The Collapsing Hierarchies

Juris Hartmanis*

87-861

September 1987

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

*This research was supported by NSF Research Grant DCR 85-20597.

J. Hartmanis*

Department of Computer Science

Cornell University

Ithaca, New York 14853

The Collapsing Hierarchies

Introduction

The last twelve months have been one of the most exciting and creative periods in the study of structural complexity theory. It seems that during this short period, more progress has been made than during the previous five years in the understanding of the structure of feasible computations. Some of the intellectual excitement can already by surmised from the proceedings of the Second Structure in Complexity Theory Conference published by the Computer Society. The full excitement though was best felt at this conference held at Cornell University during June, 1987. Besides some of the very nice results in the Proceedings, a good deal of excitement was created by more recent results obtained just before the conference. Because of the leisurely pace of the conference there were many opportunities to explain and discuss the most recent results and several of them were presented in well attended poster sessions. Finally, possibly the most exciting result solving a well known twenty- three year old open problem, was obtained right after the conference (and, we hope, inspired by the new results discussed at the conference and Ithaca's beauty).

In this column, we will discuss a cluster of these recent results about various complexity hierarchies (mostly collapsing) and the use of the census function as a unified proof technique. These results show that several recently defined hierarchies collapse, pleasantly simplifying

This research was supported by NSF Rearch Grant DCR 85-20597.

our world, and that all of these results can be derived with one proof technique using census functions. Quite surprisingly, the Boolean Hierarchy and the intertwined Query Hierarchy gained new respectability with the recent proof that the collapse of either hierarchy at any level implies that the classic Polynomial Time Hierarchy collapses to at least the third level, $PH = \Delta_3^P$. Finally, we will discuss the surprising result that the nondeterministic space bounded computations are closed under complementation. As a corollary, this shows that the context-sensitive languages are closed under complement, a problem which had been open since 1964. The proof that nondeterministic tape bounded computations are closed under complement was a surprise (few of us believed that they would be), but an even greater surprise was the elegance and simplicity of this solution again exploiting a census argument.

Census Functions and Collapsing Hierarchies

The study of computational complexity of feasible computations was substantially enriched by the definition and investigation of the Polynomial Time Hierarchy, PH. This hierarchy, the polynomial time analogue of the Kleene Hierarchy of recursive function theory [Rog 67], gave a natural classification of feasible computations above NP, either in terms of the number of alternating quantifiers over NP problems or in terms of height of NP oracle computations with NP oracles [Sto 77, Wra 77]. Since then, a bewildering variety of new complexity classes and hierarchies have been defined and investigated, reflecting and modeling different quantitative aspects of computation: various relativized computations, optimization problems, probabilistic and random computations, parallel computations, etc. In short, the structure of feasible computations is looking more and more intricate and new complexity is added almost with every conference on this topic. Unfortunately, the exact quantitative relations between most of these classes are not known and should $P = PSPACE$ most of them would be compressed into P. We should hasten to say, that it is not very likely that $P = PSPACE$ and that all the "intellectual experimental evidence" points to the conclusion that the major feasible complexity classes are all different. In structural complexity theory, we assume as a working hypothesis that PH is infinite.

On the other hand, very recently it has been shown that several of the hierarchies of complexity classes are indeed finite. These are unexpected results and, quite surprisingly, after careful study of their original proofs, they can now all be derived quite simply using census functions.

We illustrate this by outlining a proof that the so-called strong exponential hierarchy collapses. The original proof of this collapse was obtained by Lane Hemachandra [Hem 87]. Lane initially tried to construct an oracle to show that this hierarchy could be infinite in relativized worlds and the impossibility of such a construction led him to his quite complex proof that the hierarchy collapses. This illustrates an interesting use of oracles to explore the logical possibilities and indicate possible new results. The original proof was dramatically simplified by Uwe Schöning and Klaus Wagner [Sch W 87] using Kadin's lemma [Kad 87]. See also, the previous Structure in Complexity Column for a discussion of Kadin's work [Har 87].

The strong exponential hierarchy, SEH, is built up inductively:

$$E = \bigcup_{c \geq 1} DTIME[2^{cn}],\ NE,\ NP^{NE},\ NP^{NP^{NE}},\ \ldots$$

(One can also define $E' = \bigcup_{c \geq 1} DTIME[2^{n^c}]$ and observe that $E \neq E'$, but $P^{NE} = P^{NE'}$, etc).

The collapse is based on the proof that $NP^{NE} = P^{NE}$ and therefore $NP^{NP^{NE}} = NP^{P^{NE}} = NP^{NE} = P^{NE}$.

Theorem (L. Hemachandra):

$$NP^{NE} = P^{NE} \text{ and therefore } SEH = P^{NE}.$$

Proof outline following [Sch W 87]:

Clearly

$$NP^{NE} \supseteq P^{NE}.$$

To show

$$NP^{NE} \subseteq P^{NE},$$

let A be in NE and N any NP machine running in time $n^k + k$. Let U be a standard NE complete language. We will construct a deterministic polynomial time machine D such at:

$$L(N^A) = L(D^U).$$

The strategy for this proof is beautifully simple: D^U for input x, queries oracle U to find out exactly how many strings in A could be queried by $N^A(x)$. From this information D can construct a NE machine, N_c, which decides how $N^A(x)$ queries to A are answered. Now D constructs the description of an NE machine N_E which simulates $N^A(x)$ and uses N_c to answer the queries to A. Since $N^A(x)$ accepts iff $N_E(x)$ accepts, D^U queries U if N_E accepts x to determine what $N^A(x)$ does.

The following outlines this procedure in more detail.

Given x, D^U queries U, and by binary search in polynomially many queries in $|x|$ determines the exact number of strings in A up to size $|x|^k + k$ (which could be queried by N on x). Let this number be n_x.

Clearly, once n_x is known to D, it can construct in polynomial time in $|x|$ the description of an NE machine, N_c, which decides if y, $|y| \leq |x|^k + k$, *is* or *is not* in A. N_c first guesses n_x strings up to length $|x|^k + k$ and then guesses the verification that these strings are in A. For the right guesses, N_c will find exactly n_x strings in A and then it just has to check if y is or is not one of these strings.

Now D constructs the description of an NE machine, N_E, which simulates the NP machine N^A on x. If N^A queries the oracle A, N_E uses N_c to obtain the answer (since N_c will give *yes* or *no* answers for the right sequence of guesses). Thus, N_E accepts x iff N^A accepts x. As a final query D^U asks U if N_E accepts x. Thus $L(N^A) = L(D^U)$ which shows at:

$$NP^{NE} = P^{NE},$$

and therefore SEH collapses to P^{NE}.

With this proof in mind, the reader is challenged to prove the interesting fact that polynomially many queries to SAT can be replaced by $0(logn)$ sequential queries to SAT. More precisely, let $P^{SAT||}$ denote the set of languages accepted by deterministic polynomial time machines which, for each input x, $|x| = n$, can construct *one* vector query of Boolean functions, $(F_1, F_2, \ldots, F_{p(n)})$ and receive as answer a binary vector indicating which F_i are and are not satisfiable. $P^{SAT[0(logn)]}$ denotes the set of languages acceptable by a deterministic polynomial time machine which can make an input x $0[log\,|x|]$ queries to SAT. The following fact has been observed by several people, for example, see Therem 3.10 in [Hem 87].

Fact:

$$P^{SAT[0(logn)]} = P^{SAT||}$$

Another exponential hierarchy, EH, can be defined analogously to the PH either by using alternations of exponentially bounded quantifiers or as an oracle hierarchy as follows:

$$E,\ NE,\ NE^{NP},\ NE^{NP^{NP}},\ \ldots$$

which can be written as:

$$E,\ NE,\ NE^{\Sigma_1^p},\ NE^{\Sigma_2^p},\ \ldots$$

This hierarchy was discussed, for example, in [HIS 85].

Though we now know that

$$NP^{NE} = P^{NE},$$

the same proof does not show that

$$NE^{NP} = E^{NP}$$

and cannot collapse this hierarchy.

The key difference is that a P^{NE} machine can find the census function for an A in NE (by binary search successively guessing and verifying that there are at least a given number of strings in A up to a given length).

An E^{NP} machine, on the other hand, cannot find the census function of an A in NP by the same method, because an NP machine on an accepting pass can make at most polynomially many guesses and there may be many more strings in A up to length n.

Furthermore, it can be shown that this hierarchy is infinite iff the PH restricted to sparse sets is infinite [HIS 85]. In other words:

$$NE^{\Sigma_k^P} \neq NE^{\Sigma_{k-1}^P} \text{ iff there are sparse sets in } \Sigma_{k+1}^P - \Sigma_k^P.$$

We know of no oracle construction which separates EH from $EXSPACE$ nor one which makes it infinite. Nor do we know what happens to EH with probability one for random oracles.

In retrospect, the strong exponential hierarchy, SEH, turned out to be "weak" and the above hierarchy, EH, has turned out to be the better or true exponential analogue of PH.

Space Bounded Hierarchies

Several other hierarchies based on nondeterministic space bounded computations were defined and investigated following the polynomial hierarchy model. To our pleasant surprise, several of them were shown to be finite by S. Toda [To 87] and K-J. Lange, B. Jenner and B. Kirsig [LJK 87]. A very nice extension of these results and unified, easy proofs of them are given in [Sch W 87].

These results were dramatically superseded by Neil Immerman's unexpected and elegantly simple proof that nondeterministic space bounded computations are closed under complement [Imm 87]. As a corollary of this result we now know that the context-sensitive language [AHU 74, HH 74] are closed under complement; a problem which has been open since 1964. This result also shows decisively that all the nondeterministic space bounded hierarchies collapse to the first level, improving the results of [To 87], [LJK 87] and [Sch W 87].

Theorem (Immerman):

For all space constructible $s(n) \geq \log n$

$$NSPACE[s(n)] = coNSPACE[s(n)].$$

Proof Outline:

We outline the two key ideas of the proof on the special case:

$$NSPACE[n] = coNSPACE[n].$$

The first idea is that if for any $NSPACE[n]$ machine N another $NSPACE[n]$ machine N_c could compute for each x the exact number of distinct configurations N can reach from x, then an $NSPACE[n]$ machine N' could recognize $\overline{L(N)}$. Let n_x be the number of configurations $N(x)$ can reach. The recognition of $\overline{L(N)}$ by N' is done as follows: N' on x computes n_x and then cycles successively through all possible sequences $N(x)$ could reach and checks for each sequence if $N(x)$ reaches it. For the right sequence of guesses $N(x)$ will reach n_x distinct configurations and x is in $\overline{L(N)}$ iff none of these configurations is an accepting configuration of N.

The proof that the number of reachable configurations is $NSPACE[n]$ computable is shown by induction on the number of steps to reach a configuration. Let d_t be the number of configurations reached by $N(x)$ in t steps. We will describe N_c which computes d_t. Clearly, d_1 is easily computable. Given d_t, N_c will successively check for each sequence if it can be reached in one step from one of the d_t configurations reached in t steps. To do this, for each target sequence y N_c tries to guess successively d_t configurations reachable from x in t steps, and tries to verify that they are so reachable. If d_t such sequences are found and y is not reachable from any of them in one step, then go to the next target sequence, if y is reachable, add one to the d_{t+1} counter and go to next y. Combining both results we get that $NSPACE[n] = \overline{NSPACE[n]}$.

Michael Fischer has observed that if $NSPACE[s(n)]$ is closed under complement then one can easily diagonalize over these classes to get a very sharp hierarchy result just as was done for deterministic tape bounded classes [SHL 65].

Theorem: For any tape constructible $s(n) \geq logn$

$$\lim_{n\to\infty} \frac{t(n)}{s(n)} = 0$$

implies

$$NSPACE[t(n)] \neq NSPACE[s(n)].$$

It should be recalled that before we knew that these classes were closed under complement, we could not diagonalize over them and only much weaker separation results were known. Furthermore, these results were obtained in a cumbersome, ad hoc manner using translation lemmas. To appreciate the simplification of these proofs, see Chapter 12 in [AHU 74] for the cumbersome old proofs.

Historically, it is interesting to recall that Mahaney's proof that the existence of sparse many-one complete NP sets implies that $P = NP$ also exploited the census functions [Mah 80, Mah 82]. In this case, the exact census function could not be computed, but the proof used a pseudo census function and cycled through all possible values of the census function for the sparse set. An early discussion of the use of the census function can be found in [HM 80].

Similarly, the proof that $EXPTIME \neq NEXTIME$ iff there exist sparse sets in $NP - P$ uses the census function [HIS 85] and so did J. Kadin's optimal collapse of PH to $P^{SAT[0(log n)]}$ under the assumption that these exist sparse S in NP such that $NP \subseteq P^S$ [Kad 87, Har 87]. It was the proof of this last result which inspired Schöning and Wagner [Sch W 87] to derive their simple proof of the various collapses of hierarchies, some of which were superseded by Immerman's closure result [Imm 87].

Finally, one is struck by the exhaustive use of nondeterminism in Immerman's proof to compute the complement of an $NSPACE[s(n)]$ language. This brutal use of nondeterminism is reminiscent of the Book and Greibach proof that for nondeterministic time computations "two tapes are as good as any number of tapes" [BGW 70].

Boolean, Query, and Polynomial Hierarchies

One of the most recent additions to the arsenal of computational complexity hierarchies was the Boolean Hierarchy, BH, and the intertwined Query Hierarchy, QH. Both of these hierarchies have been intensively investigated and they have escaped the fate of the collapsing hierarchies. As a matter of fact, a very recent result by J. Kadin shows that if these hierarchies collapse then so does the classic PH [Kad 87a]. Since it is widely believed that PH is infinite we have to assume that BH and QH do not collapse either.

The $k-th$ level of the Query Hierarchy is given by the language accepted with k queries to SAT:

$$QH_k = P^{SAT[k]}, \ k \geq 0.$$

The Boolean Hierarchy is built up by Boolean operations over NP languages and we will not discuss here the various and interesting normal forms. Suffice to say that the first level of the BH is given by the well known class D^P defined by Papadimitriou and Yannakakis [PY 82] and it is related directly to optimization problems. A complete language for D^P is:

$$\Delta = \{F\#G \mid F \text{ in } SAT \text{ and } G \text{ in } \overline{SAT}\}.$$

Furthermore, it can be shown that QH is finite iff BH is finite.

Theorem (J. Kadin):

If the Boolean or Query Hierarchies are finite then PH is finite and

$$PH \subseteq \Delta_3^P = P^{NP^{NP}}.$$

Proof Outline:

We will sketch the key ideas for the proof that $D^P = coD^P$ implies $PH = \Delta_3^P$. The general case iterates this method and cleverly exploits special normal forms of BH to show that the same collapse of PH is implied by any collapse of BH.

Since

$$\Delta = \{F\#G \mid F \text{ in } SAT \text{ and } G \text{ in } \overline{SAT}\}$$

is a complete language for D^P, the "cleaned up complement" of Δ is complete for coD^P,

$$\tilde{\Delta} = \{F \# G \mid F \in \overline{SAT} \text{ or } G \in SAT\}.$$

Note that Δ looks like a harder language since we must test for F in SAT and G in $\overline{SAT}$. In $\tilde{\Delta}$ the *and* is replaced by an *or* and thus it suffices to find F in $\overline{SAT}$ or G in SAT.

Clearly, the easy case is if G is in SAT. This simplification will now be exploited to extract a sparse set S such at:

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

and this forces the collapse of PH.

Our assumption that $D^P = coD^P$ implies that there is a polynomial time reduction, g, of Δ to $\bar{\Delta}$. Two cases may occur.

Case 1. For a G in $\overline{SAT}$ there may exist an F, $|F| = |G|$, such that

$$g(F \# G) = F' \# G'$$

and G' in SAT. In this case, G in $\overline{SAT}$ is recognized by NP methods: guess F and verify that G' is in SAT.

Case 2. No such luck: for G_0 in $\overline{SAT}$ and for all F, $|F| = |G_0|$,

$$g(F \# G_0) = F' \# G'$$

and G' not in SAT. But then, F in SAT iff F' in $\overline{SAT}$. Equivalently, F in $\overline{SAT}$ iff F' in SAT. Thus, given G_0, we can recognize for all F, of length $|G_0|$ if F is in $\overline{SAT}$ by checking if the corresponding F' (of $F' \# G'$) is in SAT.

Therefore, we see that $D^P = coD^P$ implies that for each n either Case 1 holds for all F, with $|F| = n$, or there is a G_0 such that Case 2 holds for all such F. Thus, there is a sparse oracle S which for length n contains the proper G_0, in Case 2, or O^n indicating Case 1.

We now know that

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

With a bit of dexterity with quantifiers one can see that S is in NP^{NP} and therefore it can be generated in $P^{NP^{NP}}$. Therefore, no additional NP oracles can help since with S an NP machine can decide if F is in SAT or $\overline{SAT}$. Thus,

$$PH \subseteq \Delta_3^P.$$

□

It is fascinating that in this result the collapse of BH implies the existence of a sparse set S such that

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

and that then this sparse set forces the collapse of PH.

In the previous Structural Complexity Column [Har 87], we discussed various results about how far PH collapses if there exist sparse oracles T such that

$$NP \in P^T.$$

Clearly, the reason for all these collapses is that a sparse set with properties which can be expressed in some level of the PH (for example $NP \in P^T$ or $\overline{SAT} \in NP^S$) can be computed in the hierarchy and thus forcing the collapse. Note that sparseness is essential for this to work because no more than polynomially many elements can be guessed and tested in this manner. Thus, PH can get hold of such sparse sets (or their equivalents), but this method does not work for denser sets. An extensive use of this method can be found in [BoCo 84] where it is shown among other things that:

PH is finite iff PH^S is finite for some sparse oracle.

It should also be mentioned that Ker-I Ko has just shown that there are oracles which collapse the Polynomial Time Hierarchy to exactly k levels, for $k \geq 0$ [Ko 87].

In this area, we still do not know whether PH for random oracles is infinite with probability one. Similarly, because of the link of PH restricted to sparse sets with EH, it would be interesting to see what happens to relativized PH on sparse sets or equivalently to EH.

Finally, in the previous Structural Complexity Column, the question was raised if the Karp-Lipton-Sipser collapse of PH to Σ_2^P due to polynomial size circuits for NP (or a sparse set S such that $NP \in P^S$) is optimal. Indeed, this question has been answered by Chris Wilson in [Wil 85]. In essence, Wilson shows that even if Δ_2^P would have linear size circuits, the proof of the collapse of PH below $\Sigma_2^P \cap \Pi_2^P$ would require non-relativizing proof techniques.

References

[BoCo 84] Book, R.V., J. Balcazar, T. Long, U. Schoning and A. Selman. "Sparse Oracles and Uniform Complexity Classes", In *Proceedings 25th IEEE FOCS*, 308-311, 1984.

[BGW 70] Book, R.V., S.A. Greibach and B. Wegbreit. "Time- and Tape-Bounded Turing Acceptors and AFL's". *JCSS* 4:606-621, 1970.

[Har 87] Hartmanis, J. "Sparse Complete Sets for NP and the Optimal Collapses of the Polynomial Time Hierarchy, Structural Complexity Column". *Bulletin of EATCS*, 39:73-81, June 1987.

[Hem 87] Hemachandra. L. "The strong exponential hierarchy collapses". In *19th ACM Symposium on Theory of Computing*, 110-122, May 1987,

[HH 74] Hartmanis, J. and H.B. Hunt, III. "The LBA Problem and its Importance in the Theory of Computing", In *Complexity of Computation* (ed. R. Karp), SIAM-AMS Proceedings 7: 1-26, 1974.

[HIS 85] Hartmanis, J., N. Immerman and V. Sewelson. "Sparse sets in $NP - P$: $EXPTIME$ versus $NEXPTIME$". *Information and Control*, 65(2/3):159-181, May/June 1985.

[HM 80] Hartmanis, J. and S. Mahaney. On the "Census Complexity and Sparseness of NP-Complete Sets", Cornell University, Computer Science Department Technical Report, TR 80-416, April 1980.

[Imm 87] Immerman, N. "Nondeterministic Space is Close Under Complement", submitted for publication (also Yale University Technical Report), August 1987.

[Kad 87] Kadin, J. "$P^{NP[logn]}$ and Sparse Turing-Complete Sets for NP", In *Proceedings 2nd Structure in Complexity Theory Conference*, IEEE Computer Society Press, 33-40, June 1987.

[Kad 87a] Kadin, J. "The Polynomial Hierarchy Collapses if the Boolean Hierarchy Collapses.", Cornell University, Department of Computer Science Technical Report, TR 87-843, June 1987.

[KO 87] Ko, K. "Relativized Polynomial Time Hierarchies Having Exactly K Levels," Manuscript, June 24, 1987.

[LKJ 87] Lange, K., B. Jenner, and B. Kirsig. "The logarithmic alternation hierarchy collapses: $A\Sigma_2^L = A\Pi_2^L$". In *Automata, Languages, and Programming (ICALP 1987)*, Springer-Verlag *Lecture Notes in Computer Science* 267:529-541, 1987.

[Mah 80] Mahaney, S. "Sparse Complete Sets for *NP*: Solution of a Conjecture of Berman and Hartmanis". In *Proceedings 21st IEEE Symposium on Foundations of Computer Science*, 54-60, 1980.

[Mah 82] Mahaney, S. "Sparse Complete Sets for *NP*: Solution of a Conjecture of Berman and Hartmanis". *Journal of Computer and System Sciences* 25(2): 130-143, 1982.

[PY 82] Papadimitriou, C. H. and M. Yannakakis. "The Complexity of Facets (and Some Facets of Complexity)", *JCSS*, 28:244-259, 1982.

[Rog 67] Rogers, Jr., H. *The Theory of Recursive Functions and Effective Computability* McGraw-Hill, 1967.

[Sch W 87] Schöning, U. and K. Wagner. Collapsing Oracle Hierarchies, Census Functions, and Logarithmically Many Queries. Technical Report 140, Mathematics Institute, University of Augsburg, May 1987.

[SHL 65] Stearns, R.E., J. Hartmanis, and P.M. Lewis II. "Hierarchies of Memory Limited Computations". *IEEE Conference on Switch Circuit Theory and Logical Design*, 179-190, 1965.

[Sto 77] Stockmeyer, L. "The Polynomial-Time Hierarchy". *Theoretical Computer Sci-*

ence, 31-22, 1977.

[To 87] Toda, S. "Σ_2 $SPACE(n)$ is Closed Under Complement". Submitted for publication. 1987.

[Wil 85] Wilson, C. B. "Relativized Circuit Complexity", *JCSS*, 169-181, 1985.

[Wra 77] Wrathall, C. "Complete Sets and the Polynomial-Time Hierarchy", *Theoretical Computer Science*, 3:23-33, 1977.