ON POLYNOMIAL TIME ISOMORPHISMS OF COMPLETE SETS[+]

L. Berman

J. Hartmanis

TR 76-297

Department of Computer Science
Cornell University
Ithaca, N.Y. 14853

ON POLYNOMIAL TIME ISOMORPHISMS OF COMPLETE SETS+

J. Hartmanis and L. Berman
Department of Computer Science
Cornell University
Ithaca, NY

Abstract

In this note we show that the recently discovered NP complete sets arising in number theory, the PTAPE complete sets arising in game theory and EXPTAPE complete sets arising from algebraic word problems are polynomial time isomorphic to the previously known complete sets in the corresponding categories.

+This research has been supported in part by National Science Foundation Research Grant DCR 75-09433.

1. Introduction

The investigation of lower level computational complexity and of analysis of algorithms has been strongly influenced by the study of efficient reducibilities and the resulting discovery of complete problems in various complexity classes, [1,4,5]. The investigation of the complexity classes NP, PTAPE, and EXPTAPE has shown that they are fundamental to a real understanding of complexity theory and that complete problems for those classes appear naturally in computer science, operations research, and also in many branches of mathematics such as number theory, game theory, and abstract algebra. As a matter of fact a bewildering variety of complete problems have been found for these classes. In particular, the families NP and PTAPE have yielded suprisingly many complete problems.

In [3] polynomial time computable isomorphism (p-isomorphism) was investigated, and necessary and sufficient conditions were discovered that guarantee that a given NP complete set is polynomial time isomorphic to a standard NP complete set, say the conjunctive normal form satisfiability problem for Boolean functions Using these methods it was shown that all the well known NP complete problems are isomorphic under p-time mappings. This established that inspite of the different origins and attempted simplifications all the classical NP complete problems are essential identical. Similar p-isomorphism results were obtained for the well known PTAPE complete sets, again showing them to be essentially the same set [3].

Since then several other interesting problems have been shown to be complete for the classes NP, PTAPE, and EXPTAPE. At the Eighth Annual ACM Symposium on Theory of Computation (1976) it was shown that

(a) NP complete problems arise naturally in number theory [7],

(b) a large number of problems about winning strategies in game theory are PTAPE complete [10],

(c) certain word problems in algebra (equivalently, certain problems concerning properties of Petri nets) are complete in EXPTAPE [7].

At the FOCS (1976) it was shown that numerous questions related to divisibility of sparse polynomials are NP complete [9].

The purpose of this paper is to show that these new complete problems are polynomial time isomorphic to the corresponding classical complete problems in their respective classes.

2. Isomorphisms of NP Complete Sets

We recall that

NP = {L|L is accepted by a non-deterministic Turing machine in polynomial time}

PTAPE = {L|L is accepted by a deterministic Turing machine in polynomial tape}

EXPTAPE = {L|L is accepted by a deterministic Turing machine in 2^{cn} tape, for some c>0, and n = length of input}.

We say that A, $A \subseteq \Sigma^*$, is NP complete if and only if A is in NP and for every L in NP there exists a polynomial time computable function f such that

$$x \in L \text{ if and only if } f(x) \in A.$$

PTAPE <u>complete</u> and [illegible]APE <u>complete</u> sets are defined simil

The notion of polynomial completeness has been of enormous use in classifying recursive sets; however, it does have its limitations. The class of NP complete sets contains many sets of practical importance; and so, it is natural to study these sets more closely in an attempt to gain greater insight into whatever structural properties they possess which make them har

As an attempt to capture the notion of "polynomial structur identity" we have made the following definitions

<u>Definition</u>: Two sets A and B are <u>polynomial time isomorphic</u> (p-isomorphic) if there is a function f satisfying the following properties:

1. f is 1-1 and onto;
2. $x \in A$ iff $f(x) \in B$
3. both f anf f^{-1} can be computed in p-time.

One should note the similarity between this definition and the definition of recursively isomorphic.

Let CNF-SAT designate the set of all satisfiable Boolean formulas in conjuncture normal form. It is known that CNF-SAT is a NP complete set [1]. From Theorems 1 and 6 in [1] we can derive the following result.

<u>Theorem NP</u>: An NP complete set B is p-isomorphic to CNF-SAT if and only if there exist two p-time computable functions S_B and D_B such that

1. $(\forall x,y)[S_B(x,y)\in B \text{ iff } x\in B]$
2. $(\forall x,y)[D_B(S_B(x,y))=y]$.

Thus to determine whether an NP complete set B is p-isomorphic to the classic NP complete sets [5], such as CNF-SAT, we just have to check whether the set B admits the two p-time computable functions S_B and D_B. The function S_B is a polynomial time padding function which encodes arbitrary strings y in x while preserving the membership in the set B, and the function D_B must reverse this process by determining in polynomial time what string was encoded into x. It should be pointed out that these are very simple conditions and in part the purpose of this paper is to demonstrate how easily these conditions can be verified for different sets.

We illustrate this with an interesting new NP complete set arising from quadratic Diophantine equations. In [7] it was shown that the set

$$\text{DIOPH} = \{ax^2+by-c \mid a,b,c\geq 0 \text{ are integers and there are positive integers } x_0,y_0 \text{ such that } ax_0^2+by_0-c=0\}$$

is NP complete (where ax^2+by-c is encoded in standard binary form).

<u>Corollary</u>: DIOPH is p-isomorphic to CNF-SAT.

<u>Proof</u>: From [7] we know that DIOPH is an NP complete set. Therefore, from our previous theorem we just have to verify that there exist two p-time functions S and D satisfying the two conditions of the theorem.

Define the encoding function $S((a,b,c),n)$ as follows: Let $\hat{n}$ = the integer obtained by concatenating 1 and n treating the resulting binary string as an integer. Let n_i be the $i^{\underline{th}}$ digit of $\hat{n}$ when $\hat{n}$ is expressed in binary.

If $b \neq 0$ then

begin find smallest prime so that p does not divide b

$$\text{let } j = 2\cdot(\lfloor \log_p b \rfloor + 1)(\lfloor \log_2 \hat{n} \rfloor + 1)$$

$$n' = \left[\sum_{i=0}^{\lfloor \log_2 \hat{n} \rfloor} n_i p^{2i(\lfloor \log_p b \rfloor + 1)} \right] + p^{j-(\lfloor \log_p b \rfloor + 1)}$$

$$a' = p^j a$$

$$b' = b$$

$$c' = p^j c + bn'$$

end

if $b = 0$ then

begin if there are natural number solutions

then begin for any p-time pairing function f

$$a' = 1$$

$$b' = 0$$

$$c' = [f(f(a,c),n)]^2$$

end

```
else begin a' = 1
           b' = 0
           c' = 1 + [f(f(a,c),n)]^2
      end
end
```

We must now show that the above function $S(-,-)$ has the desired properties. If $b = 0$, the correctness of S is clear since square roots can be performed in p-time and both f and f^{-1} are in p-time by assumption.

If $b \neq 0$ the situation is less immediate. First, notice that p, the smallest prime not dividing b, can be found in p-time since for large n the product of primes less than n is $O(2^{2^n})$. Therefore, there is a prime $p < \lfloor \log \log b \rfloor$ which does not divide b and these can all be checked in time polynomial in $\log b$. This establishes that $S(-,-)$ can be computed in p-time. Note also that given $S((a,b,c),n)$ we can recover b and therefore also p.

The following observations will be useful in showing that $S(-,-)$ preserves membership in DIOPH:

1) $p^j > bn'$

2) n' is a sequence of blocks of $(\lfloor \log_p b \rfloor + 1)$ digits

when expressed in base p. It is also self delimiting, i.e. given

n' =

	$\lfloor \log_p b \rfloor + 1$	

p and b and any string which ends in bn', n' can be recovered from the end of the string. Therefore given $S((a,b,c),n)$, we can compute n. This guarantees that the decoding function D exists and so, if $S(-,-)$ preserves membership, we are done.

Let (x_0,y_0) be a solution to $ax^2+by-c+0$ then $(x_0,p^j y_0+n')$ is a solution to $p^j ax^2+by - (p^j c+bn')$.

If (x_1,y_1) is a solution to

$$ap^j x^2 + by - (p^j c+bn') = 0$$

then we claim that $(x_1,\frac{y_1-n'}{p^j})$ is a natural number solution to

$$ax^2 + by - c = 0.$$

First, notice that p^j divides $b\cdot(y_1-n')$ and p does not divide b so p^j divides (y_1-n').

$$p^j[ax_1^2+b(\frac{y_1-n'}{p^j}) - c] = ap^j x_1^2 + by_1 - (p^j c+bn') = 0.$$

So $(x_1,\frac{y_1-n'}{p^j})$ is an integer solution and it merely remains to show that $y_1-n'\geq 0$, or equivalently that $ax_1^2-c \leq 0$. Now

$$p^j(ax_1^2-c) = b(n'-y_1)$$

if $ax_1^2-c>0$ then since $y_1>0$ and $b>0$ $p^j \leq b(n'-y_1) < bn'$ a contradiction to observation 1.

Therefore, the function $S(-,-)$ satisfies the requirements of the NP Theorem and by observation 2 the needed $D(-)$ function exists. Thus

$$ax^2 + by - c\in DIOPH \text{ iff } S[ax^2 + by-c,d]\in DIOPH.$$

Therefore, the two p-time computable functions D and S have the

quired properties for DIOPH and we conclude from Theorem NP
at DIOPH is p-isomorphic to CNF-SAT, as was to be shown.

The above problem is unusual only in that the encoding and de-
ding functions are difficult to compute. This, no doubt, reflects
e complexity of the reduction used to show the problem NP-
mplete. Our next example is again drawn from questions in
lassical mathematics; however, our isomorphism results apply in a
ich more direct manner.

We consider the set

$$IV = \{(\alpha_1,\ldots,\alpha_{k1};\beta_1,\ldots,\beta_{k2}) \mid \prod_{j=1}^{k2}(x^{\alpha_j}-1) \text{ is not a factor of } \prod_{j=1}^{k2}(x^{\beta_j}-1)\}.$$

n [9] it was shown that this set is NP-complete. We now show:

<u>Theorem:</u> DIV is p-isomorphic to CNF-SAT.

<u>Proof:</u> Consider the map $S(w,y)$ defined as

$$S((\alpha_1,\ldots,\alpha_{k1};\beta_1,\ldots,\beta_{k2}),y) = (\alpha_1,\ldots,\alpha_{k1},y;\beta_1,\ldots,\beta_{k2},y)$$

It is immediate that $S(w,y)\in DIV \iff w\in DIV$. Letting $D(-)$ be the
obvious function shows us that DIV satisfies the hypothesis of
Theorem NP and our theorem is established.

We should also note that many sets which are not known to be
NP complete do have our D and S function and will therefore be
isomorphic to CNF-SAT if they turn out to be NP complete.

Theorem: If Graph Isomorphism is NP hard then it is isomorphic to CNF-SAT.

Proof: Graph isomorphism admits the S and D function and is in NP. If it should be NP hard, (i.e. if every NP set could be many one reduced to it) the hypothesis of Theorem NP would be satisfied.

Theorem: If NP = PSPACE then INEQ(0,1,+,·,*,),(,) is p-isomorphic to CNF-SAT.

Proof: If NP = PSPACE then INEQ is NP complete and so the hypothesis of theorem NP are satisfied

In a similar fashion, our results apply almost immediately to every natural set we know of which is not known to be in P.

3. Isomorphisms of PTAPE Complete Sets

From Theorems 7 and 11 in [3] we can derive a result for p-isomorphisms of PTAPE complete sets, similar to the previous result for NP complete sets.

It is known that

$L_{\Sigma^*}=\{R \mid R$ is regular expression over $\Sigma,(,),\cdot,\cup,*$ and $L(R)=\Sigma^*\}$

is a PTAPE complete set [8].

Theorem PTAPE: A PTAPE complete set B is p-isomorphic to L_{Σ^*} if and only if there exist two p-time computable functions S_B and D_B such that

1. $(\forall x,y)[S_B(c,y) \in B \text{ iff } x \in B]$
2. $(\forall x,y)[D_B(S_B(x,y)) = y]$.

In [10] a large number of new sets arising from decision problems based on finite two-person perfect-information games were shown to be PTAPE complete. We will select a few representatives of these sets and show that they are p-isomorphic to L_{Σ^*}. The reader should be able to supply similar proofs for all other PTAPE complete sets in [10]. Note that in [10] it is shown that these sets are PTAPE complete under log-tape reducability. Since log-tape computations can be performed in polynomial time we know that these sets are also complete under polynomial time reductions, as defined in this paper.

For all the games described below we say that there exists a <u>winning strategy</u> iff there exists a winning strategy for the player who starts the game. The players alternate in successive moves. We assume that the games are encoded by a simple and straight forward method, and for the sake of brevity, we will describe them without always referring to thse encodings.

1. Input is a graph. Each player on his move places a marker on an unoccupied node which is not adjacent to any occupied node. Loser is first player unable to move. $L_1=\{G|G$ is a graph with winning strategy$\}$.
2. Input is a positive (i.e. no negations are present) CNF Boolean formula A. Each player on his move chooses

a variable in A which has not yet been chosen. After all variables have been chosen the starting player wins iff A is true when all the variables chosen by him are set to true and those chosen by his opponent to false.

$L_2 = \{A \mid A \text{ CNF formula with a winning strategy}\}$.

3. Input is two collections of finite sets of integers. $\mathbf{A} = \{Ai \mid 1 \le i \le m\}$ and $\mathbf{B} = \{Bi \mid 1 \le i \le n\}$. The players take turns choosing integers from the union of all the unoccupied sets Ai and Bi. A set is said to be occupied if some integer in it has been played. The starting player wins if all sets in $\mathbf{A}$ are occupied before all sets in $\mathbf{B}$ are occupied. Any player who simultaneously occupies the last unoccupied sets of $\mathbf{A}$ and $\mathbf{B}$ loses.

$L_3 = \{(\mathbf{A}, \mathbf{B}) \mid \text{starting player has winning strategy}\}$

<u>Corollary</u>: The sets L_1, L_2 and L_3 are all p-isomorphic to L_{Σ^*}.

<u>Proof</u>: By the previous theorem we just have to show that each of these sets, L_i, admits two p-time computable functions D_i and S_i, $1 \le i \le 3$, satisfying the conditions of the theorem.

To show that such functions exist for L_1, we consider three graphs which consist of a simple cycle through four, five and six nodes, respectively. It is easily seen that for each of these graphs the second player has a winning strategy. He can pick a node so that no further play is possible in the graph. We use this fact to construct the function S_1 as follows:

let G be a description of a graph and $y \in \{0,1\}^*$, then $S_1(G,Y) = G^1$, where G^1 is a description of the graph G followed by (descriptions

:) a six-cycle graph (as a marker) followed by a sequence of lescriptions of) four and five-cycle graphs encoding the digits : y (a four-cycle denotes a "one" and five-cycle denotes a :ero"). The function S_1 is p-time computable (for any straight orward encoding of graphs) and, furthermore, G is in L_1 iff $_1$(G,y) is in L_1. To see this we just have to observe that if ere is a winning strategy in G then there is one in S_1(G,y), since ie first player starts in G and any attempt to use the additional aphs (by the second player) results in a two move sequence iich cannot change the outcome of the game. If G is not in $_1$, then neither is S_1(G,y) since starting in or entering any f the added graphs just delays the outcome of the game in G by n even number of moves. Finally, we note that there exists a -time function D_1 such that $D_1[S_1(G,y)] = y$. D_1 just checks the ighest numbered nodes for four or five-cycles until it detects six-cycle or detects that the graph is not of the form S_1(G,y). his completes the proof that L_1 is p-isomorphic to $L_{\Sigma*}$.

We now show that there exist two p-time computable functions, $_2$ and S_2, satisfying the required conditions for L_2 to be -isomorphic to $L_{\Sigma*}$. For input A and y, $y \in \{0,1\}^*$, $S_2(A,y) = A^1$, here A^1 has enough new variables to add a conjuction of two new ariables (as a marker) followed by conjuctions of three and four ariables, respectively (always using distinct variables) ncoding the zeros, and ones of the sequence y. Clearly, S_2 is -time computable and it is seen that

$$S_2(A,y) \in L_2 \text{ iff } A \in L_2$$

since if A can be made true by the first player then so can A^1 and vica versa. Finally, it is also clear that a p-time computable function D_2 exists such that $D_2[S_2(A,y)] = y$.

In the following we construct the required functions D_3 and S_3 for L_3. For given $\mathbf{A}, \mathbf{B}$ and y

$$S_3[(\mathbf{A}, \mathbf{B}), y] = (\mathbf{A}^1, \mathbf{B}^1),$$

where $\mathbf{A}^1$ and $\mathbf{B}^1$ are the old $\mathbf{A}$ and $\mathbf{B}$ augmented by new sets of integers longer than those used in $\mathbf{A}$ and $\mathbf{B}$. Exactly the same size sets are added to $\mathbf{A}$ and $\mathbf{B}$ and in the same order, but they are over different integers; the first two new (marker) sets have one element each followed by pairs of sets of two and three (new integers) encoding the digits of y. Clearly, S_3 is p-time computable and

$$(\mathbf{A}^1, \mathbf{B}^1) \text{ is in } L_3 \text{ iff } S_3[(\mathbf{A}, \mathbf{B}), y] \text{ is in } L_3,$$

since the added sets just prolong the game but cannot change the outcome. It is also seen that the required p-time computable D_3 exists. This completes the proof that L_1, L_2 and L_3 all are p-isomorphic to $L_{\Sigma*}$.

4. EXPTAPE Complete Problems

We will now show that a recently discovered EXPTAPE complete problem associated with the uniform word problem for commutatative semiproms is p-isomorphic to previously known EXPTAPE complete problems.

Let $M_{i,n}$ be an enumeration of Turing machines which run on 2^{kn} tape. We can start with a standard enumeration of all Turing machines $M_1, M_2, \ldots$ and get $M_{i,k}$ for each integer $k, k \geq 0$, by adding a subroutine to M_i which lays off 2^{nk} tape for inputs of length n and if M_i tries to use more than 2^{kn} tape it is stopped and the input is rejected, otherwise $M_{i,n}$ does the same as M_i. Let

$$L_{EXP} = \{M_{i,k}\#CODE(x_1x_2\ldots x_n) \mid M_i \text{ accepts input } x_1, x_2\ldots x_n \text{ on } 2^{kn} \text{ tape}\}$$

where for $M_{i,k}$ $CODE(x_1x_2\ldots x_n)$ is a straight forward symbol by symbol encoding of the sequence $x_1, x_2\ldots x_n$ such that the length of $CODE(x_1x_2\ldots x_n) = k\cdot c\cdot n$, where c is the number of tape squares needed to encode a tape symbol of M_i (we assume a uniform length encoding). The set L_{EXP} is acceptable on 2^n tape since $M_{i,k}$ operated in $2^{k\cdot n}$ tape and the simulation of $M_{i,k}$ on $x_1x_2\ldots x_n$ can be carried out on $2^{c\cdot k\cdot n}$ tape squares. Furthermore it is seen that L_{EXP} is a complete language in EXPTAPE, since for any L accepted by $M_{i,k}$ the p-time mapping

$$f(x_1\ldots x_n) = M_{i,k}\#CODE(x_1x_2\ldots x_n)$$

is such that

$x_1\ldots x_n$ is in L iff $M_{i,k}\#CODE(x_1x_2\ldots x_n)$ is in L_{EXP}

From Theorems 7 and 11 in [3] we easily get the following result

<u>Theorem EXPTAPE</u>: An EXPTAPE tape complete set L is p-isomorphic to L_{EXP} iff there exist two p-time computable functions D and S such that

$$(\forall xy)[\ D(x,y) \in L \text{ iff } x \in L]$$
$$(\forall xy)[\ S(D(x,y) = y].$$

Recently [2] it has been shown that the uniform word problem for commutative semi groups (UWCS) is an EXTAPE complete language and that equivalently the reachability problem in Petri sets is also complete in EXPTAPE. We now give the definition of the UWCS. Let

Σ be a finite set of generators

and let

$$\mathbf{S} = \{w_i \equiv v_i \mid w_i, v_i \in \Sigma^*,\ 1 \le i \le k\}$$

be a finite set of equivalences including ab = ba for all a,b in Σ. Note that the associativity of concatenation automatically guarantees that we have an associative and commutative operation. Then define

$$\alpha \equiv \beta(\mathbf{S}),\ \text{for } \alpha,\beta \in \Sigma^*,$$

iff by a finite number of applications of the equivalences from $\mathbf{S}$ to substrings in α we can transform α into β. let

$$\text{UWCS} = \{\langle \alpha,\beta,\mathbf{S}\rangle \mid \alpha \equiv \beta(\mathbf{S})\}.$$

orollary: UWCS is p-isomorphic to L_{EXP}.

roof: In [2] it was shown that UWCS is an EXPTAPE complete set nder logn-tape reductions. Thus UWCS is EXPTAPE complete under -time reductitions. Therefore we just have to show that UWCS dmits two p-time computable functions D and S satisfying heorem EXPTAPE.

We construct D as follows:

$$S(\langle\alpha,\beta,\mathbf{S}\rangle,y) = \langle\alpha\bar{y},\beta\bar{y},\mathbf{S}^1\rangle,$$

here we have added two new symbols to Σ, a_0 and a_1, and $\bar{y} = h(y)$ here h is defined by $h(0) = a_0$ and $h(1) = a_1$ and extended o sequences by induction $h(\epsilon) = \epsilon$ and $h(wx) = h(w)h(x)$; the only ew relations added to $\mathbf{S}$ to get $\mathbf{S}^1$ are the ones defining ommutativety, for all x in Σ $xa_0 = a_0x$ and $xa_i = ax_1$. It is een that S is p-time computable and that

$(\langle\alpha,\beta,\mathbf{S}\rangle) \in$ UWCS iff $\langle\alpha\bar{y},\beta\bar{y},\mathbf{S}^1\rangle \in$ UWCS,

ince

$$\alpha\equiv\beta(\mathbf{S}) \text{ iff } \alpha\bar{y}\equiv\beta\bar{y}(\mathbf{S}^1).$$

Finally we note that there exists a p-time computable unction S such that

$$D[S(\langle\alpha,\beta,\mathbf{S}\rangle y)] = y.$$

Thus UWCS is p-isomorphic to L_{EXP}, as was to be shown.

At the 17th STOC, Chandra and Stockmeyer presented numerous new games which are complete for EXPTAPE, the techniques of the last section show immediately that the new sets are p-time isomorphic to L_{EXP}.

References

1. Cook, S. "The Complexity of Theorem Proving Procedures", Proceedings Third Annual ACM Symposium an Theory of Computing, May 3-5, 1971, pp 151-158.
2. Cordova, E., R. Lipton and A.R. Meyer, "Exponential Tape Complete Problems for Petri Nets and Commutative Semigraphs; Preliminary Report", Proceedings Eighth Annual ACM Symposium on Theory of Computing, May 3-5, 1976, pp. 50-54.
3. Hartmanis, J and L. Berman, "On Isomorphisms and Density of NP and other Complete Sets," Proceedings Eighth Annual ACM Symposium on Theory of Computing, May 3-5, 1976, pp 30-40.
4. Hartmanis, J. and J. Simon, "on the Structure of Feasible Computations", in Advances in Computers, Vol 14 (eds. M. Rubinoff and M.C. Yovits) Academic Press, New York, 1976, pp 1-43.
5. Karp, R. "Reducabilities Among Combinatorial Problems", In "Complexity of Computer Computations" (R. Miller and J. Thatcher, eds) pp 85-104 Plenum Press New York. 1972.
6. Ladner, R.E. "On the Structure of Polynomial Time Reducibility" JACM, vol. 22, No. 1, January 1975, pp. 155-171.
7. Manders, K. and L. Adleman, "NP-Complete Decision Problems for Quadratic Polynomials," Proceeding Eighth Annual ACM Symposium on Theory of Computing, May 3-5, 1976 pp. 23-29.
8. Meyer, A.R. and L. Stochmeyer, "The Equvalence Problem for Regular Expressions with Squaring Requires Exponential Space", IEEE 13th Annual Symposium on Switching and Automata Theory, 1972, pp. 125-129,
9. Plaisted, D.A. "More NP-Hard and NP-Complete Problems Involving Polynomials and Integers", IEEE 17the FOCS, 1976.
10. Schaefer, T.J. "Complexity of Decision Problems Based in Finite Two-Person Perfect-Information Games", Proceedings Eighth Annual ACM Symposium on Theory of Computing, May 3-5, 1976, pp. 41-49.