Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis
Series: GI, Gesellschaft für Informatik e. V.

48

Theoretical Computer Science
3rd GI Conference

Darmstadt, March 1977

Springer-Verlag
Berlin · Heidelberg · New York

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis
Series: GI, Gesellschaft für Informatik e. V.

48

Theoretical Computer Science 3rd GI Conference

Darmstadt, March 28–30, 1977

Edited by
H. Tzschach, H. Waldschmidt, H. K.-G. Walter
on behalf of the GI

Springer-Verlag
Berlin · Heidelberg · New York 1977

Program Committee
Prof. Dr. J. Berstel
Prof. Dr. K.-H. Böhling
Prof. Dr. W. Brauer
Prof. Dr. E. Engeler
Prof. Dr. G. Hotz
Prof. Dr. C.-P. Schnorr

Editors
Prof. Dr. Hans Tzschach
Prof. Dr. Helmut Waldschmidt
Prof. Dr. Hermann K.-G. Walter
Institut für Theoretische Informatik
Technische Hochschule Darmsta
Magdalenenstr. 11
6100 Darmstadt/BRD

AMS Subject Classifications (1970): 68-XX, 94-XX, 02-xx, 02Dxx, 02EXX, 02Fxx, 05-04
CR Subject Classifications (1974): 5, 1

ISBN 3-540-08138-0 Springer-Verlag Berlin · Heidelberg · New York
ISBN 0-387-08138-0 Springer-Verlag New York · Heidelberg · Berlin

Printed in Germany
Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr. 2145/3140-543210

VORWORT

Die 3. GI-Fachtagung Theoretische Informatik setzt die Reihe der Vorgängertagungen über Automatentheorie und Formale Sprachen fort. Wie an den hier zusammengefaßten Berichten erkennbar ist, ist mit der Namensänderung eine gewisse Ausweitung der Themenkreise verbunden. Hier sind als Beispiel die Arbeiten über die Deadlock-Problematik zu nennen. Die Arbeiten lassen ferner die derzeitigen Schwerpunkte der Forschung auf dem Gebiet der Theoretischen Informatik erkennen. Der Tagungsband faßt die Vorträge zusammen, die auf der dritten Fachtagung vom 28. - 30. März 1977 an der Technischen Hochschule Darmstadt gehalten werden. Da wie schon bei den Vorgängertagungen an der Form der Tagung ohne Parallelsitzungen festgehalten wurde, mußte aus der erfreulich großen Anzahl von Anmeldungen eine Auswahl getroffen werden, die dem Programmkomitee in vielen Fällen schwergefallen ist.

An dieser Stelle danken die Veranstalter den Vortragenden, Teilnehmern, Helfern und allen, die zum Gelingen der Tagung beigetragen haben, herzlich. Das Bundesministerium für Forschung und Technologie hat durch seine finanzielle Förderung die Durchführung der Tagung ermöglicht. Für großzügige Unterstützung danken wir der Technischen Hochschule Darmstadt und den Spendern aus der Industrie. An den organisatorischen Arbeiten und der Vorbereitung dieses Bandes haben die Herren Dr. H. Becker und Dipl.-Math. P. Ochsenschläger tatkräftig mitgewirkt. Ihnen gilt unser Dank ebenso wie dem Springer Verlag und den Herausgebern der Reihe Lecture Notes in Computer Science für die Aufnahme des Tagungsberichts in diese Reihe.

Darmstadt, im März 1977

H. Walter
H. Waldschmidt
H. Tzschach

INHALTSVERZEICHNIS

HAUPTVORTRÄGE

VORTRÄGE IN DER REIHENFOLGE DES PROGRAMMS

On polynomial time isomorphisms of complete sets

L. Berman - J. Hartmanis

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.

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

---

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

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 essentially 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 [2].

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. Isomorhisms 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 polnomial time computable function f such that

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

PTAPE complete and EXPTAPE complete sets are defined similarly.

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 hard.

As an attempt to capture the notion of "polynomial structural identity" we have made the following defintions

Definition: Two sets A and B are polynomial time isomorphic (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 7 and 8 in [3] 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

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

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

Therefore, the two p-time computable functions D and S have the required properties for DIOPH and we conclude from Theorem NP that DIOPH is p-isomorphic to CNF-SAT, as was to be shown.

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

We consider the set

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

In [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 $S(w,y) \in \text{DIV} \iff w \in \text{DIV}$. 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.

Pf: 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

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

is a PTAPE complete set [8].

Theorem PTAPE: A PTAPE complete set B is p-isomorphic to $L_{\Sigma}*$ 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 $L_{\Sigma *}$. 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 winning strategy 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. $L_1=\{G|G \text{ is a graph with winning strategy}\}$.
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.

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

3. Input is two collections of finite sets of integers. **A** = {Ai | 1≤i≤m} and **B** = {Bi | 1≤i≤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 **A** are occupied before all sets in **B** are occupied. Any player who simultaneously occupies the last unoccupied sets of **A** and **B** loses.

   L_3 = {(**A**,**B**) | starting player has winning strategy}

Corollary: The sets L_1,L_2 and L_3 are all p-isomorphic to $L_{\Sigma *}$.

Proof: 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 of) a six-cycle graph (as a marker) followed by a sequence of (descriptions of) four and five-cycle graphs encoding the digits of y (a four-cycle denotes a "one" and five-cycle denotes a "zero"). The function S_1 is p-time computable (for any straight forward encoding of graphs) and, furthermore, G is in L_1 iff $S_1(G,y)$ is in L_1. To see this we just have to observe that if there is a winning strategy in G then there is one in $S_1(G,y)$, since the first player starts in G and any attempt to use the additional