New Developments
in
Structural Complexity Theory

Juris Hartmanis*

88-908
April 1988

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

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

New Developments in Structural Complexity Theory

J. Hartmanis*

Department of Computer Science

Cornell University

Ithaca, New York 14853

Abstract

This paper discusses the scope and goals of structural complexity theory, describes some working hypothesis of this field and summarizes (some) recent developments.

Introduction

Structural complexity theory is the study of the relations between various complexity classes and the global properties of individual complexity classes. It studies logical implications among certain unsolved problems about complexity classes, explores the power of various resource bounded reductions and the properties of the corresponding complete languages. It uses relativization to clarify the power of different access mechanisms to information, to investigate possible relationships among complexity classes and to try to assess the difficulty of unsolved problems by contradictory relativizations.

The goal of structural complexity theory is a thorough understanding of the relations between the various complexity classes and the internal structure of these complexity classes.

During the last decade, structural complexity theory has emerged as a cohesive subfield of complexity theory with a rich set of interlocking problems and results. In particular, the last few years have been very productive and exciting [Mah, Se, Ha1, Ha2, Ha3].

In the following, we describe some of these developments and their impact on structural complexity theory.

Major Complexity Classes and Hierarchies

We recall some definitions. Let P and NP denote the sets of languages accepted in deterministic and nondeterministic polynomial time, respectively. Clearly, P and NP are the dominant complexity classes in computer science and the $P = ?\, NP$ question is the most famous open problem in our field.

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

The Polynomial Time Hierarchy, *PH*, is defined in terms of *NP* and is essential in the classification of problems with more complex logical structure than *NP*.

$$\Sigma_0^P = \Pi_0^P = \Delta_0^P = P,$$

$$\Sigma_1^P = NP, \quad \Pi_1^P = coNP, \quad \Delta_1^P = P,$$

for $i > 1$,

$$\Sigma_i^P = NP^{\Sigma_{i-1}^P}, \quad \Pi_i^P = co\ \Sigma_i^P, \quad \Delta_i^P = P^{\Sigma_{i-1}^P}.$$

$$PH = \bigcup_{i \geq 1} \Sigma_i^P.$$

Equivalently, this hierarchy can be defined (in analogy to the Kleene Hierarchy) in terms of polynomially bounded quantifiers and polynomial time computable predicates [St, Wr, HU].

For example,

$L \in \Sigma_2^P$ *iff* $L = \{x \mid (\exists^P y_1)(\forall^P y_2)\, R(x, y_1, y_2)\}$,

where $\exists^P, \forall^P$ indicates that length of y_1, y_2 ($|y_1|$, $|y_2|$) is polynomially bounded in $|x|$ and $R(\cdot,\cdot,\cdot)$ is a polynomial time computable predicate.

Let

$$SPACE[L(n)] \text{ and } NSPACE[L(n)]$$

denote the sets of languages accepted by deterministic and nondeterministic $L(n)$-space bounded machines, respectively.

Let

$$PSPACE = \bigcup_{k \geq 1} SPACE[n^k] \text{ and } NPSPACE = \bigcup_{k \geq 1} NSPACE[n^k].$$

From Savitch's result [Sa], we know that

$$NPSPACE = PSPACE,$$

contrary to our strong belief that

$$P \neq NP.$$

Let

$$E = \bigcup_{k \geq 1} TIME[2^{kn}], \qquad NE = \bigcup_{k \geq 1} TIME[k^{kn}]$$

and

$$ESPACE = \bigcup_{k \geq 1} SPACE[2^{kn}].$$

Many important computational problems lie between *NP* and Σ_2^P. To obtain a "fine structure" of the problems in this range, two intertwined hierarchies have been defined. The Boolean Hierarchy, *BH*, is built up by Boolean operations on *NP* sets.

$BH(1) = NP$,

$BH(2i) = \{L \mid L = L_1 \cap \overline{L}_2 \text{ where } L_1 \in BH(2i-1),\ L_2 \in NP\}$,

$$BH(2i+1) = \{L \mid L = L_1 \cup L_2 \text{ where } L_1 \in BH(2i), L_2 \in NP\},$$
$$BH = \bigcup_{k \geq 1} BH(k).$$

For L_i in NP, we obtain
$BH(1) = NP, \; BH(2) = \{L_1 \cap \bar{L}_2\} = D^P,$

$BH(3) = \{(L_1 \cap \bar{L}_2) \cup L_3\}, \; BH(4) = \{((L_1 \cap \bar{L}_2) \cup L_3) \cap \bar{L}_4\}, \ldots$

The Query Hierarchy is defined in terms of the number of queries a p-time machine can make to *SAT* (or any *NP* complete problem).

$P^{SAT[k]}$ is the set of languages accepted by deterministic polynomial time machines making no more than *k* queries to *SAT*.

The Query Hierarchy, *QH*, is given by:

$$QH = \bigcup_{k \geq 1} P^{SAT[k]}.$$

It can be shown that the two hierarchies, *BH*, and *QH*, are intertwined and therefore *BH* is finite if and only if *QH* is finite [CH1, CH2].

The importance of these hierarchies has been substantially enhanced by the recent proof that if *BH* (or *QH*) is finite than so is *PH* [Ka2, Ka3]. Thus, by Working Hypothesis A (as stated later) we must assume that both of these hierarchies are infinite. We will return to these problems later.

Finally, let $P^{SAT[logn]}$ denote the class of languages accepted by p-time machines with $O(logn)$ queries to *SAT*. This class contains many important optimization problems solvable by binary search methods. For example, maximal clique size in a graph, chromatic number for graphs, etc. In the following, we will see that this class also plays an important role in structural complexity work.

The work in structural complexity theory proceeds under two working hypothesis. The first hypothesis is based on the strong conviction that $P \neq NP$ and the more sweeping conjecture that *PH* is infinite.

Working Hypothesis A: Any assumption that implies that $P = NP$ or that *PH* is finite is assumed to be false.

The second hypothesis is derived from the fact that many problems in complexity theory have contradictory relativizations and that no (non-trivial) problems with double relativizations have been solved [Ha4, Ho]. The classic example for such a relativization is the Baker-Gill-Solovay [BGS] result that there exist recursive oracles A and B such that:

$$P^A = NP^A \text{ and } P^B \neq NP^B.$$

Clearly, this result indicates that the $P =? NP$ problem cannot be solved by any proof techniques that relativize. This leads to our second hypothesis.

Working Hypothesis B: Any (non-trivial) problem that has contradictory relativizations is a very hard problem and cannot be solved with "current" techniques.

In an intuitive sense, a contradictory relativization is a *mini-independence result.* This has not yet been made formally precise, but we believe that it is important to clarify the implications of contradictory relativization. Particularly, since so many problems in complexity theory have contradictory relativizations. What *independence* is implied from what *proof techniques* by contradictory relativizations?

We will return to these problems when we discuss space bounded computations.

Time Bounded Computations

The last few years have been a very exciting and successful time in the development of structural complexity theory. New and interesting structural connections have been revealed between various complexity classes. New hierarchies have been defined and studied, to clarify the fine structure of feasible problems below Σ_2^P, and relations between these hierarchies and the classic polynomial hierarchy have been discovered. Some other hierarchies have been shown to collapse and surprising new insights have been gained about nondeterministic space bounded computations. Furthermore, after a careful study, many of these new results can be obtained by quite similar *counting* or *census arguments*, which yield a nice unification and deeper understanding of structural complexity. We review some of these results in this section.

There are several other interesting developments in structural complexity theory which we will not review here [GJ, Ha5, KMR, Ma2, MY]. Some very nice new insights have emerged from the study of the structure of the sets of complete problems of various complexity classes. Some of this work is motivated by the Isomorphism Conjecture [BH] which expressed belief in a beautifully simple structure of the sets of complete problems for *NP*. The recent work on complete problems for *NP* and higher complexity classes has revealed an unexpectedly rich structure of some of these sets. These problems are far from fully understood and the new results have opened an important and interesting area of research. We view the full understanding of the structure of sets of complete problems for various complexity classes as one of the major challenges for structural complexity theory.

To review some of the new developments in more detail, we recall several definitions.

A set S, $S \subseteq \Sigma^*$, is *sparse* if there exists a polynomial p such that:

$$(\forall n) \,|\{x \mid |x| \le n \text{ and } x \in S\}| \le p(n).$$

$A \subseteq \Sigma^*$ is *many-one reducible* to S if there exists a p-time computable function f such that:

$$(\forall x)\,[x \in A \Leftrightarrow f(x) \in S].$$

We write $A \le_m^P S$.

S is *many-one complete* for *NP* if S is in *NP* and for all B in *NP* $B \le_m^P S$.

Many results in structural complexity theory are expressed in terms of sparse sets. The importance of sparse sets is caused by the ability of p-time computations to count only up to a polynomial in the length of the input. Therefore, as we will see, sparse sets play a special role in the study of feasible time limited computations. If binary search arguments can be used, then the class $P^{SAT[\log n]}$ naturally enters structural complexity results.

A very elegant use of a counting argument and sparse sets can be found in Mahaney's proof [Ma1] of the following result.

Theorem (Mahaney): If a sparse set is $\leq_m^P$-complete for NP then $P = NP$.

If we postulate the existence of a sparse set S such that:

$$NP \subseteq P^S,$$

then we get, under this weaker hypothesis, a less complete collapse of the polynomial hierarchy [KL].

Theorem (Karp, Lipton, Sipser): If S is sparse and $NP \subseteq P^S$ then:

$$PH \subseteq \Sigma_2^P.$$

This result has been strengthened under stronger assumptions in several ways [Lo, Ma2, Ya]. If we assume that S is in NP, then by a nice counting argument Kadin obtained the optimal result [Ka1].

Theorem (Kadin): If S is sparse, S in NP and

$$NP \subseteq P^S,$$

then

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

The proof is based on the subtle observation that in $0(logn)$ 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 guessing $C_S(n)$ 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 $0(logn)$ queries we can accept any L in P^{SAT}. Thus,

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

This proof explicitly 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 $PH \subseteq P^{SAT\,0[logC_S(n)]}$.

This result is optimal, in the sense that there are relativized worlds in which this collapse is optimal [Ka1, Ka2].

The existence of sparse $\leq_m^P$ - complete sets for NP forced $P = NP$. The weaker assumptions of $NP \subseteq P^S$ for a sparse set forced $PH \subseteq \Sigma_2^P$ and if assumed that S is in NP then $PH \subseteq P^{SAT[logn]}$. We now explore what additional conditions will force a complete collapse, $P = NP$, in the last two results. We recall two results [HIS, HY].

Theorem: $E \neq NE$ iff there exists a sparse sets S in $NP - P$.

Theorem: $E \neq ESPACE$ iff there exist a language with polynomial size circuits in $PSPACE - P$.

Combining these results with the previous results we get the following "downward collapse" results.

Theorem:

1. $(\exists \text{ sparse } S \text{ in } NP)\ [NP \subseteq P^S \text{ and } E = NE]$
$$\Leftrightarrow$$
$$P = NP$$

2. $(\exists \text{ sparse } S)\ [NP \subseteq P^S \text{ and } E = NE^{NP}]$
$$\Leftrightarrow$$
$$P = NP$$

3. $(\exists \text{ sparse } S)\ [PSPACE \subseteq P^S \text{ and } E = ESPACE]$
$$\Leftrightarrow$$
$$P = PSPACE.$$

Note: It is not known if $E = NE$ implies $E = NE^{NP}$. There exists relativized worlds with $E = NE$, but $E \neq NE^{NP}$ [HIS].

In analogy with the polynomial hierarchy two exponential hierarchies have been studied and one has been shown to be finite.

The *strong exponential hierarchy*, *SH*, was defined as follows:

$$\Sigma_0^E = \Pi_0^E = E$$

$$\Sigma_1^E = NE, \qquad \Pi_1^E = coNE$$

$$\Sigma_k^E = NP^{\Sigma_{k-1}^E}, \qquad \Pi_k^E = co\Sigma_k^E$$

$$SH = \bigcup_{k \geq 1} \Sigma_k^E.$$

For example,

$$\Sigma_3^E = NP^{NP^{NE}}.$$

Quite surprisingly, Hemachandra has shown that this hierarchy is finite [He].

Theorem (Hemachandra):

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

The proof that $NP^{NE} = P^{NE}$ is very similar to Kadin's proof that a sparse S in NP such that $NP \subseteq P^S$ implies $NP^{NP} = P^S = P^{SAT[\log n]}$.

The only difference in the proof is that the oracle A in NE does not have to be sparse since we have polynomially many queries to compute its census; i.e. $CENSUS_A(n^k)$ is computable by a P^{NE} machine. Since the oracle can perform NE computations, it can guess $CENSUS_A(n^k)$ many strings in A and verify that it has all relevant strings. A p-machine computes such a machine which uses $CENSUS_A(n^k)$. Then, with one more query, to NE the p-machine determines if this machine accepts. For details of this proof see [Ha3, SW].

The other exponential hierarchy defined by NE, NE^{NP}, $NE^{NP^{NP}}$, etc., is not known to collapse. We know that this hierarchy is infinite iff the polynomial hierarchy is infinite when restricted to sparse sets [HIS].

Some related results can be obtained by the similar counting argument. Let $P^{SAT||}$ denote the set of languages accepted by a p-time machine which can ask one parallel query for each input, i.e., the query is a vector of formulas $(F_1, F_2, \ldots, F_{p(n)})$ and the answer is a vector showing which formulas are satisfiable and which are not. In [He] it is shown that:

$$P^{SAT||} = P^{SAT[\log n]}.$$

Clearly, this exponential saving in sequential queries over parallel queries is heavily dependent on the structure of SAT. This result fails with probability one for random oracles [Ka2].

A surprising version of this result for constant number of queries (with a much harder proof) has been obtained by Beigel [Be].

For all $k \geq 1$

$$P^{SAT[k]} = P^{SAT[2^k-1]||}.$$

Next, we consider the Boolean hierarchy and the Query hierarchy. These hierarchies gained considerable respectability when Kadin showed that their collapse implies that the Polynomial hierarchy collapses [Ka3].

Theorem (Kadin): If for any $k \geq 1$ $P^{SAT[k]} = P^{SAT[k+1]}$ then $PH \subseteq \Delta_3^P$.

The collapse of the BH or QH has a very strong structural implication for NP and $coNP$. The collapse of the hierarchies forces NP and $coNP$ to differ by a sparse set only.

Corollary (Kadin): If BH or QH are finite, then there exists a sparse set S such that

$$coNP \subseteq NP^S.$$

We can strengthen this to an if and only if result by an additional assumption which collapses NP and $coNP$.

Theorem: $|BH| < \infty$ and $NE \cap coNE = NE^{NP}$

$$\Leftrightarrow$$

$$NP = coNP.$$

Space Bounded Computations

Space bounded computations have the capability of reusing their tape and to count up to exponentially large numbers in the length of the work tape. Both these properties endow space bounded computations with capabilities which we cannot prove for time bounded computations. It is also interesting to note that time and space bounded computations behave differently under relativization. For properly chosen relativization models deterministic and nondeterministic space cannot be separated if

$$SPACE[\log n] = NSPACE[\log n].$$

We will return to this phenomenon later.

Already Savitch's result that deterministic and nondeterministic space is polynomially related verified a property for space which we do not believe to hold for time bounded computations [Sa, HU].

Theorem (Savitch): For space constructible $L(n) \geq \log n$

$$NSPACE[L(n)] \subseteq SPACE[L(n)^2].$$

This clearly implies that $PSPACE = NSPACE$. On the other hand, we strongly believe that $P \neq NP$.

During 1987 several space bounded hierarchies were shown to be finite [LKJ, To]. These results were dramatically superseded by simultaneous Immerman and Szelepcsenyi proofs that nondeterministic space is closed under complement [Imm, Sz].

Theorem (Immerman, Szelepcsenyi): For space constructible $L(n) \geq \log n$

$$NSPACE[L(n)] = \overline{NSPACE[L(n)]}.$$

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

$$NSPACE[L(n)] = coNSPACE[n]$$

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^1 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^1 is done as follows: N^1 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 the next y. Combining both results we get that $NSPACE[n] = \overline{NSPACE[n]}$.

□

It is quite surprising that this important result with such an elegant proof remained unsolved for over twenty years, in spite of considerable interest in this area. In retrospect, it seems that few people expected nondeterministic space to be closed under complement and that we just did not fully realize the power of "counting" in this context. It also seems that the difficulty of separating complexity classes and the contradictory relativization results hinting that these problems are very hard, shied people away from serious effort to solve this and other separation problems.

It is ironic, that an attempt to relativize this problem could have revealed that it is not hiding behind possible contradictory relativizations and thus could be expected to be solvable by known techniques. In the following, we discuss relativization of space bounded computations. We are interested in what relativization says about space bounded computations and vice versa.

There are several options for the access mechanism to the oracle for space bounded computations [LL, RST]. For deterministic and nondeterministic log(n) space bounded computations, we can allow both types of machines to use an additional one-way-write-only oracle tape on which they can write n^k-long queries. For brevity, denote the corresponding language classes by LOG^A and $NLOG^A$, respectively.

It is easily shown that:

$$LOG \subseteq NLOG \subseteq P.$$

But using a straightforward Baker-Gill-Soloway oracle construction [BGS, LL], one can construct an oracle A such that:

$$NLOG^A \not\subseteq P^A.$$

Similarly, we can construct oracles B and C which for this model of relativization invalidate Savitch's result:

$$NLOG^B \not\subseteq SPACE^B\,[(log(n))^2]$$

as well as the Immerman-Szelepcsenyi result,

$$NLOG^C \neq \overline{NLOG^C}.$$

To avoid these anomalies, another somewhat artificial oracle access mechanism was defined in [RST]. In this model, again there is an additional one-way-write-only oracle tape, but it is required that the computation of the oracle query must be done deterministically by both types of machines. Denote the corresponding language classes by $LOG^{D(A)}$ and $NLOG^{D(A)}$. Now both types of machines can for any given input query only polynomially many strings in A and the above anomalies are avoided. On the other hand, when we consider linear tape bounds, this model permits exponentially long queries in the input length and looks artificial.

Finally, the simplest and most natural oracle access mechanism is to limit the oracle queries to the length of the work tape. Denote the corresponding language classes by $LOG^{S(A)}$ and $NLOG^{S(A)}$. In this model, the nondeterministic machines can use nondeterministically computed queries.

The one objection to this model is that there exist oracles A such that:

$$A \notin NLOG^{S(A)}$$

since only log(n) long queries are possible on inputs of length n. This anomaly disappears for linear and larger tape bounds.

In the following we will show that, quite surprisingly, the $D(\cdot)$ and $S(\cdot)$ models of relativization behave very similarly when we deal with the possibility of separating deterministic and nondeterministic space bounded classes.

In [Wi] it was shown that:

$$LOG = NLOG \Leftrightarrow (\forall A)\ [LOG^{D(A)} = NLOG^{D(A)}].$$

This is a generalization of a result in [Si] and later in [RS]. A later publication of this result can also be found in [KiLa].

Our extensions of this result are summarized in the following two theorems.

Theorem: For space constructible $L(n) \geq log(n)$,

$LOG = NLOG$

$$\Leftrightarrow (\forall\ L(n),A)\ \ [SPACE^{S(A)}L\ [L(n)] = NSPACE^{S(A)}\ [L(n)]\ \ ]$$
$$\Leftrightarrow (\forall\ L(n),A)\ \ [SPACE^{D(A)}L\ [L(n)] = NSPACE^{D(A)}\ [L(n)]\ \ ].$$

To outline the proof of this theorem, we recall a few definitions and results.

Let GAP represent the set of directed graphs with an IN and OUT node such that there is a directed path from IN to OUT. The following was observed in [Sa].

Lemma (Savitch): GAP is complete for $NLOG$ under log(n)-space bounded reductions.

Proof: Clearly, GAP is in $NLOG$. To see that A in $NLOG$ can be reduced to GAP, let $L(N) = A$, where N is an $NLOG$ machine. For any x, we will construct a directed graph, $G_{x,N}$, polynomially large in the size of x, such that $x \in A$ iff $G_{x,N} \in GAP$. The nodes of the graph are the configurations of N: the state of the machine, the content of the work tape and the head positions on the input tape and work tape. There is a directed edge from node a to node b iff there is a legal move from configuration a to configuration b when N is working on input x. Note that x is not represented in the configurations, only the single symbol read by the input head.

Clearly, there are only polynomially many different configurations for N on x in the length of x and the graph can be computed by a deterministic log(n)-space bounded machine from input x (and printed on a one-way output tape). x is in $L(N)$ iff there is a directed path in $G_{X,N}$ from the "starting" node to the unique "accepting" node or from IN to the OUT node.

$\square$

We now show that in the $S(\cdot)$ model and $D(\cdot)$ model an oracle cannot separate LOG from $NLOG$ if $LOG = NLOG$.

Lemma (Wilson): $LOG = NLOG \Leftrightarrow (\forall\ A)\ [LOG^{S(A)} = NLOG^{S(A)}\ ]$.

Proof: The "only if" part of the proof is obvious when we set $A = \emptyset$.

If $LOG = NLOG$ then GAP is in LOG. We now show that an $S(A)$ oracle computation cannot do any harm. For an $N^{S(A)}(x)$ machine we can again compute a directed graph of configurations, which are augmented by the query tape (of length log ($|x|$)) and transitions for both *YES* and *NO* oracle answers are included and so labeled. Again, $N^{S(A)}$ accepts iff there is a path in this graph from the *IN* to *OUT* nodes using the correct *YES* and *NO* edges as determined by A.

Since $LOG = NLOG$, a $LOG^{S(A)}$ machine can recognize such graphs. The deterministic machine proceeds as a deterministic *GAP* recognizer and consults its oracle on the query edges to decide which edges are legally present in the graph, as the computation demands. Therefore,

$$LOG^{S(A)} = NLOG^{S(A)}.$$

□

The same proof works for the $D(\cdot)$ relativization model used originally by Wilson.

This proof fails for the unrestricted oracle access mechanism in which a $NLOG^A$ machine can query exponentially many strings in A, while the LOG^A machine can query only polynomially many strings.

Next we extend this result to higher deterministic and nondeterministic tape bounded classes. Not to obscure the simplicity (or elegance) of these ideas we discuss only the linearly bounded tape classes. With a few technical details, the same proof can be used for all other space constrictible bounds above $L(n) \geq log(n)$.

Lemma: $LOG = NLOG \Rightarrow (\forall A)[SPACE^{S(A)}[n] = NSPACE^{S(A)}[n]]$.

Proof: Let the linearly space bounded machines again have a read only input tape, a linearly bounded read-write work tape and a separate linearly bounded oracle tape.

Given a nondeterministic n-space bounded machine $N_i^{S(A)}$ and input x, we construct an $NLOG^{S(A)}$ machine $N_{\sigma(i)}^{S(A)}$ such that $N_i^{S(A)}$ accepts x iff $N_{\sigma(i)}^{S(A)}$ accepts $x\#^{2^{|x|}-|x|}$. By the previous lemma we know that for $N_{\sigma(i)}^{S(A)}$ there exists an equivalent deterministic log(n)-space bounded machine $M_{p(i)}^{S(A)}$. But then there exists a deterministic linear-space bounded machine $M_{\delta(i)}^{S(A)}$ equivalent to $N_i^{S(A)}$. Thus, $SPACE^{S(A)}[n] = NSPACE^{S(A)}[n]$.

□

The same proof extends to all space constructible $L(n) \geq log(n)$ as well as to the $D(A)$ model. But not to the unrestricted oracle access model.

Theorem: If $LOG \neq NLOG$ then for all space constrictible $L(n) \geq n$ there exist oracles A and B such that:

$$SPACE^{D(A)}[L(n)] = NSPACE^{D(A)}[L(n)],$$

$$SPACE^{S(A)}[L(n)] = NSPACE^{S(A)}[L(n)],$$

$$SPACE^{D(B)}[L(n)] \neq NSPACE^{D(B)}[L(n)],$$

$$SPACE^{S(B)}[L(n)] \neq NSPACE^{S(B)}[L(n)].$$

Proof: To illustrate the proof technique, we consider linearly space bounded computations for the $S(\cdot)$ model of relativization. We will show that $LOG \neq NLOG \Rightarrow (\exists A)[SPACE^{S(A)}[n] \neq NSPACE^{S(A)}[n]]$. We can think of linear-space bounded oracle machines as log(n)-space bounded machines whose input is given by the oracle. In particular, the oracle A will contain directed graphs with *IN* and *OUT* nodes. Since the query is linearly bounded, these machines on input 1^n can query the oracle about edges in these graphs for node descriptions up to size n. Clearly, this is equivalent to a log(n) space bounded machine with the same graph presented on the input tape. Since we assume that $LOG \neq NLOG$, this permits us to construct an A such that

$$\Delta = \{1^{n_i} \mid (\exists G_i)\,[\text{the node description is of size } n_i, \text{ the edge set of } G_i \text{ is in } A \text{ and } G_i \text{ is in } GAP\,]\}$$

and

$$\Delta \in NSPACE^{S(A)}[n] - SPACE^{S(A)}[n].$$

In the construction of A, to insure that the description of the G_i's do not interfere with each other, we pick a sufficiently rapidly growing sequence of integers $n_1 < n_2 < n_3 < \ldots$. For each n_i we choose a graph G_i with node description size n_i which witnesses that the i-th *LOG* machine cannot accept *GAP*. Thus,

$$G_i \in L(D_i) \Leftrightarrow G_i \notin GAP$$

and therefore

$$\Delta \in NSPACE^{S(A)}[n] - SPACE^{S(A)}[n].$$

Observe that the same oracle also separates $NSPACE^{D(A)}[n]$ from $SPACE^{D(A)}[n]$ on the same language Δ provided that we made the gaps between n_i and n_{i+1} sufficiently large so that on input 1^{n_i} the graph G_{i+1} cannot be accessed (even if the $D(A)$-oracle access mechanism allows 2^{n_i} long queries).

Finally, it is easily shown that for a *PSPACE* complete set R and space constrictible $L(n) \geq n$,

$$SPACE^{D(R)}[L(n)] = NSPACE^{D(R)}[L(n)]$$

and

$$SPACE^{S(R)}[L(n)] = NSPACE^{S(R)}[L(n)].$$

□

The above results clearly point out that under relativization the deterministic and nondeterministic space bounded computations behave completely different than the corresponding time bounded computations. This suggests that the

$$LOG =? NLOG$$

problem may be solvable by know techniques. The recent Immerman-Szelepcsenyi success of showing $NLOG = \overline{NLOG}$, similarly suggests that the space bounded computations may be easier to understand. We believe that it is an opportune time to renew an attack on the $LOG =? NLOG$ problem. It is probably simpler than the $P =? NP$ problem.

On the other hand, if $LOG \neq NLOG$ then there exist oracles which collapse and oracles which separate the higher ($L(n) \geq n$) deterministic and nondeterministic space bounded classes. This leads to the strange possibility that the proof of

$$LOG \neq NLOG$$

could be achieved by known proof techniques, but because of the contradictory relativization (even in the $S(\cdot)$-model) the

$$SPACE\,[n] =? NSPACE\,[n]$$

problem would not be so solvable.

Recall that

$$LOG \neq NLOG \Leftrightarrow GAP \text{ not in } LOG.$$

On the other hand, $NSPACE[n]$ and $SPACE[n]$ are different if and only if LOG and $NLOG$ can be separated on a sparse, easily computable subset of GAP. To see this, observe that any $NSPACE[n]$ computation can also be described by a graph in GAP. Except that this graph has nodes with description size n (as determined by the configurations of n-space bounded machine) and that the graph is computable from the input x, $|x| = n$, and the machine description. The graph is exponentially larger than $|x|$. To make this precise, let M_u be a standard universal TM and define [Ha6]

$$KS\,[\log(n), log(n)\,] = \{ x \mid (\exists y)\,[\,|y| \leq \log(|x|) \text{ and } M_u(y) = x \text{ is computed on } \log(|x|) \text{ tape }]\}.$$

Since the computation graphs for nondeterministic n-space bounded machines are computable from N and the input in polynomial time (in the size of the resulting graph), we obtain the following.

Lemma: $SPACE\,[n] \neq NSPACE\,[n] \Leftrightarrow GAP \cap KS\,[log(n), log(n)\,] \notin LOG.$

Thus $SPACE\,[n] \neq NSPACE\,[n]$ if and only if $NLOG$ and LOG are separated on the sparse set of graphs in GAP which can be quickly computed from exponentially shorter descriptions.

As stated earlier, it could happen that $NLOG \neq LOG$ and that this can be proven by known techniques, but that this would not extend to linearly space bounded computations. That is, the separation of LOG from $NLOG$ may not be easily provable on the sparse set

$$GAP \cap KS[log(n), log(n)].$$

This, for example, could happen if one could prove that *NLOG* and *LOG* differ on random graphs. Since random graphs are not compressible, this proof would not directly extend to separation of *NSPACE*[*n*] from *SPACE*[*n*].

References

[Be] Beigel, R. "Bounded Queries to *SAT* and the Boolean Hierarchy", manuscript, June 1987.

[BGS] Baker, T., J. Gill, and R. Solovay. "Relativizations of the $P =? NP$ Question", *SIAM Journal on Computing,* 4(4):431-442, December 1975.

[BH] Berman, L. and J. Hartmanis. "On Isomorphisms and Density of *NP* and Other Complete Sets", *SIAM Journal on Computing,* 6(2):305-322, June 1977.

[CH1] Cai, J. and L. Hemachandra. "The Boolean Hierarchy: Hardware Over *NP*", Technical Report, TR 85-724, Cornell Department of Computer Science, December 1985.

[CH2] Cai, J. and L. Hemachandra. "The Boolean Hierarchy: Hardware Over *NP*", *Structure in Complexity Theory,* pp. 105-124, Springer-Verlag *Lecture Notes in Computer Science,* No. 223, 1986.

[GJ] Goldsmith, J. and D. Joseph. "Three Results on the Polynomial Isomorphism of Complete Sets", *Proceedings IEEE Symposium on Foundations of Computer Science,* pp. 390-397, 1986.

[Ha1] Hartmanis, J. "The Structural Complexity Column: A Retrospective on Structural Complexity", *EATCS Bulletin,* 31:115, February 1987.

[Ha2] Hartmanis, J. "The Structural Complexity Column: Sparse Complete Sets for *NP* and the Optimal Collapse of the Polynomial Hierarchy", *EATCS Bulletin,* 32:73-81, June 1987.

[Ha3] Hartmanis, J. "The Structural Complexity Column: The Collapsing Hierarchies", *EATCS Bulletin,* 33:26-39, October 1987.

[Ha4] Hartmanis, J. "Solvable Problems with Conflicting Relativizations", *EATCS Bulletin,* 27:40-49, October 1985.

[Ha5] Hartmanis, J. "Some Observations about *NP* Complete Sets", *Fundamentals of Computational Theory,* Springer-Verlag *Lecture Notes in Computer Science,* 278:185-196, 1987.

[Ha6] Hartmanis, J. "Generalized Kolmogorov Complexity and the Structure of Feasible Computations", *Proceedings 24th Annual Symposium on Foundations of Computer Science, IEEE Computer Society,* 439-445.

[He] Hemachandra, L. "The Strong Exponential Hierarchy Collapses", *ACM Symposium of Theory of Computing,* pp. 110-122, 1987.

[HIS] Hartmanis, J., N. Immerman, and V. Sewelson. "Sparse Sets in *NP−P:EXPTIME* Versus *NEXPTIME*", *Information and Control,* 65:159-181, May/June 1985.

[Ho] Hopcroft, J.E. "Turing Machines", *Scientific American*, pp. 86-98, May 1984.

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

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

[Imm] Immerman, N. "Nondeterministic Space is Closed Under Complement", Yale University Technical Report, August 1987 (accepted for publication in SICOMP).

[Ka1] Kadin, J. "$P^{NP[logn]}$ and Sparse Turing Complete Sets for *NP*", *Proceedings 2nd Structure in Complexity Theory Conference*, pp. 33-40, Ithaca, New York, June 1987. Submitted to *Journal of Computer and Systems Sciences.*

[Ka2] Kadin, J. *Restricted Turing Reducibilities and the Structure of the Polynomial Time Hierarchy.* Ph.D. Thesis, Cornell University, February 1988.

[Ka3] Kadin, J. "The Polynomial Hierarchy Collapses if the Boolean Hierarchy Collapses", Cornell University, Department of Computer Science Technical Report, TR 87-843, June 1987.

[KiLa] Kirsig, B. and K. J. Lange. "Separation with the Ruzzo, Simon, and Tompa Relativization Implies $DSPACE[logn] \neq NSPACE[logn]$" *Information Processing Letters*, 25:13-15, 1987.

[KL] Karp, R. and R. Lipton. "Some Connections Between Nonuniform and Uniform Complexity Classes", *ACM Symposium on Theory of Computing*, pp. 302-309, 1980.

[KMR] Kurtz, S., S. Mahaney, and J. Royer. "Collapsing Degrees." *Proceedings IEEE Symposium on Foundations of Computer Science*, pp. 380-389, 1986.

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

[Lo] Long, T. "A Note on Sparse Oracles for *NP*", *Journal of Computer and System Sciences*, 24:224-232, 1982.

[LL] Ladner, R. E. and N. A. Lynch. "Relativization of Questions About Log Space Computability." *Mathematical Systems Theory*, 10:19-32, 1976.

[Ma1] 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.

[Ma2] Mahaney, S. "Sparse Sets and Reducibilities." *Studies in Complexity Theory,* ed. R.V. Book, John Wiley and Sons, Inc. New York, pp. 63-118, 1986.

[Mah] Mahaney S., ed. *Proceedings Structure in Complexity Theory*, Second Annual Conference Computer Society Press, 1987.

[MY]	Mahaney, S. and P. Young. "Reductions Among Polynomial Isomorphism Types", *Theoretical Computer Science*, 207-224, 1985.
[RS]	Rackoff, C.W. and J. I. Seiferas. "Limitations on Separating Nondeterministic Complexity Classes", *SIAM Journal on Computing,* 10:742-745, 1981.
[RST]	Ruzzo, W.L., J. Simon, and M. Tompa. "Space-Bounded Hierarchies and Probabilistic Computations", *Journal of Computer and Systems Sciences,* 28(4):216-230, November 1984.
[Sa]	Savitch, W.J. "Relationships Between Nondeterministic and Deterministic Tape Complexities", *Journal of Computer and Systems Sciences,* 4(2):177-192, 1970
[Se]	Selman, A.L., ed. *Proceedings Structure in Complexity Theory,* Springer Verlag *Lecture Notes in Computer Science*, No. 223, 1986.
[Si]	Simon, I. *On Some Subrecursive Reducibilities*. Ph.D. Thesis, Stanford University, March 1977.
[St]	Stockmeyer, L. "The Polynomial-Time Hierarchy", *Theoretical Computer Science*, 3:1-22, 1977.
[SW]	Schoening, U. and K.W. Wagner. "Collapsing Oracle Hierarchies, Census Functions, and Logarithmically Many Queries", *STACS '88* Springer-Verlag *Lecture Notes in Computer Science*, 294:91-97, 1988.
[Sz]	Szelepcsenyi, R. "The Method of Forcing for Nondeterministic Automata", *The Bulletin on the EATCS*, 33:96-100, October 1987.
[To]	Toda, S. "Σ_2 $SPACE(n)$ is Closed Under Complement", submitted for publication, 1987.
[Wi]	Wilson, C. "Relativized Circuit Complexity", *Journal of Computer and System Sciences,* 31(2):169-181, October 1985.
[Wr]	Wrathall, C. "Complete Sets and the Polynomial-Time Hierarchy", *Theoretical Computer Science*, 3:23-33, 1977.
[Ya]	Yap, C. "Some Consequences of Non-Uniform Conditions on Uniform Classes", *Theoretical Computer Science*, 26:287-300, 1983.