Computation Times of NP Sets of Different Densities

J. Hartmanis and Y. Yesha

TR 83-548
March 1983

Department of Computer Science
Cornell University
Ithaca, New York 14853

*This research was supported in part by National Science Foundation Grant MCS78-00418. Furthermore, the work of the second author was supported in part by Dr. Chaim Weizmann Post-Doctoral Fellowship for Scientific Research.

Presented at ICALP'83 Barcelona, Spain.

Computation Times of NP Sets of Different Densities

J. Hartmanis and Y. Yesha
Department of Computer Science
Cornell University
Ithaca, New York 14853

Abstract

In this paper we study the computational complexity of sets of different densities in NP. We show that the deterministic computation time for sets in NP can depend on their density if and only if there is a collapse or partial collapse of the corresponding higher nondeterministic and deterministic time bonded complexity classes. We show also that for NP sets of different densities there exist complete sets of the corresponding density under polynomial time Turing reductions. Finally, we show that these results can be interpreted as results about the complexity of theorem proving and proof presentation in axiomatized mathematical systems. This interpretation relates fundamental questions about the complexity of our intellectual tools to basic structural problems about P, NP, $CoNP$, and $PSPACE$, discussed in this paper.

This research was supported in part by National Science Foundation Grant MCS78-00418. Furthermore, the work of the second author was supported in part by a Dr. Chaim Weizmann Post-Doctoral Fellowship for Scientific Research.

Introduction

The general motivation for this work is the need and desire to understand what makes the solution of NP problems hard, provided $P \neq NP$. The fundamental question is whether the deterministic computation time required to solve NP problems could depend on the density of the set of problems under consideration. In other words, is the problem of finding satisfying assignments for Boolean formulas in conjunctive normal form, SAT, computationally hard because there are exponentially many formulas up to size n and that no one single method can solve them all easily? Or is the satisfiability problem still hard if we consider only "thinned out" sets of formulas whose density is much lower than exponential?

It has been shown recently that the structural properties of lower density sets in NP are directly determined by the relations between the corresponding higher deterministic and nondeterministic time bounded complexity classes. We cite one such result next [Ha, HIS].

A set S is said to be *sparse* if S contains only polynomially many elements up to size n, i.e. $|S \cap (\epsilon + \Sigma)^n| \leq n^k + k$. Let

$$NEXPTIME = \bigcup_{c \geq 1} NTIME[2^{cn}] \text{ and } EXPTIME = \bigcup_{c \geq 1} TIME[2^{cn}].$$

Theorem A: There exist sparse sets in $NP-P$ if and only if $NEXPTIME \neq EXPTIME$.

For related results about tally sets see [Bo].

In this paper we continue this study and show that the deterministic computation speed of sets in NP can depend on their density if and only if the corresponding higher deterministic and nondeterministic complexity classes have collapsed or partially collapsed.

We first show that there are sets of prescribed densities in NP and $PSPACE$ which are complete under polynomial time Turing reductions for all other sets of the same density in NP and $PSPACE$, respectively. We cite one such result.

Theorem B: There exists a sparse set S_0 in NP such that all other sparse sets in NP are in P^{S_0}.

This completeness result contrasts the well known results by Mahaney [Ma] and Karp-Lipton [KL]. The first result asserts that if there exists a sparse, many-one complete set for NP then $P = NP$. The Karp-Lipton result shows that if there exists a sparse set S such that $NP \subseteq P^S$, then the polynomial time hierarchy collapses to Σ_2^P. Our results show that as long as we restrict ourselves to sparse sets in NP then there exist sparse complete sets. At the same time, it is interesting to note that the same results do not seem to hold for $CoNP$, or at least they do not hold for relativized $CoNP$ computations whereas the above results hold also for relativized computations [HIS]. We also show that there are relativized computations for which there do not exist sparse sets in NP which are complete for all other sparse sets in NP under many-one polynomial time reductions.

From Theorem B we immediately obtain a proof of the previously known Theorem A as well as new results about the relation between partial collapse of higher deterministic and nondeterministic computations and the recognition speed of sparse sets.

Theorem C: All sparse sets in NP are in

$$\bigcup_{c \geq 1} TIME[n^{c \log n}]$$

if and only if

$$NEXPTIME \subseteq \bigcup_{c \geq 1} TIME[2^{cn^2}].$$

Related results are derived for sets of other densities and computation times as well as for $PSPACE$ versus NP and $PSPACE$ versus P.

From all these results we see that the deterministic time complexity of sets in NP can depend on their density if and only if the corresponding higher deterministic and nondeterministic time classes have suffered a collapse or partial collapse. Since it is our sincere conviction that the density of sets in NP and $PSPACE$ cannot affect their computation time, we are lead to the *generalized complexity hypothesis*. This conjecture asserts for NP (i.e. the *generalized NP hypothesis*) that SAT requires roughly deterministic exponential time and that the deterministic recognition time of sets in NP does not depend on their density. This clearly implies, because of our results, that the higher deterministic and nondeterministic time classes have not even partially collapsed. For example, we conjecture that there exist sets in $NEXPTIME$ which require roughly double exponential deterministic recognition time. The *generalized PSPACE hypothesis* versus P as well as NP is formulated similarly.

Intuitively, the generalized NP hypothesis asserts that the computational difficulty of finding assignments for Boolean formulas in SAT does not stem from the existence of the aggregate of such formulas, but that the difficulty is inherent even in very sparse subsets of SAT.

We give an interpretation of these results in terms of the computational complexity of doing mathematics. We assume that we are using Peano Arithmetic, F. Let

$L_1 =$ {THEOREM: "Statement of result". PROOF: b^k □ | There is a proof of

length k or less of the stated theorem in F}.

It is easily seen that L_1 is an NP complete set.

Similarly the set

$L_2 =$ {THEOREM: "Statement of result". PRESENTATION OF PROOF:

b^k □ | There is a proof of the stated theorem in F which can be

presented on tape of length k}

is $PSPACE$ complete. By presentation of proof we mean a formal writing down of the proof so that a simple proof checker can guarantee that the theorem has a proof, but we can erase any part of the proof not needed later. Thus, when the presentation is completed, the verifier knows that a proof exists, but there may not be a complete proof written down.

Clearly, $PSPACE \neq NP$ if and only if $L_2 \notin NP$ and this happens if and only if in Peano Arithmetic there are infinitely many theorems for which the difference in the length of the shortest proof and the space needed to present a proof is not polynomially bounded.

Similarly, the same relationship will hold for sparse subsets of L_2 which are in $PSPACE$, even if we are allowed to design specialized proof systems for these restricted subsets, if and only if $EXPSPACE \neq NEXPTIME$.

Corollary D: There exist sparse sets in $PSPACE-NP$ if and only if $EXPSPACE \neq NEXPTIME$ (and the quantitative difference between proof length and length of proof presentation depends on the quantitative difference between $EXPSPACE$ and $NEXPTIME$.)

Furthermore, we observe that the existence of sparse subsets of tautologies in $CoNP-NP$ implies that for these sparse subsets we cannot design special proof rules to prove in polynomial length that they are tautologies. This is so because if a sparse subset of $TAUT$ is not in NP then we know that there cannot exist a proof system which proves these formulas to be tautologies with polynomially long proofs.

We prove the following result:

Theorem E: There exists a sparse set S in P such that

$$S \cap TAUT \in CoNP-NP$$

if and only if

$$CoNEXPTIME \neq NEXPTIME.$$

Thus if and only if $CoNEXPTIME \neq NEXPTIME$ can we find a syntactically restricted sparse subset (a sparse set S in P) of Boolean formulas for which we cannot find a good proof system that would yield polynomially long proofs for formulas in $S \cap TAUT$. Furthermore, the actual length of the possible (not polynomially bounded) proofs for $S \cap TAUT$ is given by the disparity between $CoNEXPTIME$ and $NEXPTIME$.

Using a similar method we show that $EXPSPACE \neq EXPTIME$ if and only if there exists a language in $PSPACE$ which is not in P, but has polynomial size circuits (i.e., is in the class P/Poly as defined in [KL]). Formally:

Theorem F:

$$EXPSPACE \neq EXPTIME \Leftrightarrow PSPACE \cap P/Poly \neq P$$

As a corollary we get a "uniform upward separation" result for random polynomial time R and P (R is called ZPP in [G]):

Corollary G:

$$R \neq P \Rightarrow EXPSPACE \neq EXPTIME.$$

Thus we show that a separation of two low uniform complexity classes implies a separation higher up. This result is quite unique: usually separation above implies separation below, but whether in general separation below implies separation above is open. For instance:

$$NEXPTIME \neq EXPTIME \Rightarrow NP \neq P$$

But it is not known whether the assumption $NP \neq P$ can force $NEXPTIME \neq EXPTIME$ (see also [BWX], Theorem 5) or even the weaker separation $EXPSPACE \neq EXPTIME$.

From the above comments we see that the classic problems about $P=?NP=?PTAPE$, $NP=?CoNP$, etc., are really questions about the complexity of our intellectual tools, namely mathematics. Correspondingly, our work tries to address the fundamental question of what makes these problems hard and whether restricting them to subsets of lower density can make them simpler to compute. Our results show that the lower density problems can become computationally easier than the unrestricted problem if and only if there is a partial collapse of the differences between the corresponding higher complexity classes.

Sparse Complete Sets and the Structure of NP

In this section we show that the computational complexity of sets of different densities in NP and $PSPACE$ are completely determined by the relations between the corresponding higher complexity classes.

The main tool in this study will be the existence of sparse sets in NP which are complete for all other sparse sets in NP.

Theorem 1: There exists a tally set S_0 in NP, $S_0 \subseteq 1^*$, such that all sparse sets in NP are polynomial time Turing reducible to S_0, i.e.

$$\{S \mid S \text{ sparse and in } NP\} \subsetneq P^{S_0}.$$

Proof: Let A be a complete set of $NEXPTIME$ under many-one linear time reductions and let

$$S_0 = TALLY(A)=\{1^n \mid n \in 1A\}.$$

Let S be a sparse set in NP, say

$$|S \cap (\epsilon + \Sigma)^n| \leq n^{k_0} + k_0.$$

Then the set

$$B = \{(n,r) \mid |S \cap (\epsilon+\Sigma)| \geq r\} \text{ is in } NEXPTIME,$$

since for n and r represented in binary one has enough nondeterministic time to guess r strings in S, $r \leq n^{k_0} + k_0$, and verify that they are in S. Hence, B is many-one linear time reducible to A, and the corresponding set

$$B' = \{(1^n, 1^r) \mid |S \cap (\epsilon+\Sigma)^n| \geq r\}$$

is polynomial time many-one reducible to S_0. Hence, $B' \in P^{S_0}$. Since P^{S_0} is closed under complement we see that

$$B'' = \{(1^n, 1^{r_n}) \mid |S \cap (\epsilon+\Sigma)^n| = r_n\} \text{ is in } P^{S_0}.$$

Thus in P^{S_0} we can compute the exact number of elements in S up to size n, namely r_n.

Furthermore, the set

$$C = \{(n,i,j,k,d) \mid (\exists x_1 < x_2 < \ldots < x_i = x < y_1 < y_2 < \ldots < y_j)$$

$$[|y_j| \leq n \text{ and } x_r,\ y_t \in S \text{ for } 1 \leq r \leq i \text{ and } 1 \leq t \leq j$$

$$\text{and } |x| = n \text{ and the } k^{th} \text{ digit of } x \text{ is } d]\}$$

is in $NEXPTIME$ since in nondeterministic time 2^{cn} one can guess the appropriate strings, verify that they satisfy the required conditions and are in S. But then the corresponding set C' obtained by replacing (n,i,j,k,d) by $(1^n,1^i,1^j,1^k,d)$ is in P^{S_0}, by the same argument used to show that $B' \in P^{S_0}$. Since B'' is in P^{S_0}, for any x such that $|x|=n$ we can compute r_n and then, using C', check for $1 \leq i \leq r_n$, $1 \leq j \leq r_n$ such that $i+j=r_n$, whether $x=x_i$ for x_i in S. Therefore we conclude that

$$S \in P^{S_0},$$

as was to be shown. □

Later in this paper we will investigate the possibility that there exists an $S_0 \subseteq 1^*$ which is many-one complete for all sparse sets in NP, and show that there exist relativized computations for which this is not true (though Theorem 1 holds for relativized computations).

From the first theorem we immediately obtain a known result about the collapse of higher deterministic and nondeterministic time bounded complexity classes [Ha, HIS], as well as a set of new results about partial collapse of these classes.

Corollary 2: $EXPTIME = NEXPTIME$ if and only if there are no sparse sets in $NP-P$.

Proof: If $EXPTIME = NEXPTIME$ then a complete set A of $NEXPTIME$ is in $EXPTIME$ and therefore $TALLY(A) = S_0$ is in P. But then all sparse sets in NP are in P.

Conversely, if a sparse set S is in $NP-P$ then S_0 is not in P hence $A \notin EXPTIME$ and therefore $EXPTIME \neq NEXPTIME$. □

We say that a set S is P*-printable* if and only if for input 1^n in polynomial time we can print all the elements of S up to size n. Clearly, every P-printable set is sparse.

Similarly, we define a set S to be NP*-printable* if and only if there exists a nondeterministic polynomial time machine such that for input 1^n there exists a computation which prints exactly all the elements of S of length at most n, and every computation either prints exactly those elements or halts with indication of failure to print.

It is easily seen that the proofs of the previous results yield the following.

Corollary 3: $EXPTIME = NEXPTIME$ if and only if every sparse set in NP is P-printable.

Next we show that the upward separation method yields necessary and sufficient conditions also for NP-printability.

Theorem 4: $NEXPTIME = CoNEXPTIME$ if and only if every sparse set in NP is NP-printable.

Proof: Assume $NEXPTIME = CoNEXPTIME$, let S be a sparse set in NP and define

$$L = \{(n,i) \mid |S \cap (\epsilon + \Sigma)^n| \geq i\},$$

where n and i are represented in binary. Clearly, for any (n,i) in nondeterministic exponential time a machine can guess i different strings up to size n and verify that they are in S. Therefore, L is in $NEXPTIME$ and since $NEXPTIME = CoNEXPTIME$ we can use a nondeterministic exponential time machine to check if (n,i) is in $\bar{L}$. Clearly $i_n = |S \cap (\epsilon + \Sigma)^n|$ is given by $(n,i_n) \in L$ and $(n,i_n+1) \in \bar{L}$. Thus we see that

$$L' = \{(n,i_n) \mid |S \cap (\epsilon + \Sigma)^n| = i_n\} \in NEXPTIME$$

and therefore

$$L'' = \{(1^n, i_n) \mid |S \cap (\epsilon+\Sigma)^n| = i_n\} \in NP.$$

But then a nondeterministic polynomial time machine for input 1^n can print

$$x_1 < x_2 < \ldots < x_j < \ldots < x_{i_n}, \text{ for } 1 \leq j \leq i_n,\ |x_j| \leq n,\ x_j \in S,$$

by first guessing i_n and verifying that it is a correct guess and then guessing i_n distinct strings of S of length at most n and printing them if the guess is verified (if not the machine fails to print). Thus S is NP-printable.

Assume that every sparse set in NP is NP-printable and let A be a set in $NEXPTIME$. Then $TALLY(A) = \{1^n \mid n \in 1A\}$ is a sparse set in NP and therefore NP-printable, but then $\overline{TALLY(A)}$ is also in NP and we see that $\bar{A}$ is in $NEXPTIME$. But then $CoNEXPTIME \subseteq NEXPTIME$ and therefore

$$CoNEXPTIME = NEXPTIME. \ \square$$

From Theorem 4 we can obtain a further characterization of the $NEXPTIME = CoNEXPTIME$ collapse.

Corollary 5: $NEXPTIME = CoNEXPTIME$ if and only if for all sparse sets S in NP $\bar{S}$ is in NP.

Proof: Since $NEXPTIME = CoNEXPTIME$ implies that S in NP is NP-printable by Theorem 4 we immediately see that $\bar{S}$ is in NP.

Conversely, if for every sparse set T in NP $\bar{T}$ is also in NP then we see that for any sparse set S in NP the set

$$L'' = \{(1^n, i_n) \mid |S \cap (\epsilon+\Sigma)^n| = i_n\} \text{ is in } NP$$

and therefore S is NP-printable. Therefore

$$NEXPTIME = CoNEXPTIME$$

by Theorem 4. $\square$

The above completeness results for sparse sets in NP can be easily extended to $PSPACE$ versus NP and $PSPACE$ versus P. Furthermore, with an additional uniformity assumption these results generalize to denser sets in NP and without the uniformity assumption to denser sets in $PSPACE$. For a discussion of uniformity conditions on NP sets see [HIS].

Next we show that a partial collapse of the higher deterministic and nondeterministic complexity classes directly determines the computation time of the lower density sets in NP and $PSPACE$. We first prove, as an example, a special case of our general result.

Theorem 6: $NEXPTIME \subseteq \bigcup_{c\geq 1} TIME[2^{cn^k}]$ if and only if all sparse sets in NP are in

$$\bigcup_{c\geq 1} TIME[n^{c(\log n)^{k-1}}].$$

Proof: If $NEXPTIME \subseteq \bigcup_{c\geq 1} TIME[2^{cn^k}]$ then for a complete set A of $NEXPTIME$

$$TALLY(A) = S_0 \text{ is in } TIME[2^{d(\log n)^k}] = TIME[n^{d(\log n)^{k-1}}].$$

But then by Theorem 1 every sparse set S in NP is in P^{S_0} and

$$S \in P^{S_0} \subseteq \bigcup_{c\geq 1} TIME[n^{c(\log n)^{k-1}}].$$

Conversely, if every sparse set of NP is in

$$\bigcup_{c\geq 1} TIME[n^{c(\log n)^{k-1}}]$$

then so is S_0 and we see that

$$A \in TIME[2^{rn^k}],$$

for some r. But then

$$NEXPTIME \subseteq \bigcup_{c\geq 1} TIME[2^{cn^k}]. \quad \square$$

Related results can easily be derived for $PSPACE$ versus NP and $PSPACE$ versus P.

We now state without proof the generalization to any well behaved computation times.

Theorem 7: Let $f(n)\geq n$ be nondecreasing and fully-time-constructible. Then:

(1) $NEXPTIME \subseteq \bigcup_{d\geq 1} TIME[2^{d(f(dn+d))}]$ if and only if every sparse set in NP is in $\bigcup_{d\geq 1} TIME[2^{d(f(d\log n+d))}]$.

(2) $CoNEXPTIME \subseteq \bigcup_{d\geq 1} NTIME[2^{d(f(dn+d))}]$ if and only if the complement of every sparse set in NP is in $\bigcup_{d\geq 1} NTIME[2^{d(f(d\log n+d))}]$. $\square$

Results about sets of higher than polynomial density are correspondingly related to higher complexity classes below exponential time.

We say that a set S has density $\sigma(n)$ if

$$|S \cap (\epsilon+\Sigma)^n| \leq \sigma(n).$$

Theorem 8: There are no $\sigma(n)=n^{\log n}$ dense sets in

$$PSPACE-NP$$

if and only if

$$\bigcup_{c\geq 1} SPACE[2^{c\sqrt{n}}] = \bigcup_{c\geq 1} NTIME[2^{c\sqrt{n}}].$$

We can derive similar results for NP if we assume that our lower density sets are uniformly distributed. (For a more detailed discussion of uniform distributions see [HIS]).

Theorem 9: There are no $\sigma(n)=n^{\log n}$ uniformly dense sets in $NP-P$ if and only if

$$\bigcup_{c\geq 1} TIME[2^{c\sqrt{n}}] = \bigcup_{c\geq 1} NTIME[2^{c\sqrt{n}}],$$

and in this case

$$SAT \in \bigcup_{c\geq 1} TIME[2^{c\sqrt{n}}].$$

Finally, we list an illustrative result about partial collapse of subexponential complexity classes.

Theorem 10: $\bigcup_{c\geq 1} SPACE[2^{c\sqrt{n}}] \subseteq \bigcup_{c\geq 1} TIME[2^{cn^k}]$ if and only if all $\sigma(n)=n^{\log n}$ dense sets of $PSPACE$ are in

$$\bigcup_{c\geq 1} TIME[n^{c(\log n)^{k-1}}].$$

On Many-One Complete Sparse Sets

The existence of a tally set S_0 in NP such that all other sparse sets in NP are in P^{S_0} raises the question whether there exists a tally set which is many-one polynomial time complete for all sparse sets in NP.

Our results show that there exist relativized computations for which no tally set S_0 can be complete for all sparse sets in NP under many-one reductions. At the same time, it is easily seen that Theorem 1 holds for relativized computations and therefore for any oracle A there exists a sparse set complete for all other sparse sets in NP^A under Turing reducibility.

Let $\leq_T^P$ and $\leq_M^P$ denote, respectively, polynomial time Turing and many-one reductions. Let Σ_i^E denote the Σ-levels of the exponential hierarchy, i.e.

$$\Sigma_0^E = EXPTIME,\ \Sigma_1^E = NEXPTIME,$$
$$\Sigma_2^E = NEXPTIME^{\Sigma_1^P} = NEXPTIME^{SAT}, \text{ etc.}$$

We first prove a technical result which shows that for some oracle A there do not exist tally sets which are $\leq_M^P$-complete for all sparse sets in NP^A.

Lemma 11: Let $S_0 \subseteq 1^*$ and assume that for all sparse S in NP we have $S \leq_M^P S_0$. Then $NEXPTIME = \Sigma_2^E$ implies that

$$NEXPTIME = EXPTIME.$$

Proof (outline): We first observe that it is sufficient to show that every set S of the form

$$S = T \cap SAT,$$

where T is a sparse set in P, must be in P, since then by [HIS] $NEXPTIME = EXPTIME$. The assumption

$$\Sigma_2^E = \Sigma_1^E = NEXPTIME$$

implies that for any sparse set $S = T \cap SAT$, T sparse and in P, the set

$$\{(F_i,\ x_i) \mid F_i \in S \text{ and } x_i \text{ is the minimal solution of } F_i\}$$

is also in NP. For F_i in S let F_i^k denote F_i with its first k variables, $0 \leq k \leq |x_i|$, filled in with the values of its minimal solution x_i (we choose our syntax so that $|F_i^k| = |F_i|$). Then

$$S' = \{F_i^k \mid F_i \in S \text{ and } 0 \leq k \leq |x_i|\}$$

is seen to be a sparse set in NP.

Note that S does not necessarily have the self reducibility property but that S' has a weak form of this property, sufficient for the following proof.

If $S' \leq_M^P S_0 \subseteq 1^*$ then for any F in T in polynomial time we can find the minimal satisfying assignment of F if $F_i = F$ or determine that F is not in S. This is done by searching the tree of functions generated by F by partial assignments of variables (as in P. Berman's proof [Be, Ma]). We can discard any subtree which is assigned a string not in 1^* by the reduction. For subtrees with the same labels in 1^* we always pick the leftmost subtree to find x_i if $F_i = F$. Since there are only polynomially many labels in 1^* the reduction can assign, we see that the search is completed in polynomial time, either yielding the minimal solution or showing that F is not in S. □

Corollary 12: There exists an oracle A such that no tally set can be $\leq_M^P$-complete for all sparse sets in NP^A.

Proof: Since there exists an oracle A [S] such that

$$EXPSPACE^A = ... = \Sigma_2^{E(A)} = \Sigma_1^{E(A)} \neq \Sigma_0^{E(A)},$$

a relativized version of the previous lemma implies that there cannot exist a tally set $\leq_M^P$-complete for all sparse sets of NP^A. □

Furthermore, from Theorem 12 in [HIS] it follows that there exists an oracle A such that no tally set can be even $\leq_T^P$-complete for all sparse sets in $CoNP^A$.

The Computational Complexity of Mathematics

It is well known that the sets of provable theorems of sufficiently rich, axiomatized mathematical systems form complete sets for the recursively enumerable sets under recursive reductions. Thus, intuitively, we can say that the provable theorems in Peano Arithmetic form a set which is computationally as hard as any recursively enumerable set. Unfortunately, this interpretation dos not yield any real insight about the computational complexity of doing mathematics.

We believe that the proper formulation for the study of the computational complexity of mathematics and therefore the study of the computational complexity of our intellectual tools in general, is by investigating the difficulty of proving theorems by bounding the length of the desired proof. If we do this then, as will be shown below, the questions about the computational complexity of the process of doing mathematics -- finding proofs and presenting proofs -- become questions about P, NP, and $PSPACE$.

Assume that we have an axiomatized formal system F, which could be Peano Arithmetic, and that we have given a "natural" definition for the length of proofs and related concepts.

Then it is easily seen that the set

L_1 = {THEOREM: "Statement of result". PROOF: b^k □ | There is a proof of length k or less of the stated theorem in F}

is NP complete.

Similarly the set

L_2 = {THEOREM: "Statement of result". PRESENTATION OF PROOF: b^k □ | There is a proof of the stated theorem in F which can be presented on tape of length k}

is $PSPACE$ complete. By *presentation of proof* we mean a formal writing down of the proof so that a simple (polynomial time) proof checker can guarantee that the theorem has a proof, but we can erase any part of the proof not needed later. Thus, when the presentation is completed, the verifier knows that a proof exists, but there may not be a complete proof written down.

Clearly, $PSPACE \neq NP$ if and only if L_2 is not in NP and this happens if and only if in Peano Arithmetic there are infinitely many theorems for which the difference in the length of the shortest proof and the space needed to present a proof is not polynomially bounded.

The fundamental question is whether finding proofs of theorems in mathematics is hard because of the existence of the aggregate of all provable theorems so that no one method can prove them all easily or is it because "individual" theorems are hard to prove. Since we cannot give precise mathematical meaning to "computational complexity" of finding proofs for individual theorems, we replace this question by questions about sparse or supersparse subsets of the sets L_1 and L_2. Clearly this brings us right back to the main topic of this paper and shows that questions about sparse subsets of $NP-P$, $PSPACE-NP$ and $PSPACE-P$ are actually fundamental questions about the nature of mathematics. For example, we easily obtain the following result.

Corollary 13: There exists a sparse set S in P such that $L_2 \cap S \notin NP$ if and only if

$$EXPSPACE \neq NEXPTIME.$$

In the study of proof techniques special attention has been given to proving a Boolean formula a tautology. Let

$$TAUT = \{F \mid F \text{ Boolean formula in } DNF \text{ such that } (\forall x)[F(x)=1]\}.$$

Clearly, $TAUT$ is a complete set for $CoNP$ and the question whether some decision problem in $CoNP$ is not in NP when restricted to some sparse domain S in P can be shown to be equivalent to the question whether for some sparse set S in P we cannot design special proof rules with which in polynomial length will prove for any tautology in S that it is indeed a tautology. The following result gives necessary and sufficient conditions for this to be impossible.

Theorem 14: The following conditions are equivalent.

(1) $CoNEXPTIME \neq NEXPTIME$

(2) For some set L in $CoNP$ and some sparse set S in P, $L \cap S \in CoNP-NP$.

(3) For some P-printable set S, $TAUT \cap S \in CoNP-NP$.

Proof: It can easily be seen that if $S \in P$ and $L \cap S \in CoNP-NP$ then $\bar{L} \cap S$ is a sparse set in $NP-CoNP$. Thus, by Corollary 5, (2) implies (1). Clearly (3) implies (2). We now outline a proof that (1) implies (3): If (1) holds then there exists a tally set T in $CoNP-NP$. There exists a 1-1 length increasing polynomial time reduction g from T to $TAUT$ [BH]. Then $g(1^*)$ is the desired S in (3). □

The above theorem can be generalized to any well behaved computation times (rather than $NEXPTIME$ and NP) in a fashion similar to Theorem 7. We omit the details.

A Uniform Upward Separation

There are known separation results about uniform complexity classes of the form: "If two high uniform complexity classes are unequal then two corresponding lower uniform complexity classes are unequal", (a *downward* separation). For instance, if $NEXPTIME \neq EXPTIME$ then $NP \neq P$. We use our techniques to prove a quite unique *upward* separation of uniform complexity classes. Let R be random polynomial time. We prove

Theorem 15: $R \neq P$ implies $EXPSPACE \neq EXPTIME$.

Before we prove this theorem, we need a lemma.

Let $P/Poly$ be the class $\bigcup_{S \text{ sparse}} P^S$ [KL]. By a result due to Meyer (see [KL]), $P/Poly$ is equal to the class of languages having polynomial size circuits.

Lemma 16: $EXPSPACE \neq EXPTIME$ if and only if $PSPACE \cap P/Poly \neq P$.

Proof: If $EXPSPACE \neq EXPTIME$ then there exists a tally set $T \in PSPACE-P$. Clearly $T \in P/Poly$. Conversely, let $L \in PSPACE-P$ be in $P/Poly$. Then it can be shown that the family of minimal circuits $\{C_n\}$ for L can be computed in space polynomial in n. $\{C_n\}$ can be encoded by a sparse set S which is in $PSPACE$ such that $L \in P^S$, hence $S \notin P$. By [HIS], $EXPSPACE \neq EXPTIME$ follows. □

Theorem 15 now follows since $R \subset PSPACE$, and also by [A] $R \subset P/Poly$.

References

[A] L. Adleman, "Two Theorems on Random Polynomial Time", IEEE-FOCS Symp. (1978), 75-83.

[Be] P. Berman, "Relationship Between Density and Deterministic Complexity of NP-Complete Languages", 5th ICALP, Lecture Notes in Computer Science 62, Springer-Verlag, Berlin (1978), 63-71.

[BH] L. Berman and J. Hartmanis, "On Isomorphism and Density of NP and Other Complete Sets", SIAM J. on Computing (1977), 305-322.

[Bo] R.V. Book, "Tally Languages and Complexity Classes", Information and Control 26 (1974), 186-193.

[BWX] R. Book, C. Wilson, and M. Xu, "Relativizing Time and Space", IEEE-FOCS Symp. (1981), 254-259.

[G] J. Gill, "Computational Complexity of Probabilistic Turing Machines", SIAM J. on Computing 6 (1977), 675-695.

[Ha] J. Hartmanis, "On Sparse Sets in NP-P", Department of Computer Science, Cornell University, TR82-508, August 1982.

[HIS] J. Hartmanis, N. Immerman, and V. Sewelson, "Sparse Sets in NP-P: EXPTIME vs NEXP-TIME", ACM Symposium on Theory of Computing, 1983.

[KL] R.M. Karp and R.J. Lipton, "Some Connections Between Nonuniform and Uniform Complexity Classes", Proceedings 12th Annual ACM Symposium on Theory of Computing (April 1980), 302-309.

[Ma] S. Mahaney, "Sparse Complete Sets for NP: Solution of a Conjecture of Berman and Hartmanis", Proceedings 21st IEEE Foundations of Computer Science Symposium (1980), 42-49.

[S] V. Sewelson, private communication.