AD-A114 334   HARVARD UNIV CAMBRIDGE MA   F/G 12/1
DECISION ALGORITHMS FOR MULTIPLAYER GAMES OF INCOMPLETE INFORMA--ETC(U)
DEC 81 J REIF, G PETERSON   N00014-80-C-0647
UNCLASSIFIED   TR-34-81   NL

1 OF 1
AD A
114334

END
DATE
FILMED
5-82
DTIC

2

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A114 334

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Decision Algorithms for Multiplayer Games of Incomplete Information

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER
TR-34-81

7. AUTHOR(s)
John Reif
Gary Peterson

8. CONTRACT OR GRANT NUMBER(s)
N00014-80-C-0647

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Harvard University
Cambridge, MA 02138

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

12. REPORT DATE
December, 1981

13. NUMBER OF PAGES
31

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
same as above

15. SECURITY CLASS. (of this report)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
unlimited

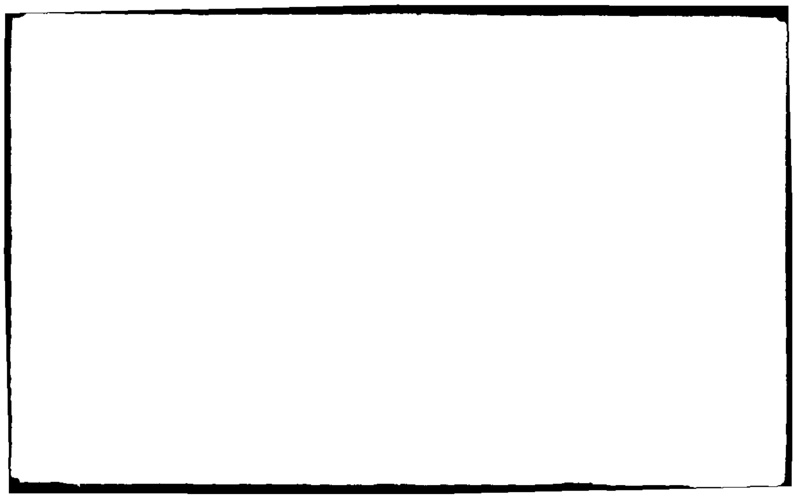
18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
games, algorithms, incomplete information, alternation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
see reverse side

20.

This paper provides algorithms for deciding the outcome for various classes of multiplayer games of incomplete information. The classes of games which our algorithms are applicable include games not previously known decidable; furthermore many of our algorithms have asymptotically optimal complexity. We apply our algorithms to provide alternative proofs of upper bounds, and new time-space tradeoffs on the complexity of multiplayer alternating machines of [Peterson and Reif, 79].



DECISION ALGORITHMS FOR MULTIPLAYER GAMES OF INCOMPLETE INFORMATION

John Reif

Gary Peterson

TR-34-81

Decision Algorithms for

Multiplayer Games of Incomplete Information

by

John Reif*

Aiken Computation Lab., Harvard University
Cambridge, Mass. 02138

Gary Peterson+

Computer Science Dept., University of Rochester
Rochester, New York 14627

December, 1981

*Research supported by National Science Foundation Grant NSF-MCS79-21024 and the Office of Naval Research Contract N00014-80-C-0647.

+Research supported in part by National Science Foundation Grant NSD-MCS79-02971.

DECISION ALGORITHMS FOR MULTIPLAYER GAMES OF INCOMPLETE INFORMATION

Abstract

This paper provides algorithms for deciding the outcome for various classes of multiplayer games of incomplete information. The classes of games which our algorithms are applicable include games not previously known decidable; furthermore many of our algorithms have asymptotically optimal complexity. We apply our algorithms to provide alternative proofs of upper bounds, and new time-space tradeoffs on the complexity of multiplayer alternating machines of [Peterson and Reif, 79].

Keywords

games, algorithms, incomplete information, alternation

1. INTRODUCTION

A multiplayer game can be specified (formal definitions are given in Section 2) by a set of positions, a relation defining possible next moves, and an assignment of the rights of players to view and modify certain components of positions. These rights do not change during the play of a game. A player may have incomplete information on the values of those components for which it has no rights to view, though it may be possible for a player to partially infer this information by observing the visible portion of positions in the previous play of the game.

A *strategy* for player i defines a single next move for player i from any given sequence of previous moves ending in a position for which it is player i's turn to move. Such strategies are called *pure*. (In contrast, a companion paper [Reif, 1981] considers *mixed* strategies where more than one moves next are allowed to be chosen with various probabilities. A strategy of player i must depend only on the visible components of previous positions for which player i made moves.

Players in a multiplayer game are partitioned into two teams, τ_0 and τ_1. Section 2 defines the *win outcome problem* for a game: Do the players of team τ_1 have a strategy yielding only finite winning plays? We also consider the *non-loss outcome problem*: does team τ_1 have a strategy yielding possibly infinite plays with no losses? Furthermore, we consider the *Markov (m(n)) outcome problem*: Given an initial position of length n, does team τ_1 have a winning strategy dependent only on the previous m(n) positions of any play? The Markov(1) outcome problem was previously considered in [Peterson and Reif, 79]. The purpose of this paper is to provide decision algorithms for these outcome problems.

Section 3 provides decision algorithms for outcome problems of certain games with a *space bound* $S(n) \geqslant \log n$ (i.e., $S(n)$ bounds the space required to compute possible next moves from a position). [Chandra, Kozen, and Stockmeyer, 78] give an algorithm for deciding the acceptance of a space bounded alternating machine which we adopt to decide in time exponential in $S(n)$ the win outcome of any $S(n)$ space bounded game of *perfect information.* [Reif, 79] gives an algorithm with time double exponential in $S(n)$ for the win outcome of any $S(n)$ space bounded *two player* game of incomplete information. Both of these algorithms were known to be asymptotically optimal. Unfortunately, there is a *three player* game with constant space bound but *indecidable* win outcome problem [Peterson and Reif, 79]. Thus we must consider a restricted class of space bounded games, if we wish decision algorithms.

A game is *hierarchical* if players of T_1 can be linearly ordered by their rights. We give a powerset construction for eliminating incomplete information in a hierarchical game with a finite space bound. This powerset construction was first used for two player games in [Reif, 79]. Hierarchical games are in a sense the largest class of games for which we can apply our powerset construction for eliminating incomplete information. In particular, the undecidability result of [Peterson and Reif, 79] for 3 player games of incomplete information can be extended to show that for any game G which is not hierarchical, there is a game $\hat{G}$ which has undecidable win and nonloss outcome problems; where G and $\hat{G}$ have the same players and each player has the same rights to view components of positions in $\hat{G}$ as in G.

Let a *clique* be a maximal set of players of a game with exactly the same rights to view components of positions. Let $\chi(G)$ be the number of cliques in team τ_1 for which the players do not have all the rights of team τ_0. Our decision algorithm for the win and nonloss outcome of a $S(n)$ space bounded hierarchical game requires deterministic time of $\chi(G)+1$ repeated exponentations on $S(n)$. By lower bound results of [Peterson and Reif, 79] our algorithms has an asymptotically optimal time bound.

A *blindfold game* is a hierarchical game where no component of a position visible to a player of team τ_1 is ever modified by a player of team τ_0. For a blindfold game with space bound $S(n)$, our decision algorithm requires deterministic space of $\chi(G)$ repeated exponentiations on $S(n)$, which again is asymptotically optimal by the lower bound result of [Peterson and Reif, 79].

Also, in Section 3, we give an algorithm for the Markov $(m(n))$ outcome problem of any $S(n)$ space bounded game; this extends a previous result of [Peterson and Reif, 79] for the case $m=1$.

The algorithm results of Section 3 imply Corollaries giving time bounds for plays induced by a winning strategy in any hierarchical game and for a Markov $m(n)$ winning strategy of any game.

Section 4 gives an algorithm for the outcome problems of any hierarchical game with both a space bound and an alternation bound (i.e., a bound on the number of times a sequence of moves of the team τ_1 alternates with moves of team τ_0).

Section 5 shows that for games with both a *time bound* $T(n) \geq n$ ($T(n)$ is the maximum number of moves per winning play from an initial position of length n) and a *branching bound* b (b upper bounds is the maximum number of possible positions divided by a single move from any position) the outcome problems can be solved in deterministic space O(T(n)b).

The Appendix defines the multiplayer alternating machines of [Peterson and Reif, 79] to which we apply our decision algorithms.

2. MULTIPLAYER GAMES OF INCOMPLETE INFORMATION

2.1 Game Definitions

A $k+1$ *player game* is a triple $G = (\mathrm{POS}, \vdash, \mathrm{VISIBLE}, \tau_1)$ where:

(i) POS is the set of *positions* with

$$\mathrm{POS} = \{0,\ldots,k\} \times R_1 \times \cdots \times R_r$$

where $R_1,\ldots,R_r$ are sets of strings over a finite alphabet.

(ii) $\vdash \subseteq \mathrm{POS} \times \mathrm{POS}$ is the *next move relation*, satisfying axioms A1, A2 given below.

(iii) VISIBLE is a mapping from $\{0,\ldots,k\}$ to (not necessarily proper) subsets of $\{1,\ldots,r\}$.

(iv) *Team* τ_1 is a subset of $\{0,1,\ldots,k\}$. The opposing team is $\tau_0 = \{0,1,\ldots,k\} - \tau_1$.

The domain of VISIBLE are the $k+1$ *players* named $0,\ldots,k$. Let $p = (i,p[1],\ldots,p[n])$ be a position in POS. Then $\mathrm{next}(p) = i$ is *the player whose turn it is to move next*. For each $j = 1,\ldots,r$, $p[j]$ is an element of R_j and is called the *j-th component of* p. We say *player* i *has the rights to view* $p[j]$ if $j \in \mathrm{VISIBLE}(i)$. For each $i = 0,\ldots,k$, let $\mathrm{vis}_i(p) = (v,b)$ where v is the list (in order of occurrence) of components of position p for which player i has rights, and b is a boolean which is 1 just if it is player i's turn to move at position p. Let $\mathrm{invis}_i(p)$ be the list of components of position p for which player i has no rights. Let $W \subseteq \mathrm{POS}$ be the set of positions from which there is no next move by any player. Let $\mathrm{POS}_i = \{p \in \mathrm{POS} \mid \mathrm{next}(p) = i\}$.

For *each* player $i \in \{0,\ldots,k\}$ we assume:

A1 If $p \in POS_i$ and $p \vdash p'$ then $invis_i(p) = invis_i(p')$.

A2 If $p,q \in POS_i \vdash W$ and $vis_i(p) = vis_i(q)$ then

$$\{vis_i(p') | p \vdash p'\} = \{vis_i(q') | q \vdash q'\}.$$

Axiom A1 states that no move by player i can modify a component of a position for which player i has no rights to view. Axiom A2 insures that if a pair of nontermination positions p,q are indistinguishable to player i, then the sets of possible next moves from p and q must be indistinguishable to player i.

2.2 Plays

For any finite string π of positions, let $last(\pi)$ be the last position of π. Fix an initial position $p_I \in POS$. A *play* is a (possibly infinite) string $\pi = p_0 p_1 \ldots$ of positions such that $p_0 = p_I$ is the initial position, $p_0 \vdash p_1$, $p_1 \vdash p_2, \ldots$ and $last(\pi) \in W$ whenever π is finite. A play p is a *loss* for player i if p is finite and $last(p) \in POS_i \wedge W$; thus it is player i's turn to move but there is no next move. A *play prefix* π is a finite null initial substring of a play; thus π indicates a finite sequence of moves from the initial position p_I. We now inductively define a sequence $vis_i(\pi)$ representing that portion of the play prefix π which is visible to player i. Let $p = last(\pi)$. If π is of length 1 then $vis_i(\pi) = vis_i(p)$. Let p' be a position such that $p \vdash p'$. If both $p \notin POS_i$ and $vis_i(p) = vis_i(p')$ then let $vis_i(\pi p') = vis_i(\pi)$, and otherwise let $vis_i(\pi) = vis_i(\pi) vis_i(p)$. Thus player i can detect the occurrence of a move only if it is a move by player i or if player i has rights to a component of the position which is modified by the move.

2.3 The Game Tree

The *game tree* GT is the set of play prefixes. The *root* of GT is the initial position p_I. Each play prefix π is considered a *node* of T. The *children* of π are the play prefixes which are of length one less than π and which are a prefix of π. Thus $\text{last}(\pi) \vdash \text{last}(\pi')$ if π' is a child of π. Let GT_i be the set of play prefixes π such that $\text{last}(\pi) \in POS_i - W$ (these are those play prefixes ending at a position for which it is player i's turn to move).

2.4 Pure Strategies

A partial function $\sigma: GT_i \to GT$ is a (pure) *strategy for player* i if

(1) for any $\pi \in GT_i, \sigma(\pi)$ is a child of π, and

(2) if $\pi, \pi' \in GT_i$ and $\text{vis}_i(\pi) = \text{vis}_i(\pi')$ then $\text{vis}_i(\sigma(\pi)) = \text{vis}_i(\sigma(\pi'))$.

The first restriction requires the strategy to make only legal moves. The second restriction insures that the strategy depends only on that portion of the play prefix which is visible to player i.

Recall that the players $0,\ldots,k$ are partitioned into disjoint sets τ_0, τ_1 which we call *teams*. A play p is a *win for team* τ_1 if it is a loss for some player on team B. A play p is *nonloss for team* τ_1 if it is not a loss for any player on team τ_1. Let a *team strategy for* τ_1 be a mapping $\sigma: (\cup_{i \in \tau_1} GT_i) \to GT$ such that for each player $i \in \tau_1$, the restriction of σ to domain GT_i is a strategy for player i. A play π is a *play by team strategy* σ if whenever π' is a prefix of π and π' is in the domain of σ, then $\sigma(\pi')$ is a prefix of π. σ is a *winning (nonloss,* respectively) *strategy for team* τ_1 if every play by strategy σ is a win (nonloss, respectively) for team τ_1. Note that all plays by a winning strategy are finite, but a play by a nonloosing strategy may be infinite.

To define Markov strategies, we must introduce some special notation. Given a play prefix π, and an integer $m \geq 0$, let $\text{last}_m(\pi)$ be the last m positions of π if $m < |\pi|$, otherwise let $\text{last}_m(\pi) = \pi$. A strategy σ for player i is *Markov(m)* if $\sigma(\pi) = \sigma(\pi')$ for all play prefixes $\pi, \pi' \in GT_i$ such that $\text{vis}_i(\text{last}_m(\pi)) = \text{vis}_i(\text{last}_m(\pi'))$. Hence player i plays by a Markov(m) strategy if its moves depend only on the visible portions of the last m positions of any play. A strategy for team τ_1 is Markov(m) if the strategy for each player $i \in \tau_1$ is Markov(m).

2.5 The Outcome Problem for a Game

Let $G = (POS, \vdash, VISIBLE, \tau_1)$ be a game. We assume the next move relation $\vdash$ is represented as a *next move transducer* which is a Turing machine transducer with an input tape and an one-way write only output tape, and possibly some work tapes. The input alphabet Σ for this transducer contains the symbols appearing in positions of POS. Given any $p \in \Sigma^*$, the next move transducer outputs the set of moves $\{(p,p') | p \vdash p'\}$ if $p \in POS$, and otherwise outputs a distinguished symbol $\$ \notin \Sigma$ if $p \notin POS$. The game G is thus finitely represented by pair containing the next move transducer and VISIBLE.

The *win outcome (Markov(m(n)) outcome, nonloss outcome,* respectively) *problem* for game G (finitely represented as above) is:

> Given an initial position $p_I \in POS$, is there a winning (winning Markov(m(n)), nonloss, respectively) team strategy for τ_1?

2.6 Complexity of Games

A move $p \vdash p'$ is an *alternation* if $p \in POS_i$ and $p' \in POS_j$ where $(i \in \tau_0$ iff $j \in \tau_1)$. Thus, $p \vdash p'$ is an *alternation* if the players to move at p, p' are on different teams.

Game G has *time bound* $T(n)$ (*alternation bound* $A(n)$, *space bound* $S(n)$, respectively) if for each position $p_I \in POS$ of length n for which team τ_1 has a winning strategy σ, there is some such σ where π contains at most $T(n)$ moves. (π contains at most $A(n)$ alternations, the next move transducer of G requires $S(n)$ work tape cells, respectively)for each play π induced by σ from initial position p_I.

The complexity bound definitions given above are relevant only to the win outcome problem for a game. When considering the Markov(m(n)) (nonloss, respectively) outcome problem, we bound space, time, and alternations only for plays of same winning Markov(m(n)) (nonloss, respectively) strategy.

2.7 Obvious Properties of the Outcomes of a Game

Since the players of team τ_0 are allowed any legal moves in response to a team strategy of τ_1, we have:

PROPOSITION 2.1. *Let* G_1 *be the* $|\tau_1|+1$ *player game derived from* G *by substituting for all the players of team* τ_0 *a single omniscient player of* τ_0 *which has rights to all components of any position, and is allowed to move the same as the players of* τ_0 *previously moved. Then the win, nonloss and Markov(m(n)) outcome of* G_1 *is the same as* G, *for any* $m(n) \geq 0$.

Recall from the Introduction that a *clique* is a maximal set φ of players such that VISIBLE(i) = VISIBLE(j) for all players $i,j \in \varphi$. It is also easy to show:

PROPOSITION 2.2. *Let* G_2 *be the game derived from* G *by substituting for each clique of* τ_1 *a single player with the same rights as the clique and allowed to move the same as the players of the clique. Then* G_2 *has the same win, nonloss, and Markov(m(n)) outcome as* G, *for any* $m(n) \geq 0$.

2.8 Restricted Classes of Games

Fix a game G with teams τ_0, τ_1.

A player i is *deterministic* if for each position $p \in POS_i$, there is at most p' such that $p \vdash p'$, G is *nondeterministic* if each player of τ_0 is deterministic. G is *deterministic* if all players are deterministic.

We now define some classes of games with restrictions to VISIBLE.

A player $i \in \tau_1$ *has perfect information of* τ_0 if VISIBLE(i) $\supseteq$ VISIBLE(j), for each $j \in \tau_0$. G is *perfect information* if all players of τ_1 have perfect information of τ_0.

G is *hierarchical* if the players of τ_1 can be ordered $\{i_1,\ldots,i_h\}$ so that VISIBLE(i_x) $\supseteq$ VISIBLE(i_{x+1}) for $1 \leqslant x < h$. Hierarchical games occur naturally in a variety of situations such as the multiprocessing games of incomplete information of [Reif and Peterson, 80] where a hierarchy of processes are generated by a sequence of fork operations.

G is *blindfold* if G is hiearchical and no player of τ_1 can view any portion of a position which is ever modified by a player of τ_0. Note that if G is blindfold then without modifying the outcome problems for G, we can by Proposition 2.1 disallow the players of τ_0 rights to view any portions of positions which are viewed by any player of τ_1, so we can assume

$$\text{VISIBLE}(i) \wedge \text{VISIBLE}(j) = \emptyset$$

for all $i \in \tau_0$, $j \in \tau_1$. The games of MASTERMIND, BATTLESHIP and BLINDFOLD-PEEK [Reif, 79] are examples of blindfold games.

3. DECISION ALGORITHMS FOR SPACE BOUNDED GAMES

This section provides algorithms for deciding the outcome of a game $G = (POS, \vdash, VISIBLE, \tau_1)$ with space bound $S(n) \geqslant \log n$ given an initial position $p_I \in POS$ and team τ_1. We will let $POS(p_I)$ be the set of positions in POS reachable from p_I by some sequence of moves of $\vdash$, and with space bound $S(n)$.

Since the positions of POS are strings over a finite alphabet:

PROPOSITION 3.1. *If* $S(n) \geqslant \log n$, *there is a constant* $c > 0$ *such that* $|POS(p_I)| \leqslant c^{S(n)}$.

It is also straightforward to show:

PROPOSITION 3.2. *The win and Markov(m(n)) outcome of any deterministic (nondeterministic, respectively) game with space bound* $S(n) \geqslant \log n$ *can be decided in deterministic (nondeterministic, respectively) space* $S(n)$. *Furthermore, the nonloss outcome of any deterministic (nondeterministic, respectively) game with space bound* $S(n) \geqslant \log n$ *can be decided in deterministic (co-nondeterministic, respectively) space* $S(n)$.

3.1 Deciding the Markov(m(n)) Outcome of a Space Bounded Game

LEMMA 3.1. *If a game* G *has a space bound* $S(n) \geqslant \log n$, *with respect to the Markov(m(n)) outcome problem, then* G *has a time bound* $2^{O(m(n)S(n))}$ *with respect to the Markov(m(n)) outcome problem.*

To prove this Lemma, we observe that there is a constant $c > 0$ such that if a Markov(m(n)) strategy σ induces a play π of length $> c^{m(n)S(n)}$, then π contains a repeated play subsequence π_1 containing $m(n) - 1$ moves. Thus π is of the form $\pi = \pi_0 \pi_1 \pi_2 \pi_1 \pi_3$ where π_0 and π_2 are either play subsequence or empty strings, and π_3 is a play subsequence. If it is a player of team τ_1 turn to move at least(π_1), then π_3 has an infinite number of occurrences of π_1 so π is infinite. Otherwise if it is a player of team τ_0 turn to move at last(π_1), then $\pi' = \pi_0 \pi_1 \pi_2 \pi_1 \pi_2 \cdots$ is an infinite play induced from σ. In either case σ is not a winning strategy. Hence a winning Markov(m(n)) strategy induces only plays of length $\leqslant c^{m(n)S(n)}$. □

We wish to determine the Markov(m(n)) outcome of a game G with space bound $S(n) \geqslant \log n$. By Lemma 3.1, a winning Markov(m(n)) strategy σ need only be defined for plays containing $\leqslant c^{m(n)S(n)}$ positions, for some constant $c > 0$. Therefore we can verify a Markov(m) strategy is winning within deterministic $2^{O(m(n)S(n))}$ space. Furthermore, σ can be represented by a function $\lambda : POS(p_I)^{m(n)} \to POS(p_I)$ such that $\sigma(\pi) = \pi p$ iff $\lambda(last_{m(n)}(\pi)) = p$ for all π in the domain of σ.
By Proposition 3.1 there can be at most $c^{(m(n)+1)S(n)}$ such functions λ. Hence we have shown:

THEOREM 3.1. *The Markov(m(n)) outcome of any game* G *with space bound* $S(n) \geqslant \log n$ *can be decided in deterministic space* $2^{O(m(n)S(n))}$.

By Lemma 3.1 and Theorem 3.1 we have:

COROLLARY 3.1. *For any* $S(n) \geqslant \log n$, *and* $k \geqslant 0$

$$MA_k\text{-SPACE}(S(n)) = MA_k\text{-SPACE,TIME}(S(n), 2^{O(S(n))})$$
$$\subseteq D\text{-SPACE}(S(n)).$$

3.2 Deciding a Space Bounded Game of Perfect Information

Let G be a game of perfect information. Fix an initial position p_I of length n.

PROPOSITION 3.3. *If team* τ_1 *has a winning strategy* σ *in game* G *of perfect information, then team* τ_1 *has a winning Markov(1) strategy.*

To prove this, suppose $\sigma(\pi) \neq \sigma(\pi')$ for play prefixes $\pi, \pi' \in GT_i$ $j \in \tau_1$ such that $vis_i(last(\pi)) = vis_i(last(\pi'))$. Then the strategy σ' is also winning if it is identical to winning strategy σ except that $\sigma'(\pi') = \sigma(\pi)$. By repeating this process, we can derive a winning Markov(1) strategy for team τ_1. □

By applying Proposition 2.3 to Lemma 3.3, we have:

LEMMA 3.2. *If a game* G *of perfect information has space bound* $S(n) \geqslant \log n$, *then* G *has time bound* $2^{O(S(n))}$. □

Now assume G has constructible space bound $S(n) \geqslant \log n$ (if $S(n)$ is not constructible, we try $S(n) = 0,1,\ldots$). Let W be the positions in $POS(p_I)$ with no next move. Given a mapping ℓ: $POS(p_I) \rightarrow \{\underline{true}, \underline{false}\}$, let $f(\ell)$ be the mapping such that for each $p \in POS(p_I)$ for which it is a player of team τ_1 turn to move, let

$$
\begin{array}{llllll}
f(\ell)(p) & = \underline{true} & \text{if} & i=0 & \text{and} & p \in W \\
 & = \underline{false} & \text{if} & i=1 & \text{and} & p \in W \\
 & = \bigwedge\limits_{p \vdash p'} \ell(p') & \text{if} & i=0 & \text{and} & p \notin W \\
 & = \bigvee\limits_{p \vdash p'} \ell(p') & \text{if} & i=1 & \text{and} & p \notin W
\end{array}
$$

Let $a = |POS(p_I)|$. By Proposition 3.1 $a = 2^{O(S(n))}$ $f(\ell)$ can be computed from ℓ in deterministic time a^2. Let $\ell_0^* = f(\ell_0^*)$ be the mapping derived by repeatedly applying f to initial mapping $\ell_0 : POS(p_I) \rightarrow \{\underline{false}\}$, until a fixea point is reached. (This computation of ℓ_0^* is similar to a procedure used by [Chandra, Kozen, and Stockmeyer, 78] to decide acceptance of space bounded alternating machines.) Note that the sequence of mappings ℓ_0, $f(\ell_0)$, $f(f(\ell_0))$,... are monotone in the sense that if $f^x(\ell_0)(p) = \underline{true}$ then $f^{x+1}(\ell_0)(p) = \underline{true}$, for any $x \geqslant 0$ and $p \in POS(p_I)$. Hence the computation converges by at most a repeated applications of f to ℓ_0 so $\ell_0^* = f^a(\ell_0)$. The total deterministic time to compute ℓ_0^* is thus $O(a^3) = 2^{O(S(n))}$.

We now show by induction on length of plays that $\ell_0(p_I) = \underline{true}$ iff team τ_1 have a winning strategy. Suppose, for some $x \geqslant 0$, and for any $p \in POS(p_I)$ that $f^x(\ell_0)(p) = \underline{true}$ iff team τ_1 have a winning strategy from p where all plays have length $< x$. (This trivially holds for $x = 0$). If $p \in W$ then by definition $f^{x+1}(\ell_0)(p) = \underline{true}$ iff it is a player of τ_0 turn to move at p iff team τ_1 have a trivial winning strategy form p. If it is a player of τ_0 turn to move at $p \notin W$ than by definition $f^{x+1}(\ell_0)(p) = \underline{true}$ iff $f^x(\ell_0)(p') = \underline{true}$ for each $p' \in POS(p_I)$ such that $p \vdash p'$ iff (by the induction hypothesis) team τ_1 has a winning strategy of length $< x$ from each $p' \in POS(p_I)$ such that $p \vdash p'$. If it is a player of τ_1 turn to move at $p \notin W$ then by definition $f^{x+1}(\ell_0)(p) = \underline{true}$ iff $f^x(\ell_0)(p') = \underline{true}$ for some $p' \in POS(p_I)$ such that $p \vdash p'$ iff (by the induction hypothesis) team τ_1 has a winning strategy of length $< x$ from some $p' \in POS(p_I)$ such that $p \vdash p'$. In either case, $f^{x+1}(\ell_0)(p) = \underline{true}$ iff team τ_1 has a winning strategy of length $< x + 1$ from p.

To determine the nonloss outcome of G, let ℓ_1^* be the mapping derived by repeatedly applying h a total of $a = 2^{O(S(n))}$ times to the mapping $\ell_1 : \mathrm{POS}(p_I) \to \{\underline{\mathrm{true}}\}$. It is again easy to show that $\ell_1^*(p_I) = \underline{\mathrm{true}}$ iff team τ_1 has a nonloss strategy in game G.

Furthermore both mappings ℓ_0^* and ℓ_1^* are computed in deterministic time $2^{O(S(n))}$. Hence we have shown:

THEOREM 3.2. *The win outcome and nonloss outcome of any game of perfect information with space bound* $S(n) \geq \log n$ *can be decided in deterministic time* $2^{O(S(n))}$.

By Lemma 3.2 and Theorem 3.2:

COROLLARY 3.7. (due to [Chandra, Kozen, and Stockmeyer, 78]). *For any* $S(n) \geq \log n$,

$$\begin{aligned} \text{A-SPACE}(S(n)) &= \text{A-SPACE,TIME}(S(n), 2^{O(S(n))}) \\ &\subseteq \text{D-TIME}(2^{O(S(n))}). \end{aligned}$$

3.3 Elimination of Incomplete Information from a Hierarchical Game

Let $G = (\mathrm{POS}, \vdash, \mathrm{VISIBLE}, \tau_1)$ be a hierarchical game. Fix an initial position $p_I \in \mathrm{POS}$ and teams τ_0, τ_1. We assume the set of positions reachable from p_I is finite. We give a method for transforming G to a game of perfect information. We will accomplish this in stages. In each stage we effectively eliminate from the game the incomplete information associated with a clique φ_1 of players in team τ_1 (though these players will remain in the game), so that in the resulting derived game G^+ the players of φ_1 have perfect information.

For any set of players of G, let

$$\text{VISIBLE}(\varphi) \;=\; \bigcup_{i\in\varphi} \text{VISIBLE}(i) \;.$$

Let $\varphi_0 \subseteq \tau_1$ be the set of players of τ_1 with perfect information of τ_0; so VISIBLE$(\varphi_0) \supseteq$ VISIBLE(i) for each player $i \in \tau_0$. Let $\chi(G)$ be the number of cliques of $\tau_1 - \varphi_0$. (This will be an important parameter of a game's complexity.) Let φ_1 be the clique of $\tau_1 - \varphi_0$ such that VISIBLE$(\varphi_1) \supseteq$ VISIBLE(i) for all players $i \in \tau_1 - \varphi_0$. We shall eliminate the incomplete information of clique φ_1.

We now derive from G a new hierarchical game $G^+ = (\text{POS}^+, \vdash^+,$ VISIBLE$^+, \tau_1)$. G^+ has the same players as G and the same teams τ_0, τ_1.

Let RIGHTS$^+(i) = \{1,\ldots,r+1\}$ for each player $i \in \varphi_1 \cup \varphi_0$ and let RIGHTS$^{+}(i)$ = RIGHTS(i) for all other players. By this assignment of rights, the players of φ_0 and φ_1 have perfect information.

For each position $p = (i, p[1], \ldots, p[r])$, in POS, and every play prefix π of G with $p = \text{last}(\pi)$, we derive a new position $P(\pi) = (i, p[1], \ldots, p[r], p[r+1])$ of POS$^+$ where the r+1 component $p[r+1]$ is $\{\text{last}(\pi') \mid \pi'$ is a play prefix with $\text{vis}_{\varphi_1}(\pi') = \text{vis}_{\varphi_1}(\pi)\}$ and where $\text{vis}_{\varphi_1}(\pi) = \text{vis}_j(\pi)$ for any $j \in \tau_1$. Intuitively, the r+1 component $p[r+1]$ of $P(\pi)$ is the set of possible positions visible after π, from the players of φ_1 point of view (where the players of φ_1 are allowed only to view $\text{vis}_{\varphi_1}(\pi)$). Note that $P(\pi) = P(\pi')$ iff $\text{vis}_{\varphi_1}(\pi) = \text{vis}_{\varphi_1}(\pi')$. We allow no next move from position $P \in \text{POS}^+$ if the r+1 component of P contains a position of POS with no next move. Hence the team τ_1 wins at P if $P = P(\pi)$ for some play π winning for τ_1. Otherwise, let $P \vdash^+ P'$ be a move of G^+ if $P = P(\pi)$ and $P' = P(\pi')$ for some child π' of π. By this definition, a move of G^+ from P simulates all possible next moves of G from any position $\text{last}(\pi')$ where $P = P(\pi)$. Fix $P_I = P(p_I)$ to be the initial position of G^+.

Note that G^+ has space bound $2^{O(S(n))}$; thus, we have an exponential blow-up in space complexity.

LEMMA 3.3. *Team* τ_1 *has a winning (nonloss, respectively) strategy in* G *from initial position* p_I *iff team* τ_1 *has a winning (nonloss, respectively) strategy* G^+ *from initial position* P_I.

Proof. We will establish a 1-1 correspondence between winning strategies in G and G^+. (A similar 1-1 correspondence can also be shown between nonloss strategies in G and G^+.)

Let σ be a winning strategy for team τ_1 in G. For each play prefix π of G^+, for which one of the players of τ_1 turn to move, let $\sigma^+(\pi) = \pi P(\sigma(\pi))$ where π is the play prefix of G from which π was derived. Suppose σ^+ induces a play π of G^+ which is not winning for team τ_1 in G^+. Then there is a play π of G induced from σ where $\text{last}(\pi)$ is contained in the $r+1$ component of $P(\pi)$ and some such π is not winning for team τ_1. But this contradicts our assumption that σ is a winning strategy. Thus σ^+ is a winning strategy.

Let σ^+ be a winning strategy for team τ_1 in G^+. For each play prefix π of G where it is one of the player of team τ_1 turn to move, let $\sigma(\pi)$ be the child of π such that $\sigma^+(\pi) = \pi P(\sigma(\pi))$ for any play prefix π of G^+ whose positions are derived from the positions of π. σ can easily be shown to be a winning strategy for G. □

Let $g(x,0) = x$ and $g(x,y+1) = 2^{g(x,y)}$ for $y \geq 0$. Thus $g(x,y)$ is derived by applying y repeated exponentations of 2 to x.

THEOREM 3.3. *The win outcome and the nonloss outcome of any hiearchical game* G *with space bound* $S(n) \geqslant \log n$ *can be decided in deterministic time* $g(O(S(n)), \chi(G)+1)$.

Proof. Fix an initial position p_I of G with length n. Lemma 3.3 can be applied $\chi(G)$ times to yield a game G* of perfect information with space bound $g(O(S(n)),\chi(G))$, and with initial position P_I^* such that team τ_1 has a winning strategy in G* from P_I^* iff team τ_1 has a winning strategy in G from P_I^*. By Theorem 3.1, G* can be decided in deterministic time $g(O(S(n)),\chi(G)+1) = 2^{g(O(S(n)),\chi(G))}$. □

By Proposition 2.3, Lemma 3.1 and Theorem 3.3, it is interesting to observe that

COROLLARY 3.3. *If hierarchical game* G *has space bound* $S(n) \geqslant \log n$, *then it suffices that a winning strategy be Markov(m(n)), where* $m(n) = g(O(S(n)),\chi(G))$. *Furthermore,* G *has time bound* $2^{O(m(n))}$.

By Theorem 3.3 and the above Corolary 3.3,

COROLLARY 3.4. *For any* $S(n) \geqslant \log n$ *and* $k \geqslant 0$

$$\begin{aligned} PA_k\text{-SPACE}(S(n)) &= PA_k\text{-SPACE,TIME}(S(n),g(O(S(n)),k)) \\ &\subseteq \text{D-TIME}(g(O(S(n)),k+1)) \ . \end{aligned}$$

THEOREM 3.4. *To win outcome (nonloss outcome, respectively) of any blindfold game* G *with space bound* $S(n) \geqslant \log n$ *can be decided in nondeterministic (co-nondeterministic, respectively) space* $g(O(S(n)),\chi(G))$.

Proof. Fix an initial position p_I of G of length n. As in Theorem 3.3, Lemma 3.3 can be applied $\chi(G)$ times to yield a game G^* of perfect information with space bound $g(O(S(n)),\chi(G))$. But since G is blindfold, any winning or nonloss strategy of team τ_1 can be made oblivious to moves of team τ_0. Thus G^* can be transformed to a nondeterministic game G^N by allowing team τ_1 to nondeterministically choose a strategy τ_1 (this can be done nondeterministically in space $g(O(S(n)),\chi(G)))$ and then allowing team τ_0 to iteratively choose each possible strategy σ_0 of team τ_0 (this can be done deterministically in space $g(O(S(n)),\chi(G)))$; during the simulated play of the game G^* team τ_0 must move by strategy σ_0 and the players of τ_1 must move by strategy σ_1. The win outcome (nonloss outcome, respectively) of G^N can be decided in nondeterministic (co-nondeterministic, respectively) space $g(O(S(n)),\chi(G))$ by Proposition 3.2.

By Theorem 3.4 and Corollary 3.3,

COROLLARY 3.5. *For any* $S(n) \geqslant \log n$ *and* $k \geqslant 0$,

$$\begin{aligned} BA_k\text{-SPACE}(S(n)) &= BA_k\text{-SPACE,TIME}(S(n),g(O(S(n)),k)) \\ &\subseteq N\text{-SPACE}(g(O(S(n)),k)). \end{aligned}$$

4. DECISION ALGORITHMS FOR GAMES WITH BOTH ALTERNATION AND SPACE BOUNDS

THEOREM 4.1. *If a game* G *of perfect information has space bound* $S(n) \geq \log n$ *and alternation bound* $A(n)$*, then the win outcome and nonloss outcome of* A *can be decided in deterministic space* $(A(n) + S(n))S(n)$.

Proof. (We utilize here an algorithm attributed to A. Borodin by [Chandra, Kozen, and Stockmeyer, 78], for deciding acceptance of alternating machines with space and alternation bounds.) Let $p_I \in POS$ be an initial position of length n. We assume $S(n)$ is constructible (otherwise try $S(n) = 0,1,\ldots$). Let $POS(p_I) \subseteq POS$ be those positions reachable from p_I with space $\leq S(n)$.

Given any two positions $p, p' \in POS(p_I)$ let $PATH(p,p')$ ($APATH(p,p')$, respectively) be the predicate that holds iff there is a sequence of moves from p to p', with no alternations (except the last move is an alternation, respectively) and where all positions visited are in $POS(p_I)$. Also, let $DIVERGE(p)$ be the predicate that holds for a position $p \in POS(p_I)$ if $PATH(p,p')$ and either

(1) $PATH(p',p')$, or

(2) There is a move from p' to a position with space $> S(n)$.

Thus $DIVERGE(p)$ holds for each p iff there is an infinite nonalternating play from p. Membership in $POS(p_I)$ and the predicates PATH, APATH, and DIVERGE can be decided in nondeterministic space $S(n)$, and hence in deterministic space $S(n)^2$ by [Savitch, 70].

We now define a recursion procedure $DECIDE(p,a)$:

If $p \in POS_0$ then goto [1] else goto [2].

[1] In this step, the moves of player in team τ_0 are decided. If DIVERGE(p) then return false. Else if $a = 0$ then return true. Else, deterministically consider each $p' \in POS(p_I)$ such that APATH(p,p') holds. If DECIDE(p',a-1) holds for all such p' then return true else return false.

[2] Here the moves of players in team τ_1 are decided. If $a = 0$ then return false. Else, deterministically consider each $p' \in POS(p_I)$ such that APATH(p,p') holds. If DECIDE(p',a-1) holds for any such p' then return true else return false.

It can be shown that for each $p \in POS(p_I)$, DECIDE(p,a) holds iff team τ_1 has a winning strategy from position p where all plays have $\leq a$ alternations. Hence, DECIDE(p_I,A(n)) decides the outcome of G.

Note that each invocation of the procedure DECIDE can be implemented in deterministic space $S(n)$ and there are at most $A(n)$ recursive cells. Also, $S(n)^2$ global space is required to compute the predicates APATH and DIVERGE. Thus the total space requirement is $(A(n) + S(n))S(n)$.

The procedure for deciding the nonloss outcome of G is similar, except that we delete "If DIVERGE(p) then return false" from [1] and add "If DIVERGE(p) then return true" as the first statement of [1]. □

COROLLARY 4.1. (due to Borodin) *For* $S(n) \geq \log n$ *and* $A(n) \geq 0$,

$$\text{A-SPACE,ALT}(S(n),A(n)) \subseteq \text{DSPACE}((A(n) + S(n))S(n)).$$

THEOREM 4.2. *If hierarchical game* G *has space bound* $S(n) \geq \log n$ *and alternation bound* $A(n)$, *then the outcome and nonloss outcome of* G *can be decided in deterministic space* $(A(n) + 1)g(O(S(n)),\chi(G))$.

Proof. As in Theorem 3.3, we can apply Lemma 3.3 $\chi(G)$ times to yield a game of perfect information G^* with space bound $g(O(S(n),\chi(G))$ where G^* has the same outcome as G. Since the construction of Lemma 3.3 introduces no new alternations, $A(n)$ is an alternation bound of G^*. Thus by Theorem 4.1, the outcome of G^* can be decided in deterministic space $(A(n)+1)g(O(S(n)),\chi(G))$. □

COROLLARY 4.2. *For* $S(n) \geqslant \log n$, $A(n) \geqslant 0$, *and any* $k \geqslant 1$,

$$PA_k\text{-SPACE,ALT}(S(n),A(n)) \subseteq \text{D-SPACE}((A(n)+1)g(O(S(n)),k)) \ .$$

5. A DECISION PROCEDURE FOR TIME AND BRANCH BOUNDED GAMES

Let G be a game of incomplete information with time bound $T(n) \geqslant n$. Let G have *branch bound* $b \geqslant 0$ if for each position $p \in POS$, $b \geqslant |\{p' | p \vdash p'\}|$. Fix an initial position p_I of length n. We assume $T(n)$ is constructible (otherwise try $T(n) = 0,1,\ldots$). To decide the win outcome of G, we need only choose each strategy σ for team τ_1 and verify that this team wins for any play π induced from σ. But each such play π has at most $T(n)$ moves and can be stored in space $S(n)\log(b)$. This space also suffices to deterministically verify that σ is Markov$(m(n))$. Thus we have shown:

THEOREM 5.1. *The win outcome and Markov(m(n)) outcome of any multiplayer game of incomplete information with time bound* $T(n) \geqslant n$, *and branch bound* b *can be decided in deterministic space* $S(n)\log(b)$.

COROLLARY 5.1. *For any* $T(n) \geqslant n$,

$$MPA_k\text{-TIME}(T(n)) \subseteq \text{D-SPACE}(S(n)).$$

6. CONCLUSION

This paper gives decision algorithms for various classes of games of incomplete information. In light of the lower bound results of [Peterson and Reif, 79] our algorithms are in many cases asymptotically optimal.

Our algorithms are applied in this paper to obtain upper bounds and time-space tradeoffs on the complexity of multiplayer alternation machines of [Peterson and Reif, 79]. Our algorithm for space bounded games of incomplete information is also utilized [Reif and Peterson, 80] for deciding certain formulae of a multiprocess logic of incomplete information. We believe that our method given in Section 3.3 for elimination of incomplete information in a hierarchical game, may be of use in many other problems where ambiguity and incomplete information arises; for example the understanding of natural language.

It would be worthwhile to investigate heuristic techniques for deciding games of incomplete information. We suggest here the use of α-β search on the game tree of the game G^* obtained in Theorem 3.3 by eliminating incomplete information in a hiearchical game G (the α-β search technique avoids exploration of the entire game tree).

BIBLIOGRAPHY

Chandra, A.K., D.C. Kozen and L.J. Stockmeyer, "Alternation," Research Report RC 7489, IBM, Yorktown Heights, N.Y., Jan., 1978.

Even, S. and Tarjan, R.E., "A combinatorial problem which is complete in polynomial space," 7th Annual ACM Symp. on Theory of Computing, May 1976, Hershey, pp. 41-49.

Fraenkel, A.S., Garey, M.R., Johnson, D.S., Schaefer, T.J. and Yesha, Y., "The complexity of checkers on an $N \times N$ board-preliminary report," Proc. 19th IEEE Symp. on Foundations of Computer Science, Oct. 1978, pp. 55-64.

Hartmanis, J. and R.E. Stearns, "On the computational complexity of algorithms," *Trans. Amer. Math. Soc.* 117 (1965), pp. 285-306.

Hopcroft, J.E. and Ullman, J.D., *Introduction to Automata, Theory, Languages and Computation*, Addison-Wesley Publ. Comp., 1979.

Jones, N.D., "Blindfold games are harder than games with perfect information," *Bulletin of the European Assoc. for Theoretical Comp. Sci.*, No. 6, Oct. 1978, pp. 4-7.

Jones, N.D., "Space-bounded reducibility among combinatorial problems," *J. Comput. System Sci.* 11, (1975), pp. 68-85.

Lichtenstein, D. and Sipser, M., "GO is Pspace Hard," *JACM* 27 (1980), pp. 393-401.

Lichtenstein, D. and Sipser, M., "Computing a perfect strategy for $n \times n$ chess requires time exponential in N," 8th ICALP, Acre, Israel, (July 1981), pp. 278-293.

Peterson, G.L. and J.H. Reif, "Multiple Person Alternation," Proc. 20th IEEE Symp. on Foundations of Computer Science, October 1979.

Peterson, G.L., "Succinct representation random strings, and complexity classes," 21st Foundations on Foundations of Computer Science, (Oct. 1980), pp. 86-95.

Reif, J.H., "Universal two person games of incomplete information," 11th Annual ACM Symposium for Theory of Computing, 1979.

Reif, J.H. and G.L. Peterson, "A dynamic logic of multiprocessing with incomplete information," Proceedings 7th Annual ACM Symposium on Principles of Programming Languages, Las Vegas, (Jan. 1980), pp. 193-202.

Reif, J.H., "Algorithms for mixed strategies in multistage games of incomplete information," in preparation.

Savitch, W.J., "Relationships between nondeterministic and deterministic tape complexities," *J. Comput. Sci.* 4, (1970), pp. 177-192.

Schaefer, T.J., "Complexity of some two person perfect-information games," *J. Comp. and Sys. Science* 16, No. 2, pp. 188-225.

Stockmeyer, L.J., "The polynomial-time hierarchy," *Theoretical Computer Science* 3, (1977), pp. 1-22.

Stockmeyer, L.J. and Meyer, A.R., "Word problems requiring exponential time: Preliminary report," Proceedings 5th ACM Symp. on Theory of Computing, 1973, pp. 1-9.

Stockmeyer, L.J. and Chandra, A.K., "Provably difficult combinatorial games," *SIAM J. Computing* 8, No. 2 (May 1979), pp. 151-174.

APPENDIX: MACHINE DEFINITIONS

Multiperson alternation is defined here from

(1) a restricted nondeterministic Turing machine N, where we

(2) partition each state and assign players various rights to view and modify tapes and portions of states, and then we

(3) introduce a multiplayer game, called a computation game, which is used to define acceptance for the resulting multiperson alternating machine.

(NOTE: These three steps can be viewed as an algorithm for defining a game-like machine from a nondeterministic machine. It will be obvious that we could have begun with any type of nondeterministic machine. For example, if we begin with a nondeterministic RAM than we must assign players rights to view and modify various memory registers rather than types. Acceptance of the resulting multiplayer RAM would be again defined by a computation game.)

A1. Definition of a Nondeterministic Machine

A *nondeterministic Turing machine* (N-Tm) is a tuple

$$N = (Q, q_I, \Sigma, \Gamma, \#, b, t, \delta)$$

where Q is a finite set of *states*,

$q_I \in Q$ is the *initial state*

Σ is a finite set of *input symbols*

Γ is a finite set of *tape symbols*

$\#, b \in \Gamma - \Sigma$ are the distinguished *endmarker* and *blank* symbols

r is the number of tapes

$\delta \subseteq (Q \times \Gamma^t) \times (Q \times \Gamma^t \times \{\underline{\text{left}}, \underline{\text{right}}, \underline{\text{static}}\}^t)$
is the *transition relation.*

The operation of this machine is defined in the usual manner; however we do not define acceptance here (since we will give a nonstandard definition of acceptance later).

The tapes are named $1,\ldots,t$. Tape 1 is the read-only *input tape*. Given input string $\omega \in \Sigma^*$, input tape initially contains $\#\omega\#$, with the input tape head scanning the first symbol of ω. We assume there are no transitions past on an endmarker $\#$. The $t-1$ *work tapes* $2,\ldots,t-1$ initially contain two-way infinite strings of the blank symbol b. The *contents* of a tape are given as (X,Y) where X is the nonblank suffix of the portion of the tape to the left of the scan head, and Y is the nonblank prefix of the portion of the tape just under and to the right of the scan head.

A2. Definition of Multiplayer Alternation

A k+1-*player alternation machine* (MPA_k-TM) is a pair $M = (N,\text{VISIBLE})$ where:

(1) $N = (Q,q_I,\Sigma,\Gamma,\#,b,t,\delta)$ is a N-TM with restrictions given below.

(2) VISIBLE is a mapping from $\{0,\ldots,k\}$ to (not necessarily proper) subsets of $\{1,\ldots,r\}$.

We require that the state set $Q \subseteq \{0,\ldots,k\} \times Q_1 \times \cdots \times Q_s$ where $Q_1,\ldots,Q_s$ are finite sets and $s = r-t$. A *configuration* is a sequence $(i,q_1,\ldots,q_s,(X_1,Y_1),\ldots,(X_t,Y_t))$ where $q = (i,q_1,\ldots,q_s)$ is the current state, and (X_j, Y_j) are the current contents of tape j, for $j = 1,\ldots,t$. The *initial configuration* has the initial state q_I and the tape contents initialized as for N. Let POS be the set of all configurations. The *next move relation* $\vdash \subseteq \mathrm{POS} \times \mathrm{POS}$ is the binary relation on configurations defined in the usual manner from the transition function δ. We require that the *computation game* $G^M = (\mathrm{POS}, \vdash, \mathrm{VISIBLE})\ \tau_1$ be a $k+1$ player game satisfying axioms A1 and A2 of Section 2.1, with teams $\tau_0 = \{0\}$ and $\tau_1 = \{1,\ldots,k\}$.

(NOTE: In a previous paper [Peterson and Reif, 78] we have an equivalent definition of MPA-TMs where we are not given VISIBLE explicitly, but allow each player i a maximal set VISIBLE (i) which satisfies Axioms A1 and A2.)

We say M *accepts* input string $\omega \in \Sigma^*$ if players $\tau_1 = \{1,\ldots,k\}$ have a winning strategy in G^M from the initial configuration. Let the *language of* M be $L(M) = \{\omega \in \Sigma^* | \omega$ is accepted by $M\}$.

A3. Machine Intuition

To aid the reader's intuition, we introduce some (redundant) terminology common to complexity theory. We have already used *configurations* for the positions of a computation game G^M. Each play of G^M is called a *computation sequence* and the game tree T is the *computation tree*. The computation sequences induced by a winning term strategy form an *accepting subtree* of T. The player 0 is called the $\forall$-player and the players i, $1 \leq i \leq k$, are called the $\exists i$-players. (They form the *existential team* τ_1.) The *accepting states* (*rejecting states*, respectively) are those states in Q which have 0-th component 0 (not 0, respectively) and from which there is no state transition.

A4. Machine Types

Machine types are named in Figure 1, depending on the type of their computation games. It can be seen that our machine definitions are equivalent to those standard definitions appearing in the referenced papers.

A5. Complexity Bounds for Machines

M has *space bound* $S(n)$ (*time bound* $T(n)$, *alternation bound* $A(n)$, respectively) if for each input string $\omega \in \Sigma^n$ accepted by M, there exists an accepting subtree T' such that no tape contains $> S(n)$ non-blank tape cells each computation sequence $\pi \in T'$ contains $\leq T(n)$ moves, π contains $\leq A(n)$ alternations, respectively). Thus, the computation game G^M has space bound $O(S(n+O(1)))$, time bound $T(n+O(1))$, and alternation bound $A(n+O(1))$, respectively.

For any $\alpha \in \{D,N,A,BA_k,PA_k,MPA_k\}$ we let α-TIME$(T(n))$ (α-SPACE$(S(n))$, α-SPACE,TIME$(S(n),T(n))$,α-SPACE,ALT$(S(n),A(n))$, respectively) denote the class of languages accepted by α-TMs in time $T(n)$ (space $S(n)$, simultaneous space $S(n)$ and time $T(n)$, simultaneous space $S(n)$ and alternations $A(n)$, respectively).

Computation Game Type	Machine Type	Abbreviated Name	Reference
deterministic	deterministic TM	D-TM	[Hopecroft and Ullman,79]
nondeterministic	nondeterministic TM	N-TM	"
perfect information	alternating machine	A-TM	[Chandra, Kozen, and Stockmeyer, 79]
2 player blindfold	blind alternating machine	PA_1-TM	[Reif,79]
2 player incomplete information	private alternating machine	PA_1-TM	"
k+1 player Markov(1)	k+1 player Markov alternating machine	MA_k-TM	[Peterson and Reif, 79]
k+1 player blindfold	k+1 player blind alternating machine	BA_k-TM	"
k+1 player hierarchical	k+1 player private alternating machine	PA_k-TM	"
k+1 player incomplete information	k+1 player alternating machine	MPA_k-TM	"

Figure 1. Multiplayer alternating machines defined in increasing generality.