JOURNAL OF COMBINATORIAL THEORY, Series A 31, 199–214 (1981)

Computing a Perfect Strategy for $n \times n$ Chess Requires Time Exponential in n

AVIEZRI S. FRAENKEL*

Department of Applied Mathematics,
The Weizmann Institute of Science, Rehovot, Israel

AND

DAVID LICHTENSTEIN[†]

Department of Computer Science, The University of California,
Berkeley, California 94720

Communicated by the Managing Editors

Received May 15, 1979

> 'Tis all a chequer-board of nights and days
> where destiny with men for pieces plays;
> hither and thither moves and mates and slays
> and one by one back in the closet lays.
>
> The *Rubaiyat* of Omar Khayyam

It is proved that a natural generalization of chess to an $n \times n$ board is complete in exponential time. This implies that there exist chess positions on an $n \times n$ chessboard for which the problem of determing who can win from that position requires an amount of time which is at least exponential in $\sqrt{n}$.

1. INTRODUCTION

Among all the games people play, chess towers as the most absorbing and widely played. Indeed, if attention is restricted to two-person games of perfect information without chance moves played outside the Orient, the ever

---

* Initial work done while visiting the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801.

[†] Present address: Department of Computer Science, Yale University, New Haven, Connecticut 06520.

0097-3165/81/050199–16$02.00/0

rejuvenating interest in the 1500 year old game has a quality of depth and breadth well beyond that of any potential rival. It is noteworthy, then, that in the long string of complexity results for games, chess had yet to appear. Recently J. Storer announced that chess on an $n \times n$ board is Pspace-hard [10]. See also J. M. Robson [7]. We will show that a natural generalization of chess to $n \times n$ boards is complete in exponential time, the first such result for a "real" game. This implies that there exists $d > 0$ and infinitely many positions π such that an algorithm for deciding whether White (Black) can win from that position requires at least $c^{|\pi|^d}$ time-steps to compute, where $c > 1$ is a constant, and $|\pi|$ is the size of π. Generalized chess is thus provably intractable, which is a stronger result than the complexity results for board games such as Checkers, Go, Gobang and Hex which were shown to be Pspace-hard [1, 3, 5, 6].

We let generalized chess be any game of a class of chess-type-games with one king per side played on an $n \times n$ chessboard. The pieces of every game in the class are subject to the same movement rules as in 8×8 chess, and the number of White and Black pawns, rooks, bishops and queens each increases as some fractional power of n. Beyond this growth condition, the initial position is immaterial, since we analyze the problem of winning for an arbitrary board position.

Unfortunately, our constructions seem to violate the spirit of 8×8 chess, in much the same way as the complexity proofs for Checkers, Go, and Gobang and Hex mentioned above. Typical positions in our reduction do not look like larger versions of typical 8×8 chess endgames. Although we have not tried to answer questions of reachability, it seems offhand as though players would have a hard time trying to reach our board positions from any reasonable starting position. (Reachability may not seem quite as unfeasible, perhaps, if we recall the chess rule stating that a pawn reaching the opposite side of the board can become any piece of the same color other than pawn or king [4].) What we can say, however, is that certain approaches for deciding whether a position in 8×8 chess is a winning position for White may not be very promising, namely, those aproaches which work for arbitrary positions and generalize to $n \times n$ boards. Such approaches use time exponential in n, and hence can be useful only if the exponential effect had not yet been felt for $n = 8$.

Thus, while we may have said very little if anything about 8×8 chess, we have, in fact, said as much about the complexity of deciding winning positions in chess as the tools of reduction and completeness in computational complexity allow us to say.

Our result is in line with the suggestion to demonstrate the complexity of interesting board games by imbedding them in families of games [8]. An interesting corollary of our result is that if Pspace $\neq$ Exptime, as the conjecture goes, then there is no polynomial bound on the number of moves necessary to execute a perfect strategy. This is so because Pspace $\subseteq$ Exptime, and the "game-tree" of chess can be traversed in endorder to determine the win–lose–tie membership of each node (game position). Though this takes an exponential amount of time, the memory requirement at each step is only the depth $p(n)$ of the tree—which is kept on a stack—and the description of a terminal position. Thus, if $p(n)$ is polynomial, then the game is in Pspace. Since chess is complete in Exptime, it belongs to the hardest problems there, hence it lies in Exptime – Pspace if Pspace $\neq$ Exptime.

For the sake of the uninitiated, we now give a short informal introduction to the basic notions of computational complexity. Let S be a subclass of decision problems (i.e., problems whose answer is "Yes" or "No"). For decision problems π_1, π_2, we say that π_1 is *polynomially transformable* (or *reducible*) to π_2 (notation: $\pi_1 \propto \pi_2$), if there exists a function f from the set of instances of π_1 to the set of instances of π_2 such that:

(i) I is an instance of π_1 for which the answer is "Yes" if and only if $f(I)$ is an instance of π_2 for which the answer is "Yes".

(ii) $f(I)$ is computable by a polynomial time algorithm in the size of I (a "polynomial time algorithm").

A decision problem π is *S-complete* if:

(i) $\pi \in S$,

(ii) for every $\pi' \in S$, $\pi' \propto \pi$.

A decision problem π is *S-hard* if (ii) holds but (i) does not necessarily hold. A decision problem is *intractable* if it cannot be decided by a polynomial time algorithm.

A *nondeterministic algorithm* is an "algorithm" which can "guess" an existential solution, such as a path in a tree and then verify its validity by means of a deterministic algorithm.

Important classes of decision problems are the class P of all decision problems π with (deterministic) algorithms whose running time is bounded above by a polynomial in the size $|\pi|$ of π; the class NP (nondeterministic polynomial) of all decision problems π with nondeterministic algorithms whose running time is bounded above by a polynomial in $|\pi|$; the class Pspace of all decision problems π whose algorithms require an amount of memory space bounded above by a polynomial in $|\pi|$; and the class Exptime of all decision problems π with (deterministic) algorithms whose running time is bounded above by an exponential function in $|\pi|$. The following basic relations hold:

$$P \subseteq NP \subseteq \text{Pspace} \subseteq \text{Exptime}.$$

It is not known whether any of these inclusions is proper, except that

$P \neq$ Exptime. Furthermore, NP and Pspace are not known to contain any intractable decision problems, but Exptime is.

From the definition of $\propto$ it follows that if $\pi_1 \propto \pi_2$, then $\pi_2 \in P$ implies $\pi_1 \in P$. Therefore the S-complete problems for any S are the "hardest" problems of S. In prticular for $S =$ Exptime, the S-complete problems are all intractable. For further details and a formal treatment of this topic the reader is referred to Garey and Johnson [2].

2. The Reduction

Let Q be the following question: Given an arbitrary position of a generalized chess game on an $n \times n$ chessboard from our class of chess games, can White (Black) win from that position? Following [2], we define Exptime to be the set of decision problems with time-complexity bounded above by $2^{p(n)}$ for some polynomial p of the input size n. Since in chess there are six distinct pieces of each color, the number of possible configurations in $n \times n$ chess is bounded above by 13^{n^2}, hence $Q \in$ Exptime. We shall show that $G_3 \propto Q$, where G_3 is the following Boolean game proved complete in exponential time by Stockmeyer and Chandra [9]. Throughtout, W (B) stands for White (Black). As usual, a *literal* is a Boolean variable or its complement.

Every position in G_3 is a 4-tuple (τ, W-LOSE(X, Y), B-LOSE(X, Y), a), where $\tau \in \{W, B\}$ denotes the player whose turn it is to play from the position, W-LOSE $= C_{11} \vee C_{12} \vee \cdots \vee C_{1p}$ and B-LOSE $= C_{21} \vee C_{22} \vee \cdots \vee C_{2q}$ are Boolean formulas in 12DNF, that is, each C_{1i} and each C_{2j} is a conjunction of at most 12 literals ($1 \leqslant i \leqslant p$, $1 \leqslant j \leqslant q$); and a is an assignment of values to the set of variables $X \cup Y$. The players play alternately. Player W (B) moves by changing the value of precisely one variable in X (Y). In particular, passing is not permitted. W (B) loses if the formula W-LOSE (B-LOSE) is true after some move of player W (B). Thus W can move from (W, W-LOSE, B-LOSE, a) to (B, W-LOSE, B-LOSE, a') iff B-LOSE is false under the assignment a (otherwise the game already terminated previously), and a and a' differ in the assignment of exactly one variable in X. If W-LOSE is true under the assignment a', then W just lost. A player who violates any of the game's rules loses immediately.

In order to show $G_3 \propto Q$, we have to simulate G_3 on an $n \times n$ chessboard. Specifically, the goal is to construct a position on the board where only one rook and two queens per variable can move. All other pieces are deadlocked. Each rook is permitted to be in only one of two positions, which have the meaning of assigning the values of 1 (T) or 0 (F) to the corresponding variable. The positioning of the deadlocked pieces forces the queens to move through predefined "channels" in order to reach the opponent's king, and the positioning of the rook determines one of two possible avenues through which a queen may pass. The overall construction is such that those and only those truth-assignments to the variables which win the game G_3 for *W* (*B*) lead the queens of *W* (*B*) to win the generalized chess game from the constructed position.

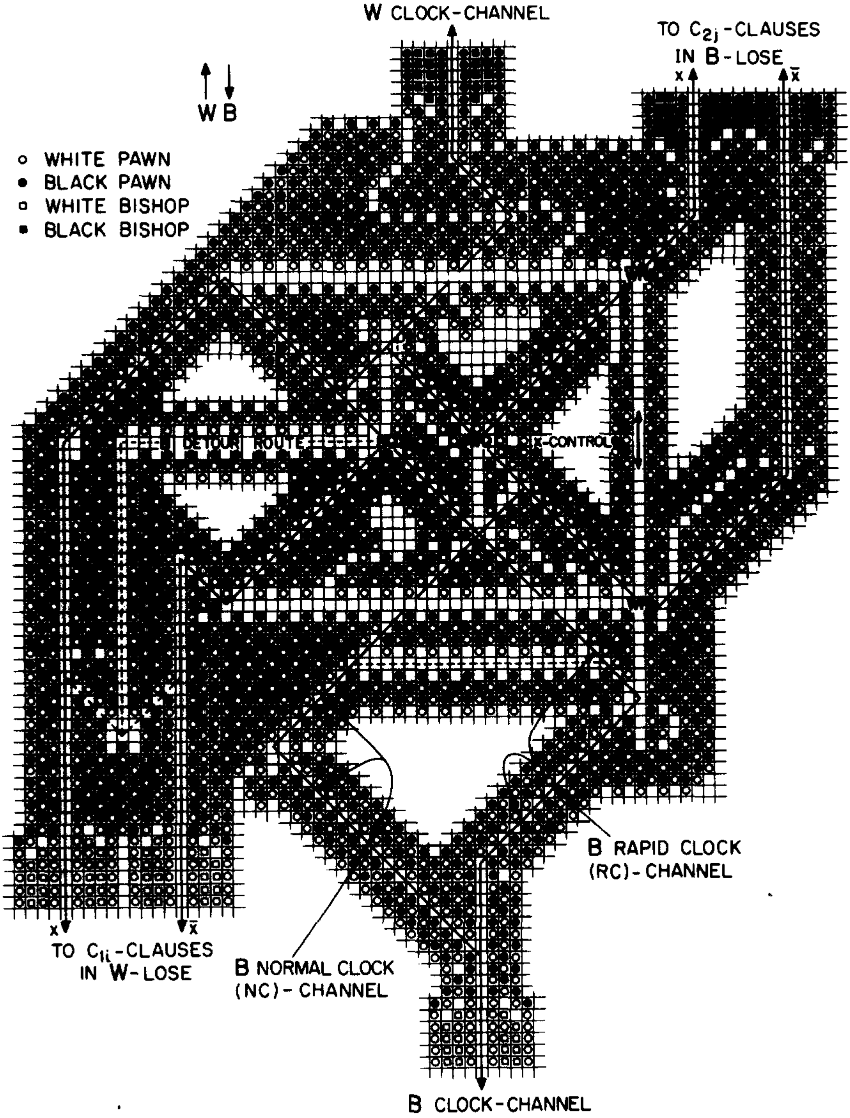

FIG. 1. White Boolean controller.

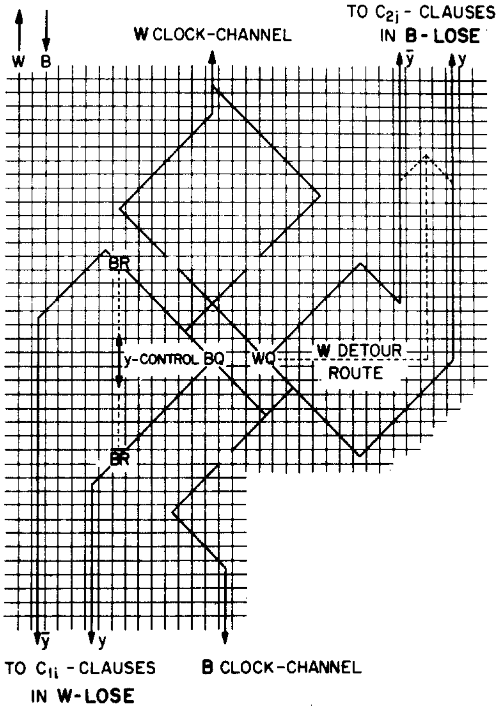

FIG. 2. Schema of black Boolean controller.

Our basic structure is the *Boolean controller.* Figure 1 (2) illustrates a *W* (*B*) Boolean controller for a variable $x \in X$ ($y \in Y$). White circles are *WP*'s (*W* pawns), black circles *BP*'s (*B* pawns), white square *WB*'s (*W* bishops), black squares *BB*'s (*B* bishops), and *WR*, *BR*, *WQ*, *BQ* stand for *W* rook, *B* rook, *W* queen, *B* queen, respectively. If *WR* is at its south position in the *WR*-channel, as in Fig. 1, also called *x-position*, then the value of x is 1. If *WR* is at the north position of the *WR*-channel, denoted by $\overline{WR}$ in Fig. 1, also called *$\bar{x}$-position*, then the value of x is 0. A similar convention is adopted for Fig. 2 which is indicated only schematically because a *B* Boolean Controller (*BBC*) is obtained from a *W* Boolean Controller (*WBC*) by an interchange $C_{1i} \leftrightarrow C_{2j}$, $x \leftrightarrow y$, $\bar{x} \leftrightarrow \bar{y}$ and $W \leftrightarrow B$ throughout, followed by a 180° rotation. (Here and below, C_{1i} (C_{2j}) denotes a typical clause of *W*-LOSE (*B*-LOSE).)

There is one *W* (*B*) Boolean Controller for each $x \in X$ ($y \in Y$). In normal play, *W* (*B*) moves his *WR* (*BR*) between the x-position and the $\bar{x}$-position (y- and $\bar{y}$-position) in any *W* (*B*) Boolean Controller until the game G_3 will have been decided. If *W* (*B*) does not abide by these rules, then his opponent can win via the *B* (*W*) Normal Clock (*NC*) or the *B* (*W*) Rapid Clock (*RC*) mechanisms detailed below.

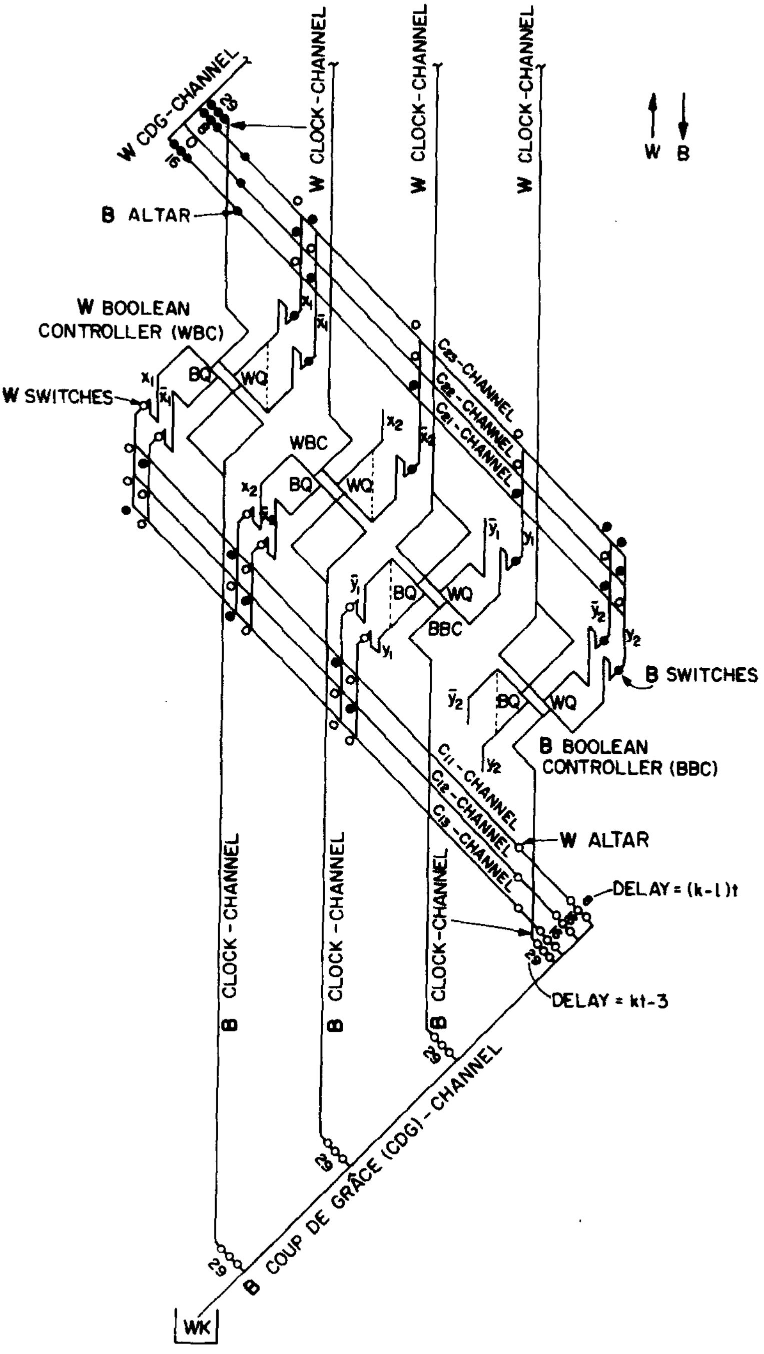

FIG. 3. Global view of the construction for the case W-LOSE $= C_{11} \vee C_{12} \vee C_{13}$, $C_{11} = \bar{x}_1 \wedge x_2 \wedge \bar{y}_1$, $C_{12} = \bar{x}_2 \wedge y_1$, $C_{13} = x_1 \wedge x_2$ B-LOSE $= C_{21} \vee C_{22} \vee C_{23}$, $C_{21} = x_1 \wedge y_2$, $C_{22} = \bar{x}_1 \wedge \bar{x}_2 \wedge y_1 \wedge \bar{y}_2$, $C_{23} = x_1 \wedge \bar{x}_2 \wedge y_1$.

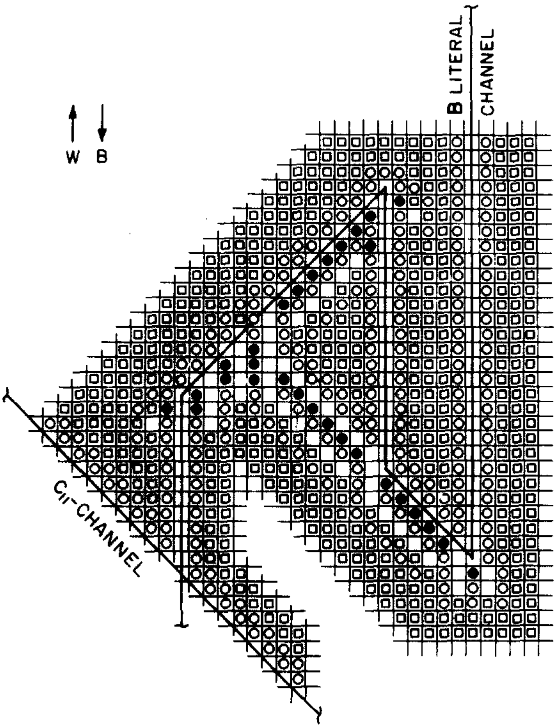

FIG. 4. One-way *W* Switch.

A global view of the construction is shown in Fig. 3. Let k be the largest number of literals in any "And-Clause" in *W*-LOSE and *B*-LOSE. Let C_{1l} in *W*-LOSE be an And-Clause consisting of l literals for some $1 \leqslant l \leqslant k \leqslant 12$. Suppose that $C_{1l} = 1$ after a move of *W*. Now $C_{1l} = 1$ if and only if there are l *B* queens which can reach C_{1l}-channel intersections not under attack in $t = 8$ moves each: two moves in the *WBC* (Fig. 1) or *BBC* (Fig. 2), one move for reaching the *W Switch* (Fig. 4), four moves in the *W* Switch and one last move for reaching the C_{1l}-channel. These l *B* queens now proceed down this channel, where $l - 1$ of them are captured at the *W Altar* (Fig. 5), and the lone survivor passes through a *W* delay-line from where it emerges into the *B Coup De Grâce* (*CDG*)-channel to checkmate the *W* king (*WK*) (Fig. 6).

The *W* (*B*) Switch (Fig. 4) is designed to let a single *B* (*W*) queen pass from a *W* or *B* Boolean Controller to the C_{1l} (C_{2j})-channels. When a *BQ* comes down a *WBC* or a *BBC* to an as yet untraversed *W* Switch, it captures the *WP* on the longer diagonal path and then proceeds down unperturbed to the C_{1l}-channels. If, however, a *BQ* attempts to pass the *W* Switch

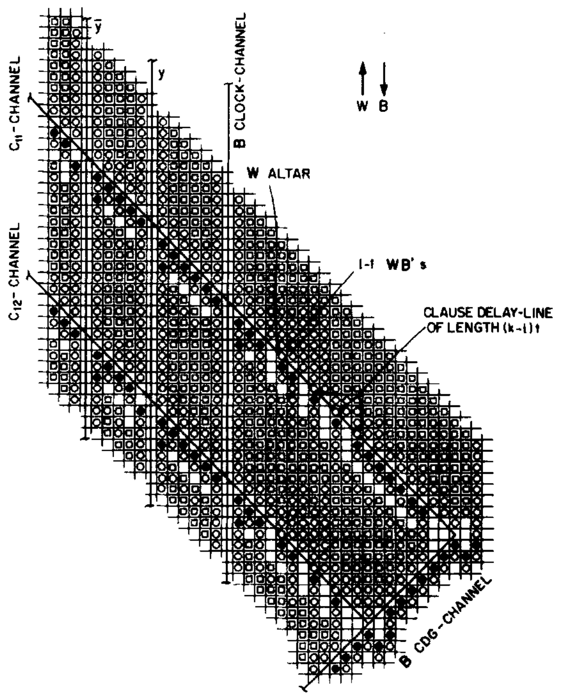

FIG. 5. *W* Channel crossings, *W* Altar and Clause-channel delay-lines.

in the opposite direction, whether previously traversed or untraversed, then, on reaching the northeast corner of the longer diagonal path, the *WP* just underneath the captured *WP* goes north by one square and thus opens up a line of more than *k* *WB*'s effectively covering the shorter diagonal path of the Switch, making it impassable.

The crossings of Clause-channels with a Clock-channel and two Literal-channels can be observed from the western part of Fig. 5. If $\bar{y} \in C_{11}$, $\bar{y} \notin C_{12}$, say, then a *BQ* coming down the $\bar{y}$-channel can stop unperturbed at the intersection—called *island*—with the C_{11}-channel. But if it tries to come to rest at the intersection with the C_{12}-channel, called *through-intersection*, then it is promptly captured by a *WP*. The situation is reversed for a *BQ* coming down the y-channel if, say, $y \notin C_{11}$, $y \in C_{12}$. On the other hand, a *BQ* coming down a Clock-channel cannot stop unattacked at any crossing with a C_{1i}-channel; all its intersections with Clause-channels are through-intersections.

We remark that if a literal is not used in *W*-LOSE (*B*-LOSE), its channel is truncated prior to reaching the *W* (*B*) Switch (Fig. 3).

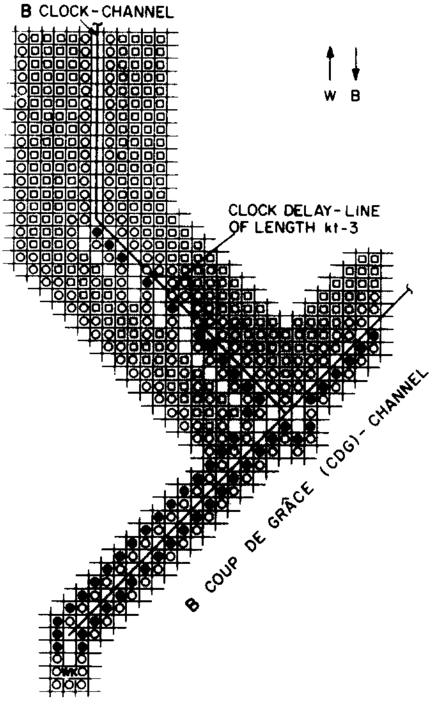

FIG. 6. Lower part of *B* Clock-channel with Clock delay-line impinging on *B* *CDG*-channel.

Every channel-segment has length at least $U \equiv 2(k(t+1)+2)$, and the shields around each channel, including truncated ones, also have thickness at least U. The reason for this will become clear later. (In the figures, some segments seem short and some shields thin, which is the result of emphasizing the main features at the expense of the secondary ones. But it should be kept in mind that the true length of segments and thickness of shields is at least U throughout.)

3. THE WINNING SCENARIO

As was mentioned above, if C_{1l} contains l literals and $C_{1l} = 1$ following a move of *W*, then there are l *BQ*'s each of which can reach the C_{1l}-channel in $t = 8$ moves. The strategy of *B* is to first move all l *BQ*'s into the C_{1l}-channel and then to move each of them as far down the C_{1l}-channel towards the *B* *CDG*-channel as *W* permits. The first *BQ* to pass has to capture the *WP* located at the *W* Altar which is backed up by a line containing precisely $l-1$ *WB*'s (Fig. 5). Thus *W* will capture j of the *BQ*'s for some $0 \leqslant j < l$. Then the $(j+1)$-th *BQ* captures a *W* piece at the *W* Altar after $lt + j + 1$

moves: each of the l BQ's requires t moves to reach the C_{1i}-channel and $j+1$ of them make one capture move each. After the $(j+1)$th BQ captures a W piece at the Altar, it spends $(k-l)\,t$ moves in a W delay-line consisting of $(k-l)\,t$ WP's. Two additional moves are spent for reaching and riding the B CDG-channel. Using this strategy, B thus requires $lt+j+1+(k-l)\,t+2=kt+j+3$ moves for checkmating the WK.

Following the departure of the first BQ from its vantage point on some Boolean Controller towards a C_{1i}-channel, the WQ on the same Boolean Controller can enter the W Clock-channel. Each Clock-channel contains a delay-line of kt-3 moves (Fig. 6). Since W also captures j BQ's in the C_{1i}-channel and there are six additional moves for entering and leaving the W Clock-channel and riding the W CDG-channel, W can checkmate the BK (B king) after $kt+j+3$ moves. Thus B wins with a margin of one move. Since $j<l\leqslant k$, B can in fact checkmate the WK in at most $k(t+1)+2$ $(=U/2)$ moves. Every other move of W, from among the limited moves available to him, is also doomed to failure. This is shown in the next section.

If, after W's move which made $C_{1i}=1$, W switches his WR between the x-position and the $\bar{x}$-position on some WBC, thus possibly unsatisfying W-LOSE, B can still select the values satisfying W-LOSE by using the B *Detour Route* (Fig. 1). This requires an additional move of B, but since also W lost one move in his extra WR switching maneuver, the move balance between B and W is preserved, and B can still win.

Now suppose that B starts to move BQ's towards some C_{1i}-channels before the game G_3 has been decided. B's only chance to win is to transfer at least l BQ's to some C_{1i}-channel if clause C_{1i} comprises l literals, since this is the only way a BQ can enter the B CDG-channel. W's response is as follows:

(i) Whenever a BQ in a WBC advances to its first station towards an x ($\bar{x}$)-channel while WR is in the $\bar{x}$ (x)-position, then WR captures BQ.

(ii) The first time a BQ moves—either from a WBC or a BBC—without passing a WR-threatened position, W activates its W Clock in the same Boolean Controller, then captures BQ's whenever possible, either as per (i) above or in the C_{1i}-channels as detailed below, otherwise proceeding down the W Clock-channel.

(iii) Whenever a BQ stops at a through-intersection, it is captured by a WP, subsequently by a WB. Whenever a BQ comes to a W Altar, it is captured by a WB.

Thus B must in fact transfer l BQ's to *islands* of the C_{1i}-channel. Note that the rth BQ requires t'_r moves to reach any such island, where $t'_r=t$ or $t+1$. Since $C_{1i}=0$, we have $t'_r=t+1$ for some r. Now B spends l moves in the BQ–WB battles at the W Altar, $(k-l)\,t$ moves in the channel delay-line and two moves for reaching and riding the *B* *CDG*-channel. Thus *B* requires at least $\sum_{r=1}^{l} t_r' + (k-l)t + l + 2 \geqslant lt + 1 + (k-l)t + l + 2 = kt + l + 3$ moves to checkmate the *WK*. Now *W* spends $l-1$ moves in capturing *BQ*'s and $kt+3$ moves in the *W* Clock and *W* *CDG*-channels. Thus *W* can checkmate the *BK* in $kt+l+2$ moves, fewer moves than *B* needs, and so *W* wins.

4. "ILLEGAL" MOVES

The above analysis—except the last part—was based on the assumption that the players do in fact simulate G_3. We call a move "illegal" if it is a legal move in generalized chess, but is either not part of the simulation of G_3 altogether, or is part but is taken at the wrong time for a proper simulation of G_3. Below we consider the nonobvious "illegal" moves.

I. *The WBC*

There are only six pieces that can move: *WR*, *WQ*, *BQ*, two *BP*'s and one *WP* (Fig. 1).

A. *Moves of WR*. (i) Suppose that while the game G_3 is still undecided, *WR* leaves the *WR*-channel from its normal x or $\bar{x}$-position, going east or west. (This has the bizarre effect of making both $x=1$ and $\bar{x}=1$ as far as *B*-LOSE is concerned, but leaving x unchanged in *W*-LOSE.)

If *WR* stops in the line of sight of *BQ*, then *BQ* captures *WR*. The timing, as is easy to verify, is such that even if *WR*'s move made *B*-LOSE true, *B* can now win via the *B* *RC*-channel except that if *WQ* moved to the x-position after *BQ* captured *WR*, then *BQ* has to back up to the *B* *NC*/*RC*-channel intersection and win via the *B* *NC*-channel. If *WR* stops elsewhere, then *BQ* goes directly to the *WR*/*B* *RC*-channel intersection and wins via the *B* *RC*-channel.

(ii) Suppose that while G_3 is still undecided, *WR* stops within the *WR*-channel at some location other than the x or $\bar{x}$-position. (This has the effect of making $x=1$ and $\bar{x}=1$ in both *B*-LOSE and *W*-LOSE.) If this location is the intersection with the *B* *RC*-channel, then *BQ* captures *WR* and wins again via the *B* *RC*-channel. Otherwise a *BP* captures *WR*. If now *W* moves his queen to the x-position, then *BQ* goes to the *B* *NC*/*RC*-channel intersection and then wins via the *B* *NC*-channel (even if *B*-LOSE is now true). Otherwise *BQ* can again win via the *B* *RC*-channel.

B. *Moves of WQ*. (i) Suppose that while G_3 is still undecided, *WQ* moves northwest to the intersection with the *W* Clock-channel. Then *BQ* will capture *WQ*, since otherwise *W* can win via its Clock mechanism. Even if *W* now makes *B*-LOSE true, *B* can win by moving southeast to the intersection with the *B* *NC*-channel and then proceeding down this channel.

(ii) Suppose that *WQ* moves as in (i) in some *WBC R*, but the move is made after *W*-LOSE has been made true previously by *W*. If *BQ* in *R* is required for winning, *B* moves it out towards the C_{1i}-channels. Otherwise *B* continues with his normal winning strategy, ignoring *W*'s move altogether.

(iii) Suppose that while G_3 is still undecided, *WQ* moves down vertically. If it comes to rest at the *B NC*/*RC*-channel intersection, *B* will capture it with his *BQ* which will subsequently proceed down the *B NC*-channel and win. Otherwise *WQ* is captured by a *BP*. Even if *W* now makes *B*-LOSE true, *B* can win with his *BQ* via the *B NC*-channel.

(iv) Suppose that *WQ* moves as in (iii), but the move is made after *W*-LOSE has previously been made true by *W*. Then *B*'s strategy is essentially the same as in (ii), so we omit it.

(v) Once *BQ* has left a *WBC*, *WQ* can neither pass through the Literal-channels in *W*-LOSE nor through the *B* Clock-channel, because of the *BP*'s defending the channel corners. An attempt by *WQ* to advance in parallel to some of these channel segments from the outside, by gnawing its way along the shielding *WP*'s and then slipping in at a suitable corner, is simply ignored by *B*, since the length of each channel-segment is at least *U*, which is about twice as long as it takes *B* to win. Also *WQ* cannot skip from channel to channel by penetrating through channel-shields, since these have thickness at least *U*.

(vi) Suppose that after *W*-LOSE has previously been made true by *W*, and *WR* is in the $\bar{x}$-position, *WQ* moves to the x-position in some *WBC R*. If *BQ* in *R* is required for winning, *B* will now move it towards the C_{1i}-channels via the *B* Detour Route. Otherwise *B* continues with his normal winning strategy.

If under the same assumption *WR* is in the x-position and *WQ* advances towards the $\bar{x}$-position by capturing the *BP* just southwest of the $\bar{x}$-position, then provided *BQ* of *R* is required for winning, *BQ* moves out towards the C_{1i}-channels via the x-channel. If *BQ* is not required for winning, *W*'s move is ignored as before.

C. *Moves of BQ*. The moves (*B*i)–(*B*v) have obvious counterparts for *BQ* in a *WBC* and move (*B*vi) has a counterpart in a *BBC*, so we omit the details. Only in the counterpart of (Bii) a slightly new situation may arise: Suppose that *BQ* moved to the *B NC*/*RC*-channel intersection and *WQ* then advanced towards the $\bar{x}$-position—since *WQ* is required for winning—first capturing the *BP* just southwest of the $\bar{x}$-position. If *BQ* now moves to the original position of *WQ*, then *WQ* captures *BQ* and then continues down the x-channel towards the C_{2j}-channels. Otherwise *WQ* continues directly down the x-channel. A similar situation can arise in the counterpart of (*B*iv), which *W* handles also in the way just described.

Suppose that *BQ* advances to its first station towards an x ($\bar{x}$)-channel while *WR* is in the $\bar{x}$ (x)-position. The case where this is done before G_3 has been decided was dealt with at the end of the previous section. If *BQ* makes a move of this type after *B* made *B*-LOSE true, it is ignored by *W*, who continues with his normal winning strategy.

D. *Moves of the pawns.* (i) Suppose that while G_3 has not yet been decided, the *BP* just west of the *B NC*/*RC*-channel intersection or the *BP* two squares north of it, moves south. Then *WQ* goes northwest to a point one square southeast of the *W* Clock intersection (call this square *K*). *W* can now win via his Clock since *B* loses one move on account of blocking the entrance to the *B* Clock-channel with his own *BP*.

(ii) Suppose that while G_3 has not yet been decided, the *WP* just south of *K* moves north onto *K*. Then *BQ* moves southeast to the middle of the first leg of the *B RC*-channel, from where it can win by going west to the *B NC*-channel.

II. *Preventing Backlash*

Suppose that *B*, either before G_3 has been decided or after it has been decided in *W*'s favor, assembles a squadron of *BQ*'s in the C_{1i}-channels in an attempt to break back into some *B* Clock-channels or into some Literal-channels, with the aim of reaching the C_{2j}-channels via some Boolean Controllers. If *B* succeeds in capturing even one of the *WQ*'s needed for a normal winning strategy of *W*, the game's outcome is not clear anymore.

Now *W* commences executing his normal winning strategy at the latest one move after the first *BQ* is moved towards the C_{1i}-channels. Assume first that *B* attempts to break back via some *B* Clock-channels. *B* needs $t+1$ moves to place a *BQ* at a C_{1i}/B Clock-channel intersection, which is a through-intersection. Then *W* will capture *BQ* there. After *B* moved $k+1$ *BQ*'s to such through-intersections and *W* captured them (the first with a *WP*, subsequent ones with *WB*'s, see Fig. 5), *B* spent $(k+1)(t+1)$ moves; and *W* spent $(k+1)\,t$ moves pursuing his normal winning strategy and $k+1$ moves capturing *BQ*'s at their prospective backlash points. Since shields have thickness at least $U > k+1$, *W* has a sufficient supply of bishops to do the latter. (Note that in Fig. 5 the true distance between the three vertical channels is much larger than shown.) It is thus seen that in at most $k-t+2 \leqslant 6$ additional moves *W* wins. If *B* attempts to break back via some Literal-channels, then it again takes $t+1$ moves to place a *BQ* at a C_{1i}/Literal-channel intersection, which may be an island. At least three additional moves are made by *BQ* before it is captured by a *WB* in a *W* Switch. Thus a fortiori *W* wins by pursuing his normal winning strategy and capturing (at most $k+1$) *BQ*'s which try to break back.

5. POLYNOMIALITY OF TRANSFORMATION

Recall our earlier notation: p (q) is the number of And-Clauses in W-LOSE (B-LOSE) and $m = |X| + |Y|$. The subscripts i of the literals x_i and y_i are encoded in binary. Therefore the length of W-LOSE (B-LOSE) has magnitude about $12p \log p$ ($12q \log q$), and the input size is thus $O((p+q)\log(pq))$. Clearly $m \leqslant 12(p+q)$.

For each variable our construction requires a constant amount of chess-pieces: The Boolean Controller, four Literal-channels, two Clock-channels and four Switches associated with a variable require a constant amount of chess-pieces since each channel-segment has length $O(k(t+1))$ which is a constant, and the shields around each channel also have thickness $O(k(t+1))$. Thus the sequence of m Boolean Controllers, oriented in a general northwest to southeast direction (Fig. 3), has length $O(m) = O(p+q)$. Therefore also the Clause-channels and CDG-channels have length $O(p+q)$ each. The total thickness of the Clause-channels with their shields is also $O(p+q)$. It follows that the construction can be realized on a square board of side $n = O(p+q)$, and so the transformation is polynomial.

Note. If we provide Switches in the Clock-channels in addition to those in the Literal-channels, we can replace the bishop shields around the Clause-channels by pawn shields. The Switches themselves can be redesigned so that they can operate without bishops. If, in addition, we back up the Altars by queens instead of bishops, it seems possible to avoid using bishops altogether. This leads to the possibility that $n \times n$ German checkers ("Dame") can be proved Exptime-complete by a method similar to the above proof. (In German checkers a piece reaching the opposite side of the board essentially becomes a queen rather than a king. We are told that this is the rule also for the version of the game as played in the USSR.) Of course also other board games (such as $n \times n$ Go) may be Exptime-complete.

ACKNOWLEDGMENTS

We are much indebted to J. M. Robson for putting his finger on a number of weak spots in earlier drafts of the paper. We also wish to thank the referee for his constructive criticism and comments.

REFERENCES

1. A. S. FRAENKEL, M. R. GAREY, D. S. JOHNSON, T. SCHAEFER, AND Y. YESHA, The complexity of checkers on an $n \times n$ board—preliminary report, *in* "Proceedings, 19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Mich., October 1978," pp. 55–64, IEEE Computer Society, Long Beach, Calif., 1978.

2. M. R. GAREY AND D. S. JOHNSON, "Computers and Intractability: A Guide to the Theory of *NP*-Completeness," Freeman, San Francisco, 1979.
3. D. LICHTENSTEIN AND M. SIPSER, *Go* is polynomial-space hard, *J. Assoc. Comput. Mach.* **27** (1980), 393–401; also appeared in "Proceedings, 19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Mich., October 1978," pp. 48–54, IEEE Computer Society, Long Beach, Calif., 1978.
4. M. E. MORRISON (Ed.), "Official Rules of Chess," 2nd ed., David McKay, New York, 1978.
5. S. REISCH, Gobang ist PSPACE-vollständig, *Acta Inform.* **13** (1980), 59–66.
6. S. REISCH, Hex ist PSPACE-vollständig, *Acta Inform.* **15** (1981), 167–191.
7. J. M. ROBSON, "*N* by *N* Chess is Pspace-Hard," TR-CS-80-09, Computer Science Dept., Australian National University, 1980.
8. H. SAMELSON (Ed.), Queries, No. 4 (iii), *Notices Amer. Math. Soc.* **24** (1977), 190–191.
9. L. J. STOCKMEYER AND A. K. CHANDRA, Provably difficult combinatorial games, *SIAM J. Comput.* **8** (1979), 151–174.
10. J. STORER, A note on the complexity of chess, *in* "Proceedings 1979 Conference on Information Sciences and Systems," pp. 160–166, Dept. of Electr. Eng., Johns Hopkins University, Baltimore.