ON POLYNOMIAL TIME COMPUTABLE PROBLEMS

T. Kasai
Dept. of Computer Science, The University of Electro-communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182, JAPAN

A. Adachi
Dept. of Academic and Scientific Program, IBM, JAPAN

Abstract. In this paper, we investigate problems which require $O(n^k)$ time, for each interger k, where n is the size of input. Also, we present a number of problems which require exponential time.

1. Introduction

A number of complete problems in various complexity classes are reported. Jones and Laaser [5] showed some familiar problems which are complete in deterministic polynomial time with respect to log space reducibility. A great number of familiar problems have been reported which are complete in NP (nondeterministic polynomial time) [1], [3], [6]. Even and Tarjan [4] considered generalized Hex and showed that the problem to determine who wins the game if each player plays perfectly is complete in polynomial space. Schaefer [9] derived some two-person games from NP complete problems which are complete in polynomial space. However compelling the circumstantial evidence may be, no one has yet been able to prove that NP complete or polynomial space complete problems are actually intractable, and also there are few authors to distinguish the degree of the polynomial when the time complexity of a given algorithm is polynomial. There is general agreement that if problems have no polynomial time algorithm they are intractable. Some problems, however, in the class P are not always tractable. In this point of view, it is significant to find the concrete degree of the polynomial which represent the time complexity of a problem.

In this paper, a technique to obtain a lower bound of the time complexity of problems is introduced. We consider k-pebble games for each integer k, which involves moving pebbles according to certain rules, initially k pebbles are placed on certain places. The goal of the game is to put a pebble on a particular place. We show that the problem to determine whether or not the first player has a forced win in a (2k+1)-pebble game requires $O(n^k)$ time. A rough discussion such as to determine whether or not a given problem belongs to NP is independent of the machine model and the way of defining the size of problems, since any of the commonly used machine models can be simulated by any other with only a polynomial loss in running time and no matter what criteria the size is defined, they differ from each other by polynomial order. However, in precise discussion, for example, in the discussion whether the computation of a problem requires $O(n^k)$ time or $O(n^{k+1})$ time, the complexity heavily depends on machine models and the definition of the size of problems.

From these points, we introduce a somewhat stronger notion of reducibility. Using this reducibility, we show that the game, so called "Cat and 2k+2 Mice problem" requires $O(n^k)$ time. The basic results are also applied to show that certain problems are complete in exponential time. We consider a game, so called "Chinese checkers game," and a game similar to the "Towers of Hanoi." It has been shown that the winning strategy problems of these games are exponential time complete.

2. Preliminaries

In this section, some fundamental notions are introduced for the study of polynomial time computability.

Definition 2.1 A two-person game G is a tripple $G = (X,E,s)$, where:

(1) X is a finite set, an elenent of X is called a position,

(2) E is a subset of $X \times X$, an element of E is called a rule, we sometimes write $x \to y$ for an element (x,y) of E,

(3) s, in X, is called the starting position.

At the beginning of the game, the first player is in position s. If $(x,y) \in E$ for any $x, y \in X$, then a player in his turn may move the position x to the position y. The winner is the player who makes the other player unable to move.

<u>Definition 2.2</u> A <u>pebble game</u> is a quadruple $G = (X,R,S,t)$, where:

(1) X is a finite set of nodes, the number of node is called the <u>order</u> of G,

(2) $R \subseteq \{(x,y,z) \mid x,y,z \in X, x \neq y, y \neq z, z \neq x\}$ is called a set of <u>rules.</u>

(3) S is a subset of X; the number of nodes in S is called the <u>rank</u> of G,

(4) t is a node in X, called the <u>terminal node.</u>

A pebble game of rank k is simply called k-pebble game.

A pebble game is played by two plyers, P_1 and P_2, who alternatively move pebbles on the pebble game, with P_1 playing first. At the beginning of the pebble game, pebbles are placed on all nodes of S. If $(x,y,z) \in R$ and pebbles are placed on x,y but not on z, then we can move a pebble from x to z. The winner is the player who puts a pebble on the terminal node or who makes the other player unable to move.

More formally the two-person game represented by a pebble game is stated as follows.

<u>Definition 2.3</u> A <u>two-person game induced from a pebble game</u> $G = (X,R,S,t)$ is a triple $\overline{G} = (\overline{X},E,S)$, where:

(1) $\overline{X} = \{A \mid A \subseteq X, \#(A) = \#(S)\}$, $\#(A)$ denotes the number of elements in A,

(2) $E = \{(A,B) \mid A,B \in \overline{X}, t \notin A, (x,y,z) \in R, x,y \in A, z \notin A, B = (A - \{x\}) \cup \{z\}\}$.

Throughout this paper, by Turing machines, we mean a single tape Turing machine.

<u>Definition 2.4</u> We denote by N the set of all natural numbers. A <u>problem</u> is a triple (Σ,L,σ), where Σ is an alphabet, L is a subset of Σ^*, and σ is a computable function from Σ^* to N. When Σ and σ are understood, we simply write L instead of (Σ,L,σ). The function σ is called the <u>size function</u> of L, and for each $w \in \Sigma^*$, $\sigma(w)$ is the <u>size</u> of w.

<u>Definition 2.5</u> Let L Σ^* be an arbitrary problem. Suppose that σ is the size function associated with L. Then L is said to be <u>$T(n)$ time computable with respect to the size</u> iff there exists a Turing machine such that for any input x, the computation terminates within $T(\sigma(x))$ steps. We say that a problem L <u>requires $T(n)$ time</u> iff for any T', if $\inf_n T'(n)/T(n) = 0$ then L can not be solved within $T'(n)$ time. In general, we write $T' \ll T$ if $\inf_n T'(n)/T(n) = 0$ for two

functions T and T'.

A Turing machine M is said to be <u>$T(n)$ time bounded with respect to the size σ</u> iff for any input x, the computation terminates within $T(\sigma(x))$ steps. When the size is defined to be the length of inputs, i. e., $\sigma(x) = |x|$, we simply say that M is <u>$T(n)$ time bounded</u>.

<u>Definition 2.6</u> Let T and Z be functions from N to N, such that for any $m, n \in N$, if $m < n$ then $Z(m) < Z(n)$. Let L_1 and L_2 be problems. Let σ_1 and σ_2 be sizes of L_1 and L_2, respectively. Then L_1 is <u>(T,Z)-reducible</u> iff there exists a function f from Σ_1^* to Σ_2^* which satisfies the following:

(1) For any $x \in \Sigma_1^*$, $x \in L_1$ iff $f(x) \in L_2$,

(2) There exists a Turing machine which computes the function f, and for any x, the computation of f terminates within $T(\sigma_1(x))$ steps,

(3) For any $x \in \Sigma_1^*$, $\sigma_2(f(x)) < z(\sigma_1(x)$.

<u>Lemma 2.1</u> Let L_1 and L_2 be problems with the size function σ_1 and σ_2, respectively. Suppose that L_1 is (T,Z)-reducible to L_2 and $T \ll T_1$. If the computation of L_1 requires T_1 time with respect to the size σ_1, then the computation of L_2 requires $T_1 \cdot Z^{-1}$ time with respect to the size σ_2, where $T_1 \cdot Z^{-1}$ stands for the composition of the function T_1 and the inverse function Z^{-1}.

<u>3. Problems which require $O(n^k)$ time.</u>

In this section we consider two problems which require $O(n^k)$ time.

<u>Definition 3.1</u> For each integer k, <u>k-pebble game problem</u> is the problem when a k-pebble game is played by two persons to determine whether the first player has a wining strategy, that is, a way to win the game. In the k-pebble game problem, the size is defined to be the number of nodes.

<u>Theorem 3.1</u> The $2k+1$-pebble game problem requires $O(n^k)$ time.

The proof is done by showing that any $O(n^k)$ time computable problem is $(O(n \log n), O(n))$-reducible to the $2k+1$ pebble game problem. The outline of the proof is as follows: From a given $O(n^k)$ time bounded Turing machine M and an input x of length n, we construct $2k+1$ pebble game such that M accepts x if and only if the first player has a forced win in the $2k+1$ pebble game, and furthermore, the construction can be performed under $(O(n \log n), O(n))$-reducibility.

Definition 3.2 A cat and mice game is a 5-tuple $G = (X,E,S,a,t)$ where: X is a finite set of nodes, $E \subseteq X \times X$ is the set of edges, S is a subset of X, and a and t are elements of $X-S$. The size of G is defined to be the number of nodes. If the number of elements of S is k, then G is called a cat and k-mice game. The game is played on the directed graph (X,E). At the beginning of the game, mice occupy all nodes of S, and the cat occupies the node a. The cat and one of the mice alternate moves according to a single edge of the graph with the cat moving first. The cat wins if the cat and one of the mice occupy the same node. The mice wins if one of the mice reaches the goal t before being caught.

Theorem 3.2 The cat and $2k+2$ mice game problem requires $O(n^k)$ time.

The proof is done by showing that the $2k+1$ pebble game problem is $(O(n \log n), O(n))$-reducible to the cat and $2k+2$ mice problem.

4. Problems which requier exponential time.

In this secrion, the basic results are applied to show that certain games are exponential time complete. From Theorems 3.1 and 3.2, it follows the followings.

Theorem 4.1 (i) The two-person pebble game problem requires exponential time. (ii) The cat and mice problem requires exponential time.

Definition 4.1 A Chinese checkers game is $G = (N,E,W,B,t)$, where N is a finite set of nodes, $E \subseteq N^2$ is the set of edges, W and B are subsets of N such that $W \cap B = \emptyset$, and t is an element of N.

A Chinese checkers game G is a game played on the graph (N,E) between two players, White and Black. White moves first. Initially, White stones are placed on each node of W and black stones are placed on each node of B. Suppose that (x,y) and (y,z) are edges of E. If there are a white stone on x, a black stone on y and no stone on z, then White in his turn can move the stone from x to z. Similarly, if a black stone is on x, a white stone is on y and no stone is on z, then Black can move the black stone from x to z. The player wins if after his move he has a stone on his color on the node t or the other player cannot move any stone of his color.

Theorem 4.2 The problem to determine whether there is a winning strategy in a Chinese checkers game require exponential time.

Definition 4.2 Let Z be the set of integers. A peg game is $G = (V,m,n)$, where $m, n \varepsilon N$ and V is a finite subset of Z^n such that $(v_1,\cdots,v_n) \varepsilon V$ implies $v_1+v_2+\cdots+v_n = 0$.

A peg game can be considered as the game described as follows. There are n pegs fixed upright of a board, and m disks. Each disk has a hole in its center. An element $y = (y_1,\cdots,y_n)$ of N^n represents that y_i disks are threaded on the i-th peg, $i = 1, 2,\cdots, n$. A rule $v = (v_1,\cdots,v_n) \varepsilon V$ means that for each i, if $v_i \geqq 0$ then we put v_i disks on the i-th peg, and if $v_i < 0$ we remove-v_i disks from the i-th peg. Initially, all disks are threaded on the first peg. The object of the game is to transfer all disks to the n-th peg. In the two-person game, when two players althernatively move disks by the rules, the player wins if after his move all disks are on the n-th peg.

Theorem 4.3 A two-person peg game problem requires exponential time.

References

[1] A. V. Aho, J. E. Hopcroft and J. D. Ullman. The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] A. K. Chandra and L. J. Stockmeyer, Alternation, Proc. 17th Ann. IEEE Symp. on Foundation of Computer Sciences, 1976, pp.98-108.

[3] S. A. Cook, The complexity of theorem-proving procedures, Proc. 3rd ACM Symp. on Theory of Computing, 1971, pp.151-158.

[4] S. Even and R. R. Tarjan, A combinatorial problem which is complete in polynomial space, J. Assc. Comput. Mach., 23(1976), pp.710-719.

[5] W. D. Jones and W. T. Laaser, Complete problems for deterministic

polynomial time, Theoretical Comput. Sci., 3(1977), pp.105-117.

[6] R. M. Karp, Reducibility among combinatorial problems, Complexity of Computer Computations, R. E. Miller and J. w. Thatcher, eds., Plenum Press, New York, 1972, pp.85-104.

[7] T. Kasai, A. Adachi and S. Iwata, Classes of pebble games and complete problems, SIAM J. Comput. 8(1979) pp.574-586.

[8] T. Kasai and A. Adachi, A characterization of time complexity by simple loop programs, J. Comput. System Sci., 20(1980) pp.1-17.

[9] T. J. Schaefer, Complexity of decision problems based on finite two-person perfect-information games, Proc. 8th Ann. ACM Symp. on Theory of Computing, 1976, pp.41-49.