Annual Structure in Complexity Theory Conference will be held at Cornell University, June 16-19, 1987. This conference will be followed by the Second Annual Symposium on Logic in Computer Science also held at Cornell, June 22-25, 1987. This promises to be an exciting summer at Cornell for theoretical computer science and the intellectual contours of structural complexity work should become even better defined. Following von Emde Boas' axiom that "scopes of conferences are determined more by subject lists and program committee constituencies than by phrases", we list the program committees for the first and second Structure in Complexity Theory Conferences and the topic list for the second conference.

> *1st Conference*: Ron Book; Juris Hartmanis; Harry Lewis; Steve Mahaney, co-chair; Ken McAloon; Alan Selman, co-chair; Mike Sipser; Peter von Emde Boas; and Paul Young.
>
> *2nd Conference*: Shafi Goldwasser; Juris Hartmanis; Neil Immerman; Deborah Joseph; Steve Mahaney, chair; Uwe Schoning; Alan Selman; Mike Sipser; Larry Stockmeyer; and Peter van Emde Boas.

Topics:

Structure of complexity classes	Properties of complete sets
Resource-bounded reducibilities	Theory of relativizations
Applications of recursion theory	Random and interactive proof systems
Kolmogorov complexity	Cryptographic complexity
Applications of finite model theory	Independence results

The first Structure in Complexity Theory Conference was proceeded by two previous conferences with a strong structuralist flavor. The first one was organized by Carl Smith and Paul Young at Purdue University in 1981 and the second one, organized by Ron Book, was held in Santa Barbara in 1983. Three of the papers presented at the Santa Barbara conference on Computational Complexity Theory constitute Studies in Complexity Theory edited by Ron Book [Bo]. Of these papers, Mahaney's "Sparse Sets and Reducibilities" gives a particularly nice overview of the role played by sparse sets in recent work in structural complexity. Here one should also mention Uwe Shoning's delightful "Complexity and Structure" [Sch], which is derived from his Habilitations Schrift and gives a nice exposition of structural complexity results leading to and surrounding topics studied by the author.

We hope that this short Retrospective on Structural Complexity helps focus attention on this area of research and entices some of the readers to investigate it further and participate in its future growth.

Request

To make this structural complexity column as interesting and informative as this area deserves, please send me relevant information about new results, trends, conjectures, publications, conferences, and other related topics.

References

[Bo] Book, R.V. (Editor), Ker-I, Ko, S.R. Mahaney, K. McAloon. Studies in Complexity Theory. Pitman, London, John Wiley and Sons, N.Y., 1986.

[HY] Hartmanis, J. and Y. Yesha, "Computation Times of NP Sets of Different Densities". Theoretical Computer Science, Volume 34, 17-32, 1984.

[Sch] Schoning, Uwe. "Complexity and Structure". Springer-Verlag, Lecture Notes in Computer Science, Volume 211, 1985.

[Se] Selman, A.S. (Editor), "Structure in Complexity Theory", Springer-Verlag, Lecture Notes in Computer Science, Volume 223, 1986.

[Tr] Traktenbrot, B.A. "A Survey of Russian Approaches to Perebor (Brute-Force Search) Algorithms". Annals of the History of Computing, Volume 6, 384-400, October 1984.

STRUCTURAL COMPLEXITY COLUMN

J. Hartmanis
Department of Computer Science
Cornell University
Ithaca, NY 14853

Sparse Complete Sets for NP and the Optimal Collapse of the Polynomial Hierarchy

Introduction.

The goal of computational complexity theory is a complete understanding of the quantitative laws that govern computing. Of particular practical interest to computer science is a thorough understanding of the computational complexity of the feasible computations, that is, the computations below *PSPACE*. Clearly, the best known open problem in this area is the classic $P = ?NP$ problem, which has loomed unanswered over computational complexity research since its formulation in 1971. As a matter of fact, none of the related separation problems among the major complexity classes, defined by different resource bounds, have been solved. Among them are:

$$SPACE\ [log\ n] = ?\ NSPACE\ [log\ n] = ?\ P = ?\ NP = ?$$

$$PSPACE = ?\ EXPTIME = ?\ NEXPTIME = ?\ EXPSPACE, \text{ etc.}$$

On the other hand, during the last ten years, the work in structural complexity theory, has revealed many interesting results about relations between different properties of these complexity classes, explored relations and implications between various unsolved problems and yielded many results about relativized complexity classes.

In this column we will review a branch of this research which very recently reached, in a definite technical sense, an optimal conclusion.

Sparse Complete Sets for NP

All of our experience with problems in NP suggest the inescapable conclusion that $NP \neq P$ and furthermore, that the complete problems in NP may indeed require exponential time for their solutions. Thus, we assume, as a working hypotheses, that $P \neq NP$ and furthermore, that the polynomial time hierarchy is infinite, implying that $PH \neq PSPACE$.

The study of sparse complete sets for NP is motivated by the desire to understand how much information is needed to solve NP problems. This research has yielded many technical results, some new proof techniques and a much deeper overall understanding of the information needed to solve NP problems, if we assume that polynomial-time hierarchy is infinite, i.e. $|PH| = \infty$.

For the sake of completeness, we summarize the necessary concepts and definitions [GJ 79, HU 79, Sto 77].

Let P and NP denote the class of all deterministic and nondeterministic polynomial time languages, respectively. The polynomial time hierarch PH is defined inductively:

$$\Sigma_1^P = NP \quad , \quad \Pi_1^P = coNP,$$

$$\Sigma_{k+1}^P = NP^{\Sigma_k^P} \quad , \Pi_{k+1}^P = co\Sigma_{k+1}^P \quad \text{and}$$

$$\Delta_{k+1}^P = P^{\Sigma_k^P} \quad \text{for} \quad k \geq 1.$$

$$PH = \bigcup_{k \geq 1} \Sigma_k^P.$$

A set S is *sparse* iff there exists a polynomial $p(n)$ which bounds the number of elements in S up to size n.

A set S in NP is $\leq_m^P$*-complete* for NP iff for every L in NP there exists a polynomial time function f (i.e. $f \in PF$) such that

$$x \in L \Leftrightarrow f(x) \in S.$$

A set S is $\leq_T^P$-*hard* for NP iff

$$NP \subseteq P^S;$$

if such a set is in NP, then it is said to be $\leq_T^P$-*complete.*

The key difference between $\leq_m^P$-completness (also called "many-one completness") and $\leq_T^P$-completness (or "Turing completness") is that in the first case one just performs an easy translation of one problem into another, without using any information about solutions of NP problems. For Turing reductions, one can query the complete problem S and use the information for subsequent queries polynomially often. Thus, one has to expect stronger consequences from the existence of sparse $\leq_m^P$-complete sets for NP than $\leq_T^P$-complete sets. This is indeed true, as described below.

The study of many-one NP complete sets, which will not be reviewed here, culminated in Mahaney's elegant proof of the following result [Mah 80, Moch 82].

Theorem. There exist sparse many-one NP complete sets iff $P=NP$.

Thus there is no sparse NP set into which we can translate all other NP problems, unless $P=NP$.

The situation for Turing-hard and Turing-complete sets is more complex and we review some of these developments below.

It was observed by Albert Meyer and reported in [BH 77] that:
There exists a sparse set S such that $NP \subseteq P^S$ iff there exist polynomial size circuits for SAT.

We say that a language L *has polynomial size circuits* iff there exists a polynomial $p(n)$ and a sequence of circuits C_1, C_2, $C_3 ..., C_n, ...$ such that for all n $|C_n| \leq p(n)$ and for all x, $|x| \leq n$,

$$C_n(x) = 1 \Leftrightarrow x \in L.$$

No uniformity conditions are imposed on how the circuits are obtained nor is it required that the sparse set S is in NP.

Some time later, Karp and Lipton [KL 80] studied problems which can be solved in polynomial time with polynomial amounts of advice and referred to them as $P/poly$. It is easy to see that all these definitions are equivalent.

Lemma: There exists a sparse set S such that $NP \subseteq P^S$ iff $NP \subseteq P/poly$ iff NP has polynomial size circuits.

The difficult to read Karp-Lipton paper makes the beautiful observation that if $NP \subseteq P/poly$, then the polynomial hierarch collapses to Σ_3^P. This result, with Sipser's help, was extended to the following theorem. We state this result in terms of sparse sets to create an easy transition to the subsequent results.

Theorem: If there exists a sparse set S such that $NP \subseteq P^S$ then $PH = \Sigma_2^P$.

To indicate the main idea of this proof we use the equivalent polynomial size circuit formulation. They key idea is that with an $\exists$ quantifier one can guess the existence of a polynomial size circuit, C_n, for SAT and then use an $\forall$ quantifier to verify that the circuit works properly. This is done as follows:

C_n functions properly iff

$$C_n(0) = 0 \ \text{ and } \ C_n(1) = 1$$

and

$$(\forall F)\ [|F| \leq n,\ \ C(F) = 1 \ \Leftrightarrow\ C(F_0) = 1 \vee C(F_1) = 1].$$

Where F_0 and F_1 are the formulas obtained from F by setting the first variable to 0 and 1, respectively.

From this we see that with two polynomially bounded quantifiers $\exists\forall$ we can obtain a polynomial size circuit for SAT if such a circuit exists. Equivalently, the correct polynomial size circuits for SAT can be recognized by NP^{SAT} machines. Thus they are in Σ_2^P. Once the circuit is obtained it can be used to successively eliminate the innermost quantifiers, because each $\exists$ and $\forall$ followed by a polynomial time predicate can be translated, by Cook's theorem, into a Boolean formula which then can be tested on the circuit. Thus $PH \subseteq \Sigma_2^P$.

Under the stronger assumption that a sparse set is Turing-complete for NP, Mahaney was able to prove a deeper collapse [Mah 86].

Theorem: If there exists a sparse S in NP such that $NP \subseteq P^S$, then

$$PH \subseteq P^S = P^{SAT}.$$

The proof of this result is based on two observations. First, since S is sparse and in NP, a P^S (or P^{SAT}) machine by binary search can find bit by bit all strings in S up to size n in $p(n)$ many steps. Once S is known up to sufficiently long strings, we can compute what a NP^S machine does in polynomial time and therefore successively eliminates all quantifiers in PH. Thus $PH \subseteq P^{SAT}$.

Long extended Mahaney's result by imposing weaker conditions on the sparse set S [Lon 82]. See also [Yap 83].

Theorem: If there exists a sparse set S in P^{SAT} such that $NP \subseteq P^S$, then $PH \subseteq P^{SAT}$.

Quite surprisingly, Mahaney's result can be improved to yield a much deeper collapse.

Let $P^{SAT[logn]}$ denote the set of languages accepted in polynomial time with no more than $O(logn)$ queries to SAT.

Very recently, Jim Kadin of Cornell University has shown that sparse Turing-complete sets force PH to collapse to $P^{SAT[logn]}$. This is quite a surprising result in that a p-time machine with logn queries cannot determine the relevant strings in S (as it was required by Mahaney's proof). As a matter of fact, this result gives new insight about how oracle information can be accessed, contributing to our growing understanding of information access mechanisms [Kad 86, Kad 87a].

Theorem: If there exists a sparse set S in NP such that $NP \subseteq P^S$ then $PH \subseteq P^{SAT[logn]}$.

To indicate the key ideas of Kadin's proof we will show that

$$P^{SAT} \subseteq P^{SAT[logn]}$$

from which, by Mahaney's result, it follows that

$$PH \subseteq P^{SAT[logn]}.$$

The proof is based on the subtle observation that in $O(logn)$ queries a P^{SAT} machine can compute, by binary search, the census function $C_S(n)$ for S, which gives the exact number of strings in S up to length n. Once $C_S(n)$ is known, an NP machine can recognize $x \in \overline{S}$, for $|x| \leq n$, by gussing $C_S(n)$ strings in S, verifying that they are in S and then checking if x is different from all $C_S(n)$ strings.

Therefore, any NP^S machine (or NP^{SAT} machine) can be replaced by an NP machine which has the census information, $C_S(n)$, and uses as subroutines two NP machines for (short) strings in S and $\overline{S}$, respectively. But for this machine, with one query to S (or SAT), we can determine if it accepts a given string. Thus with $O(logn)$ queries we can accept any L in P^{SAT}. Thus

$$PH \subseteq P^{SAT} \subseteq P^{SAT[logn]}.$$

This proof relates the depth of the collapse of PH to the density of the sparse oracle. For an oracle S in NP with $C_S(n) \leq n^k$, such that $NP \subseteq P^S$ we have $PH \subseteq P^{SAT\, O[logC_S(n)]}$.

In essence, we use all but one of the queries to compute the census function of S and then complete the computation with one more query.

The difference between Mahaney's and Kadin's results is that the first method queried SAT to obtain all the strings in S up to length n and the second proof only needed $O(logn)$ queries to get the census function from which to derive a special NP machine (for each n) whose action on the input can be determined with one more query.

Personally I was impressed by Mahaney's "collapse" and was quite surprised by Kadin's result. Since Kadin's result looked the best possible (as Mahaney's did before), I challenged

Kadin to try to prove his result optimal in some relativized worlds. It should be noted that all previous results hold in relativized worlds.

Indeed Kadin was able to show that there exist base oracles for which his result is the best possible [Kad 87a].

Theorem: There exists a recursive set B such that NP^B has sparse Turing complete sets and therefore

$$PH^B \subseteq [P^B]^{NP^B[logn]},$$

but for any f such that

$$\lim_{n\to\infty} \frac{f(n)}{logn} = 0$$

we have

$$[P^B]^{NP^B[logn]} \not\subseteq [P^B]^{NP^B[f(n)]}.$$

This result requires quite a subtle oracle construction and proves that, at least in some relativized worlds, the census information is necessary and sufficient for the collapse of PH. It should also be emphasized that this result relays on the assumption that S is in NP.

In this context, it would be interesting to determine whether there are relativized worlds in which the Karp-Lipton-Sipser collapse, for sparse set S (not in NP) such that $NP \subseteq P^S$, yielding $PH \subseteq \Sigma_2^P$, is optimal. The same question can be asked about T. Long's result.

As with many relativization results, we know that problems that can be relativized in two contradictory ways are usually, though not always, hard to solve [Har 85]. In essence, we know that "standard" diagonalization techniques can not resolve these problems.

In general. relativization techniques are useful in exploring logical possibilities, assessing the problem difficulty and giving further insights in the study of different access mechanisms to information. It should be emphasized, that many relativization results in complexity theory should be viewed as an exploration of the properties of various access mechanisms to information and the relations of different restrictions and structuring of this information.

For example, the original Baker-Gill-Solovey [BGS 75] relativization results can be viewed as demonstrating with an example $P^A \neq NP^A$, that the NP-access mechanism is more powerful than a P-access mechanism if the the information in A is not too complete and one has to search through all of it. On the other hand, the $P^{PSPACE} = NP^{PSPACE}$ example shows that the difference of these access mechanisms can be overwhelmed by very complete information available in the oracle. Thus we see that the power of the P and NP access mechanisms depends on the nature of the accessible information. Similarly, the results described earlier, indicate that if the difference in NP and P access mechanism can be eliminated by a sparse set, then alternating quantification in the polynomial hierarchy can be stopped after two steps; the additional quantifiers do not gain greater expressive power.

As indicated earlier, the proof that a problem can be relativized in two contradictory ways serves today in theoretical computer science almost the same role as proving a problem NP hard in the study of algorithms. If a problem is NP hard, we are very unlikely to solve in reasonable time sufficiently big instances of this problem. Similarly, the contradictory relativization (of a sufficiently "rich" problem) is a good indication that it can not be solved with our current mathematical techniques.

Finally, we will report one more interesting relationship between low and high collapses and sparse sets discovered by Kadin. The question is whether we can replace two queries to SAT with one more complex query. That is, is

$$P^{SAT[1]} = P^{SAT[2]} ?$$

Again the answer involves the collapse of the polynomial hierarchy and key sparse sets. The following result, among several stronger results, was derived by Kadin [Kad 87b].

Theorem: If $P^{SAT[1]} = P^{SAT[2]}$ then

$$P^{SAT[1]} = P^{SAT[log]}$$

and

$$PH = \Delta_3^P.$$

In other words, the collapse of $P^{SAT[1]}$ and $P^{SAT[2]}$ induces two other collapses, first $logn$ queries can be replaced by one query and higher up PH collapses to Δ_3^P. The proofs of these two collapses are different and so far it is not known what happens between

$$P^{SAT[1]} = P^{SAT[logn]} \text{ and } \Delta_3^P = PH.$$

For example, do the above collapses imply that

$$P^{SAT} = NP^{SAT[logn]}?, \quad NP^{SAT} = P^{SAT}? \ etc.$$

Again, one could try to construct oracles to see if there exist relativized worlds which have the two separate collapses but still not collapse, for example, P^{SAT} and $NP^{SAT[logn]}$. If so, then we have to assume that the unrelativized problem is very hard.

In this connection, it is interesting to recall Krentel's result [Kr 86] that for polynomial time functions

$$PF^{SAT} = PF^{SAT[logn]} \Rightarrow P = NP.$$

The corresponding result for P^{SAT} and $P^{SAT[logn]}$ is not known and it fails in some relativized worlds.

So far we have seen that the existence of sparse NP hard and NP complete sets force various collapses of the polynomial time hierarch. Since $P^{SAT[1]} = P^{SAT[2]}$ also forces PH to collapse, the natural question is whether any key sparse sets are associated with this collapse. This answer was also provided by Kadin [Kad 87b].

Theorem: If $P^{SAT[1]} = P^{SAT[2]}$ then there exists a sparse set S such that

$$\overline{SAT} \in NP^S.$$

We recall that the existence of a sparse set S such that

$$NP \subseteq P^S \quad \Rightarrow \quad PH \subseteq \Sigma_2^P.$$

Analogously, we can show [Yap 83, Kad 87b] the following.

Theorem: If there exists a sparse set T such that

$$coNP \subseteq NP^T \quad then \quad PH \subseteq \Delta_3^P.$$

Again these results show interesting trade-offs between different access mechanisms, namely P and NP, and consequences of the existence of special sparse sets. Note that $\overline{SAT} \in P^S$ implies that $NP \subseteq P^S$ and $coNP \subseteq P^S$. Similarly, $\overline{SAT} \in NP^T$ implies that $coNP \subseteq NP^T$. In the first case this implied $PH \subseteq \Sigma_2^P$, in the second case, $PH \subseteq \Delta_3^P$.

At the present, it is not known whether the above discussed collapses of PH are optimal in some relativized worlds, and thus reflect real differences in the access power of P and NP to sparse sets.

Conclusion

The results summarized in this column bring the study of the depth of collapse of the polynomial time hierarchy, under the weight of the (assumed) existence of sparse Turing complete sets for NP, to a partial conclusion. Clearly, this is only a small part of structure complexity theory, but, I believe, it is an interesting part of the search for a deeper understanding of nature of computation.

As a matter of fact, so far all the work in structural complexity theory has to be viewed in military parlance, only as an "intellectual reconesonce in force" until the real intellectual battle will lead to the first quantitative separation of some of the major complexity classes.

References

[BGS 75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP question. *SIAM Journal on Computing 4*, pp. 431-442, 1975.

[BH 77] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing 6*, pp. 305-322, 1977.

[GJ 79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[Har 85] J. Hartmanis. Solvable problems with conflicting relativizations. *Bulletin of the European Association for Theoretical Computer Science*, pp. 40-49, Oct. 1985.

[HU 79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[Kad 86] J. Kadin. *Deterministic Polynomial Time with O(log(n)) Queries.* Technical Report TR-86-771, Cornell University, August 1986.

[Kad 87a] J. Kadin. $P^{NP[log]}$ and sparse Turing complete sets for NP. 1986. Accepted for *Structure in Complexity Theory, 1987.*

[Kad 87b] J. Kadin. Private communication.

[KL 80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *ACM Symposium on Theory of Computing*, pp. 302-309, 1980.

[Kr 86] M. Krentel. The complexity of optimization problems. *Proceedings ACM Symposium on Theory of Computing*, pp.67-76, 1986.

[Lon 82] T.J. Long. A note on sparse oracles for NP. *Journal of Computer and System Sciences 24*, pp. 224-232, 1982.

[Mah 80] S. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. In *Proceedings IEEE Symposium on Foundations of Comuter Science*, pp. 54-60, 1980.

[Mah 82] S. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. *Journal of Computer and Systems Sciences, 25*, pp. 130-143, 1982.

[Mah 86] S. Mahaney. Sparse sets and reducibilities. In *Studies in Complexity Theory*, ed. R.V. Book, John Wiley and Sons, Inc., New York, pp. 63-118, 1986.

[Sto 77] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science, 3, pp. 1-22, 1977.*

[Yap 83] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Sicence 26,* pp. 287-300, 1983.