Some Observations About *NP* Complete Sets

Juris Hartmanis[*]

87-817

March 1987

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

[*] This research has been supported in part by National Science Foundation Grant CCR 8520597.

Some Observations About *NP* Complete Sets

Juris Hartmanis[†]

Department of Computer Science

Cornell University

Ithaca, New York

Abstract

In this paper, we summarize and extend some recent results about the properties of *NP* complete sets and related results about the structure of feasible computations.

Introduction

In the early seventies, the work of Cook, Karp and Levin established the importance of the complexity classes P, NP and $PSPACE$ by showing that a variety of important natural computational problems were complete for these classes. This initiated an intensive study of these complexity classes and created a veritable gold-rush fever in the search for complete problems. In particular, NP complete problems were found in many different areas of computer science, operations research, pure mathematics and other sciences. Today, many hundreds of natural NP complete problems are known as well as a wide collection of P and $PSPACE$ complete problems, firmly establishing the dominant importance of these classes to computer science. At the same time, many questions about these classes have not been answered, effort not withstanding. It is quite surprising that the whole awesome, intellectual arsenal of mathematics seems to have no results or tools with which to attack the separation problem:

$$P = ?\ NP = ?\ PSPACE.$$

In particular, the P and NP separation problem has to be viewed today as one of the most important open problems in theoretical computer science and possibly in all of mathematics. This problem is really a question about the quantitative computational difference between the difficulty of

[†] This research has been supported in part by National Science Foundation Grant CCR 8520597.

finding a proof for a theorem and checking the correctness of a given proof. Therefore, it is indeed a fundamental question about the quantitative nature of mathematics.

During the last decade, a variety of new, more or less natural, complexity classes below *EXPSPACE* have been defined and investigated. Some of these classes are defined to model the complexity of probabilistic computations and others to investigate the complexity of parallel computations. The composite computational complexity world described by the many new complexity classes reveals an intricate, interlocking world of unexpectedly rich fine structure below *EXPSPACE* and, particularly below *PSPACE*. At the same time, these new complexity classes have added a bewildering set of questions about the properties of and relations between these classes for which so far few solutions have been found.

The purpose of this paper is to summarize and extend some of the recent results about the properties of *NP* complete sets and related results about the structure of the feasible computations mentioned above.

Many-One Complete Sets

We assume that the reader is familiar with the basic ideas and results of computational complexity theory as, for example, presented in [GJ 79, HU 79]. For the sake of completeness and to establish notation, we summarize the basic concepts of our discours.

Let P and NP denote, respectively, the families of languages accepted by deterministic and nondeterministic polynomial time bounded Turing machines. *PSPACE* is the family of languages accepted by polynomial space bounded Turing machines. *EXPTIME* and *NEXPTIME* are, similarly, the families of languages accepted by deterministic and nondeterministic, $T(n) = 2^{cn}$, $c \geq 1$, bounded Turing machines.

The ***polynomial time hierarchy***, PH, is defined inductively:

$$\Sigma_1^P = NP, \ \Pi_1^P = CoNP$$

$$\Sigma_k^P = NP^{\Sigma_{k-1}^P}, \ \Pi_k^P = co \ \Sigma_k^P \text{ for } k > 1.$$

A language A in NP is *many-one complete* for NP iff for any B in NP there exists a polynomial time computable function f (i.e. f in PF) such that:

$$x \in B \ \ iff \ \ f(x) \in A,$$

we write $A \leq_m^P B$.

A language A is *Turing complete* for NP iff $NP \subseteq P^A$, we write $A \leq_T^P B$.

The completeness for other families of languages is defined similarly.

Two languages A and B are *polynomial time isomorphic (p-isomorphic)* iff there exists a bijection, f, such that f and f^{-1} are in PF and

$$x \in A \ \ iff \ \ f(x) \in B,$$

we write $A \cong_p B$.

Clearly, these concepts are modeled on the corresponding, earlier developed, ideas from recursive function theory, where recursiveness corresponds to polynomial time computability and recursive enumerability corresponds to NP-ness. Quite surprisingly, the research on feasible computations seems to show that the awesomely powerful computations used in recursive function theory (i.e., recursive reductions and recursive isomorphisms) have created a far less complex world by suppressing the fine structure which we believe exists when we limit the computational power to define complexity classes and the corresponding reductions.

It is well known from recursive function theory [Ro 67] that all recursively enumerable complete sets as well as all the complete sets in each level of the Keene Hierarchy are, respectively, recursively isomorphic. Analogously, on closer inspection, all the natural NP complete sets show great structural similarities. The comparison and study of the known NP complete sets led to the following formalization which captures the above observed similarity between NP complete sets.

A set A, $A \subseteq \Sigma^*$ has (polynomial time) *padding functions* iff there exist $D(\ , \ )$ and $S(\ )$ in PF such that:

$$\forall \ (x, y \in \Sigma^*) \ [D(x,y) \in A \ \ iff \ \ x \in A]$$

$$\forall\ (x, y \in \Sigma^*)\ [S \bullet D(x,y) = y].$$

It can easily be shown that the well known NP complete problems all have padding functions. For example, this is quickly verified for SAT, the set of satisfiable Boolean formulas in conjunctive normal form. Using padding functions, we are led to the following characterization of NP complete sets p-isomorphic to SAT [BH 77].

Theorem: An NP complete set A is p-isomorphic to SAT iff A has padding functions.

Since all the known NP complete sets had padding functions and since all r.e. complete sets are isomorphic, there were strong indications that all NP complete sets may be p-isomorphic and thus the belief that there is basically only one isomorphism type of NP complete sets. These considerations led in 1976 to the following conjecture [BH 77]:

Isomorphism Conjecture: All NP complete sets are p-isomorphic.

This conjecture has also been referred to in the literature as the *Berman-Hartmanis Conjecture* [Mah 80, Mah 82, Bet 78, Kur 83]. Note that, $P = NP$ implies that there exist NP complete sets that are not p-isomorphic and therefore the above conjecture implies that $P \neq NP$, indicating that the conjecture may be very hard to resolve.

The isomorphism conjecture implies several subconjectures. We know that sets of strongly different densities cannot be p-isomorphic. To make this precise, we say that A, $A \subseteq \Sigma^*$, is *sparse* iff there exists a k such that for all n:

$$|A \cap \Sigma^n| \leq n^k + k.$$

The isomorphism conjecture implies that no sparse set can be NP complete and, in particular, that no set A over a unary alphabet, $A \subseteq a^*$, can be NP complete.

The last conjecture was proven by P. Berman [BE 78]:

Theorem: If SAT is $\leq_m^P$-reducible to A, $A \subseteq a^*$, then $P = NP$.

Berman's proof elegantly exploits the self-reducibility of SAT to prune the tree of all possible functions obtained from F by partially assigned variables, to a polynomial size subtree by means of the reductions of formulas in the tree to $A \subseteq a^*$, thus, placing SAT in P.

By a considerably subtler argument, partially inspired by the techniques in [KL 80], Mahaney in his Ph.D. dissertation showed that no sparse set could be NP complete [Mah 80].

Theorem: If A is sparse and $SAT \leq_m^P A$ then $P = NP$. Therefore, if $P \neq NP$ there are no sparse $\leq_m^P$-complete NP languages.

This result is invitively very satisfying, in that it asserts that if $P \neq NP$, then NP problems cannot be solved with "small amounts of information", i.e., there is no polynomial in n size look up table to solve satisfiability for formulas up to size n.

This also showed that if $P \neq NP$ then the isomorphism conjecture will not be disproved by a simple density argument.

If $P \neq NP$, then we know that any sparse set in NP is an incomplete set. The existence of incomplete sets in $NP - P$, if $P \neq NP$, was shown by Ladner [La 75] by using the earlier developed *delayed diagonalization* technique. At the same time, the Ladner incomplete sets have not been shown to be sparse and it is known that in some relativized worlds, they cannot be sparse [Kur 85, HIS 85].

The existence of sparse and therefore incomplete sets in $NP - P$ was resolved in [HIS 85, Har 83a] by linking their existence to higher computational complexity classes.

Theorem: Sparse sets exist in $NP - P$ iff $EXPTIME \neq NEXPTIME$.

For related results about tally sets, see [Bo 74]. A nice summary of the study of sparse sets in structural complexity theory can be found in [Mah 86].

The above result has been extended [Li 85] to show that if computably arbitrary sparse sets exist in $NP - P$ then the corresponding deterministic and nondeterministic complexity classes are different, i.e.:

$$\bigcup_{k \geq 1} TIME\ [T(n)^k] \neq \bigcup_{k \geq 1} NTIME\ [T(n)^k],$$

for all real-time computable $T(n) \geq n$.

Clearly, from the Tracktenbrot-Borodin Gap theorem, we know that there exist computable (but not real-time computable) monotonically increasing functions, $F(n) \geq n$, such that:

$$\bigcup_{k \geq 1} TIME\ [F(n)^k] = \bigcup_{k \geq 1} NTIME[F(n)^k].$$

This extension of the above theorem has interesting proof theoretic implications about separating P from NP. Any proof separating P from NP by a "generalized diagonalization" that can be slowed down to separate P from NP by arbitrarily sparse sets, yields a proof that all higher deterministic and nondeterministic classes are separated (provided the above result can be formulized in the system). For further discussion of related proof techniques see [Koz 80, Re 86].

The consequences of the existence of two non-isomorphic NP complete sets have been investigated by Mahaney in [Mah 83].

Theorem: If A and B are NP complete and $A \not\equiv_P B$, then there exist infinitely many pair wise non-isomorphic NP complete sets.

The proof constructs from A and B and a new complete set C for NP such that C is not p-isomorphic to A and B, by an ingenious scheme of composing C from alternating segments of A and B.

The structure of the degrees of NP complete sets, assuming that there exist non-isomorphic complete sets, is studied in [MY 85].

If there exists an NP complete set A which is not p-isomorphic to SAT, then we know that A does not have polynomial time padding functions. Several possible such NP complete sets have been suggested in the literature; see for example [JY 85, Har 83b]. A natural, possibly NP complete set, not isomorphic to SAT, is suggested by generalized Kolmogorov complexity [Har 83b].

Let M_u be a standard universal Turing machine. Define:

$$K[n^{\frac{1}{2}},\ n^3] = \{x \mid (y)\ [\ |y| \leq |x|^{\frac{1}{2}} \text{ and } M_u\ (y) = x \text{ in } |x|^3 \text{ steps}]\}.$$

Clearly, $K[n^{\frac{1}{2}}, n^3]$ is in NP, it does not seem to be in P and it could be NP complete. Quite surprisingly, relativization suggests that this may not be easy to prove (as shown recently by the author).

Theorem: There exist oracles A and B such that $K^A[n^{\frac{1}{2}}, n^3]$ is NP complete and K^B $[n^{\frac{1}{2}}, n^3]$ is not NP complete.

Doubts About the Isomorphism Conjecture

Though the proof that sparse sets can not be $\leq_m^P$ - complete for NP, provided $P \neq NP$, proved the weaker versions of the isomorphism conjecture, some serious doubts have been expressed about this conjecture.

Stuart Kurtz constructed an oracle A such that $P^A \neq NP^A$ and for which there exist non isomorphic $\leq_m^P$ - complete sets in NP^A [Kur 83]. A related construction of non-isomorphic sets will be discussed later in the connection with the Joseph and Young One-Way conjecture. At the same time, it has been observed that the construction of an oracle which supports the isomorphism conjecture seems much harder (and has not yet been fully achieved). For very recent work on this topic see [GJ 86].

Joseph and Young have raised very serious objections to the isomorphism conjecture [JY 85]. Their objections to the conjecture seem to stem from a careful analysis why the proof that all r.e. complete sets are recursively isomorphic can not be extended to the NP complete sets [Ro 67]. They conclude that the existence of non-isomorphic NP complete sets is linked to the existence of one-way functions.

One-Way Conjecture: There exist non isomorphic NP complete sets *iff* there exist one-way functions.

A *one-way-function* is a one-to-one, polynomially honest function f in PF whose inverse is not in PF.

A function f is *polynomially honest* iff there exists a k such that

$$\forall\ (x)[|\ x\ |^{\frac{1}{k}} \leq |f(x)| \leq |\ x\ |^k + k].$$

The existence of one-way functions is of great interest to cryptography and it has been linked to the properties of the complexity class UP defined by Valiant.

An NP-machine N is *categorical* iff for all x there exists at most one accepting path for $N(x)$.

$$UP = \{L(N_i) \mid N_i \text{ is categorical}\}.$$

The link between UP and one-way-functions, as observed by several authors, including [SG 84], is given by the following.

Theorem: There exist one-way-functions iff $P \neq UP$.

The class UP is little understood; for example, it is not known whether there exists a UP complete language. In [HH 86a] it was shown that there exist oracles A and B such that $P^A \neq UP^A \neq NP^A$, $P^B \neq UP^B \neq NP^B$ and that there exist complete languages in UP^A and there are no complete languages in UP^B. Furthermore, it was shown that if UP has compete languages then there exists a set U in P such that U contains only Boolean formulas which have at most one satisfying assignment and

$$U \cap SAT \textit{ is } UP \textit{ complete.}$$

For related results, see [Sip 82, HI 85].

We now show that there is a reasonable oracle A (i.e., $P^A \neq NP^A$) for which the One-way Conjecture is contradicted. As corollaries, we show that there are relativized worlds, A, in which there are NP^A-complete sets that are not p^A-isomorphic and, indeed, are not isomorphic even under tremendously powerful isomorphisms.

To achieve our goal we first construct a special oracle. For detailed proofs see [HH 87].

Theorem: There is a set $A = PSPACE \oplus B$ such that

1. $P^A = UP^A \neq NP^A$, and

2. B has only strings of lengths from the widely spaced set E.

$$E = \cup_{i \geq 0} e_i, \; e_0 = 10^{10}, e_i = 2^{2^{2^{2^{2^{2^{2^{2^{e_{i-1}}}}}}}}} \quad \text{for } i > 0.$$

Proof outline: The proof exploits the techniques of [BGS 75], of [Rac 82], of Cai and Hemachandra's relativizations of the counting hierarchy [CH 86], and of Hartmanis and Hemachandra's proof that there is an oracle for which $P \neq UP \neq NP$ yet UP has complete languages [HH 86a].

The construction of the desired, very sparse B which separates P^A from NP^A will not be given here in detail since it is a reasonably straight forward construction.

The proof that $P^A = UP^A$ is more interesting and we outline a method following [Rac 82]. We will show that for any categorical N_i^A, $L(N_i^A)$ is in P^A. Use $PSPACE$ to find if for some valid value of B' there is an accepting computation of $N_i^{PSPACE \oplus B'}(x)$; if not, reject x. Use PSPACE to get the path, say $\boldsymbol{path}_0$. Query all elements in the path, let S_0 be all elements queried on the path, and let W_0 be the elements on which the path was wrong (disagreed with B). If the path was never wrong, we have a true accepting path, so accept x.

Similarly, use PSPACE to find if, for some valid value of B' consistent with our knowledge about the elements of S_o, there is an accepting computation of $N_i^{PSPACE \oplus B'}(x)$; if not, reject x. Use PSPACE to get the path, say $\boldsymbol{path}_1$. Query all elements in the path, let S_1 be all elements queried on the path, and let W_1 be the elements on which the path was wrong (disagreed with B). If the path was never wrong, we have a true accepting path, so accept x.

Keep repeating this. The process finishes quickly. Why? Each $\boldsymbol{path}_k$ must *conflict* with each of the paths $\boldsymbol{path}_0$, $\boldsymbol{path}_1$, ..., $\boldsymbol{path}_{k-1}$, since we were robustly categorical over all valid extensions. Note that $(j, l \textit{ with } j \neq l)$ $[W_j \cap W_l = \emptyset]$, since no mistake is made twice. Thus, $\boldsymbol{path}_k$ must conflict with $\boldsymbol{path}_0$ on some element that is both in W_0 of $\boldsymbol{path}_0$ and $S_k - W_k$ of $\boldsymbol{path}_k$. Similarly, it conflicts with $\boldsymbol{path}_1$ on some element that is both in W_1 of $\boldsymbol{path}_1$ and in $S_k - W_k$ of

$path_k$, and so on. But since the W_j's are disjoint, we take up $k - 1$ spaces of $S_k - W_k$ just to disagree with the previous paths. Thus, the process goes on at most until we examine $| x |^i + i$ paths. At that point, we either have eliminated all paths (so $N_i^A(x)$ rejects) or we have found a path consistent with our oracle (so $N_i^A(x)$ accepts). Thus, we have accepted an arbitrary UP^A language in P^A, so $P^A = UP^A$.

We have shown $P^A = UP^A \neq NP^A$.

We now use this result to show the following:

Theorem: There is an oracle A that is:

1. reasonable, i.e., $P^A \neq NP^A$,
2. allows no one-way functions, i.e., $P^A = UP^A$, and
3. contradicts the Isomorphism Conjecture, i.e., there are $\leq_m^{p,A}$-complete sets for NP^A that are not p^A-isomorphic.

Proof Outline: The canonical complete set for A, $Univ_A$, described below, is quite dense. We construct a second complete set, U_A, that has huge stretches over which it contains no strings. From this, it follows that the two sets can not be p^A-isomorphic.

Let A be the set constructed in the previous *Theorem*. Thus $P^A = UP^A \neq NP^A$, and A has the special form described in the *Theorem*. Let:

$Univ_A = \{1\#x\#N_i\#\ padding \mid N_i^A(x) \text{ accepts in at most } |padding| \text{ steps}\} \cup \{0\#y | y \in \Sigma^*\}$.

Universal set $Univ_A$ is canonical complete set for A.

Our second complete set U_A will, at lengths l "near" an element of E, code all strings of $Univ_A$ of length less than l. At lengths l "far" from an element of E, U_A will contain *no* elements. We say a length l is "near" an element of E if for some $z \in E$ we have:

$$l \in [\log z, 2^{2^z}].$$

Claim 1: It can now be shown without too much difficulty that U_A and $Univ_A$ are not p^A-isomorphic.

Claim 2: U_A is $\leq_m^{p,A}$ -complete for NP^A. Clearly, $U_A \in NP^A$, since $Univ_A \in NP^A$ and U_A just codes in $Univ_A$ strings at certain lengths. Let $L \in NP^A$. We show $L \leq_m^{p,A} U_A$. Given x we must reduce "$x \in L$?" to a question of membership in U_A. Simply put, if x is close to an e_i we can reduce to a nice universal string coded into U_A, and if x is not close to any e_i, we can discover by brute force all relevant strings of B and then use $P^{PSPACE} = NP^{PSPACE}$ to determine if $x \in L$. We leave the details given in [HH 87] to the reader.

From the above results, we can easily derive the following:

Corollary: There is an oracle A for which there are $\leq_m^{p,A}$-complete sets for NP^A that are not p^A-isomorphic. As a matter of fact, these sets are not isomorphic by any primitive recursive isomorphism.

Very recently, Kurtz, Mahaney and Royer [KMR 86] have obtained some exciting results about the structure of complete sets in EXPTIME. Clearly, in EXPTIME one has enough computational power to diagonalize over all p-time reductions and p-isomorphisms and so one can expect that one can construct special complete languages. Quite surprisingly, this is not easy to show and was finally achieved in [KMR 86].

An *m-degree* is a collection of sets equivalent under $\leq_m^p$-reductions. An m-degree is *collapsing* iff its members are p-isomorphic.

Thus, the Isomorphism Conjecture can be restated that the NP complete sets form a coilapsing m-degree.

By beautifully subtle diagonalization arguments, the following is shown:

Theorem: EXPTIME has complete sets whose:

1. m-degree contains infinitely many 1-degrees
2. m-degree collapses.

For proofs and a complete summary of related work, see [KMR 86]. These results show that the structure of EXPTIME is quite complex. Similarly, though not yet proven, we now have to expect that the structure of NP and in general, the structure of the feasible computations (say,

below EXPSPACE) is very complex and little understood at this time. It is the realization which motivates and inspires the work in structural complexity.

It is ironic to reflect that recursive function theory, which has inspired and guided much of the work in computational complexity, had revealed a far simpler structure about r.e. complete sets and related concepts. It seems as if by using the most powerful reductions and isomorphisms recursive function theory has eliminated the fine structure which we are now finding in the world of the feasible computations. It is still only a partially understood field of research, but one that promises new challenges, many surprises and which we must understand.

On Turing Complete Sets

So far, we have seen that the existence of sparse $\leq_m^P$**-complete sets for** NP **implies that** $P = NP$ **and these results can be shown to hold for all relativizations of** NP. **The situation is quite different when we consider** $\leq_T^P$**-complete or hard sets for** NP.

First of all, there exist an oracle A **such that** $P^A \neq NP^A$ **and** NP^A **has sparse** $\leq_T^P$**-complete sets, as shown by Mahaney.**

Theorem: There exists a sparse set $B, B \subseteq (0+1)^*$, such that for $A = PSPACE \oplus B$,

$$P^A \neq NP^A$$

and the sparse set:

$$Pr(B) = \{x \mid x = x'\#^k \ ,\ \ x' \text{ is a prefix of } x \text{ in } B \text{ and} \mid x \mid = \mid x'\ \#^k \mid \}$$

is Turing complete for NP^A.

Proof: By a straight forward [BGS 75] diagonalization argument, we can construct a sparse set B such that for $A = PSPACE \oplus B$

$$P^A \neq NP^A .$$

It is easily seen that for a sparse set B, $Pr(B)$ is a sparse set. Furthermore,

$$NP^A = P^{A \oplus Pr(B)}.$$

To see this, we just have to observe that fore any NP machine, N^i running in time $n^k + k$ and any x, a deterministic polynomial time machine with oracle $A \oplus Pr(B)$ can determine by binary search, all possible strings in B up to length $n^k + k$. After that, with the relevant strings of B known, with one deterministic query to $PSPACE$ it can be determined if x is accepted by N_i^A. Thus, NP^A has a sparse $\leq_T^P$-complete language.

As for most problems (but clearly not all [Har 85]), that can be diagonalized in two contradictory ways, the above result suggests that it may be very difficult to show that the existence of a sparse $\leq_T^P$-complete set for NP implies that $P = NP$.

At the same time, it is known that the existence of sparse $\leq_T^P$-hard or complete sets would have very dramatic consequences by collapsing the Polynomial Time Hierarchy, PH.

First, we point out an interesting connection between sparse oracles for NP and polynomial size circuits.

We say that a language A, $A \subseteq (0 + 1)^*$, has *polynomial size circuits* if there exist a k and a sequence of circuits C_1, C_2, C_3, $\cdots$ such C_i has i input lines, one output line and:

$$\forall\ (i, x)\ \ [\,|C_i| \leq i^k + k \ \text{ and } [\,|x| = i \ \ \Rightarrow\ \ [C_i(x) = 1 \ \ x \in A]].$$

The following result has been observed by A. Meyer and reported in [BH 77].

Theorem: A has polynomial size circuits iff $A \in P^S$ for a sparse S.

Karp, Lipton and Sipser showed that if NP has a sparse oracle S, $NP \subseteq P^S$, then PH is finite [KL 80].

Theorem: If S is sparse and $NP \subseteq P^S$ then $PH \subseteq \Sigma_2^P = \Pi_2^P$.

Theorem: If S is sparse and $EXPTIME \subseteq P^S$ then $EXPTIME = \Sigma_2^P \neq P$.

Stronger results have been obtained by Mahaney if S is in NP.

Theorem: If a sparse set S is in NP and $NP \subseteq P^S$, then:

$$PH \subseteq P^S.$$

This result can be established by determining by binary search, in polynomially many queries in n all the strings in S up to length n. Clearly, once the relevant strings of S are known, in polynomially time, we can accept any set in PH by successive elimination of qualifiers using S.

The above result seemed to be optimal, since polynomially many queries to S are needed to obtain all the strings in S up to length n, in order to solve the problems in PH.

Quite surprisingly, far fewer queries to a sparse S are sufficient to solve all problems in PH. The best, and very likely the optimal result in this area, was obtained very recently by J. Kadin [Kad 86a and 86b].

Let $P^{A[logn]}$ denote the family of languages accepted by deterministic polynomial time machines that for an input of length n make at most $c \bullet \log n$ queries to the oracle A.

Theorem: If for a sparse set S in NP

$$NP \subseteq P^S,$$

then

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

This result is proven by the very subtle observation that for input x, $|x| = n$, in $c \log n$ queries we can determine the census of S up to size $n^k + k$, i.e., $|S \cap \Sigma^{\leq n^k + k}| = C_S(n^k + k)$.

Once the relevant census is known, only one more query to SAT is required to determine if a nondeterministic machine N_i, guessing $C_S(n^k + k)$ strings of S and verifying that it has guessed the correct strings, will accept x.

Very recently, Kadin has shown that there are relativized worlds where his result is indeed optimal. That is, for a base oracle A (not indicated here), $P \neq NP$ and if $NP \subseteq P^S$, then $PH \subseteq P^{SAT\,[\log C_S(n)]}$, but $PH \not\subseteq P^{SAT\,[F(n)]}$ for any $F(n)$ such that:

$$\underset{n \to \infty}{limsup} = \frac{F(n)}{\log C_S(n)} = 0.$$

In other words, we need enough queries to determine the census function of S to capture PH.

Complete Sets for Sparse Sets in NP

We know that there exist sparse sets in $NP-P$ iff $EXPTIME \neq NEXPTIME$ and, furthermore, that if there exists a sparse $\leq_m^P$-complete set for NP then $P = NP$, a $\leq_T^P$-complete sparse set implies that $PH \subseteq P^{SAT[\log n]}$.

At the same time, intuitively we feel that sparse sets in NP should not be as hard to recognize as SAT, since they cannot be $\leq_m^P$-complete (if $P \neq NP$) and that they "contain less information".

The following result gives a precise technical meaning to these intuitive ideas [Har 83b].

Let M_u be a standard universal Turing Machine and define:

$$K[logn, n^2] = \{x \mid (\exists y)\ |y| \leq \log |x| \text{ and } M_u(y) = x \text{ in } |x^2| \text{ steps}\}.$$

In [HY 84] it was shown that $K[log\ n, n^2] \cap SAT$ is $\leq_T^P$-complete for all sparse sets in NP.

Theorem: $\{S \mid S \text{ in } NP \text{ and } S \text{ is sparse}\} \subseteq P^{SAT \cap K[log\ n,\ n^2]}$.

It is not known whether any $\leq_m^P$-complete sets exist for all sparse sets in NP.

The above result shows that sparse sets in NP can be recognized in polynomial time with an oracle consisting of the "easily compressable" formulas in SAT. Furthermore, the results discussed earlier imply that if for the Kolmogorov simple formulas (i.e., $F \in K[log\ n, n^2]$) it is easy to decide whether they are satisfiable then $EXPTIME = NEXPTIME$.

It is also interesting to observe that by means of generalized Kolmogorov complexity, we can characterize all the sparse sets that separate P from NP [HH 86b].

Theorem: If $P^A = NP^A$ and S is sparse, then:

$$P^{A \oplus S} = NP^{A \oplus S}$$

iff for some c,

$$S \subseteq K^A[clogn + c,\ n^c + c].$$

Clearly, if we could characterize all the sets which separate P from NP then we would have solved the $P = ?NP$ problem.

The limitations of sparse oracles is investigated in [BBL*84].

Theorem:

1. PH is finite iff there exists a sparse S such that PH^S is finite.

2. $PH \neq PSPACE$ iff there exists a sparse oracle S such that $PH^S \neq PSPACE^S$.

The proofs of these results are based on the observation that the higher levels of the polynomial time hierarchy, just as in PSPACE, we have enough computing power to "pull-down" S and print all relevant strings of S on tape. Thus, the power of the sparse oracle is the same for PH and $PSPACE$ and they can differ iff they differ in the unrelativized case. Clearly, if $PH \neq PSPACE$ then they are different for $S = \emptyset$, which is sparse.

This result complements nicely [Yao 85].

Finally, very recently, to explore the full logical freedom of relativization, the author has constructed an oracle A such that $NP^A \neq coNP^A$ but:

$$PH^A = [P^A]^{NP^A[1]} = PSPACE^A,$$

where $P^{NP^A[1]}$ denotes the family of languages accepted in polynomial time with one oracle query to NP^A.

Conclusion

The results reviewed in this paper indicate a surprisingly rich structure of the complexity classes of feasible computations and raise many new questions about the quantitative nature of computations. These problems seem to be mathematically very hard and their solution, leading to a deeper understanding of the computational comaplexity of feasible computations and the computational nature of mathematics itself, is a major challenge to computer science and mathematics.

I believe that none of us who started the systematic study of computational complexity in the early sixties, fully realized the richness and magnitude of the research area we had entered. Nor do I believe that any of us expected to run into so many hard problems so soon. At the same time, we now fully realize the importance of these problems and see them as central to the full understanding of the quantitative nature of computing and mathematics.

References

[BE 78] P. Berman. Relationships between density and deterministic complexity of NP-complete languages. *Proceedings of the 5th International Colloquium on Automata, Languages, and Programming,* Springer-Verlag *Lecture Notes in Computer Science,* 62, pp. 63-71, 1978.

[Bet 78] R.V. Book, et al, "Inclusion Complete Tally Languages and the Hartmanis-Berman Conjecture", Math. Systems Theory 11, pp. 1-8, 1978.

[BBL *84] J. Balcazar, R. Book, T. Long, U. Schoening, and A. Selman. Sparse oracles and uniform complexity classes. In *Proceedings IEEE Symposium on Foundations of Computer Science,* pp. 308-313, 1984.

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

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

[Bo 74] R.V. Book. Tally Languages and Complexity classes. *Information and Control 26,* pp. 186-193, 1974.

[CH 86] J. Cai and L. Hemachandra. The Boolean Hierarchy: hardware over *NP*. In *Structure in Complexity Theory* Springer-Verlag *Lecture Notes in Computer Science #233,* pp. 105-124, 1986.

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

[GJ 86] J. Goldsmith and D. Joseph. Three results on the polynomial isomorphism of complete sets. In *Proceedings IEEE Symposium on Foundations of Computer Science,* pp. 390-397, 1986.

[GS 84] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. In *Proceedings IEEE Symposium on Foundations of Computer Science,* pp. 495-503, 1984.

[Har 83a] J. Hartmanis. On Sparse Sets in $NP-P$: *Information Processing Letters 16,* pp. 55-60, 1983.

[Har 83b] J. Hartmanis. Generalized Kolmogorov complexity and the structure of feasible computations. In *Proceedings IEEE Symposium on Foundations of Computer Science,* pp. 439-445, 1983.

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

[HH 86a] J. Hartmanis and L. Hemachandra. Complexity classes without machines: on complete languages for UP. In *Automata, Languages, and Programming (ICALP 1986)* Springer-Verlag *Lecture Notes in Computer Science #226,* pp. 123-135, 1986.

[HH 86b] J. Hartmanis and L. Hemachandra. On Sparse Oracles Separating Feasible Complexity Classes. In *STACS: 3rd Annual Symposium on Theoretical Aspects of Computer Science,* Springer-Verlag *Lecture Notes in Computer Science #210,* pp. 321-333, 1986.

[HH 87] J. Hartmanis and L. Hemachandra. One-Way Functions, Robustness, and the Non-Isomorphism of NP Complete Sets. Department of Computer Science, Cornell University, Ithaca, New York. Technical Report TR 86-796. To be presented at *Structure in Complexity Theory,* 1987.

[HI 85] J. Hartmanis and N. Immerman. On complete problems for NP $\cap$ coNP. In *Automata, Languages, and Programming (ICALP 1985),* Springer-Verlag *Lecture Notes in Computer*

Science #194,pp. 250-259, 1985.

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

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

[HY 84] J. Hartmanis and Y. Yesha. Computation times of NP sets of different densities. *Theoretical Computer Science*, 34, pp. 17-32, 1984.

[JY 85] D. Joseph and P. Young. Some remarks on witness functions for non-polynomial and non-complete sets in NP. *Theoretical Computer Science*, 39, pp. 225-237, 1985.

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

[Kad 86b] J. Kadin. $P^{NP[log]}$ and sparse Turing complete sets for NP. 1986. Accepted for *Structure in Complexity Theory*, 1987.

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

[KMR 86] S. Kurtz, S. Mahaney, and J. Royer. Collapsing degrees. In *Proceedings IEEE Symposium on Foundations of Computer Science*, pp. 380-389, 1986.

[Kol 65] A. Kolmogorov. Three approaches for defining the concept of information quantity. *Prob. Inform. Trans.*, 1, pp. 1-7, 1965.

[Koz 80] D. Kozen. Indexing of Subrecursive classes. *Theoretical Computer Science*, pp. 277-301, 1980.

[Kur 83] S. Kurtz. A relativized failure of the Berman-Hartmanis conjecture. 1983. Unpublished manuscript.

[Kur 85] S. Kurtz. Sparse Sets in $NP-P$: Relativizations. *SIAM Journal of Computing 14*, pp. 113-119, 1983.

[La 75] R. Ladner. On the Structure of polynomial time reducibility. *Journal of the Association for Computing Machinery* 22, pp. 155-171, 1975.

[Li 85] Ming Li. Lower Bounds in Computational Complexity, Ph.D. Thesis, Cornell University, Ithaca, New York, 1985.

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

[Long 85] T.J. Long. On restricting the size of oracles compared with restricting access to oracles. *SIAM Journal on Computing*, 14, pp. 585-597, 1985.

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

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

[Mah 83] S. Mahaney. On the number of p-isomorphism classes of NP-complete sets. *Proceedings of the 22nd IEEE Symposium on Foundations of Computer Science*, pp. 130-143, 1981.

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

[MY 85] S. Mahaney and P. Young. Reductions among polynomial isomorphism types. *Theoretical Computer Science*, pp. 207-224, 1985.

[Rac 82] C. Rackoff. Relativized questions involving probabilistic algorithms. *Journal of ACM*, 29, pp. 261-268, 1982.

[Re 83] K. Regan. On diagonalization methods and the structure of language classes. *Proceedings FCT '83*, Springer-Verlag *Lecture Notes in Computer Science #158*, pp. 368-380, 1983.

[Re 86] K. Regan. On the separation of complexity classes. Ph.D. dissertation, Oxford University, August 1986.

[Ro 67] H. Rogers, Jr. "Theory of Recursive Functions and Effective Computability", McGraw-Hill, New York, NY, 1967.

[Sch 86] U. Schoening. *Complexity and Structure.* Springer-Verlag *Lecture Notes in Computer Science #211,* 1986.

[Sip 82] M. Sipser. On relativization and the existence of complete sets. In *Automata, Languages, and Programming (ICALP 1982),* Springer-Verlag *Lecture Notes in Computer Science #140,* pp. 523-531, 1982.

[SG 84] A.L. Selman and J. Grollmann. Complexity measures for public-key cryptosystems. *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science,* pp. 495-503, 1984.

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

[Yao 85] A. Yao. Separating the Polynomial-time hierarchy by oracles. In *Proceedings IEEE Symposium on Foundations of Computer Science,* pp. 1-10, 1985.