CLASSES OF PEBBLE GAMES AND COMPLETE PROBLEMS

T. Kasai

Mathematical Sciences Department
IBM Thomas J. Watson Reasearch Center
Yorktown Heights, New York 10598

A. Adachi

Academic and Scientific Programs
IBM Japan Ltd.
Roppongi, Tokyo, 106 Japan

S. Iwata

Department of Information Science
Sagami Institute of Technology
Fujisawa, Kanagawa, 251 Japan

A "pebble game" is introduced and some restricted pebble games are considered. It is shown that in each of these games the problem to determine whether there is a winning strategy (two-person game) is harder than the solvability problem (one-person game). We also show that each of these problems is complete in well known complexity classes.

Key words: Exponential time; log-space; NP; pebble game; polynomial space; polynomial time; Turing machine; two-person game; winning strategy.

1. Introduction

A number of complete problems in various complexity classes are reported. Jones and Laaser [8] showed some familiar problems which are complete in deterministic polynomial time. A great number of familiar problems have been reported which are complete in NP (nondeterministic polynomial time) [1,3,9]. Even and Tarjan [6] 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 [10] derived some two-person games from NP complete problems which are complete in polynomial space.

We introduce a "pebble game" in which player(s) move pebbles according to the rules to put a pebble on a particular place. We show that when the game is played by two persons the problem to determine whether there is a winning strategy is complete in exponential time, and when played by one person, the problem to determine whether one can put a pebble on a particular place (called the solvability problem) is complete in polynomial space. Then we consider various classes of restricted pebble games and study their complexity classes. In particular, it has been shown that the problem of determining whether there is a winning strategy in a game played by two persons is harder in a sense than the solvability problem played by one person.

Our results are summarized in Table 1, where NLOGSPACE, P, NP, PS, EXP stand for nondeterministic log space, deterministic polynomial time, nondeterministic polynomial time, polynomial space, deterministic exponential time respectively.

2. Preliminaries

In this section, the basic objects with which we are concerned are reviewed. For additional details and background, see [1,2,7].

By Turing machine, we mean a machine with a finite-state control, a two-way read only input tape, and a single two-way read-write work tape; the machine halts whenever it enters the accepting state.

Let w be the input to a Turing machine, and $|w| = n$. DTIME(T(n)) is defined to be the class of languages accepted by deterministic Turing machines within T(n) time. NTIME(T(n)) is defined analogously for nondeterministic Turing machines. Similarly, DSPACE(S(n)) and NSPACE(S(n)) are defined to be the classes of languages accepted within S(n) space by deterministic and nondeterministic Turing machines, respectively. Now, let

$$\text{NLOGSPACE} = \bigcup_{i\geq 0} \text{NSPACE}(i\cdot \log n), \qquad (1)$$

Table 1. Our Results

	solvability problem (played by one person)	winning strategy problem (played by two persons)
pebble game of fixed rank	NLOGSPACE complete	P complete
acyclic pebble game	NP complete	PS complete
pebble game	PS complete	EXP complete

$$P = \bigcup_{i\geq 0} \mathrm{DTIME}(n^i), \quad (2)$$

$$NP = \bigcup_{i\geq 0} \mathrm{NTIME}(n^i), \quad (3)$$

$$PS = \bigcup_{i\geq 0} \mathrm{DSPACE}(n^i) = \bigcup_{i\geq 0} \mathrm{NSPACE}(n^i) \quad (4)$$

$$EXP = \bigcup_{i\geq 0} \mathrm{DTIME}(2^{n^i}). \quad (5)$$

Let Γ be a set of tape symbols. A function $f: \Gamma^* \to \Gamma^*$ is computable in log-space if and only if there is a deterministic Turing machine M additionally equipped with a one-way write-only output tape such that for any input w of M, M halts with f(w) on its output tape, having scanned no more than $\log(|w|)$ work tape symbols. Let $L, L' \subseteq \Gamma^*$. L is log-space reducible to L' if and only if there is a function f computable in log-space such that for any input w, $w \in L$ if and only if $f(w) \in L'$. For a class of languages C, a language L is called C complete if L is in C, and L' is log-space reducible to L for any language L' in C.

Definition [2]. A k-tape alternating Turing machine (ATM for short) is a 8-tuple $M = (Q,\Sigma,\Gamma,\delta,b,q_1,q_a,U)$ where:

(1) Q, q_1, q_a are the finite set of states, initial state, accepting state, respectively, $q_1, q_a \in Q$.

(2) Σ, Γ are the finite set of input symbols and the set of tape symbols respectively, with $\Sigma \subsetneq \Gamma$.

(3) b, in $\Gamma - \Sigma$, is the blank symbol.

(4) δ, the next move function, maps a subset of $Q\times\Sigma\times\Gamma^k$ to subsets of $Q\times\Gamma^k\times\{-1,0,+1\}^{k+1}$. An element of $\{-1,0,+1\}^{k+1}$ represents changes of head locations of the input tape and k work tapes.

(5) U is a set of universal states, $U \subseteq Q$.

(6) Q - U is a set of existential states.

The ATM has a read only input tape, with the reading head initialized to the first symbol of the input. A configuration of the ATM consists of the state, head positions and contents of the k+1 tapes. Each of k work tapes is initially blank. A move of the ATM consists of reading one symbol from each of k+1 tapes, and then writing one symbol on each work tape and moving the heads as allowed by δ, along with a state-change of the ATM. If C is a configuration of M, let $\mathrm{Next}_M(C)$ denote the set of possible configurations after one move of M. We say a configuration is existential (respectively universal, initial, accepting) if the state of the ATM in that configuration is an existential (respectively universal, initial, accepting) state.

Let C be a configuration of an ATM. A value v(C) of C is either true or false defined by followings:

(1) If C is an accepting configuration, v(C) is true.

(2) If C is an existential configuration but not an accepting configuration, and there is $C' \in \mathrm{Next}_M(C)$ such that v(C') is true, then v(C) is true.

(3) If C is a universal configuration but not an accepting configuration, and for every configuration $C' \in \mathrm{Next}_M(C)$, v(C') is true, then v(C) is true.

M accepts the input w if and only if $v(C_0)$ is true where C_0 is the initial configuration.

We say an ATM M is standard if (1) M has only one work tape with the head initialized to the first cell of the tape, (2) if a configuration C of M is existential (universal), then every configuration $C' \in \mathrm{Next}_M(C)$ is universal (existential), (3) the initial state is existential and the accepting state is universal, and (4) $\mathrm{Next}_M(C) = \emptyset$ if and only if C is an accepting configuration.

We let ATIME(T(n)) and ASPACE(S(n)) denote the classes of languages accepted by ATM's within time T(n) and within space S(n), respectively.

Lemma [Chandra and Stockmeyer,2]. Let $S(n) \geq \log n$ and $T(n) \geq n$. If $L \in \mathrm{ASPACE}(S(n))$, then L is accepted by a standard ATM within space S(n). If $L \in \mathrm{ATIME}(T(n))$, then L is accepted by a standard ATM within time $O(T^2(n))$.

Theorem 1 [Chandra and Stockmeyer,2].

$$EXP = \bigcup_{i\geq 0} \mathrm{ASPACE}(n^i), \quad (6)$$

$$PS = \bigcup_{i\geq 0} \mathrm{ATIME}(n^i), \quad (7)$$

$$P = \bigcup_{i\geq 0} \mathrm{ASPACE}(i\cdot\log n). \quad (8)$$

3. Pebble games

Definition. A pebble game is a quadruple $G = (X,R,S,t)$ where:

(1) X is a finite set of nodes; the number of nodes is called the order of G.

(2) $R = \{(x,y,z): x,y,z \in X,\ x \neq y,\ y \neq z,\ z \neq x\}$ is called a set of rules. For $A, B \subseteq X$, we write $A \vdash B$ if $(x,y,z) \in R$, $x,y \in A$, $z \notin A$, and $B = (A - \{x\}) \cup \{z\}$. We say the move $A \vdash B$ is made by the rule (x,y,z). A symbol $\overset{*}{\vdash}$ denotes the reflexive and transitive closure of $\vdash$.

(3) S is a subset of X; the number of nodes in S is called the rank of G.

(4) t is a node in X, called the terminal node.

A pebble game is said to be solvable if there exists $A \subseteq X$ such that $S \overset{*}{\vdash} A$ and $t \in A$.

A pebble game can be regarded as a game of moving pebbles on a graph. Initially pebbles are placed on all nodes of S. If $(x,y,z) \in R$ and pebbles are placed on x, y and not on z, then we can move a pebble from x to z. The game is solvable if we can place a pebble on the terminal node t by moving pebbles according to rules.

A pebble game played by two persons is a game between two players, P_1 and P_2, who alternately move pebbles on the pebble game, with P_1 playing first. The winner is the first player who can put a pebble on the terminal node, or who can make the other player unable to move.

By the term "one-person pebble game problem," we mean the problem to determine for a given pebble game G, whether G is solvable. By "two-person pebble game problem," we mean the problem to determine for a pebble game G, whether the first player has a winning strategy when G is played by two persons.

Theorem 2. Two-person pebble game problem is EXP complete.

Proof. It should be clear that this problem is in EXP.

Let $M = (Q,\Sigma,\Gamma,\delta,b,q_1,q_a,U)$ be a standard ATM such that only $p(n)$ cells are available on the work tape for some polynomial p in n, where n is the length of input w of M. Since $\mathrm{EXP} = \bigcup_{i \geq 0} \mathrm{ASPACE}(n^i)$ by Theorem 1, to prove the theorem, it suffices to construct a pebble game G' such that the construction is performed within log-space, and M accepts w if and only if the first player P_1 wins the game G'. Let $Q = \{q_1,\ldots,q_s\}$ and let $w = w_1w_2\ldots w_n$ ($w_i \in \Sigma$, $i = 1,2,\ldots,n$) be the input.

Let $G' = (X',R',S',t')$ where:

(1) $X' = \{[q,i,\ell]: q \in Q,\ 1 \leq i \leq n,\ 1 \leq \ell \leq p(n)\}$
$\cup \{[\ell,a]: 1 \leq \ell \leq p(n),\ a \in \Gamma\}$
$\cup \{[q,i,\ell,a,a']: q \in Q,\ 1 \leq i \leq n,\ 1 \leq \ell \leq p(n),\ a,a' \in \Gamma\}$
$\cup \{s_1,\ s_2,\ t'\}$.

(2) R' is defined as follows:

(2.1) for each $q \in Q$, $a \in \Gamma$, i ($1 \leq i \leq n$), ℓ ($1 \leq \ell \leq p(n)$),

(2.1.1) if $\delta(q,w_i,a)$ contains (q',a',d',d''), $a \neq a'$ then let
$([q,i,\ell],[\ell,a],[q,i,\ell,a,a'])$,
$([\ell,a],[q,i,\ell,a,a'],[\ell,a'])$,
$([q,i,\ell,a,a'],[\ell,a'],[q',i+d',\ell+d''])$
be elements of R',

(2.1.2) if $\delta(q,w_i,a)$ contains (q',a,d',d'') then let
$([q,i,\ell],[\ell,a],[q',i+d',\ell+d''])$
be an element of R'

(2.2) for each i ($1 \leq i \leq n$), ℓ ($1 \leq \ell \leq p(n)$), let $([q_a,i,\ell],s_1,s_2)$ be elements of R',

(2.3) let (s_2,s_1,t') be an element of R'.

(3) $S' = \{[q_1,1,1],\ s_1\}$
$\cup \{[\ell,b]: 1 \leq \ell \leq p(n)\}$.

A pebble being on a node of the form $[q,i,\ell]$ represents that the current state of the ATM M is q, and that the current head positions of the input tape and the work tape are on the i-th cell and on the ℓ-th cell respectively. A pebble being on a node $[\ell,a]$ represents that symbol a is written on ℓ-th cell of the work tape, and a pebble being on a node of the form $[q,i,\ell,a,a']$ means that M is to change symbol a to a' on the ℓ-th cell of the work tape and that M is in state q at the head position i on the input tape. The nodes s_1, s_2, t' are auxiliary nodes. Thus pebbles being on all nodes of S' imply the initial configuration of M. It is clear that the construction can be performed within log-space. We now show that M accepts w if and only if the player P_1 wins the game G'.

Suppose that M accepts w. Then the value of the initial configuration of M is true. Thus, for every true-valued existential configuration C_1, there is a true-valued universal configuration C_1' such that $C_1' \in \mathrm{Next}_M(C_1)$, and for every true-valued universal configuration C_2 except the accepting configurations, $C_2' \in \mathrm{Next}_M(C_2)$ implies that C_2' is a true-valued existential configuration. Each move of M corresponds either to three consecutive moves of G' induced by the rules in (2.1.1) or to one move induced by (2.1.2). In case a move of M changes a symbol on the work tape, player P_1 moves a pebble by the first rule of (2.1.1) corresponding to a move of M from a true-valued existential configuration to a true-valued universal configuration. Then player P_2 has to move a pebble by the second rule of (2.1.1). After that, P_1 moves a pebble by the third rule of (2.1.1). In case the move of M does not change the symbol on the work tape, P_1 moves a pebble by a rule of the form (2.1.2). Then it is P_2's turn. Whatever rules P_2 may choose, the moves correspond to moves of M from a true-valued universal configuration to a true-valued existential configuration.

Since the initial configuration is existential and the accepting configurations are universal, P_1 can first place a pebble on a node of the form $[q_a,i,\ell]$, then P_2 has to place a pebble on s_2, and P_1 can place a pebble on the terminal node t' of G'.

Therefore, P_1 has a winning strategy.

Suppose that M does not accept w. Then the value of the initial configuration of M is false. In this case, player P_2 can always move pebbles to nodes in G' which correspond to false-valued configurations of M. Thus, P_1 can not win.

Corollary. One person pebble game problem is PS complete.

Proof. It should be clear that the problem is in PS. Let M be a nondeterministic Turing machine which accepts an input w, $|w| = n$, within polynomial space. We construct a pebble game G

such that M accepts w if and only if G is solvable. Note that if all universal states of an ATM are treated as existential states, then the ATM behaves as a nondeterministic Turing machine. Hence we can treat the ATM in the proof of Theorem 2 as a nondeterministic Turing machine. Now let G be the pebble game constructed in the proof of Theorem 2; then it is clear that M accepts w if and only if G is solvable.

Definition. A pebble game $G = (X,R,S,t)$ is acyclic if the digraph (X,E) is acyclic where, $E = \{(x,z), (y,z): (x,y,z) \in R\}$.

Theorem 3. Two-person acyclic pebble game problem is PS complete.

Proof. The maximum number of moves made in an acyclic pebble game $G = (X,R,S,t)$ is less than $|X|\cdot|S|$, since each pebble can move at most $|X|$ times. Thus this problem is in PS.

Let M be a $p_0(n)$-time bounded standard ATM, where n is the length of an input w of M, and p_0 is a polynomial in n. Then we can construct a pebble game $G_1 = (X_1,R_1,S_1,t_1)$ as in the proof of Theorem 2 such that M accepts w if and only if the first player can win in G_1 within $p_1(n) = 3\cdot p_0(n)$ moves. Now we construct an acyclic pebble game $G_2 = (X_2,R_2,S_2,t_2)$ such that the first player wins in G_2 if and only if the first player wins in G_1 within $p_1(n)$ moves. Let $G_2 = (X_2,R_2,S_2,t_2)$ where:

$$X_2 = \{[x,i]: x \in X_1,\ 0 \leq i \leq p_1(n)\},$$

$$R_2 = \{([x,i],[y,j],[z,\max(i,j)+1]): (x,y,z) \in R_1,\ z \neq t_1,\ 0 \leq i,j < p_1(n)\}$$
$$\cup\{([x,i],[y,j],[z,p_1(n)]):(x,y,z) \in R_1,\ z = t_1,\ 0 \leq i,j < p_1(n)\},$$

$$S_2 = \{[x,0]: x \in S_1\},$$

$$t_2 = [t_1,p_1(n)].$$

It is obvious that the pebble game G_2 is acyclic. It is also obvious that the first player has a winning strategy in G_2 if and only if the first player has a winning strategy in G_1 within $p_1(n)$ moves. Thus M accepts w if and only if the first player has a winning strategy in G_2. Note that the construction of G_2 from M is performed within log-space. Since $PS = \bigcup_{i\geq 0} ATIME(n^i)$ by Theorem 1, the problem is PS complete.

Corollary. One-person acyclic pebble game problem is NP complete.

Proof. Since the maximum number of moves made in an acyclic pebble game $G = (X,R,S,t)$ is less than $|S|\cdot|X|$, the solvability problem is in NP. We can show that for any nondeterministic Turing machine M, there is an acyclic pebble game G_2 such that M accepts input w within polynomial time in $|w|$ if and only if G_2 is solvable, by the same construction method as in the proof of Theorem 3.

Definition. Let $G = (X,R,S,t)$ be a pebble game. G is called a pebble game of fixed rank if the number of nodes in S is fixed.

Theorem 4. Two-person pebble game problem of fixed rank is P complete.

Proof. It is clear that the problem is in P. Let $M = (Q,\Sigma,\Gamma,\delta,b,q_1,q_a,U)$ be a log n space bounded standard ATM, where n is the length of the input $w = w_1w_2\ldots w_n$ of M. Now let $G = (X,R,S,t)$ be a pebble game of rank 3 where:

(1) $X = \{[q,i,\ell]: q \in Q,\ 1 \leq i \leq n,\ 1 \leq \ell \leq \log n\}$
$\cup\{[\alpha]: |\alpha| = \log n,\ \alpha \in \Gamma^*\}$
$\cup\{[q,i,\ell,a,a']: q \in Q,\ 1 \leq i \leq n,\ 1 \leq \ell \leq \log n,\ a,a' \in \Gamma\}$
$\cup\{s_1, s_2, t\}$.

(2) R is defined as follows:

(2.1) for each $q \in Q$, i ($1 \leq i \leq N$), $a \in \Gamma$, ℓ ($1 \leq \ell \leq \log n$), $\beta,\gamma \in \Gamma^*$ such that $|\beta a\gamma| = \log n$, $|\beta| = \ell-1$,

(2.1.1) if $\delta(q,w_i,a)$ contains (q',a',d',d''), $a \neq a'$, then let
$([q,i,\ell],[\beta a\gamma],[q,i,\ell,a,a'])$,
$([\beta a\gamma],[q,i,\ell,a,a'],[\beta a'\gamma])$,
$([q,i,\ell,a,a'],[\beta a'\gamma],[q',i+d',\ell+d''])$
be elements of R,

(2.1.2) if $\delta(q,w_i,a)$ contains (q',a',d',d'') then let
$([q,i,\ell],[\beta a\gamma],[q',i+d',\ell+d''])$
be an element of R,

(2.2) for each i ($1 \leq i \leq n$), ℓ ($1 \leq \ell \leq \log n$), let
$([q_a,i,\ell],s_1, s_2)$
be elements of R,

(2.3) let (s_2,s_1,t) be an element of R.

(3) $S = \{[q,1,1],[bb\ldots b],s_1\}$.

It can be shown that M accepts w if and only if the first player wins the game G by a similar argument as in the proof of Theorem 2. Note that the construction of G is performed within log-space and that the rank of G is 3. Since $P = \bigcup_{i\geq 0} ASPACE(i\cdot\log n)$ by Theorem 1, the problem is P complete.

Corollary. One-person pebble game problem of fixed rank is NLOGSPACE complete.

Proof. Clearly this problem is in NLOGSPACE. We can construct a pebble game G of rank 3 as in the proof of Theorem 4 such that a log n space bounded nondeterministic Turing machine accepts input if and only if G is solvable. Note that the construction of G is performed within log-space, and an ATM would behave as a nondeterministic Turing machine if each universal state is treated as an existential state.

References

1. Aho, A. V., Hopcroft, J. E. and Ullman, J. D., The Design and Analysis of Computer Algorithms, (Addison-Wesley, Reading, Mass., 1974).

2. Chandra, A. K. and Stockmeyer, L. J., Alternation, Proc. 17th Ann. IEEE Symp. on Foundation of Computer Sciences, 98 (1976).

3. Cook, S. A., The complexity of theorem-proving procedures, Proc. 3rd Ann. ACM Symp. on Theory of Computing, 151,(1971).

4. Cook, S. A., An observation on time-storage tradeoff, J. Comput. Syst. Sci., 9, 213 (1974).

5. Cook, S. A. and Sethi R., Storage requirements for deterministic polynomial time recognizable languages, J. Comput. Syst. Sci., 13, 25 (1976).

6. Even, S. and Tarjan, R. E., A combinatorial

which is complete in polynomial space, J. ACM, 23, 710 (1976).

7. Hopcroft, J.E. and Ullman, J. D., Formal Languages and Their Relation to Automata, (Addison-Wesley, Readind, Mass., 1969).

8. Jones, N. D. and Laaser, W. T., Complete problems for deterministic polynomial time, Theoretical Comp. Sci., 3, 105 (1977).

9. Karp, R. M., Reducibility among combinatorial problems, in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., p. 85 (Plenum Press, New York, 1972).

10. Schaefer, T. J., Complexity of decision problems based on finite two-person perfect-information games, Proc. 8th Ann. ACM Symp. on Theory of Computing, 41 (1976).