ON CENSUS COMPLEXITY

AND SPARSENESS OF NP COMPLETE SETS

by

J. Hartmanis and S. Mahaney

TR 80-416

Department of Computer Science
Cornell University
Ithaca, New York 14853

ON CENSUS COMPLEXITY AND SPARSENESS OF NP COMPLETE SETS

J. Hartmanis and S. Mahaney
Department of Computer Science
Cornell University
Ithaca, New York 14853

Abstract

In this note we show that if there are sparse NP complete sets with a polynomial time computable census function then P = NP. We also derive related results about the complexity of the census function for context-sensitive languages and log n-tape bounded languages.

1. Introduction

It was conjectured in [1] that all NP complete sets are isomorphic under polynomial time mappings and it was shown that all the <u>known</u> NP complete sets are indeed polynomial time isomorphic. If all NP complete sets are polynomial time isomorphic then P $\neq$ NP and no sparse sets can be NP complete; a set $A \subseteq \Sigma^*$, is <u>sparse</u> if there exists a k such that for all n

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

In particular, it was conjectured in [1] that no set over a single letter alphabet can be NP complete if P $\neq$ NP.

This last conjecture was proven correct in [2] and in [3]. This result was extended to show that if there exists a sparse set complete (or hard) for CO-NP then again P = NP. These results were further refined and extended in [7] where, among other results, it was shown that the existence of a sparse complete (or hard) set for PTAPE implies that P = PTAPE.

At the present, it still appears very unlikely that there could exist sparse NP complete sets. Such sets would reveal that there exists a radically different class of NP problems and they would provide a means of solving NP complete problems with a (precomputed) polynomially long oracle tape [4].

Though the original conjecture about the existence of sparse NP complete sets remains open, we show in this note further conditions under which the existence of sparse NP complete sets implies that P = NP, and establish other results relating the complexity of the census function of a set to properties of nondeterministic computations.

For a set $A \subseteq \Sigma^*$, let the census function F_A be given by

$$F_A(n) = \{x \mid x \in A \text{ and } |x| \le n\}.$$

Clearly, if $F_A(n)$ is polynomial time computable then so is

$$F_A^*(n) = |A \cap \Sigma^n| .$$

We show that if there exists a sparse NP complete set A with a polynomial time computable census function, F_A, then P = NP.

By similar methods we show that the context-sensitive languages are closed under complementation if and only if there exists a complete context-sensitive language, A, whose census function can be computed on a nondeterministic linearly bounded automaton. We also discuss log n-tape languages.

Since it is currently conjectured that P ≠ NP and that there exist context-sensitive languages whose complements are not context-sensitive languages, we conjecture that the deterministic and nondeterministic languages differ in that deterministic languages have complete sets with easily computable census functions and that this is not the case for nondeterministic languages.

2. Census Functions and Sparseness

In this section we show that if there exists a sparse NP complete set whose census function is polynomial time computable then P = NP.

Theorem 1: If $A \subseteq \Sigma^*$, is a sparse set complete for NP, and the census function F_A is polynomial time computable, then

$$\text{CO-NP} = \text{NP}.$$

Proof: Since A is complete for NP, $\bar{A} = \Sigma^{*}-A$ is complete for CO-NP and we will show that $\bar{A}$ is in NP.

To recognize $\bar{A}$ in nondeterministic polynomial time consider the following nondeterministic machine $M_{\bar{A}}$:

for input x the machine computes $F_A^{*}(|x|) = n$ and nondeterministically guesses n different sequences of length $|x|$, $x_1, x_2, \ldots, x_n$. It then verifies, using the nondeterministic polynomial time algorithm for A, that all the sequences are in A. The input x is accepted if and only if $x \neq x_i$, $1 \le i \le n$.

Clearly, for every input x in $\bar{A}$ there is a sequence of guesses which will show in polynomial time that x is in $\bar{A}$ and any such sequence of verifiable guesses identifies only elements of $\bar{A}$.

Thus $\bar{A}$, which is complete for CO-NP, is in NP and, therefore,

$$\text{CO-NP} \subseteq \text{NP},$$

but then A is in CO-NP and we see that

$$\text{CO-NP} = \text{NP},$$

as was to be shown.

□

Combining Theorem 1 with Fortune's result [3] yields:

Corollary 2: If A is a sparse, NP complete set with a polynomial time computable census function F_A, then

$$P = NP.$$

Proof: Under the above hypothesis, by Theorem 1 we know that

$$NP = CO\text{-}NP$$

But then A is a sparse complete set for CO-NP which, by Fortune's result [3], implies that

$$P = NP.$$

□

It is interesting to note that in the above result we made essential use of the fact that A is an NP complete set, whereas the results cited before [2,3,7] used only the fact that the sparse sets were NP or CO-NP hard.

The above result can be extended to nondeterministically computable census functions as done in the next section for context-sensitive languages.

It is our conjecture that for no NP complete set, sparse or otherwise, is the census function computable in polynomial-time. At the same time, so far we have not been able to derive any interesting implications about NP from the assumption that the census function is polynomial-time computable. The situation is different for context-sensitive languages, as shown in the next section.

3. Context-Sensitive Languages

In this section we show that the existence of complete context-sensitive languages with easily computable census functions implies that the context-sensitive languages are closed under complements. Let CSL and DCSL denote the families of nondeterministic and deterministic context-sensitive languages, respectively.

We say that A is <u>complete for CSL</u> if any B in CSL can be reduced to A by a deterministic log n-tape transducer which does not increase the length of the output by more than a constant factor times the input length, i.e., $|f(x)| \le C\ |x|$.

Under this definition it is quite easily seen that there do not exist sparse, complete languages for CSL.

We say that a function $f: N \to N$, is <u>computable by a nondeterministic linearly bounded automaton</u> (ndlba) iff there exists a ndlba, M, such that if M halts on input a^n then $M(a^n) = f(n)$.

<u>Theorem 3</u>: The context-sensitive languages are closed under complement if and only if there exists a complete context-sensitive language, A, with the census function, F_A, computable by a ndlba.

<u>Proof</u>: If F_A^* is computable on a ndlba then $\bar{A}$ is recognizable by the following ndlba M:

> for input x' M computes $F_A^*(|x|)=N$. It then guesses in increasing order (by lexiographic ordering) N different strings of length $|x|$ and for each string, using the ndlba for A, checks that x_i is in A and $x_i \neq x$. x is acceptable iff all guesses are properly verified.

Clearly x in $\bar{A}$ is accepted by M for some sequence of proper guesses and any x accepted is in $\bar{A}$. Thus $L(M) = \bar{A}$, where M is a ndlba.

Conversely, if the context-sensitive languages are closed under complementation then for any complete language A, A and $\bar{A}$ are recognized by ndlba, M_A and $M_{\bar{A}}$, respectively. To compute $F_A(n)$, we combine M_A and $M_{\bar{A}}$ in an ndlba, M, which exhaustively runs through all strings x, $|x|=n$, say in increasing order, and simultaneously checks if x in A or $\bar{A}$, counting the elements found to be in A. If all strings are found to be in A or $\bar{A}$ then M outputs the number of elements found to be in A. Clearly, M is a ndlba which computes F_A^*, as was to be shown.

□

4. LOG-Tape Bounded Computations

For the sake of completeness we discuss what is known about sparse complete sets for nondeterministic log n-tape bounded computations (under restrictions by deterministic log n-tape transducers [5]). We denote the families of languages accepted by nondeterministic and deterministic log n-tape automata by NLOG and LOG, respectively.

First of all, we conjecture that there do not exist sparse complete sets for NLOG (and therefore NLOG ≠ LOG), though we have not been able to prove this. As a matter of fact, we have not been able even to prove that if $A \subseteq a^*$ is complete for NLOG then NLOG = LOG. The following easily verifiable fact, was observed in [6].

Fact 4: If A, $A \subseteq a^*$ is complete for NLOG then,

$$\text{NLOG} = \text{LOG iff DCSL} = \text{CSL}.$$

Theorem 5: If A, $A \subseteq a^*$, is complete for NLOG and F_A is computable on a nondeterministic log n-tape automaton, then NLOG is closed under complementation.

every language in NLOG can be recognized by an log n-tape automaton which scans its input deterministically and uses nondeterministic computations after this phase without looking at the input.

For any input a^n we use such an automaton to store k in binary form on its work tape, compute (nondeterministically) $F_A(n)=N$, then guesses on the available work tape successively N integers $n_1 < n_2 < \dots < n_N \le n$ and tries to verify (successively) that a^{n_i} is in A for $1 \le i \le N$. If the verification succeeds, then

$$a^n \text{ is in } \bar{A} \text{ iff } n \neq n_i,\ 1 \le i \le N,$$

which is checked for each successive guess. Thus we see that $\bar{A}$ is accepted by a nondeterministic log n-tape automaton, as was to be shown.

□

References

1. L. Berman and J. Hartmanis, "On Isomorphisms and Density of NP and Other Complete Sets," SIAM J. Comput., 6 (1977), pp. 305-323.

2. P. Berman, "Relationships Between Density and Deterministic Complexity of NP Complete Languages," Fifth Int. Symp. on Automata, Languages and Programming, Udine, Italy, Springer-Verlag Lecture Notes in Computer Science, Vol. 62, 1978, pp. 63-71.

3. S. Fortune, "A Note on Sparse Complete Sets," SIAM J. Comput., 8 (1979), pp. 431-433.

4. J. Hartmanis "Feasible Computations and Provable Complexity Properties," CBMS-NSF Regional Conference Series in Applied Math., Vol. 30, SIAM, Philadelphia, PA, 1978.

5. J. Hartmanis, N. Immerman, and S. Mahaney, "One-Way Log-Tape Reductions," IEEE 19th Annual Symposium on Foundations of Computer Science, October 16-18, 1978, pp. 65-72.

6. J. Hartmanis and S. Mahaney, "Languages Simultaneously Complete for One-Way and Two-Way Log-Tape Automata," Department of Computer Science, TR 80-413, Cornell University, Ithaca, NY.

7. A.R. Meyer and M.S. Paterson, "With What Frequency are Apparently Intractable Problems Difficult?" to be published.