JOURNAL OF COMPUTER AND SYSTEM SCIENCES **29**, 274–301 (1984)

The Complexity of Two-Player Games of Incomplete Information*

JOHN H. REIF†

Aiken Computation Laboratory, Harvard University, Cambridge, Massachusetts 02138

Received December 20, 1981; revised March 1, 1984

Two-player games of *incomplete information* have certain portions of positions which are private to each player and cannot be viewed by the opponent. Asymptotically optimal decision algorithms for space bounded games are provided. Various games of incomplete information are presented which are shown to be *universal* in the sense that they are the hardest of all reasonable games of incomplete information. The problem of determining the outcome of these universal games from a given initial position is shown to be complete in doubly exponential time. "Private alternating Turing machines" are defined to be a new type of alternating Turing machines related to games of incomplete information. The space complexity $S(n)$ of these machines is characterized in terms of the complexity of deterministic Turing machines, with time bounds doubly exponential in $S(n)$. *Blindfold games* are restricted games in that the second player is not allowed to modify the common position. Asymptotically optimal decision algorithms for space bounded blindfold games are provided. Various blindfold games are also shown to have exponential space complete outcome problems and to be universal for reasonable blindfold games. "Blind alternating Turing machines" are defined to be private alternating Turing machines with restrictions similar to those in blindfold games. The space complexity of these machines is characterized in terms of the complexity of deterministic Turing machines with a single exponential increase in space bounds.

1. INTRODUCTION

A two-player game G consists essentially of disjoint sets of positions for two players named 0 and 1, plus relations specifying legal next-moves for the players. We assume positions are strings over a finite alphabet. A *position* P contains portions which are *private* to each player (invisible to their opponent) and the remaining portions of P are *common* and may be publicly viewed by both players. The set of legal next-moves for a given player must be independent of the opponent's private portions of positions.

* A preliminary version of this paper appeared as "Universal Games of Incomplete Information" in the 11th Annual ACM Symposium for Theory of Computing, 1979.

† This work was supported in part by Office of Naval Research, Contract N00014-80-C-0647 and National Science Foundation Grant MC79-21024.

0022-0000/84 $3.00

The game G is of *perfect information* if no position contains a private portion. On the other hand, a game is *blindfold* if player 0 never modifies the common portion of a position.

For example, consider the game PEEK of Fig. 1a. (PEEK was first described in Stockmeyer and Chandra [18].) A *position of* PEEK consists of a box with two open ends and containing various plates stacked horizontally within. The plates are perforated by holes of uniform size in various places. The top and bottom of the box are also perforated with holes. Each plate contains a knob on one of the open ends of the box, and the plate may slide horizontally to either of two locations: "in" or "out." Once "out," a plate can only be pushed "in," and *vice versa*. The players stand at the two open ends of the box. A *move* by a player $a \in \{0, 1\}$ consists of grasping a knob from his side and pushing the corresponding plate either "in" or "out." The player may also pass. If just after a move of player a the plates are aligned so that the player can "peek" through a sequence of holes from the top to the bottom, then the

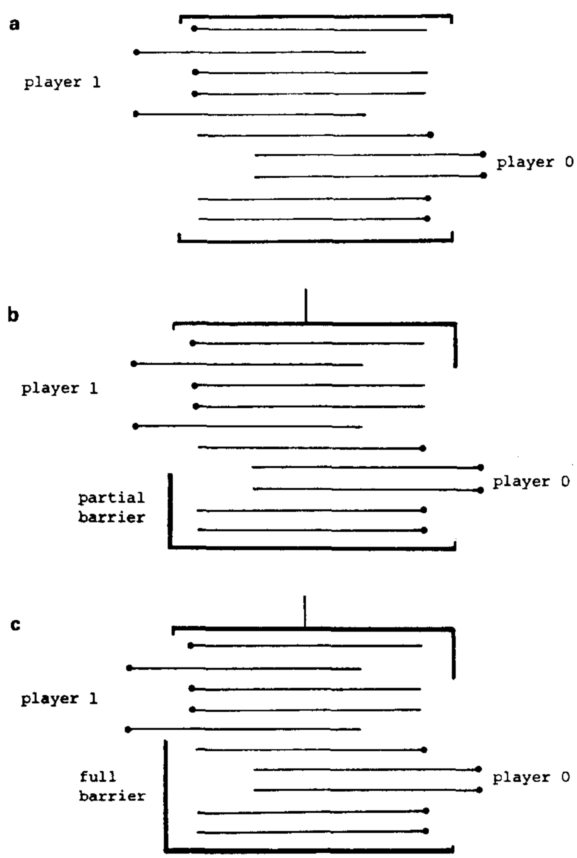

FIG. 1. (a) A position of PEEK; (b) a position of PRIVATE-PEEK; (c) a position of BLIND-PEEK.

player *wins*. PEEK is a game of perfect information: each player knows the pattern of holes on the plates and can view the location of all the plates.

To introduce *private portions of positions*, we place partial barriers on both ends of the box, as in Fig. 1b. These barriers hide the location of some, but perhaps not all, of the opponent's plates. Also, we place a barrier on top of the box, as in Fig. 1b, so each player only can view half of the top of the box. Both players are still aware of the pattern of holes on each plate. However, each player can attempt to "peek" through the box only from their half of the top of the box. Let PRIVATE-PEEK be the resulting game of incomplete information. By requiring that the barriers on the side of player 1 obscure the locations of *all* the opponent's plates, we have the blindfold game BLIND-PEEK (see Fig. 1c).

The *outcome problem for a game G* is the problem of determining the existence of a winning strategy for player 1, given an initial position. If no *a priori*-bound is placed on the size of positions of games, the outcome problem is undecidable (see the computation games of Sect. 3). We consider a game to be *reasonable* if its space bound for positions is $O(n)$.

Given a class of games $\mathscr{C}$, a game G is *universal* to $\mathscr{C}$ if (1) $G \in \mathscr{C}$ and (2) the outcome problem for each $G' \in \mathscr{C}$ is log-space reducible (see Stockmeyer and Meyer [17]; a log-space reduction is always polynomial time) to the outcome problem for G. The game PEEK was shown universal to reasonable games of perfect information in Stockmeyer and Chandra [18]. We show BLIND-PEEK is universal for all reasonable blindfold games, and that PRIVATE-PEEK is universal for all reasonable games. While the outcome problem for PEEK is complete (with respect to log-space reductions) in exponential time, the outcome problem for BLIND-PEEK is complete in exponential space, and the outcome problem for PRIVATE-PEEK is complete in double exponential time.

A game with an easy-to-compute next-move relation can be considered to be a computing machine. Game G *accepts input* ω, depending on the outcome of the game from an initial position containing ω. Games of perfect information related in this way to the *alternating machine* (**A-TM**) of Chandra, Kozen, and Stockmeyer [1] in which existential states (identified with player 1) alternate with universal states (player 0) during a computation. A *nondeterministic Turing machine* (**N-TM**) is related to a game of perfect information with the second player absent, and a *deterministic Turing machine* (**D-TM**) is related to a game of perfect information with at most a single next-move from any position.

In this paper we introduce two new types: *private* and *blind alternating machines*. We add to an A-TM certain work tapes private to universal states (player 0); the machine cannot read the private tapes while in existential states. The result is a private alternating machine (**PA-TM**), as in Fig. 2. For a blind alternating machine (**BA-TM**) we restrict a PA-TM so that the universal states can write only on their private tapes, and on no other tapes. *Acceptance* of input strings by these machines is defined by the outcome in corresponding computation games.

Let $\mathscr{F}$ be a set of functions on variable n. For each $\alpha \in \{D, N, A, PA, BA\}$, let $\alpha SPACE(\mathscr{F})$ be the class of languages accepted by α-TMs within some space bound

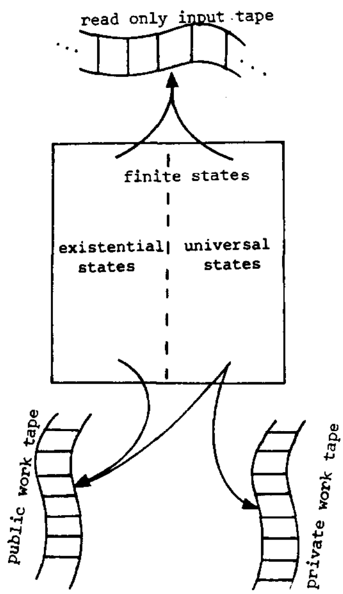

FIG. 2. An alternating Turing machine with a tape private to the universal states.

in $\mathscr{F}$, and let $\alpha TIME(\mathscr{F})$ be the class of languages accepted by α-TMs within some time bound in $\mathscr{F}$, Let EXP($\mathscr{F}$) be the set of functions

$$\{c^{F(n)} \mid c > 0 \quad \text{and} \quad F(n) \in \mathscr{F}\}.$$

We drop the set brackets in the above notation if $\mathscr{F}$ is a singleton set and let EXP($f(n)$) denote EXP($\{f(n)\}$). For example, the polynomial functions $\mathrm{POLY}(n) = \{n^c \mid c \geqslant 1\}$ can be defined in this notation as $\mathrm{POLY}(n) = \mathrm{EXP}(\log n)$.

Chandra, Kozen, and Stockmeyer [1] relate the space and time complexity of A-TMs and D-TMs as follows:

For each function $S(n) \geqslant \log n$,

$$ASPACE(S(n)) = DTIME(\mathrm{EXP}\ (S(n)))$$

$$ATIME(\mathrm{EXP}\ (S(n))) = DSPACE(\mathrm{EXP}\ (S(n))).$$

We characterize the space complexity PA-TMs and BA-TMs in terms of the time and space complexity of A-TMs and D-TMs as follows (see Fig. 3):

For each function $S(n)$,

$$\begin{aligned} BASPACE(S(n)) &= ATIME(\mathrm{EXP}(S(n))) \\ &= DSPACE(\mathrm{EXP}(S(n))), \\ PASPACE(S(n)) &= ASPACE(\mathrm{EXP}(S(n))) \\ &= DTIME(\mathrm{EXP}(\mathrm{EXP}(S(n)))). \end{aligned}$$

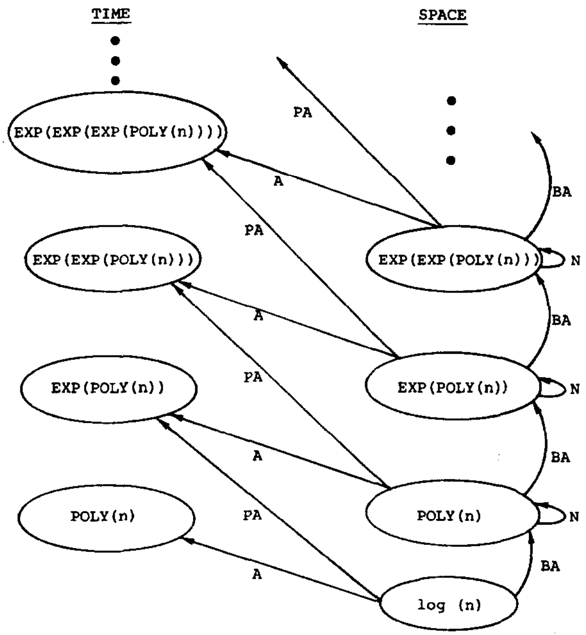

FIG. 3. Complexity jumps for α-TMs from αSPACE to deterministic time and space; $\alpha = A$ for "alternation;" $\alpha = BA$ for "blind alternation;" $\alpha = PA$ for "private alternation;" $\alpha = N$ for "nondeterministic."

Also, the time complexity of PA-TMs, BA-TMs, and A-TMs are all roughly the same:

$$\begin{aligned} PATIME(\mathrm{EXP}(S(n))) &= BATIME(\mathrm{EXP}(S(n))) \\ &= ATIME(\mathrm{EXP}(S(n))) \\ &= DSPACE(\mathrm{EXP}(S(n))). \end{aligned}$$

This paper is organized as follows: the next section defines games of incomplete information; Section 3 introduces our PA-TMs and BA-TMs; Section 4 presents decision algorithms for space bounded games, and also games with both alternation and space bounds; Section 5 gives lower bounds on the complexity of space bounded games; Section 6 considers the complexity of time bounded PR-TMs and BA-TMs; Section 7 describes certain propositional formula games which are universal for reasonable games; and Section 8 concludes the paper with mention to extensions and applications of this work to multiplayer games and multiprocessing, in collaboration with Gary Peterson.

2. Two-Player Games of Incomplete Information

2.1. *Game Definitions*

A (two-player) *game* is a tuple $G = (\mathrm{POS}, \vdash)$, where

(i) POS is the set of *positions*, with $\mathrm{POS} = \{0, 1\} \times PP_0 \times PP_1 \times CP$ and PP_0, PP_1, CP are sets of strings over a finite alphabet.

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

The *players* are named 0, 1. If $p = (a, pp_0, pp_1, cp)$ is a position in POS, then p is composed of a number $a \in \{0, 1\}$ indicating which player's turn is next, a portion pp_0 which is *private to player* 0, a portion pp_1 which is *private to player* 1, and a *common portion* cp.

For $a \in \{0, 1\}$, let POS_a be the set of positions with 1st component a; thus POS_a are the positions for which it is player i's next move.

Informally, a player wins by making the last move. Thus the object of the game is to force the opponent into a position from which there is no next move. Formally, let the set of *winning positions* be $W = \{p \in \mathrm{POS} \mid p \vdash p' \text{ for no } p'\}$. If $p \in W \wedge \mathrm{POS}_0$ then p is a *winning position* for player 1.

Given a position $p = (a, pp_0, pp_1, cp)$ let $\mathrm{vis}_1(p) = (\mathrm{i}, pp_1, cp)$ be the *portion visible to player* 1 and let $\mathrm{priv}_1(p) = pp_1$ be the *portion of p private to player* 1 ($\mathrm{vis}_0(p)$ and $\mathrm{priv}_0(p)$ are defined similarly, with 0 in place of 1).

The idea of imperfect information is captured in the following two axioms. Axiom 1 asserts that a player cannot modify the portion of the position private to his opponent. Axiom 2 asserts that a player's possible moves next are independent of the portion of the position private to his opponent.

A1. If $p \in \mathrm{POS}_1$ and $p \vdash p'$ then $\mathrm{priv}_0(p) = \mathrm{priv}_0(p')$.

A2. If $p, q \in \mathrm{POS}_1 - W$ and $\mathrm{vis}_1(p) = \mathrm{vis}_1(q)$ then $\{\mathrm{vis}_1(p') \mid p \vdash p'\} = \{\mathrm{vis}_1(q') \mid q \vdash q'\}$.

We also assume both A1 and A2 hold with 0 exchanged with 1.

2.2. *Plays and Strategies for Games*

For any finite string π of positions, let $\mathrm{last}(\pi)$ be the last position of π. Fix an *initial position* $p_I \in \mathrm{POS}$. 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 $\mathrm{last}(\pi) \in W$ whenever π is finite. A play π is said to be a *win for player* 1 if π is finite and $\mathrm{last}(\pi) \in \mathrm{POS}_0 \wedge W$.

A *play prefix* π is a finite nonnull initial substring of a play. Intuitively, a play prefix represents a sequence of legal moves starting from an initial position. Note that the players need not alternate. After any play prefix π, the sequence $\mathrm{vis}_1(\pi)$ represents the extent of player 1's knowledge about the game play to date. We define $\mathrm{vis}_1(\pi)$ inductively. Let $p = \mathrm{last}(\pi)$. If π is of length 1 then $\mathrm{vis}_1(\pi) = \mathrm{vis}_1(p)$. Suppose we are given $p' \in \mathrm{POS}$ such that $p \vdash p'$. The move $p \vdash p'$ is a *private move of player* 0 if both $p \in \mathrm{POS}_0$ and $\mathrm{vis}_1(p) = \mathrm{vis}_1(p')$. If the move is private then we let $\mathrm{vis}_1(\pi p') = \mathrm{vis}_1(\pi)$ (intuitively, this first case ensures that player 1 cannot detect moves of player 0 which do not modify that portion of p visible to player 1), and otherwise $\mathrm{vis}_1(\pi) = \mathrm{vis}_1(\pi), \mathrm{vis}_1(p)$.

The *game tree* T is the set of play prefixes. The *root* of T is the initial position p_I. Each play prefix π is considered a *node* of T. The *children* of π are those play prefixes π' of length one more than π and such that π is a prefix of π'. Let T_1 be the set of play prefixes π such that $\mathrm{last}(\pi) \in \mathrm{POS}_1 - W$; thus it is player 1's turn to move at $\mathrm{last}(\pi)$.

A *strategy* for player 1 is a function $\sigma \colon T_1 \to T$ such that

(1) for any $\pi \in T_1, \sigma(\pi)$ is a child of π, and

(2) if $\pi, \pi' \in T_1$ and if $\mathrm{vis}_1(\pi) = \mathrm{vis}_1(\pi')$ then $\mathrm{vis}_1(\sigma(\pi)) = \mathrm{vis}_1(\sigma(\pi'))$.

Thus σ is a rule for player 1 to select his next move. Condition (2) says that this selection must be made only on the basis of the knowledge player 1 has about the progress of the game to date. (Note that this is not implied from axiom A2.) A play π is a *play induced by 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 called a *winning strategy* for player 1 iff every play by strategy σ is a win for player 1.

The *outcome problem* for game G is: given an initial position $p_I \in \mathrm{POS}$, is there a winning strategy for player 1?

Note that although games such as checkers and Go have standard initial positions, their rules may be readily generalized to $n \times n$ boards. Initial positions of games are not always fixed in this paper, since we shall wish to consider the outcome problem for games, given *arbitrary* initial positions. This allows us a meaningful notion of the complexity of the outcome of these games. The complexity of various generalized games of perfect information is considered in Schaefer [15], 1978, Even and Tarjan [2], Fraenkel, Garey, and Johnson [3], Lichtenstein and Sipser [8], Fraenkel and Lichtenstein [4], Stockmeyer and Chandra [18]. The complexity of a blindfold game was first considered in Jones [6].

To model a game like two-player poker, in which players do not have perfect information even at the start, we may simply add an initial move which allows player 0 to choose both player's cards; this is justified by Proposition 2.1, given below.

It should be clear that the outcome of a game is not affected if player 0 is allowed to "cheat," by viewing the private portions of player 1's positions. For each position $p \in \mathrm{POS}$, let p^c be the position derived from p by making common to *both* players that portion of p originally private to player 1. Let G^c be the game so derived from game G. It follows immediately from our definition of strategies that

PROPOSITION 2.1. *Player* 1 *has a winning strategy in G from initial portion p_I iff player* 1 *has a winning strategy in G^c from initial position p_I^c.*

Nevertheless, the outcome of *probabilistic strategies*, as defined in Reif [11] are highly dependent on the existence of private positions of both players.

2.3. *Special Types of Games*

A strategy σ is *Markov* if $\sigma(\pi)=\sigma(\pi')$ for all play prefixes π, π' such that $\text{last}(\pi)=\text{last}(\pi')$. Markov strategies are independent of previous play except for the current position. Thus it suffices to consider a Markov strategy to be a mapping from the current position to the next position.

A game is *perfect information* if the private portions of any position is the fixed value *null*, so that the only nontrivial information in a position is the common portion. Thus $\text{POS} \approx \{0, 1\} \times CP$. For example, chess, checkers, and Go are all games of perfect information.

PROPOSITION 2.2. *In any game of perfect information, if player* 1 *has a winning strategy* σ, *then player* 1 *has a winning Markov strategy.*

To prove this proposition we define a strategy σ' such that for any play prefix $\pi, \sigma'(\pi)=\sigma(\pi')$, where π' is the lexically minimal play prefix such that $\text{last}(\pi)=\text{last}(\pi')$. Then σ' is winning for player 1 if σ was. On the other hand, strategies for games of incomplete information must generally depend on previous play to determine the possible private positions of the opponent.

A game is *blindfold* if the common portions of p and p' are the same whenever $p \in \text{POS}_0$ and $p \vdash p'$; thus there is no interchange of information from player 0 to player 1 in a blindfold game. Some examples of blindfold games are given in Section 7. Also see Jones [6]. The traditional German game of blind chess is not truly a blindfold game since there is a gradual transfer of positional knowledge when players are informed of illegal moves.

A game is *solitaire* if on any play prefix on which player 1 has made at least one move, the remaining moves of player 0 are deterministic. An initial sequence of moves of player 0, preceding any move of player 1, may allow player 0 to develop its private portion of the position. For example, Battleship, Mastermind, and of course the card game of solitaire are all solitaire games.

A game is *nondeterministic* if $\text{POS}_0 \wedge (\text{POS} - W)$ is empty. Note that a nondeterministic game can always be made a game of perfect information without modifying its outcome, by simply letting the private portions of positions be in the common portion of positions.

A game is *deterministic* if the next move relation $\vdash$ is, hence for each portion $p \in \text{POS}$, there is at most one position $p' \in \text{POS}$ such that $p \vdash p'$.

2.4. *Complexity Bounds on Games*

Let $G=(\text{POS}, \vdash)$ be a game. Let us assume for any position $p \in \text{POS}$, the positions $\{p' \mid p \vdash p'\}$ are ordered $\vdash_1(p),\ldots, \vdash_d(p)$ so that $\vdash_i(p)$ is the ith position derived by a next-move from p. A *next-move transducer for* $\vdash$ is a one-to-one encoding function f that maps positions into Σ^* for some finite alphabet Σ. The next-move transducer, when given initial position p_I, i, and $f(p)$ for some position p, produces $f(\vdash_i(p))$. The space used to write p_I is not counted in the space of the transducer, but the space used to write $f(p)$ is counted.

Let a move $p \vdash p'$ be an *alternation* if $p' \notin W$ and either ($p \in \mathrm{POS}_0$ and $p' \in \mathrm{POS}_1$) or ($p \in \mathrm{POS}_1$ and $p' \in \mathrm{POS}_0$).

Game G has *time bound* $T(n)$ (*alternation bound* $A(n)$, *space bound* $S(n)$, respectively) if on each position $p_I \in \mathrm{POS}$ of length n from which player 1 has a winning strategy, there is some σ such that for each play π induced from σ, π has $\leqslant T(n)$ moves (π has $\leqslant A(n)$ alternations, the next move transducer requires $\leqslant S(n)$ work tape cells for the moves of π, respectively).

It is interesting to note that any game with a fixed initial position and finite time or space bound, can be represented as a physical object with a finite game board and a finite set of tokens for marking positions.

Let G be a *reasonable game* if it has space bound $O(n)$.

3. Private and Blind Alternating Machines

The *alternating machine* proposed by Chandra, Kozen, and Stockmeyer [1] has a natural correspondence to games of perfect information. The states of alternating automata are named either *universal* or *existential*. The sequencing between existential and universal states corresponds to the alternation of moves by players in the play of a game.

We introduce here a new type of alternating machine with private tapes which have a natural correspondence to games of incomplete information. In fact, we will define the languages accepted by these machines by the existence of winning strategies for the corresponding computation games.

Let a *private alternating machine* (**PA-TM**) be a tuple

$$M = (S, Q, q_I, \Sigma, \Gamma, \#, b, t, t_p, \delta),$$

where

S is a finite set,

$Q \subseteq \{0, 1\} \times S$ is the *state set*,

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

Σ, Γ are the finite sets of *input* and *tape* symbols with $\Sigma \subseteq \Gamma$,

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

t is the number of tapes and t_p is the number of private tapes,

$\delta \subseteq (Q \times \Gamma^t) \times (Q \times \Gamma^{t-1} \times \{\text{left, right, static}\}^t)$ is the *transition relation*, with restrictions given below.

If $q = (a, s)$ is a state in Q, then q is composed of a number $a \in \{0, 1\}$ and a *common portion* $s \in S$. If $a = 1$ then q is an *existential state* and otherwise if $a = 0$ then q is a *universal state.*

There is a read-only *input tape*. Initially the input tape contains $\#\omega\#$, with the input tape head scanning the first symbol of ω, where $\omega \in \Sigma^*$ is an input string. (We assume there are no transitions past the endmarkers $\#$.) There are also $t-1$ *work tapes*, initially containing two-way infinite strings of the blank symbol b. The tapes $1,..., t_p$ are *private work tapes*; they can only be written on from a universal state, and the transitions from each existential state are independent of the contents of the private work tapes (these restrictions to δ are made precise below). The other $t-t_p-1$ tapes are *common work tapes* and might be written on from any state of Q. The *contents* of a tape are given as (L, R), where L is the nonblank suffix of the portion of the tape to the left of the scan head, and R is the nonplank prefix of the portion of the tape just under and to the right of the scan head.

We now define the *computation game* $G^M = (\mathrm{POS}, \vdash)$, where POS are the positions (to be defined) of M and the next moves $\vdash \subseteq \mathrm{POS} \times \mathrm{POS}$ are as defined by the transition relation δ of M. The player 1 which makes moves from existential states is called the *existential player* and the player 0 which makes moves from universal states is called the *universal player.*

Let a *position* of M be a tuple $p = (a, pp_0, pp_1, cp)$, where

(i) $a \in \{0, 1\}$ indicates that the current state is either existential ($a = 1$) or universal ($a = 0$),

(ii) the *portion* pp_0 *private to the universal player* contains the contents of the private tapes,

(iii) the portion pp_1 private to the existential player 1 is *null.*

(iv) the *common portion* cp is a pair whose first part is the common portion of the state, and whose second part is the contents of the common work tapes.

Thus the portion $\mathrm{vis}_1(p)$ *visible to the existential player* is all of p but the contents of the private tapes, and the portion $\mathrm{vis}_0(p)$ *visible to the universal player* is all of p. (This is justified by Proposition 2.1.) We require G^M to satisfy axioms (A1), (A2); this gives us our required restrictions on the transition function δ of M. (NOTE. We may further decompose each of the states into a *private component*, with restrictions to the transition relation just as given here for the private tapes. This additional complication given in our original (Reif [11]) definition of *PA*-TMs is not required as long as there is at least one cell of one private tape which may be used to store the state of the universal player.)

For any input string $\omega \in \Sigma^*$, let the *initial position* $p_I(\omega)$ have initial state q_I and tape contents initialized as described above. We introduce some (redundant) terminology to aid the reader's intuition. The *accepting states* are those universal states with no successors. The *rejecting states* are those existential states with no successors. Each play of G^M is called a *computation sequence* and the game tree T called a *computation tree*. The input string $\omega \in \Sigma^*$ is *accepted by M* if the existential player has a winning strategy. The computation sequences induced by a winning strategy form an *accepting subtree* of T. Let the *language of M* be $L(M) = \{\omega \in \Sigma^* \mid \omega \text{ is accepted by } M\}$.

The PA-TM is a natural generalization of various *types* of machines previously described in the literature. If M has no private tapes, it is an *alternating machine* (**A-TM**) as described by Chandra, Kozen, and Stockmeyer [1]. These have computation games which are of perfect information. If M is further restricted to allow only those universal states which are accepting (i.e., have no successors), then it is a *nondeterministic Turing machine* (N-TM) as is now common in the literature. If the transition relation of M is still further restricted to be deterministic, then we have a *deterministic Turing machine* (D-TM or just TM), the machine originally envisioned by Turing.

We now define still another type of machine. Let a **BA-TM** be a PA-TM restricted so that the universal player can never modify the common portion of any position, i.e., can never write on nor move the heads of the common tapes nor modify the common portion of the state. (Note that this property is easy to decide from inspection of the transition relation of M.) The computation game of BA-TM is by definition a blindfold game. Thus we have defined for each *game type* g in $\mathscr{G} = \{$incomplete information, blindfold, perfect information, nondeterministic, deterministic$\}$ a corresponding *machine type* $m(g)$ in $\mathscr{M} = \{$private alternating, blind alternating, alternating, nondeterministic, deterministic$\}$ with computation game of type g.

The winning strategies of computation games can be recursively enumerated, and thus the language of each PA-TM and BA-TM is recursively enumerable. Also, the D-TMs accept all the recursively enumerable sets and each D-TM is a PA-TM and a BA-TM. Hence we have

THEOREM 2.1. *The PA-TMs and BA-TMs each accept precisely the recursively enumerable sets.*

We next consider the computational complexity of PA-TMs and BA-TMs. 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 is an accepting subtree T' such that no tape has more than $S(n)$ nonblank cells on any configuration (each computation sequence $\pi \in T'$ has at most $T(n)$ moves, each $\pi \in T'$ has at most $A(n)$ alternations, respectively). Thus computation game G_M has space bound $O(S(n + O(1)))$ (time bound $T(n + O(1))$, alternation bound $A(n + O(1))$, respectively) if M has space bound $S(n)$ (time bound $T(n)$, alternation bound $A(n)$, respectively).

By the usual tape encoding techniques (where we encode each $2/\varepsilon$ consecutive work tape cells as a $2/\varepsilon$-tuple in a new tape alphabet), we have a constant space compression result:

THEOREM 3.2. *For any $\varepsilon > 0$ and machine M of machine type g space bound $S(n)$, there is a machine with space bound $\varepsilon S(n)$ that accepts the same language as M, and with the same machine type g as M with no additional tapes or alternations.*

We also have a constant speed-up result:

THEOREM 3.3. *For any $\varepsilon > 0$ and any machine M of any type in $\mathscr{G}$, with time* *bound $T(n)$ such that* $\inf_{n\to\infty} T(n)/n = \infty$ *and at least one tape, there is a machine of the same type as M with time bound $\varepsilon T(n)$ and the same number of tapes, that accepts the same language as M.*

Proof. There is a constant d upper bounding the number of next possible moves from any given position of M. Thus there are at most d^t positions of M reachable after t moves from any given position. We construct a simulating machine M' of the same type as M and which accepts the same strings as M. As in Theorem 3.2, we encode each t consecutive cells of each tape of M as a t-tuple in the tape alphabet of M'. M' will have a distinguished state associated with each of the 2^{d^t} possible strategies of the existential player within the next t existential moves from any given position. Also, M' will have an additional tape, private to the universal player, which will contain a counter Δ, where $0 \leqslant \Delta \leqslant t$. A move by the universal player of M is *private* if it does not modify the common portion of the position. Given input string $w \in \Sigma^n$, the simulation will proceed in at most $T(n)/t$ phases, where in each phase M' simulates t nonprivate steps of M. At the start of a phase, Δ is set to t. Then M' moves one cell left, two cells right, and then one cell left on each of its tapes so as to determine the current relevant tape contents. Then the existential player of M' is allowed, by a single state transition, to choose its strategy for the next t existential steps of M (if no such strategy exists, M' rejects).

The universal player of M' then executes a series of rounds, each of which requires only a single step of M' and furthermore each is undetectable to the existential player of M'. At the start of a round, we can inductively assume that $t - \Delta$ is the number of nonprivate steps of M so far simulated by M' during this phase. On this round the universal player of M' simulates $t' = \min(t, \Delta)$ steps of M (some of these steps may be existential; for these moves the strategy previously chosen by the existential player of M' is used. At the end of the round, Δ is privately subtracted by $t' - t_p$, where t_p is the number of steps of the round which are private to the universal player. If now $\Delta > 0$ then we proceed to the next round, and otherwise we terminate the round. After the last round, the universal player makes visible to the existential player all (if any) the modifications to the common portion of the position which were made on the simulated nonprivate moves during this phase. M' makes four additional moves of the tape heads: (left, twice right, and left again) to update the tapes, and then the simulation proceeds to the next phase. M' makes at most 10 steps for every t steps of M, and the total time bound of M' is $n + \lceil n/t \rceil + 10\lceil T(n)/t \rceil \leqslant \varepsilon T(n)$ if $n \leqslant \frac{9}{20}\varepsilon T(n)$ and we let $t = 20/\varepsilon$. On the other hand, there are only a constant number of inputs of length $n > \frac{9}{20}\varepsilon T(n)$, and for these inputs we can use the finite state control to decide acceptance within time n. ∎

Next we show that the computation games of various types of machines are universal for the corresponding classes of games. Fix some functions $S(n) \geqslant \log n$ and $A(n)$ and let g be a game type in $\mathscr{G}$. Let $\mathscr{C}$ be the class of games of fixed game type g with space bound $S(n)$ and alternation bound $A(n)$. Let us assume that the set of positions derived by a single move from any given position of length n, can be computed in deterministic space $MS(n)$. For each game $G = (\mathrm{POS}, \vdash)$ of $\mathscr{C}$, let B_G

be the deterministic log space mapping from positions in POS to their binary string representation. Let N_G be a binary string encoding the deterministic space $MS(n)$ next move transducer for $\vdash$.

Clearly, there is a machine $M_{\mathscr{C}}$ such that for each game $G \in \mathscr{C}$ and position p of G, M accepts $(N_G, B_G(p))$ iff player 1 has a winning strategy in G from initial position p. Thus $M_{\mathscr{C}}$ decides the outcomes of all the games of $\mathscr{C}$. Furthermore, $M_{\mathscr{C}}$ has corresponding machine type $m(g)$ (i.e., its computation game is of type g) has tape alphabet $\{0, 1, b, \#\}$, space bound $S(n) + MS(S(n))$, and alternation bound $A(n)$. If $MS(S(n)) = O(S(n))$ then by Theorem 3.2, $M_{\mathscr{C}}$ need to have only space bound $S(n)$. Thus we have shown:

THEOREM 3.4. *If $MS(S(n)) = O(S(n))$ then the computation game $G^{M_{\mathscr{C}}}$ is a universal game for the game class $\mathscr{C}$.*

By applying the space compression Theorem 3.2, we have

COROLLARY 3.4. *For each game type $G \in \mathscr{G}$, if $\mathscr{R}$ is the class of reasonable games (i.e., with space bound $S(n) = n$) of type g, then there is a linear space bounded machine $M_{\mathscr{R}}$ of corresponding type $m(g)$ such that $G^{M_{\mathscr{R}}}$ is a universal game for $\mathscr{R}$.*

4. Decision Algorithms for Space Bounded Games

It is easy to show

THEOREM 4.1. *Any deterministic (nondeterministic, respectively) game with space bound $S(n) \geqslant \log n$ can be decided in deterministic (nondeterministic, respectively) space $O(S(n))$.*

By applying the result of Savitch [14] we can easily show

COROLLARY 4.1. *Any nondeterministic game with space bound $S(n) \geqslant \log n$ can be decided in deterministic space $O(S(n)^2)$.*

We consider now in turn decision algorithms for deciding games of perfect information, then games of incomplete information, and finally blindfold games.

4.1. *Deciding a Game of Perfect Information*

THEOREM 4.2. *For any $S(n) \geqslant \log n$, the outcome of any game G of perfect information with space bound $S(n)$ can be decided in deterministic time $2^{O(S(n))}$.*

This result will be utilized in Section 4.2. For completeness, we give here an algorithm similar to a procedure previously given by Chandra, Kozen, and Stockmeyer [1] for determining acceptance of an alternating machine with a space bound. We assume $S(n)$ is constructible (else try the method below with $S(n) = 0, 1,\ldots$).

Let $G = (\mathrm{POS}, \vdash)$. Given an initial position p_I of length n, we construct a set $\mathrm{POS}(p_I)$ of all positions reachable by moves of G from p_I and with space $\leqslant S(n)$. Since G has position size bound $S(n)$ there must be a constant c independent of n such that $|\mathrm{POS}(p_I)| \leqslant c^{S(n)}$.

We will also construct a sequence of mappings from $\mathrm{POS}(p_I)$ to {**true, false**}. Initially, let $l(p) = \mathbf{false}$ for each $p \in \mathrm{POS}(p_I)$. We then compute a new mapping $f(l)$ such that for each $p \in \mathrm{POS}(p_I)$,

$$\begin{aligned} f(l)(p) &= \mathbf{false} && \text{if } p \in W \wedge \mathrm{POS}_1, \\ &= \bigvee_{p \vdash p'} l(p') && \text{if } p \in \mathrm{POS}_1 - W, \\ &= \mathbf{true} && \text{if } p \in W \wedge \mathrm{POS}_0, \\ &= \bigwedge_{p \vdash p'} l(p') && \text{if } p \in \mathrm{POS}_0 - W. \end{aligned}$$

Let l^* be the mapping derived by repeatedly applying f to l until there is no change. This requires at most $|\mathrm{POS}(p_I)|$ iterations and $2^{O(S(n))}$ deterministic time per iteration, since we have assumed that the next-moves in all games are computable in linear space. Thus $2^{O(S(n))}$ total time is required. Then we can show there is a 1–1 correspondence between Markov strategies σ of player 1 and labelings l^* constructed by the above process. In particular, the positions mapped by l^* to **true** correspond to the positions appearing in winning plays induced by some such σ, and vice versa. Thus we can show $l^*(p_I) = \mathbf{true}$ iff player 1 has a winning Markov strategy for p_I. By Proposition 2.2, Markov strategies suffice.

Since any labeling l^* of Theorem 4.2 with $l^*(p_I) = \mathbf{true}$ corresponds to a winning Markov strategy whose plays are each of length $\leqslant 2^{O(S(n))}$, we have

COROLLARY 4.2. *If G is a game of perfect information with space bound $S(n) \geqslant \log n$ then G has time bound $2^{O(S(n))}$.*

4.2. *Eliminating Incomplete Information from a Game*

We now give a powerset construction for transforming a game $G = (\mathrm{POS}, \vdash)$ of incomplete information into a game $G^+ = (\mathrm{POS}^+, \vdash^+)$ of perfect information whose positions are *sets* of positions of G. (The construction is somewhat reminiscent of the subset construction in finite state automata.) Our decision algorithms will rely on this construction, which entails an exponential blow-up in space complexity. In Section 5.3, we show that, in the worst case, such a complexity blow-up must occur.

Fix some initial position $p_I \in \mathrm{POS}$. We will assume that the set of positions reachable by moves from p_I is finite. For each play prefix π of G we construct a position $P(\pi)$ of G^+ with common portion the set $\{\mathrm{last}(\pi') \mid \pi'$ is a play prefix with $\mathrm{vis}_1(\pi) = \mathrm{vis}_1(\pi')\}$. (This is the set of current possible positions after π, from player 1's point of view, by viewing only $\mathrm{vis}_1(\pi)$.) Let the private portions of $P(\pi)$ be *null* (thus G^+ is a game of perfect information) and let the next player to move in $P(\pi)$ be the same as in last(π). Note that if $\text{vis}_1(\pi) = \text{vis}_1(\pi')$, then the next player to move in last(π) is the same as in last(π'). Hence

$$P(\pi) = P(\pi') \qquad \text{iff} \quad \text{vis}_1(\pi) = \text{vis}_1(\pi').$$

We allow no next-move from $P(\pi) \in \text{POS}^+$ if it is player 1's turn to move and π is some play prefix of G with last(π) $\in W$. (Thus player 0 wins at $P(\pi)$ for any π which is winning for player 0.) Otherwise, we let $P(\pi) \vdash^+ P(\pi')$ be a move of G^+ if π, π' are play prefixes of G and π' is a child of π. (Thus, moves of G^+ from $P(\pi)$ simulate all possible moves of G from position last(π).) Fix $P(p_I)$ to be the initial position of G^+.

THEOREM 4.3. *Player* 1 *has a winning strategy in G from initial position p_I iff player* 1 *has a winning strategy in G^+ from $P(p_I)$.*

Proof. We establish a 1–1 correspondence between winning strategies of G and winning Markov strategies of G^+.

Case 1. Let σ be a winning strategy for player 1 in G. For each play prefix π^+ of G^+, where it is player 1's turn to move at last(π^+), let $\sigma^+(\pi^+) = \pi^+ P(\sigma(\pi))$ for any play prefix π of G such that $P(\pi) = \text{last}(\pi^+)$. σ^+ is now shown by contradiction to be a winning Markov strategy for G^+. Suppose π^+ is a play of G^+ induced from σ^+ but π^+ is not winning for player 1. Then there is a play π of G induced from σ, where $P(\pi) = \text{last}(\pi^+)$, and such that π is not winning for player 1. But this contradicts our assumption that σ is winning.

Case 2. On the other hand, let σ^+ be a winning strategy for player 1 in G^+. By Proposition 2.2, we can assume without loss of generality that σ^+ is a Markov strategy. For each play prefix π of G, where it is player 1's turn to move in last(π), let $\sigma(\pi)$ be a child of π such that $\sigma^+(\pi^+) = \pi^+ P(\sigma(\pi))$ for any play prefix π^+ of G^+ such that $P(\pi) = \text{last}(\pi^+)$. Again, σ can easily be shown by contradiction to be a winning strategy for G. ∎

Note that we do not yet have a space bound for G^+. Next we give a decision algorithm for G. We show our algorithm can be executed by an alternating machine whose computation game is essentially G^+ and whose space bound is $2^{O(S(n))}$.

ALGORITHM A.

Input a game $G = (\text{POS}, \vdash)$ of incomplete information, with initial position p_I.
$P \leftarrow \{p_I\}$
WHILE true **DO**
  $P' \leftarrow \{p' \mid p \vdash p', p \in P\}$
  $W(P) \leftarrow \{p \in P \mid p \vdash p' \text{ for no } p'\}$
  $V \leftarrow \{\text{vis}_1(p) \mid p \in P'\}$
  IF $P \subseteq \text{POS}_1$ **THEN**

```
        BEGIN
          COMMENT player 1's move
          IF W(P) ≠ ∅ THEN L1: REJECT
          ELSE L2: v ← an existentially chosen element of V
        END
      ELSE
        BEGIN
          COMMENT player 0's move with P ⊆ POS_0
          IF W(P) = P THEN L3: ACCEPT
          ELSE L4: v ← a universally chosen element of v
        END
      P ← {p' ∈ P' | vis_1(p') = v}
    OD
```

Intuitively, the algorithm tests for the existence of a winning strategy for player 1 by simulating all possible plays by all possible strategies simultaneously. Trial strategies are extended existentially, one step at a time. At each step all possible moves of player 2 are simulated to determine whether the strategy is adequate so far. If not, it is rejected; if so, it is continued to be extended. The invariant of the while loop is that P is a set of the form

$$\{\text{last}(\pi') \mid \text{vis}_1(\pi) = \text{vis}_1(\pi'), \pi' \text{ is a play prefix from } p_I\}$$

for some play prefix π from p_I in G. Thus, P is equivalent to the common portion of a position of the game G^+. This loop invariant also implies that either $P \subseteq \text{POS}_1$ or $P \subseteq \text{POS}_0$, since it can be determined from the visible portion of a position whose turn it is.

We have four conditions within the body of the while statement. In the case $L1$ is reached, it is player 1's turn and player 0 had a sequence of moves against this partial strategy that lead to a position $p \in \text{POS}_1 \wedge W$. In this case we must reject, since the partial strategy has been shown inadequate. At $L2$, it is player 1's move and he has a next-move from every possible position. In this case the strategy is extended existentially one move step in all possible ways. In the cases $L3$ or $L4$ are reached, player 0 has the initiative. At $L3$, he has no next-move, so this branch of the trial strategy is winning for player 1. At $L4$, player 0's next-move is chosen universally among all possible, reflecting the fact that any strategy of player 1 must fail them all.

Thus Algorithm A implements the game G^+ by use of an alternating machine. Algorithm A accepts exactly when there is a winning strategy for player 1, since the algorithm establishes a one-to-one correspondence between these winning strategies and finite accepting subtrees of the computation tree of the alternating machine.

Since there are no more than $2^{O(S(n))}$ positions of G reachable from $p_I, |P| \leqslant 2^{O(S(n))}$, so Algorithm A can be executed by an alternating machine with space bound $2^{O(S(n))}$.

THEOREM 4.4. *The outcome of any game G of incomplete information with space bound $S(n)$ can be decided by an alternating machine with space bound* $2^{O(S(n))}$.

By Theorems 4.2 and 4.4 we have

THEOREM 4.5. The outcome of any game G of incomplete information with space bound $S(n)$ can be decided in deterministic time $2^{2^{O(S(n))}}$.

4.3. *A Decision Algorithm for Blindfold Games*

We show here

THEOREM 4.6. *Any blindfold game G with space bound $S(n)$ can be decided in nondeterministic space* $2^{O(S(n))}$.

Proof. Let $G = (\text{POS}, \vdash)$ as in the proof of Theorem 4.4. Since the game is blindfold, the cardinality of V, in step $L4$ of Algorithm A, is always exactly 1.

Let Algorithm A be modified to A$'$ by substituting at step $L4$ "let v be the unique element in V." The resulting Algorithm A$'$ is obviously *nondeterministic* (since we utilize only existential choice). We claim that if G is blindfold, then Algorithm A$'$ accepts iff player 1 has a winning strategy. To see this, we simply observe that since the game is blindfold, the moves chosen by player 1 in its winning strategy are oblivious to any moves by player 0. ∎

4.4. *Games with Both Alternation and Space Bounds*

THEOREM 4.7. *For any game G of perfect information with alternation bounds $A(n)$ and space bound $S(n) \geqslant \log n$, the outcome of G can be decided in deterministic spaces $(A(n) + S(n))\, S(n)$.*

Proof. By Theorem 3.2 we can show that the outcome of G can be decided by an alternation machine M with alternation bound $A(n)$ and space bound $S(n)$. Borodin has shown (see Chandra, Kozen, and Stockmeyer [1]) that the acceptance problem for M can be decided in space $(A(n) + S(n))\, S(n)$. ∎

Now let G be a game of incomplete information with alternation bound $A(n)$ and space bound $S(n) \geqslant \log n$. Fix an initial position of length n. By Theorem 4.3, the game G^+ of perfect information has the same outcome as G, and by the proof of Theorem 4.4, G^+ is the computation game of an alternating machine with space bound $2^{O(S(n))}$. G^+ has the same alternation bound $A(n)$ as G. Thus by Theorem 4.7,

THEOREM 4.8. *For any game G of incomplete information with alternation bound $A(n)$ and space bound $S(n) \geqslant \log n$, the outcome of G can be decided in deterministic space* $(A(n) + 1)\, 2^{O(S(n))}$.

5. Lower Bounds on the Complexity of Space Bounded Games

To derive our lower bounds, we use the technique of encoding computations of a standard type of machine into one of our new types of machines. Then we can apply hierarchy results known for the standard type of machine, to obtain the desired lower bounds for our new types of machines.

5.1. *Lower Bounds on Games of Perfect Information*

This technique was utilized by Chandra, Kozen, and Stockmeyer [1] to obtain lower bounds for games of perfect information. They show

Theorem 5.1. *For each* $S(n) \geqslant \log n$,

$$ASPACE(S(n)) \supseteq DTIME(\mathrm{EXP}(S(n)))$$

(*see definition of* EXP *in the Introduction*).

By applying their version of Theorem 4.1, they have

Corollary 5.1. *For each* $S(n) \geqslant \log n$,

$$ASPACE(S(n)) = DTIME(\mathrm{EXP}(S(n))).$$

This is an elegant characterization of the power of space bounded alternation. We aim to derive such characterizations for private and blind alternations.

5.2. *Lower Bounds for Blindfold Games*

Theorem 5.2. *For each* $S(n) \geqslant \log n$,

$$BASPACE(S(n)) \supseteq NSPACE(\mathrm{EXP}(S(n))).$$

Proof. Let M be an N-TM with an input string $\omega \in \Sigma^n$. We assume M has a constructible space bound $c^{S(n)}$ for some constant $c > 0$. (If it is not constructible, we try the simulation below for $S(n) = 0, 1, \ldots$. If a player wins within the allotted space then the simulation halts, accepting if the existential player wins, and rejecting if the universal player wins. Otherwise if the space $2^{S(n)}$ is exceeded then the play restarts with the space $S(n)$ incremented by 1.) Let $\vdash$ be the next move relation of M. It will be useful to assume that for each position p of M, that is, neither accepting nor rejecting, there are exactly d next-moves (where d is a constant dependent only on M) $\vdash_1(p), \ldots, \vdash_d(p)$. We consider the configurations of M to be strings over a finite alphabet Δ. Let $D = \{1, \ldots, d\}$ be considered symbols disjoint from Δ and let $\Delta' = \Delta \cup D$.

We now construct a BA-TM M_1 with space bound $S(n)$. The players will alternate on each move. M_1 will require a unique state for each symbol in Δ'. Let the existential player of M_1 choose (by entering the appropriate states) a string of the form $p_0 r_1 p_1 r_2, ..., r_k p_k$, where $r_1, ..., r_k \in D$ and $p_0, ..., p_k \in \Delta^*$. Let the universal player of M_1 choose to privately (by use of a private tape) verify that one of the following conditions is violated:

(i) p_0 is the initial configuration of M,

(ii) p_k contains the accepting state of M, or

(iii) $p_i = \vdash_{r_i}(p_{i-1})$ for $i = 1, ..., k$.

Note that if (i), (ii), and (iii) all hold then the string chosen by the existential player of M_1 is an accepting computation (if the r_i are ignored). This is the goal of the existential player of M_1. The universal player of M_1 is trying to verify that the string chosen by the existential player is *not* an accepting computation.

(Note that it is essential that the universal player of M_1 *privately* choose to verify (i), (ii), or (iii), or otherwise the existential player of M_1 could "cheat" by observing which of (i), (ii), or (iii) are tested and then varying the choice of string $p_0 r_1 p_1 r_2, ..., r_k p_k$ so that not all of (i), (ii), and (iii) hold for any choice of the string.)

To verify (i) is violated, the universal player of M_1 may utilize $\log n$ cells of a private tape for a pointer to symbols of the input string ω. It is trivial to verify the case (ii) is violated. For the case (iii) it is useful to define for each $r \in D$, a function $F_r : \Delta' \times \Delta' \times \Delta' \times \Delta' \to \Delta'$, such that for each $a_{-1} a_0 a_1 a_2 \in \Delta'$, if $a_0 \in D$ then $F_r(a_{-1}, a_0, a_1, a_2) = a_0$ and otherwise if $a_{-1} a_0 a_1 a_2$ are the $j-1, j, j+1, j+2$ symbols of string $r'p'rp$ then $F_r(a_{-1} a_0 a_1 a_2)$ is the jth symbol of the string rp, where $p = \vdash_r(p')$ for configurations p, p', and $r' \in D$. (Thus F_r checks that p follows correctly from p' on taking the rth transition.)

To verify (iii) is violated, let the universal player choose to store on a private tape $a_{-1} a_0 a_1 a_2$ which are the $j-1, j, j+1, j+2$ symbols of $r_{i-1} p_{i-1} r_i p_i$ for some i, $1 \leqslant i \leqslant k$, and some j, $1 \leqslant j \leqslant \text{length}(p_i)$. The universal player must then test that $F_{r_i}(a_{-1} a_0 a_1 a_2)$ is the jth symbol of the string $r_i p_i r_{i+1}$. (Note that the universal player just privately guesses when to start checking during some point during the play of the game and so the *BA*-TM does not have to write down i.) The total space cost is thus $S(n)$, since $j \leqslant 2^{O(S(n))}$. We let M_1 accept only if the universal player cannot verify either (i), (ii), or (iii) has been violated. Thus M_1 accepts iff there exists an accepting computation $p_0 p_1, ..., p_k$ of M. Clearly M_1 is blindfold since the moves of the existential players are completely oblivious to the move of the universal players. ∎

By combining Theorems 4.6 and 5.2 we have

COROLLARY 5.2. *For each* $S(n) \geqslant \log n$,

$$BASPACE(S(n)) = NSPACE(\text{EXP}(S(n))).$$

5.3. *Lower Bounds for Games of Incomplete Information*

The reader may inquire: did the proof of Theorem 5.2 utilize the full power of private alternating machines? Indeed, it did not, since the simulation game was blindfold. The following theorem uses a similar construction, but also employs the dynamic interaction between the existential and universal player possible in general games of incomplete information

THEOREM 5.3. *For each* $S(n) \geqslant \log n$,

$$PASPACE(S(n)) \supseteq ASPACE(\mathrm{EXP}(S(n))).$$

Proof. Let M be an A-TM with input string $\omega \in \Sigma_n$. We assume M has constructible space bound $c^{S(n)}$, if for some constant $c \geqslant 1$ (otherwise try $S(n) = 0, 1, \ldots,$ as described in the proof of Theorem 5.2). Let Δ, Δ', and F be defined just as in Theorem 5.2. The proof is similar, however, here we construct in deterministic $\log n$ space a PA-TM M_2 with space bound $S(n)$ which accepts iff M accepts.

We will require again a unique state of M_2 for each symbol of Δ'; all other states will be associated with a null symbol. We also again let the player alternate on each move. The players will choose (by entering the appropriate states) a string of the form $p_0 r_1 p_1 r_2, \ldots, r_k p_k$, where $r_1, \ldots, r_k \in D$ and $p_0, \ldots, p_k \in \Delta^*$. All these symbols will be chosen by the existential player, except that if p_{i-1} contains a universal state, then the universal player publically chooses $r_i \in D$ by writing r_i on a public tape (this has the effect of creating d branches on the game tree, since the subsequent choice of p_i by the existential player may be very dependent on observation of the universal player's choice of r_i). Again we require the universal player to privately (by use of a private tape) attempt to verify that one of the cases (i), (ii), or (iii) is violated.

Note that if the cases (i), (ii), or (iii) hold for each choice of the r_i's then the existential player of M_2 has chosen a set of string which (if the r_i symbols are ignored) are an accepting subtree (i.e., these strings are the accepting computation sequences induced by a winning strategy for the existential player in the game G^M). This is the goal of the existential player of M_2, and we let M_2 accept if this goal is achieved. Otherwise, if the universal player finds a violation of (i), (ii), or (iii), then M_2 rejects. ∎

Combining Theorems 4.5, 5.1, and 5.3 we have

COROLLARY 5.3. *For each* $S(n) \geqslant \log n$,

$$\begin{aligned} PASPACE(S(n)) &= ASPACE(\mathrm{EXP}(S(n))) \\ &= DTIME(\mathrm{EXP}(\mathrm{EXP}(S(n)))). \end{aligned}$$

As a consequence of Corollary 3.4, and the results of this section, we have

(1) a space n bounded PA-TM M whose computation game G^M is universal for all reasonable games.

(2) a space n bounded BA-TM M' whose computation game $G^{M'}$ is universal for all reasonable blindfold games.

By the hierarchy theorem for deterministic time complexity (Hartmanis and Stearns [5] we have

COROLLARY 5.4. *There is a $c > 1$ such that if any D-TM decides the outcome of G^M in time $T(n)$, then $T(n) > 2^{cn/\log n}$.*

By space hierarchy results,

COROLLARY 5.5. *There is a $c > 1$ such that if any D-TM decides the outcome of $G^{M'}$ in space $S(n)$, then $S(n) > c^{n/\log n}$.*

6. TIME BOUNDED BLIND AND PRIVATE ALTERNATING MACHINES

Let $\Sigma_{A(n)}^{T(n)}$ be the class of languages accepted by alternating machines with time bound $T(n)$, with alternation bound $A(n)$, and existential initial state. We now characterize the time complexity of blind and private alternating machines in terms of the time complexity of alternating machines.

THEOREM 6.2. *For each $T(n)$ such that* $\inf_{n\to\infty} T(n)/n = \infty$,

$$BATIME(T(n)) = \Sigma_2^{T(n)}.$$

Proof. Let M be a BA-TM with time bound $T(n) \geqslant n$ and input string $\omega \in \Sigma^n$. Since the existential player of M is oblivious to any move by the universal player of M, it might just as well have chosen its moves at the start of the computation, and stored them into a consecutive sequence of tape cells. By Theorem 3.3, this can be done in time $T(n)/2$ if we augment the tape alphabet so that each pair of moves of the existential player is represented by a distinct symbol. Next, we let the universal player choose all its moves and attempt to verify the resulting play is not accepting. By Theorem 3.3 this can also be done in time $T(n)/2$ using the augmented tape alphabet. Thus the resulting machine M' has time bound $T(n)$ and accepts just the strings accepted by M. Note that the moves of the existential player of M' precede all the moves of the universal player. Thus, all portions of positions of the universal player can be considered common, so M' is an alternating machine. Thus we have shown $BATIME(T(n)) \subseteq \Sigma_2^{T(n)}$. (Note that this simulation is not particularly space efficient since M' may now require at least space $T(n)/2$.)

To show $\Sigma_2^{T(n)} \subseteq BATIME(T(n))$, we first observe that if M_1 is an A-TM, where all the moves of the existential player precede all moves of the universal player then the existential player is oblivious to any subsequent moves of the universal player. If M_1 has time bound $T(n)$, then since it has only one alternation, M_1 can be speeded up by a factor of two to $T(n)/2$ without introducing any further alternations. Let M_2 be the BA-TM derived from M_1 by introducing a new private tape for each original tape on which the universal player did any writing or head movement operations. Each tape operation of the existential player must be simulated, in the next succeeding step, by the universal player on these new private tapes. This slows the simulation time by a factor of two, down to time $T(n)$, and introduces $T(n)/2$ alternations. The resulting blind alternating machine M_2 accepts just the strings accepted by M_1. ∎

The following results were first given in Peterson and Reif [9] in a more general context of multiplayer games.

THEOREM 6.3. *For any* $T(n)$, *such that* $\inf_{n\to\infty} T(n)/n = \infty$,

$$PATIME(T(n)) = ATIME(T(n)).$$

Proof. First observe that any A-TM is a PA-TM, so $PATIME(T(n))$ contains $ATIME(T(n))$. On the other hand, let M be a PA-TM with time bound $T(n)$ and input string $\omega \in \Sigma^n$. We can assume a constant d bounding the maximum number of common portions of positions possible from a single position of M.

We require a set Γ' of $d+1$ special new tape symbols for M', one for each set of next-moves of M which are indistinguishable to the existential player, and also one distinguished symbol designating a "pass" move. We construct an A-TM M' which simulates M in two stages. In the first stage of the simulation, the existential and universal players alternatively write symbols of Γ' on consecutive cells of a new tape of M'. The existential player is allowed to terminate this stage at any time. In the next stage, the universal player attempts to verify that there is some play π of M from the initial position and consistent with previously chosen moves, such that π is not winning for the existential player. If so, the machine M' rejects, and otherwise M' accepts. The total time for these two phases is $3T(n)$, but this can be speeded up to $T(n)$ by Theorem 3.3. ∎

7. Universal Games on Propositional Formulas

In this section we construct various propositional formula games which are universal for reasonable games. These games and the reductions between them are generalizations of work on games of perfect information in Stockmeyer and Chandra [18].

Boolean variables take on values 1, 0 representing **true**, **false**, respectively. Let a *literal* be a boolean variable or its negation. Let a propositional formula F be in k-*conjunctive* (*disjunctive*) *normal form* if F consists of a conjunction (disjunction) of formulas $F_1, F_2, \ldots, F_j$ with each F_i a disjunction (conjunction, respectively) of at most k literals.

We now consider games on propositional formulas which we show are universal for all reasonable games.

Let $\mathbf{G}^1$ be the game in which a position contains a propositional formula $F(X, Y^C, Y^{P0}, Y^{P1}, a, s)$ in 5-conjunctive normal form, with X^C, Y^{P0}, Y^{P1} each sequences of variables and a, s individual variables, and also a truth assignment to its variables. The formula F and the truth assignment to the variables of X, Y^C, a, s are common to both players 1 and 0, but the truth assignment to the variables of Y^{P0}, Y^{P1} are private to player 0.

Player 1 moves by setting a to 1 and choosing a new truth assignment for the variables of X. Player 0 moves by (a) setting a to 0, (b) setting s to the complement of its previous truth assignement, and (c) then choosing a new truth assignment for the variables of Y^C, Y^{Ps}. The formula F is not modified by these moves, except for the changes in the truth assignment to its variables. The loser is the first player whose move yields a truth assignment for which the formula F is **false**.

LEMMA 7.1. *G^1 is universal for reasonable games of incomplete information.*

Proof. Let M be a PA-TM with space bound n. Let $\omega \in \Sigma^n$ be an input string to M. We encode each position of G^M as a bit vector of length $n' = O(n)$ (where the constant multiple depends only on the size of the tape alphabet of M), so that bits $1, 2, \ldots, k$ are those of $\mathrm{vis}_1(p)$ (the portions of p common to both the existential and universal players), and the bits $k+1, \ldots, n'$ contain those portions of p private to the universal player.

Using the techniques of Stockmeyer [16], we may construct a linear size propositional formula $\mathrm{NEXT}(Z_1, Z_2, T)$, where Z_1, Z_2, T are sequences of variables each of length n' and such that: if Z_1 encodes (by some fixed encoding which is computable in $O(\log n)$ space by a D-TM) a position p_1 then there exists an assignment to the variables of T such that $\mathrm{NEXT}(Z_1, Z_2, T)$ is **true** if and only if Z_2 encodes some position p_2 derived from p_1 by a move of M.

We introduce new sequences of variables X, Y^C, Y^{P0}, Y^{P1} of length m, m, l, l, where $m = k + n'$ and $l = n' - k$. Let $Y = Y^C Y^{P0} Y^{P1}$. Let $X[i, j]$ denote $X(i), X(i+1), \ldots, X(j)$ for any $1 \leqslant i \leqslant j \leqslant m$.

For distinct $s, \bar{s} \in \{0, 1\}$, let $\mathrm{NEXT}_{0,s}(X, Y)$ be the formula derived from $\mathrm{NEXT}(Z_1, Z_2, T)$ by substituting $X[1, k]$, $Y^{P\bar{s}}[1, l]$ for Z_1, substituting $Y^C[1, k]$, $Y^{Ps}[1, l]$ for Z_2, and substituting $Y^C[k+1, m]$ for T. Also, let $\mathrm{NEXT}_{l,s}(X, Y)$ be derived from $\mathrm{NEXT}(Z_1, Z_2, T)$ by substituting $Y^C[1, k]$, $Y^{Ps}[1, l]$ for Z_1, substituting $X[1, k]$, $Y^{Ps}[1, l]$ for Z_2, and substituting $X[k+1, m]$ for T. As usual we consider player 1 to be identified with the existential player of M and player 0 to be identified with the universal player of M. Without loss of generality, we assume the players move in strictly alternating order, and the first player to move is existential. Then for each $a \in \{0, 1\}$, $\mathrm{NEXT}_{a,s}$ defines legal moves by player a on switch variable $s \in \{0, 1\}$.

Now we consider the formula

$$\begin{aligned} F(X, Y^C, Y^{P0}, Y^{P1}, a, s) = {} & (a \wedge s \rightarrow \mathrm{NEXT}_{1,1}(X, Y)) \wedge (a \wedge \neg s \rightarrow \mathrm{NEXT}_{1,0}(X, Y)) \\ & \wedge (\neg a \wedge s \rightarrow \mathrm{NEXT}_{0,1}(X, Y)) \\ & \wedge (\neg a \wedge \neg s \rightarrow \mathrm{NEXT}_{0,0}(X, Y)). \end{aligned}$$

F can easily be put in 5-conjunctive normal form of size $O(n)$ and is constructable in $O(\log n)$ space by D-TM.

Let $P_I(\omega)$ be the initial configuration of M on input ω. Initially let $s = a = 1$. Also, initially let the variables $Y^C[1, k]\ Y^{P1}[1, l]$ be assigned to encode $P_I(\omega)$ and let all other variables be assigned arbitrarily. Let formula F and this initial truth assignment be the initial position of game G^1. The player 1 wins game G^1 if and only if player 1 (the existential player) wins the computation game G^M if and only if M accepts input ω. Thus we have a log-space reduction from the acceptance problem for M to the outcome problem for G^1. By Corollary 3.4, G^M is universal for universal games, so we conclude that G^1 is universal for reasonable games. ∎

Let $\mathbf{G}^2$ be the game in which each position contains formulas $\mathrm{WIN}_1(U, V^C, V^P)$ and $\mathrm{WIN}_0(U, V^C, V^P)$ in disjunction normal form and truth assignments to the sequences of variables of U, V^C, V^P.

The formulas WIN_1 and WIN_0 and truth assignments to variables $U \cup V^C$ are viewed commonly by both players, but the truth assignment to the variables of V^P are private to player 0. Player 1 moves by changing the truth assignment to at most one variable of U, while player 0 moves by changing at most one variable of V^C, V^P. Player $a \in \{0, 1\}$ *wins* if formula WIN_a is **true** after a move by player a.

THEOREM 7.1. *G^2 is universal for reasonable games of incomplete information.*

Proof. We now introduce sequences of variables U^A, U^B, V^A, V^B of length $m' = 4m + 2l + 4$. Let $U = U^A \cdot U^B$ and let $V = V^A \cdot V^B$. The values of the sequences of variables X, Y defined in the previous construction will, in legal plays of our game G^2, be contained in U, V as in Figs. $4_{1,0}, 4_{0,1}, 4_{1,1}, 4_{0,0}$. The private portion V^P of V has the value of Y^{P0}, Y^{P1} and V^C contains the values of other elements of V.

For each $s \in \{0, 1\}$ and player $a \in \{0, 1\}$, let $\mathrm{NEXT}'_{a,s}(U, V)$ be the formula derived from formula $\mathrm{NEXT}_{a,s}(X, Y)$ by substituting variables as in Fig. $4_{a,s}$.

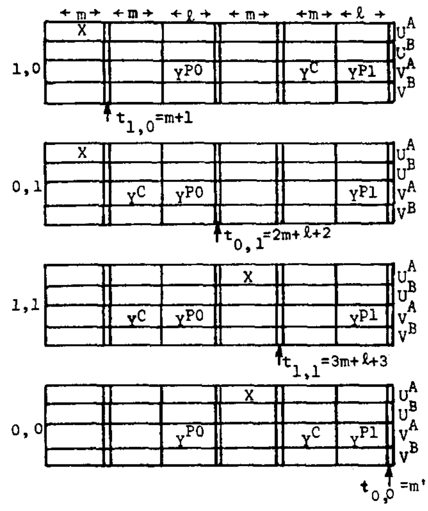

FIGURE 4

We describe a *legal play* such that if players 1 and 0 play legally then player 1 wins if and only if M accepts input ω. Let a *legal cycle* be a play which satisfies the following *restriction* L for $i = 1, 2,..., m'$:

player 0 changes the truth assignment of either $V^A(i)$ or $V^B(i)$.

player 1 changes the truth assignment of either $U^A(i)$ or $U^B(i)$.

within the legal cycle we also require *restriction* L$'$ to hold: for distinct $s, \bar{s} \in \{0, 1\}$ and each i, $(t_{1,\bar{s}} \bmod m') < i \leqslant t_{0,s}$, player 0 assigns variables so that $\mathrm{NEXT}_{0,s} = \mathbf{true}$ when $i = t_{0,s}$, and for $(t_{0,s} \bmod m') < i \leqslant t_{1,s}$, player 1 assigns variables so that $\mathrm{NEXT}_{1,s} = \mathbf{true}$ when $i = t_{1,s}$. Thus M accepts input ω if and only if player 1 has a winning strategy within legal players satisfying restrictions L and L'. The following construction forces legal play by both players.

We now introduce some notation for operations on sequences of Z, Z' of boolean variables of length m'. Let $\oplus$ be the boolean exclusive-or operative and let

$$Z \oplus' Z' = (Z(1) \oplus Z'(1),..., Z(n) \oplus Z'(n)),$$

and let

$$\Delta Z = (\neg(Z(n) \oplus Z(1)), Z(1) \oplus Z(2),..., Z(n-1) \oplus Z(n)).$$

Also let $\text{TH-TWO}(Z) = \vee_{1 \leqslant i < j \leqslant m'}(Z(i) \wedge Z(j))$ be the threshold-two function. For simplicity of notation we define formulas $U' = \Delta(U^A \oplus' U^B)$, and $V' = \Delta(V^A \oplus' V^B)$, which are sequences which locate boundaries between contiguous 0's or 1's. To detect illegal play we define

$$\mathrm{ILL}_1 = \text{TH-TWO}(U') \vee \bigvee_{1 \leqslant i \leqslant m'} (U'(i) \wedge V'(i+1) \wedge \neg V'(i-1))$$

$$\mathrm{ILL}_0 = \text{TH-TWO}(V') \vee \bigvee_{1 \leqslant i \leqslant m'} (V'(i) \wedge U'(i+2) \wedge \neg U'(i)).$$

Thus $\mathrm{ILL}_a = \mathbf{true}$ just if player $a \in \{0, 1\}$ has violated restriction L for a legal cycle.

For each player $a \in \{0, 1\}$, let

$$\mathrm{ILL}'_a = \bigvee_{s \in \{0,1\}} (U'(t_{a,s}) \wedge V'(t_{a,s}) \wedge \neg \mathrm{NEXT}'_{a,s}(U, V)).$$

$\mathrm{ILL}'_a = \mathbf{true}$ just if restriction L' has been violated by player a. Finally let $\mathrm{WIN}_0 = \mathrm{ILL}_1 \vee \mathrm{ILL}'_1$ and $\mathrm{WIN}_1 = \mathrm{ILL}_0 \vee \mathrm{ILL}'_0$. Formulae WIN_0 and WIN_1 can easily be put in disjunctive normal form of size $O(n^2)$ and can be coded into binary strings of length $O(n^2 \log n)$.

Given input $\omega \in \Sigma^n$, let p_1 be the initial position of formula game G^1 defined previously. Let the initial position p_2 of formula game G^2 contain formulas $\mathrm{WIN}_0, \mathrm{WIN}_1$ as defined above with the initial truth assignment of p_1 as in Fig. $4_{1,1}$ and $U' = V' = (1, 0, 0,..., 0)$ initially. It can be shown player 1 wins game G^2 from initial position p_2 if and only if M accepts ω. Thus by Corollary 3.4, G^2 is also a formula game universal for all reasonable games. ∎

Let G^{2B} be the blindfold game derived from formula game G^2 by requiring that the common variable sequence V^C be empty.

THEOREM 7.2. *G^{2B} is universal for all reasonable blindfold games.*

Proof. To show this, we need only note that if M is restricted to BA-TM, then the universal player can never modify the common tape. Hence the common variables V^C in our previous construction contain no information relevant to a configuration of M though they are useful to insure legal play. Hence the variables V^C used in the construction may be added to the variables V^P private to player 0. The result then follows from our proof of Theorem 7.2. ∎

Note. The game G^2 is essentially identical to the game PRIVATE-PEEK defined in the Introduction. The variables can be put in 1–1 corresponding with the plates in PRIVATE-PEEK game box. Furthermore, the variables of V^P correspond to the plates not visible to player 1. The clauses of WIN_0 and WIN_1 be put in 1–1 correspondence with locations of holes which perforate the plates so that player $a \in \{0, 1\}$ can peek through from the top to the bottom of the box iff a clause of WIN_a is satisfied.

Also, the formula game G^{2B} is essentially the game BLIND-PEEK described in the introductory section. Thus we conclude by Theorem 7.1 and 7.2,

(1) PRIVATE-PEEK is a universal reasonable game.

(2) BLIND-PEEK is a universal reasonable blindfold game.

Our log-space reduction from the computation game G^M to the game G^2 has an $O(n \log n)$ length bound. Thus by Corollary 5.4 and Theorem 7.1,

COROLLARY 7.1. *There is a $c > 1$ such that if a D-TM decides the outcome of G^2 or PRIVATE-PEEK in time $T(n)$, then*

$$T(n) > 2^{c^{\sqrt{n/\log^3 n}}}.$$

Also by Corollary 5.5 and Theorem 7.2,

COROLLARY 7.2. *There is a $c > 1$ such that if a D-TM decides the outcome of G^{2B} or BLIND-PEEK in space $S(n)$, then $S(n) > c^{\sqrt{n/\log^3 n}}$.*

8. CONCLUSION

This paper has considered the computational complexity of two player games of incomplete information. Our general conclusion is that if the space is bounded by $S(n)$, then their outcome is an exponential more difficult to decide than for games of perfect information with space bound $S(n)$. Because of our lower bounds, our decision algorithms for games of incomplete information are asymptotically optimal.

It would be interesting to extend our results for the game of PRIVATE-PEEK to prove other games, such as "blindfold chess" are universal for all reasonable games of incomplete information. The complexity of blindfold pursuit games on digraphs were considered in an early draft of this paper (Reif [11]).

It is also interesting to note that our technique of introducing private storage to an alternating machine, resulting in a PA-TM, could also be applied to any other basic parallel machine type, such as a parallel RAM. In that case each processor might have a private set of registers.

In Peterson and Reif [9] we investigate the complexity of *multiple* player games of incomplete information. Our general conclusions for multiperson games with a position size bound $S(n)$ are:

(1) if the division of private information is not restricted, then the outcome problem is undecidable even for 3 player games;

(2) however, the multiplayer games are decidable if the private information is hierarchically divided among the players; and each additional player increases the complexity of the outcome problem by a further exponential.

Reif and Peterson [13] also gave decision algorithms for various classes of multiperson games of incomplete information. Peterson [10] applied the complexity results of Peterson and Reif [9] to succinctness of string representation.

Applications of multiplayer games of incomplete information to distributed multiprocessing problems and a related multiprocess logic are described in Reif and Peterson [12].

Acknowledgments

Penelope first interested me in certain mouse vs. feline pursuit games considered in a previous draft of this paper. Paul Spirakis pointed out to me the importance of non-Markov strategies in games of incomplete information. The referees made useful suggestions which significantly improved this paper. Evan Cohn made a number of comments on a later draft of this paper.

References

1. A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer, "Alternation," Research Report RC 7489, IBM, Yorktown Heights, N.Y., Jan., 1978.
2. S. Even and R. E. Tarjan, A combinatorial problem which is complete in polynomial space, *in* "Proceedings, 7th Annual ACM Sympos. on Theory of Computing, May, 1976, Hershey," pp. 41–49.
3. A. S. Fraenkel, M. R. Garey, D. S. Johnson, T. J. Schaefer, and Y. Yesha, The complexity of checkers on an $N \times N$ board—Preliminary report, *in* "Proceedings, 19th IEEE Sympos. on Foundations of Computer Science, Oct. 1978," pp. 55–64.
4. A. S. Fraenkel and D. Lichtenstein, Computing a perfect strategy for $n \times n$ chess requires time exponential in N, "Proceedings, 8th Internat. Colloq. Automata. Lang. & Programming, Acre, Israel, July 1981," pp. 278–293.

5. J. Hartmanis and R. E. Stearns, On the computational complexity of algorithms, *Trans. Amer. Math. Soc.* **117** (1965), 285–306.
6. N. D. Jones, Blindfold games are harder than games with perfect information, *Bull. European Assoc. for Theoret. Comput. Sci.*, **6** (1978), 4–7.
7. N. D. Jones, Space-bounded reducibility among combinatorial problems, *J. Comput. System Sci.* **11** (1975), 68–85.
8. D. Lichtenstein and M. Sipser, Go is Pspace hard, *J. Assoc. Comput. Mach.* **27** (1980), 393–401.
9. G. L. Peterson and J. H. Reif, Multiple person alternation, *in* "Proceedings, 20th IEEE Sympos. on Foundations of Computer Science, October 1979."
10. G. L. Peterson, Succinct representation of random strings, and complexity classes, *in* "Proceedings, 21st IEEE Sympos. on Foundations of Computer Science, Oct. 1980," pp. 86–95.
11. J. H. Reif, Universal two person games of incomplete information, *in* "Proceedings, 11th Annual ACM Sympos. for Theory of Computing, 1979."
12. J. H. Reif and G. L. Petersen, A dynamic logic of multiprocessing with incomplete information, *in* "Proceedings, 7th Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Jan. 1980," pp. 193–202.
13. J. H. Reif and G. L. Peterson, "Decision Algorithms for Multiplayer Games of Incomplete Information," TR34-81, Aiken Computation Lab., Harvard University, Cambridge, Mass., 1981.
14. W. J. Savitch, Relationships between nondeterministic and deterministic tape complexities, *J. Comput. Sci.* **4** (1970), 177–192.
15. T. J. Schaefer, Complexity of some two person perfect-information games, *J. Comput. System Sci.* **16**, No. 2, (1978), 188–225.
16. L. J. Stockmeyer, The polynomial-time hierarchy, *Theoret. Comput. Sci.* **3** (1977), 1–22.
17. L. J. Stockmeyer and A. R. Meyer, Word problems requiring exponential time: Preliminary report, *in* "Proceedings 5th ACM Sympos. on Theory of Computing, 1973," pp. 1–9.
18. L. J. Stockmeyer and A. K. Chandra, Provably difficult combinatorial games, *SIAM J. Comput.* **8**, No. 2 (1979), 151–174.

Printed in Belgium