BLINDFOLD GAMES ARE HARDER THAN GAMES WITH PERFECT INFORMATION

by

Neil D. Jones

DAIMI PB-80
June 1977

Institute of Mathematics University of Aarhus
DEPARTMENT OF COMPUTER SCIENCE
Ny Munkegade - 8000 Aarhus C - Denmark
Phone 06-12 83 55

BLINDFOLD GAMES ARE HARDER THAN GAMES WITH PERFECT INFORMATION

Neil D. Jones

Computer Science Department
University of Aarhus, Denmark
(on leave from The University of
Kansas, Lawrence, Kansas 66045)

Recently several researchers ([5], [4], [6], [3]) have shown an interest in the complexity of determining the existence of winning strategies in various games. The purpose of this note is to show that this problem is (probably) much more difficult for games in which the players lack perfect information about the state of the game. Familiar examples of games of this type include Kriegspiel (blindfold chess) and Battleship. In particular, we show that a simple one token game on graphs requires polynomial space to analyze in its blindfold version, although polynomial time is known to be sufficient for the version with perfect information.

It was shown in [5] that the problem of determining the existence of a winning strategy in a one-token game is log-space complete for P (see [1] for definitions of P, PSPACE, etc.). The game there may be naturally represented in terms of finite automata as follows. Let $M = (S, \Sigma, \delta, q_0, F_1)$ be a finite automaton, and let $F_2 \subseteq S$. We interpret F_1 and F_2 as the sets of winning states, for players 1 and 2 respectively. Initially a token is placed on state q_0. The players move alternately, each in his turn choosing a symbol a from Σ. If the token is on state p, it is then moved to state $\delta(p, a)$.

Research partially supported by NSF grant number

A player wins by causing the token to be moved to one of his winning states. At each step both players know which state the token occupies, so that this is a game with perfect information.

A "blindfold" version of the game is obtained by requiring that the selection of symbols from Σ be done without knowledge of the current position of the token. Thus a strategy for one player is simply a sequence $a_1 \ldots a_n \in \Sigma^*$. This strategy wins for player 1 if $\delta(q_0, a_1 b_1 a_2 \ldots b_{n-1} a_n) \in F_1$ or $\delta(q_0, a_1 b_1 \ldots a_n b_n) \in F_1$ for all $b_1, \ldots, b_n \in \Sigma$;

It wins for player 2 if $\delta(q_0, b_1 a_1 \ldots b_n a_n) \in F_2$ or $\delta(q_0, b_1 a_1 \ldots b_{n-1} a_n b_n) \in F_2$ for all $b_1, \ldots, b_n \in \Sigma$.

Let BF be the set of all instances of this game in which the first player has a winning strategy. We now show that BF is PSPACE-hard. This is done by a reduction from the known PSPACE-complete set

$$\{ R \mid R \text{ is a regular expression over } \{0,1\} \text{ and } L(R) \neq \{0,1\}^* \}$$

Given a regular expression R_1 an equivalent nondeterministic finite automaton $M^R = (K, \{0,1\}, \tau, q_0, F^R)$ may be built by standard methods (e.g. [1]) with size polynomially related to the size of R. Now consider the game corresponding to the <u>deterministic</u> finite automaton

$$M = (K \cup K \times \{0,1\} \cup \{\$, ¢\} ; K \cup \{0,1\}, \delta, q_0, \{\$\} \cup K - F^R)$$

where $F_2 = \{ ¢ \}$

$\delta(p, a) = \langle p, a \rangle$	if $p \in K$, $a \in \{0,1\}$
$\delta(p, a) = ¢$	if $p, a \in K$
$\delta(\langle p, a \rangle, q) = q$	if $p \in K$, $a \in \{0,1\}$ and $q \in \tau(p,a)$
$\delta(¢, a) = ¢$	if $a \in K \cup \{0,1\}$
$\delta(p, a) = \$$	for any other $p \in K \times \{0,1\} \cup \{\$, ¢\}$, $a \in K \cup \{0,1\}$

When it is player 1's move, the token will be on a state (call it p) of M^R: He chooses a symbol $a \in \{0,1\}$ and the token is moved to $\langle p, a \rangle$. Player 2 then chooses one of the states q in $\tau(p, a)$ and the token is moved to q. If player 1 chooses an $a \notin \{0,1\}$, or player 2 chooses a

$q \notin \tau(p, a)$ the token goes to ¢ or \$ respectively resulting in an immediate win for the other player.

Now suppose M is in BF, so player 1 has a winning strategy. A win can only occur by a move of player 2, so there must be a sequence $a_1, \ldots, a_n \in \{0, 1\}$ such that $\delta(q_0, a_1 q_1 \ldots a_n q_n) \in \{\$\} \cup K - F^R$ for all $q_1, \ldots, q_n \in K$.

Now let $q_1, \ldots, q_n$ now be any sequence of states of M^R such that $q_{i+1} \in \delta(q_i, a_{i+1})$ for $0 \leq i < n$. By definition of M_1 $\delta(q_0, a_1 q_1 \ldots a_n q_n) = q_n \in K - F^R$, so $q_0, q_1, \ldots, q_n$ is not a state sequence causing M^R to accept $a_1 \ldots a_n$. But $q_1, \ldots, q_n$ was arbitrary, so $a_1 \ldots a_n \notin L(M^R)$ and so $L(R) \neq \{0, 1\}^*$.

Conversely, $L(R) \neq \{0, 1\}^*$ implies M is in BF.

The reduction is clearly polynomial, so BF is PSPACE-hard.
It is not difficult to see that BF is also in PSPACE, and so complete.

It may be of interest from a game-playing point of view to note that BF remains PSPACE-complete even if restricted to the more practically interesting planar graphs. The reason is that by [2] we may assume that M^R is planar ; it is straightforward to build M so that it also is planar.

It is to be expected that exponential space will be required to analyze the complexity of blindfold versions of games whose perfect information versions have PSPACE complexity, such as generalized HEX [4].

REFERENCES

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

2. Book, R.V. and Chandra, A.K., Inherently nonplanar automata, Acta Informatica 6, 89-94 (1976)

3. Chandra, A.K. and Stockmeyer, L.J., Alternation, Proc. 17th Symp. Found. Comp. Sc., October 1976 98-108.

4. Evan, S. and Tarjan, R.E., A combinatorial problem which is complete in polynomial space, 7th Annual ACM Symp. on Theory of Computing, May 1975, 66-71.

5. Jones, N.D. and Laaser, W.T., Complete problems for deterministic polynomial time, 6th Annual ACM Symp. on Theory of Computing, May 1974, 40-46.

6. Schaefer, T.J., Complexity of decision problems based on two-person perfect information games, 8th Annual ACM Symp. on Theory of Computing, May 1976, 41-49.