Annals of Pure and Applied Logic 65 (1993) 149–184
North-Holland



# Infinite games played on finite graphs

Robert McNaughton

*Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA*

Communicated by A. Nerode
Received 1 December 1992
Revised 9 March 1993

*Abstract*

McNaughton, R., Infinite games played on finite graphs, Annals of Pure and Applied Logic 65 (1993) 149–184.

The concept of an infinite game played on a finite graph is perhaps novel in the context of an rather extensive recent literature in which infinite games are generally played on an infinite game tree. We claim two advantages for our model, which is admittedly more restrictive. First, our games have a more apparent resemblance to ordinary parlor games in spite of their infinite duration. Second, by distinguishing those nodes of the graph that determine the winning and losing of the game (winning-condition nodes), we are able to offer a complexity analysis that is useful in computer science applications.

## 1. Introduction

This paper[1] investigates certain games of infinite duration that have only finitely many configurations. Before we begin to discuss these games, which are games of perfect information, let us review the well known finite-duration games of perfect information. The two players in any such game move in turn each with full knowledge of everything relevant to the game that has occurred previously; the play of the game and its outcome are determined completely by the moves of the players. There are many such games; *tic-tac-toe, checkers, chess* and *go* are four examples. Excluded are all games of cards in which the players do not show their hands, all games in which opposing players move simultaneously (neither one knowing the other's move until after both moves are completed) and all games in which the play is partly or wholly determined by chance events.

Typically a game of perfect information is played on a board, where the players move by moving pieces about according to certain rules. It simplifies our thinking about these games to think of all of them as board games. For example, we can

[1] Written with support from the National Science Foundation, Grant Number CCR-9114725.

think of *tic-tac-toe* as played with tokens and a three-by-three board rather than with pencil and paper.

For our purposes the concept of configuration is an important one in the study of a game. We must make it precise enough so that it includes all information about any situations of the game relevant to the way the game continues, including all strategic considerations. In particular, it must tell which of the two players is to make the next move. The concept of *configuration* in a board game is characterized completely by saying the following: configuration $C_1$ is different from configuration $C_2$ if and only if either one player has the move in $C_1$ and the opponent has the move in $C_2$, or else there is at least one piece that is in one location in $C_1$ but in another location in $C_2$.

A board game has finitely many configurations, although any game worthy of serious devotion has a large enough number of configurations so that people cannot play the game with strategies based on exhaustive search. Thus *tic-tac-toe* is not worthy of serious devotion, but each of our other three examples is. Indeed in each of these games the number of configurations seems to be large enough so that present-day computers could not do an exhaustive search of the game tree in the normal course of computer usuage.

But the number of configurations is always finite, which means that any board game can theoretically be converted to a finite directed graph, each node of which represents a configuration of the game. The result of this conversion we call a finite graph game.

More precisely, a *finite graph game* has a finite bipartite directed graph whose set $Q$ of nodes is partitioned into two sets: $R$, the set of nodes from which Red moves, and $B$, the set from which Black moves. The game has a placemarker which is moved from node to node along the directed edges to mark the progress of the play. When the placemarker is on a node of $R$, Red moves by moving the placemarker along one of the edges to a node of $B$, whereupon it is Black's turn to move. Black moves in a similar way, always leaving the placemarker on a node of $R$.

The graph has a starting node $q_0$ from which the play always starts and two disjoint sets of winning-condition nodes $W_R$ and $W_B$. If the placemarker ever lands at a node of $W_R$ or $W_B$ then the game ends with a win for Red or Black, respectively.

In many board games, and hence in many graph games, a play can end in a draw, e.g., whenever the placemarker reaches a node outside of $W_R \cup W_B$ having no outgoing edges, or whenever the play gets into a recognizable loop. However, there do exist board games and graph games in which no play can end in a draw. It is not difficult to see that in any such graph game (and hence in any such board game) one of the two players has a winning strategy. We shall not prove this well known fact; but we shall prove a similar and slightly more difficult fact as Theorem 3.3 in Section 3.

All graphs in this paper are finite. The word 'finite' in the term 'finite graph

game' does not refer to the finiteness of the graph, but to the fact that any play of the game that is not a draw is of finite duration. This paper will study *infinite graph games,* that is, games of infinite duration played on finite graphs. Winning and losing in these games will depend on the *permset* of the infinite play, i.e., the set of nodes visited infinitely often by the placemarker. Technical definitions will be given in the next section.

Playing to infinity is certainly a mathematical fiction. The real phenomena we have in mind are game-like situations that go on for an indefinite period of time. Instead of an outcome that occurs once at some termination point, there are certain desirable and undesirable things that happen continually. If certain combinations of things happen predominantly over a period of time then one player wins, whereas other combinations make the opponent the winner. The notion of infinite duration is simply a mathematical idealization of what in our ordinary thinking is an indefinite duration.

An example is the running of a business, where the objective is to remain in business and maintain a profit. Ordinarily there is no point at which the business manager wins and often there is no point at which there is a decisive loss. We might depict this enterprise as an infinite game, which the business manager (playing against the world) wins if and only if there are infinitely many periods when the business shows a substantial profit and only finitely many periods when it shows a substantial loss.

The study of infinite games has application to Computer Science, which at this date seems to be its main reason for existence. Most theories of computation have been concerned with computation as an act that produces a result upon termination of the computation itself; all computations that do not terminate are put into the same category, namely, failure. This conception is justified when user programs are the object of study, since in most computational situations users are willing to leave the scene when they get their answers.

But the operating system of a multi-user computer needs another kind of theory of computation. When one user is satisfied there are generally several other programs still running, and more that are about to start. For such an operating system, computation is an on-going process, ideally without termination, like the running of a business. Generally speaking, the application of infinite games to theoretical computer science is to process-oriented theory rather than to problem-oriented theory.

The paper [10] proposes that programmers writing concurrent programs think of themselves as playing games with a computer. Because the action of the computer is nondeterministic (or appears so to the programmer), it should be regarded as a 'devilish opponent trying its level best to execute the program in such a way that the program specification is violated'. Since programmers write for runs that are unpredictably long, it is appropriate for the games that describe this process to be of infinite duration.

At present we have nothing more to say, beyond these mere suggestions, about

possible applications of infinite games. We admit that these two suggestions are superficial in that neither the running of a business nor the work of a computer operating system is wholly a game of perfect information.

We shall define the concept of *infinite graph game* precisely in Section 2, prove some preliminary theorems in Section 3 and then state and prove our main theorem in Section 4, to the effect that in any infinite graph game one player has a winning strategy of a certain kind. The proof of the main theorem is constructive, giving us an algorithm that determines the winner and the winning strategy. Section 5 discusses the computational complexity of that algorithm.

Section 6 focuses on an interesting subclass of the class of strategies and gives sufficient condition on a game for it to have a winning strategy in the subclass. Between Section 2 and Section 6 the notion of infinite duration is never questioned, but in Section 7 a partial answer is given to the important question of how a winner in an infinite graph game might be declared after a finite amount of time.

Finally, in Sections 8 and 9 we relate our work to other papers in the literature. In Section 8 we compare our work to what we call abstract infinite games, as introduced in 1953 by Gale and Stewart [4]. In Section 9 we trace the history of our notion of infinite graph game; briefly, it originated as a question about formal logic in a 1960 paper by Büchi [1], was developed by Landweber [7, 8], and was put forth in a fully and overtly published form in a 1969 paper [2] by both authors, with a proof of something like our main theorem. Our theorem and its proof are based on a 1982 paper by Gurevich and Harrington [6], aided by a 1990 follow-up paper by Yakhnis and Yakhnis [12].

## 2. The concept

In most of the remainder of this paper (namely, through Section 7) the word 'game' when it occurs without a modifier will mean an infinite graph game. Formally, a *game* $\mathcal{G}$ is an ordered sextuple $(Q, R, B, E, W, \Omega)$, where $Q$ is the set of nodes of a bipartite directed graph, $R$ and $B$ are the subsets from which Red and Black, respectively, may move, $E$ is the set of directed edges of the graph, $W \subseteq Q$ is the subset of winning-condition nodes and $\Omega$ the set of winning subsets of $W$ (in a sense to be described). We stipulate that

1. $R \cup B = Q \neq \emptyset$ and $R \cap B = \emptyset$,
2. for each $e \in E$ there exist $r \in R$ and $b \in B$ such that either $e = (r, b)$ or $e = (b, r)$,
3. for each $p \in Q$ there is at least one $q \in Q$ such that $(p, q) \in E$,
4. $W \subseteq Q$, and
5. $\Omega \subseteq 2^W$.

At any time of a play of the game the placemarker for the game is on a node; if that node is in $R$ then it is Red's turn to move; if in $B$ then it is Black's turn. The

player moves by moving the placemarker along one of the edges in the indicated direction from that node to another node, where it is then the other player's turn by Condition 2. By Condition 3, each player always has at least one possible move to make. The players move in turn forever.

We follow the usual practice in the game theory literature of reserving the word 'game' to mean the rules of the game; for our purposes a game is completely specified when $Q$, $R$, $B$, $E$, $W$ and $\Omega$ are specified. A *play* of a game is a particular infinite history in which the two players move according to the rules of that game.

We define the *permset* of any play of the game to be the subset of the nodes of $W$ that are visited by the placemarker infinitely often during the play. If, for any play of the game, the permset of that play is a member of $\Omega$ then Black wins that play; if not then Red wins. (Note that a game with $W = \emptyset$ is trivial: if $\Omega = \{\emptyset\}$, Black always wins, regardless of the play; if $\Omega = \emptyset$, Red always wins.)

We make several remarks about this definition: First, we do not include mention of the node where the placemarker is placed to begin the game. Thus one of our games might be thought of as a set of games as games are usually conceived, since any node could be designated as the initial node for play.

Second, we could have omitted Condition (2) from the definition. Had we done so, we would have had to make only trivial modifications in our theory. Some examples could then have been simplified. But, on the whole, our present concept seems better in that it is in accord with our ordinary notion of game, in which the players move in turn.

Third, the players Black and Red are symmetrical; any game can be transformed into a game in which the roles of the two players are interchanged by (1) changing $R$ to $B$, (2) changing $B$ to $R$, and (3) replacing $\Omega$ by $2^W - \Omega$.

The fourth and final remark about our definition is that we could have excluded the subset $W$ as a constituent part of the concept of game. This simplification would have been made possible by stipulating $\Omega$ to be a subset of $2^Q$. We chose not to do this because of our interest in the complexity of computing the winning strategies of our games. Our results show that the time of computation is roughly exponential in the size of the set $W$ and polynomial in the size of the set $Q$. It is therefore important not only to distinguish $W$ from $Q$ but also to make $W$ as small as possible.

As an example, consider the game whose graph is shown in Fig. 1. In this game, $B = \{b_1, b_2, b_3\}$, $R = \{r_1, r_2, r_3\}$, $E$ is as shown, $W = \{r_1, b_1\}$ and $\Omega = \{W, \emptyset\}$. (In the figures the nodes of $W$ have double circles). Thus Black wins if and only if either $r_1$ and $b_1$ are visited only finitely often or else they are each visited infinitely often; Red wins if and only if one of these two nodes is visited infinitely often and the other finitely often.

Black has a winning strategy for this game, wherever the placemarker is placed initially. The strategy is as follows: from $b_3$, move to $r_2$ if $b_1$ has never been visited or if $r_1$ has been visited since the last time $b_1$ was visited; in all other

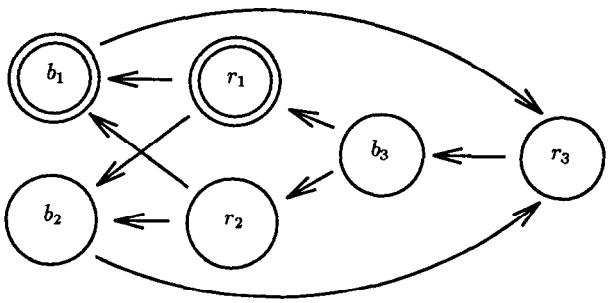


Fig. 1. Example of a game.

circumstances move to $r_1$. If Black plays this strategy then Red can cause $b_1$ and $r_1$ to be visited only finitely often (e.g., by always moving from $r_1$ or $r_2$ to $b_2$) or he can cause them both to be visited infinitely often (e.g., by always moving to $b_1$). But, in playing against Black's strategy, Red cannot play so that one node of $W$ is visited infinitely often while the other is visited only finitely often. So he cannot win.

Black's winning strategy for this game calls upon her to make her move at any time in consideration of the latest node of $W$ that was visited. Thus her strategy directs her to move in consideration of the *latest visitation record* (LVR). The definition of this concept is as follows: For a game $\mathcal{G} = (Q, R, B, E, W, \Omega)$ we stipulate that the null sequence is the LVR for the first moment of time. Therefore, if the placemaker is on a node not in $W$ then the LVR for that moment is the same as it was for the preceding moment. But if the placemarker is at node $w_i \in W$ and the LVR for the preceding moment is $w_{j_1}, \ldots, w_{j_k}$ then we stipulate: (1) If $w_i$ has never been visited before in that play then $w_{j_1}, \ldots, w_{j_k}, w_i$ is the LVR for the new moment. (2) If $w_i$ has been visited before and $w_i = w_{j_h}$ then

$$w_{j_1}, \ldots, w_{j_{h-1}}, w_{j_{h+1}}, \ldots, w_{j_k}, w_{j_h}$$

is the LVR for that moment. (The concept of LVR is taken from [6], where it is called LAR.)

Note that if $w_k$ occurs after $w_h$ in the LVR then $w_k$ was last visited after $w_h$ was last visited. And any node of $W$ that does not occur in the LVR has not yet been visited.

An *LVR strategy* for a player in a game is a rule that specifies the edge to move the placemarker along, given the node on which the placemarker is currently placed and the LVR for that time. Thus an LVR strategy in any of our games is a function over a finite domain. In practice, we do not define this function over all nodes in the graph that the player can play from, but only over those nodes that could be reached given that the player has already played according to the strategy, the opponent having played freely.

The winning strategy for Black described above in the game of Fig. 1 is an LVR strategy. The interesting thing about that game is that Black has no winning strategy that specifies moves without regard to the past. For example, if from $b_3$ Black always moves to $r_1$ (to $r_2$) then Red can win by always moving from $r_1$ to $b_2$ (from $r_2$ to $b_1$).

We shall prove in Section 4 that, in every game and for every node $q \in Q$, one of the players has a winning LVR strategy given that the play begins at $q$. Thus, although a player who has a winning strategy must remember something about what has happened in the past, that player need not remember everything that has happened. A player with an unlimited memory capacity would have no advantage. In [6] this feature of a winning strategy is called 'forgetful determinacy'.

Generally, the set of nodes from which Black can win and the set from which Red can win are both nonempty, although the game of Fig. 1 is an exception. More typical is the game of Fig. 2, in which $R = \{r_1, r_2, r_3\}$, $B = \{b_1, b_2, b_3, b_4\}$, $W = \{r_2, r_3\}$ and $\Omega = \{W, \emptyset\}$. As in Fig. 1, Red wins if and only if one of the nodes of $W$ is visited finitely often and the other infinitely often. Black has a winning strategy if play begins at $b_1$, $r_1$, or $b_2$: she always moves from $b_1$ to $r_1$ and from $b_2$ to $r_1$. On the other hand, Red has a winning strategy if play begins at $r_2$, $b_3$, $r_3$, or $b_4$: he always moves from $r_2$ to $b_3$ and from $r_3$ to $b_4$.

Incidentally, an interesting fact about the game of Fig. 2 is that neither of the two winning strategies need refer to the latest visitation record; for any node $p$ that a player visits more than once in forcing a winning play, that player can make the same move from $p$ every time. We call such strategies *no-memory winning strategies,* since the winning player can move without remembering anything about the history of the placemarker. We shall regard no-memory strategies as degenerate LVR strategies. It is significant that if a player has a winning strategy in the kind of finite-duration game discussed in Section 1 then that player has a winning no-memory strategy.

We note that a finite-duration game can be regarded as a special case of a game of infinite duration. The same can be said of games that have both infinite winning plays and finite winning plays. For example, it might be that whenever a certain node is visited play terminates with one of the two players, say Black, declared the winner. Such a node could simply be converted to an inescapable loop of two nodes, adding one of them to the set $W$, and adding the unit set of

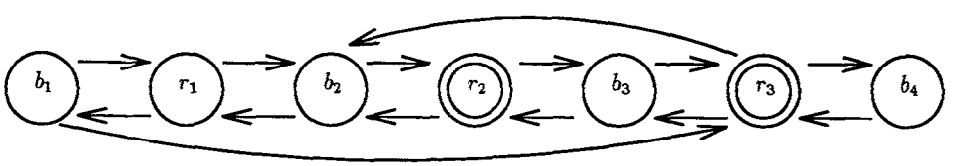


Fig. 2. Another game.

that node to $\Omega$. By successively carrying through this type of operation, any game that has plays of finite duration can be converted to an equivalent game all of whose plays are of infinite duration. Thus the concept of a game with both finite-duration winning conditions and infinite-duration winning conditions is a trivial generalization of our concept.

We say that $f$ is a *winning strategy from a set* $A \subseteq Q$ for a player of a game if $f$ is a winning strategy for that player given that the placemarker is placed initially at any node of $A$. Note that when we say, e.g., that Red has a winning strategy playing from a node $p$, that does not imply that $p \in R$; for $p \in B$, that Red has a winning strategy playing from $p$ simply means that, no matter how Black plays from $p$, Red can thereafter move according to the strategy and win. Indeed, if the set of all nodes from which Red has a winning strategy is nonempty then that set must have both $B$ nodes and $R$ nodes. And, of course, the same is true for Black.

## 3. Some preliminary theorems

In our theorems and other discussion, when we mention a game $\mathcal{G}$ we shall assume, usually without explicit mention, that $\mathcal{G} = (Q, R, B, E, W, \Omega)$.

**Theorem 3.1.** *If a player of a game $\mathcal{G}$ has LVR strategies $f$ and $g$ such that, for sets $A, C \subseteq Q$, $f$ is a winning LVR strategy playing from $A$ and $g$ is a winning LVR strategy playing from $C$, then there is a winning LVR strategy $h$ for that player playing from $A \cup C$. Furthermore, if $f$ and $g$ are no-memory strategies then so is $h$.*

**Proof.** Let $\alpha_g$ be the set of all ordered pairs $(L, p)$ such that $p$ is a node from which that player moves and $L$ is an LVR that occurs sometime when $p$ is visited in some play from $C$ when that player is using the strategy $g$. Then a winning LVR strategy $h$ from $A \cup C$ for the player is as follows:

$$h(L, p) = \begin{cases} g(L, p) & \text{if } (L, p) \in \alpha_g, \\ f(L, p) & \text{otherwise.} \end{cases}$$

If the player plays $h$ for a play beginning at a node of $C$, then it is easily established that for any move that the player makes thereafter at a node $p$ with LVR $L$, $h(L, p) = g(L, p)$; thus the player wins.

However, if the play is initially from a node of $A - C$ then the play will begin with $h(L, p) = f(L, p)$ and will continue as such until the first time (if ever) that $(L, p) \in \alpha_g$. At that point the player will move according to $h(L, p) = g(L, p)$ and will do so forever.

In the latter case, the play will be like some play from $C$ except for some finite portion of each; thus the permset will be the same, which is a win for the player.

On the other hand, if $(L, p)$ is never in $\alpha_g$ then $h(L, p) = f(L, p)$ forever. Since the play originates from a node of $A$, again the player will win. □

**Corollary.** *If a player of a game has, for each $p \in A \subseteq Q$, a winning LVR strategy playing from $p$ then the player has a winning LVR strategy playing from $A$. Furthermore, if all of the former are no-memory strategies then so is the latter.*

We note that the proof of Theorem 3.1 makes no use of the meaning of 'LVR'. Both Theorem 3.1 and the Corollary would be valid if any other class of strategies were to replace the class of LVR strategies.

**Theorem 3.2.** *If a player of a game $\mathcal{G}$ has a winning LVR strategy $f$ playing from $p \in Q$, and $q \in Q$ is visited in some play from $p$ in which that player always follows $f$, then that player has a winning LVR strategy $f'$ playing from $q$. Furthermore, if $f$ is a no-memory strategy then so is $f'$.*

**Proof.** Let $P_1$ be a finite portion of some play of the game beginning at $p$ and ending at node $q$, in which the player is playing according to strategy $f$; let $L_1$ be the LVR for the moment at the end of $P_1$. Let $P_2$ be the finite portion of a play from $q$, let $s$ be the node visited at the end of $P_2$ and let $L_2$ be the LVR (according to $P_2$) at that moment. We can compute from $L_1$ and $L_2$ the LVR $L_3$ for the moment at the end of the play $P_1P_2$ (i.e., the play that is the concatenation of $P_1$ and $P_2$): $L_3 = L_1', L_2$, where $L_1'$ is the sequence $L_1$ with those nodes occurring in both $L_1$ and $L_2$ deleted. We fix the play $P_1$ and the sequence $L_1$, and we allow $P_2$ to vary, giving us variable $s$ and $L_2$ from which an $L_3$ is determined. We write $L_3 = \beta(L_2)$ to express the functional relationship of $L_3$ to $L_2$.

Then a winning LVR strategy $f'$ that the player can use to play from node $q$ is $f'(s, L_2) = f(s, \beta(L_2))$. In other words, we are having the player (when playing from $q$) play as if he or she were playing from $p$ by prefixing the imaginary prefix $P_1$ to the actual play. The permset of any infinite play beginning from $q$ will be the same as it would have been had the play begun at $p$ since these two plays differ only in an initial finite portion. Thus, since $f$ is a winning strategy playing from $p$, so is $f'$ playing from $q$. □

We can summarize Theorems 3.1 and 3.2 as follows.

**Corollary.** *If $U_R$ ($U_B$) is the set of all nodes from which Red (Black) has some winning LVR strategy or other than Red (Black) has a uniform winning LVR strategy playing from $U_R$ ($U_B$). Furthermore, if $p \in U_R$ and $q$ is a node visited during a play from $p$ in which Red is a winning LVR strategy, then $q \in U_R$ (and similarly for $U_B$ and Black). Furthermore, these two assertions are true also for no-memory strategies.*

In the sequel we shall use Theorem 3.1 and 3.2 and the two corollaries without explicit reference.

We shall frequently discuss the situation in which some $A \subseteq Q$ is given for a game and we wish to determine the set $N$ of all nodes from which a player has a

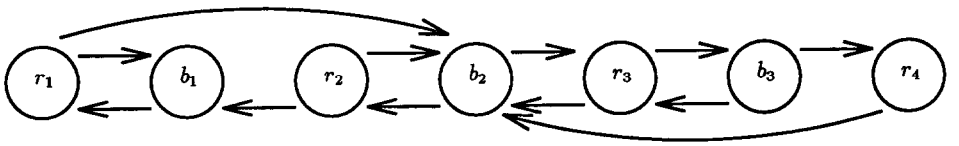


Fig. 3. $A = \{r_2, b_2\}$.

strategy to force the placemarker to be in $A$ after *one or more moves*. We call attention to the possibility that a node may be in $A - N$. An example of this is given in Fig. 3, in which $R = \{r_1, r_2, r_3, r_4\}$ and $B = \{b_1, b_2, b_3\}$. If $A = \{r_2, b_2\}$ and the player is Black then $N = \{b_2, r_3, b_3, r_4\}$. Although $r_2$ is in $A$, it is not in $N$: if Red moves from $r_2$ to $b_1$ then Black cannot force the placemarker back into $A$.

However, if we talk about the set $N'$ of nodes from which a player has a strategy to force the placemarker to be in $A$ after zero or more moves then (for any $A$ and any game) $A \subseteq N'$; indeed, $N' = N \cup A$. It follows that $A \subseteq N$ if and only if $N = N'$.

If $A$ is a good place to be then the player has an interesting advantage when $A \subseteq N$: if the placemarker is ever in $A$ then the player can play so that either it stays in $A$ forever or it always comes back into $A$ after leaving. An example in which $A \subseteq N$ is $A = \{b_2\}$ for Black in Fig. 3.

**Theorem 3.3.** *In a game $\mathcal{G}$ let $D$, $H \subseteq Q$, $D \cap H = \emptyset$, and let $N$ ($N'$) be the set of nodes outside of $H$ from which a player has a no-memory strategy to force the placemarker to be in $D$ after one or more moves (after zero or more moves) while keeping it out of $H$. Then* (1) *$N$ and $N'$ are constructible from $\mathcal{G}$, $D$ and $H$. And* (2) *from $Q - H - N$, the opponent has a no-memory strategy to insure that the placemarker either reaches $H$ sometimes without having entered $D$ or stays forever in $Q - H - N - D$ after the first move.*

**Proof.** Without loss of generality, assume the player is Red and the opponent is Black. From the remarks preceding this theorem, $N \subseteq N'$ and any $p \in N' - N$ must be in $D$.

We begin by constructing $N'$. To this end, we define a function $\gamma: 2^{Q-H} \to 2^{Q-H}$ as follows: for any $X \subseteq Q - H$, $\gamma(X)$ is the set of all $p \in Q - H$ such that at least one of the following holds:

1. $p \in X$.
2. $p \in R$ and, for some $(p, p') \in E$, $p' \in X$.
3. $p \in B$ and, for all $(p, p') \in E$, $p' \in X$.

Note that, for $X \subseteq Q - H$, $\gamma(X)$ is the set of all nodes $p \in Q - H$ such that either $p$ is in $X$ or else from $p$ Red can force the placemarker to be in $X$ after one move.

We then define $D_0 = D$ and $D_{i+1} = \gamma(D_i)$ for all $i$. Clearly, for each $i \geqslant 0$, $D_i \subseteq N'$. Note that, for each $i$, $D_i \subseteq D_{i+1}$ and if $D_i = D_{i+1}$ then $D_i = D_x$ for all

$x \geq i$. Putting $k = |Q - H - D|$ we thus get $D_k = D_x$ for all $x \geq k$. For any $p \in Q - H - D_k$, Red cannot force the placemarker into $D_k$, and hence cannot force it into $D$. Therefore, $D_k = N'$.

We next determine the set $N' - N$. By definition of $N'$ and $N$, for all $p \in N'$, $p$ is outside $N$ if and only if $p \in D$ and Black can force the placemarker in one move from $p$ into $Q - N'$; in other words, if and only if either $p \in B \cap D$ and at least one edge from $p$ goes into $Q - N'$ or else $p \in R \cap D$ and all such edges go into $Q - N'$. We can easily determine all such nodes, and we can construct $N$ from $N'$ by removing all of them. Requirement (1) of Theorem 3.3 is satisfied.

Finally we note that, for each $p \in Q - H - N$, if $p \in B$ then there is an edge from $p$ into $Q - N - D$ and if $p \in R$ then all edges from $p$ go into $Q - N - D$ (otherwise, in each case, $p$ would be in $N$). Thus from $Q - H - N$, Black has the strategy required by (2) of our theorem. □

**Corollary.** *If a player has a strategy from some node to accomplish either of the two objectives mentioned in Theorem* 3.3 *then that player has a no-memory strategy to achieve that objective.*

The proof of Theorem 3.3 gives an algorithm for computing the set $N$ or $N'$, Red's no-memory strategy from this set, and Black's no-memory strategy from $Q - H - N$ or $Q - H - N'$, given the game $\mathcal{G}$ and sets $D$, $H \subseteq Q$, such that $D \cap H = \emptyset$. This algorithm will be an important part of the solution algorithm in the constructive proof of the Main Theorem in Section 4, and will play an important role in Section 6.

Let $\mathcal{G}$ be a game, and let $Q'$ be an arbitrary subset of $Q$. The *pseudogame of* $\mathcal{G}$ *determined by* $Q'$ is $(Q', R', B', E', W', \Omega')$ where $R' = R \cap Q'$, $B' = B \cap Q'$, $E' = E \cap (Q' \times Q')$, $W' = W \cap Q'$ and $\Omega' = \Omega \cap 2^{W'}$. If this pseudogame is a game then it is called the *subgame of* $\mathcal{G}$ *determined by* $Q'$.

**Theorem 3.4.** *The pseudogame of a game* $\mathcal{G}$ *determined by* $Q' \subseteq Q$ *is a subgame if and only if for all* $p \in Q'$ *there is a* $q \in Q'$ *such that* $(p, q) \in E$. *Any winning play for a player in a subgame of* $\mathcal{G}$ *is a winning play for the same player in* $\mathcal{G}$.

**Proof.** It is clear that a pseudogame satisfies conditions 1, 2, 4 and 5 in the definition of 'game'. Hence it is a game if and only if it satisfies condition 3, which is what the first statement of the theorem states. The second statement is clear from the definition of $\Omega'$. □

We close this section with a theorem that will be useful in the following sections.

**Theorem 3.5.** *For any* $A \subseteq Q$, $\emptyset \neq A \neq Q$, *if one of the two players has a strategy from A to keep the placemarker in A forever, then the pseudogame determined by A is a subgame.*

**Proof.** From the strategy we infer that, from any node of $A$ from which the player moves, there is at least one move that advances the placemarker to a new node of $A$; and all of the opponent's moves from any node of $A$ have this property. Theorem 3.4 then gives us the result. □

## 4. Proof of algorithmic solvability

Given a game $\mathcal{G}$, we define $U_R$ ($U_B$) to be the set of all nodes from which Red (Black) has a winning LVR strategy. The *solution* to $\mathcal{G}$ is a determination of $U_R$, $U_B$, Red's winning strategy playing from $U_R$ and Black's winning strategy playing from $U_B$. This section consists of a proof of the following, which is the main theorem of this paper. The proof is adapted from Gurevich and Harrington [6] (see also [12]).

**Theorem 4.1.** *In a game $\mathcal{G}$, $U_R \cap U_B = \emptyset$ and $U_R \cup U_B = Q$; furthermore, $U_R$, $U_B$, the winning strategy for Red playing from $U_R$ and the winning strategy for Black playing from $U_B$ can be effectively determined from $\mathcal{G}$.*

**Proof.** If $W = \emptyset$ then the game is trivial, as noted in Section 2. For the remainder of the proof we shall assume that $W \neq \emptyset$. Let $W = \{w_1, \ldots, w_n\}$.

That $U_R \cap U_B = \emptyset$ is obvious. The constructive proof of our theorem is by induction on $m = |Q|$. For the basis $m = 2$, the sets $R$, $B$ and $E$ are unique up to isomorphism. Putting $R = \{r\}$ and $B = \{b\}$, $E$ must be the set $\{(r, b), (b, r)\}$. Whatever $W$ and $\Omega$ may be, Red and Black never have any choice, and $W$ is always the permset. So either $U_R = Q$ and $U_B = \emptyset$, or vice versa, and the winning strategy is obvious.

As an inductive hypothesis, assume for any $m \geqslant 3$ that Theorem 4.1 is true for all games with fewer than $m$ nodes, and let $\mathcal{G}$ be a game in which $|Q| = m$. For each $i \leqslant n$, define $N_{i,B}$ ($N_{i,R}$) to be the set of nodes, excluding $w_i$, from which Black (Red) has an LVR strategy to keep out of $w_i$ forever and to win the game. Clearly, any play from a node of $N_{i,B}$ in which Black plays the indicated strategy will be entirely inside $N_{i,B}$; similarly for $N_{i,R}$ and Red. Define $N_B$ ($N_R$) to be the set of all nodes from which Black (Red) has a strategy to force the play into

$$\bigcup_{i=1}^{n} N_{i,B} \qquad \left(\bigcup_{i=1}^{n} N_{i,R}\right)$$

in zero or more moves. By the Corollary to Theorem 3.3 this strategy can be a no-memory strategy. Clearly, $N_{i,B} \subseteq N_B$ and $N_{i,R} \subseteq N_R$ for each $i$. Also $N_B \subseteq U_B$ and $N_R \subseteq U_R$.

We must prove that the $N_{i,B}$'s, $N_B$, the $N_{i,R}$'s and $N_R$ can be effectively determined. Let $X_i$ be the set of nodes from which Red can force the placemarker

to $w_i$ in zero or more moves. By Theorem 3.3, the set $X_i$ is constructible. For any $p \in Q - X_i$ there is an edge $(p, q)$ for some $q \in Q - X_i$ (otherwise $p$ would be in $X_i$). It follows by Theorem 3.4 that the pseudogame determined by the set $Q - X_i$ is a subgame of $\mathcal{G}$, with fewer than $m$ nodes (since $w_i \in X_i$). By the inductive hypothesis we can determine both the set $Z$ of nodes from which Black has a winning LVR strategy $f$ in the subgame and $f$ itself. Since Red has no move out of $Q - X_i$ in $\mathcal{G}$, $f$ is also a winning LVR strategy for Black playing from $Z$ in $\mathcal{G}$. It follows that $Z \subseteq N_{i,B}$.

Now the LVR strategy for Black from $N_{i,B}$ to keep out of $w_i$ forever and win would also have to avoid $X_i$. It follows, by definition of $Z$, that $N_{i,B} \subseteq Z$. With the result of the previous paragraph we get $N_{i,B} = Z$. Thus $N_{i,B}$ is constructible.

From the $N_{i,B}$'s, the set $N_B$ and an accompanying strategy are easily constructed from $\bigcup_{i=1}^{n} N_{i,B}$ by Theorem 3.3. Similarly, the $N_{i,R}$'s and $N_R$ with accompanying strategies can be effectively determined.

The method of our proof will be to deal with the game $\mathcal{G}$ by peeling off $N_R$ and $N_B$ leaving us with what we shall prove is a subgame having $Q - N_R - N_B$ as its set of nodes. In this subgame a new $N_R$ and $N_B$ is peeled off, etc. Finally we get a game in which either $N_R \cup N_B = \emptyset$ or $N_R \cup N_B = Q$, which can then be handled directly. We put $Q_N = Q - N_R - N_B$.

**Lemma 1.** *If $Q_N \neq \emptyset$ then the pseudogame of $\mathcal{G}$ determined by $Q_N$ is a subgame. Any winning strategy in the subgame for a player from $p \in Q_N$ can be augmented to become a winning strategy in $\mathcal{G}$ for the same player playing from $p$.*

**Proof.** Black has no strategy to force the placemarker from any $p \in Q - N_B$ into $N_B$ (otherwise $p$ would be in $B_B$). Thus, by Theorem 3.3, Red has a strategy from $Q - N_B$ to remain in $Q - N_B$ forever. And thus, by Theorem 3.5, the pseudogame determined by $Q - N_B$ is a subgame $\mathcal{G}'$ of $\mathcal{G}$. Likewise Red has no strategy from $Q - N_R$, and hence no strategy from $Q_N = Q - N_B - N_R$, to force the placemarker into $N_R$. So (by similar reasoning) the pseudogram of $\mathcal{G}'$ determined by $Q_N$ is a subgrame of $\mathcal{G}'$ and hence a subgame of $\mathcal{G}$, giving us the first statement of Lemma 1.

The second statement of Lemma 1 follows from the fact that a move by either player out of $Q_N$ constitutes a forfeiture of the game (since Red can move out of $Q_N$ only into $N_B$, where Black has a winning strategy, and Black can so move only into $N_r$, where Red has a winning strategy). □

**Lemma 2.** *If $N_B = N_R = \emptyset$ and $W \in \Omega$ then, from all of $Q$, Black has a winning LVR strategy (i.e., $U_R = \emptyset$, $U_B = Q$). Similarly, if $N_B = N_R = \emptyset$ and $W \in 2^W - \Omega$ then Red has a winning strategy from all of $Q$.*

**Proof.** We prove only the first sentence of our lemma; the second sentence will follow by symmetry. Accordingly, we assume $N_B = N_R = \emptyset$ and $W \in \Omega$. Thus if all the nodes of $W$ are visited infinitely often then Black wins.

We define the *designated integer* of any given moment of the play of a game to be either the smallest integer $i$, $1 \leq i \leq n$, such that $w_i$ has never been visited in the play so far or, if all nodes have been visited, the $i$ such that $w_i$ is first in the LVR. In the latter case, note that the placemarker's last visit to $w_i$ was earlier than its last visit to $w_j$, for any $j \neq i$.

For each $i$, let $A_i$ be the set of nodes from which Black has a no-memory strategy to force the placemarker to $w_i$ in zero or more moves. Put $Q_i = Q - A_i$. Note that all edges from $Q_i$ to $A_i$ are from $R$ nodes, and from any such node there must be another edge leading to another node of $Q_i$. It follows by Theorem 3.4 that the pseudogame determined by $Q_i$ is a subgame $\mathcal{G}_i$ of $\mathcal{G}$. (For the purposes of Case II in the proof of Theorem 5.4 of in Section 5, we observe that $\mathcal{G}_i$ lacks the node $w_i$, and hence has fewer winning-condition nodes than $\mathcal{G}$.)

We claim that Red has no winning strategy in $\mathcal{G}_i$ playing from any node of $Q_i$. To prove this claim, we assume that $p$ is a node of $Q_i$ from which Red has a winning LVR strategy $f_i$ in $\mathcal{G}_i$. We let $U_{i,R}$ be the set of all nodes of $Q_i$ from which Red can play $f_i$ and be guaranteed a win. Since $p \in U_{i,R}$ we know that $U_{i,r} \neq \emptyset$. Let $f$ be the LVR strategy in $\mathcal{G}$ for Red defined as follows: for any LVR $L$ of $\mathcal{G}$ let $(L)_i$ be the LVR of $\mathcal{G}_i$ obtained from $L$ by deleting all winning-condition nodes of $\mathcal{G}$ that are not nodes of $\mathcal{G}_i$; for any LVR $L$ and node $q$ of $U_{i,R}$, $f(L, q) = f_i((L)_i, q)$; for $q \in A_i$, and for $q \in Q_i - U_{i,R}$, $f(L, q)$ is not defined. Now, since in $\mathcal{G}$ Black has no moves from $Q_i$ into $A_i$, it follows that if Red plays $f$ from $U_{i,R}$ in $\mathcal{G}$ then the placemarker will remain in $U_{i,R}$ and Red will win the play. So $f$ is a winning LVR strategy for Red playing from $U_{i,R}$ in $\mathcal{G}$, which insures that the placemarker keeps out of $w_i$, implying that $U_{i,R} \subseteq N_R$ and thus contradicting the hypothesis that $N_R = \emptyset$. Our argument proves that Red has no winning LVR strategy playing from any node of $Q_i$ in the subgame $\mathcal{G}_i$.

But since $w_i \notin Q_i$, the game $\mathcal{G}_i$ has fewer than $m$ nodes. By the inductive hypothesis, therefore, Black must have in $\mathcal{G}_i$ a winning LVR strategy $g_i$ playing from every node. Let the strategy $h_i$ in the game $\mathcal{G}$ be defined from $g_i$ the way the strategy $f$ is defined from $f_i$ above. If Black plays $h_i$ from $Q_i$ in the game $\mathcal{G}$ one of two things will happen: (1) Red may play forever so that the play never leaves $Q_i$, in which case Black wins. Or (2) Red may move into $A_i$, where $h_i$ is not defined. But if this happens Black can force the placemarker to $w_i$.

The winning LVR strategy for Black playing from all nodes is as follows: At any time her choice will be based on the designated integer $i$, and she plays according to the paragraph above. That is to say, as long as the placemarker is in $Q_i$ she plays $h_i$; but if it reaches $A_i$ she forces it to $w_i$. As a result, either she wins without ever visiting $w_i$ or the placemarker eventually reaches $w_i$. If it reaches $w_i$ there is a new designated integer $i'$ and Black then repeats with $i'$ in place of $i$. And so on.

We note that this strategy, call it $f$, is an LVR strategy. Where $L$ is any LVR of $\mathcal{G}$, let the integer $\phi(L)$ be defined as follows: if $L$ has length $n$ then $w_{\phi(L)}$ is the first node in $L$; if $L$ has length less than $n$ then $\phi(L)$ is the smallest $i$ such that $w_i$

is not in $L$. The strategy $f$ can thus be defined as follows: for any LVR $L$ of $\mathcal{G}$ and node $p \in Q$, (1) if $\phi(L) = i$ and $p \in A_i$ then $f(L, p)$ is as directed by the no-memory strategy that forces the placemarker into $w_i$; (2) if $\phi(L) = i$ and $p \in Q_i$ then $f(L, p) = g_i((L)_i, p)$ where $(L)_i$ is $L$ with those nodes not in $Q_i$ deleted. Thus $f$ is an LVR strategy.

If Black plays according to this strategy the permset will depend on how Red plays. For any $i$ and for any time when $w_i$ is first in the LVR, it may be possible for Red to play so that the placemarker never reaches $w_i$. But if he does then Black wins. On the other hand if Red can always, and does always, permit the appropriate $w_i$ to be visited then such a play will have $W$ as its permset, since each integer $i$, $1 \leq i \leq n$, will be the designated integer infinitely often. In that case also Black wins, since $W \in \Omega$. □

We can now complete the inductive step and, with it, the proof of the Main Theorem. Given a game $\mathcal{G}$ where $|Q| = m$, we construct $N_R$ and $N_B$ and take $Q_N = Q - N_R - N_B$. If $Q_N = \emptyset$ then $U_R = N_R$, $U_B = N_B$ and the strategies can be effectively determined. If $N_R \cup N_B = \emptyset$, Lemma 2 gives us our result.

But if neither $N_R \cup N_B$ nor $Q_N$ is empty then we consider the subgame $\mathcal{G}_N$ of $\mathcal{G}$ determined by $Q_N$. Clearly, $|Q_N| < |Q|$, and so by the inductive hypothesis there are sets $U'_R$ and $U'_B$ (where $U'_R \cup U'_B = Q_N$ and $U'_R \cap U'_B = \emptyset$), and LVR strategies $f'_R$ and $f'_B$ such that $f'_R$ ($f'_B$) is a winning LVR strategy for Red (Black) playing from $U'_R$ ($U'_B$) in $\mathcal{G}_N$. By Lemma 1, we can obtain a winning LVR strategy for Red (Black) playing from $U'_R$ ($U'_B$) in the game $\mathcal{G}$. By Theorem 3.1, we can construct a winning LVR strategy for Red (Black) playing from $U_R = U'_R \cup N_R$ (from $U_B = U'_B \cup N_B$) in $\mathcal{G}$, since there is a winning LVR strategy for Red (Black) playing from $N_R$ ($N_B$). This concludes the proof of Theorem 4.1. □

The algorithm to solve a given game is thus a recursive one. The remainder of this section will exhibit parts of the algorithm explicitly. Assume that $FO_{R,0}(\mathcal{G}, X)$ ($FO_{B,0}(\mathcal{G}, X)$) is a function whose value when $X \subseteq Q$ is the set of all nodes from which Red (Black) can force the placemarker to be in $X$ after zero or more moves. The proof of Theorem 3.3 makes it clear how to write a subalgorithm that computes these functions. Let $SUB(\mathcal{G}, X)$ be a function whose value when $X \subseteq Q$ is the pseudogame of $\mathcal{G}$ determined by $X$. (We shall be using this function only when we know that the pseudogame will be a subgame.)

Fig. 4 shows the recursive procedure that computes the functions $U_B(\mathcal{G})$ and $U_R(\mathcal{G})$ (the sets of nodes of $Q$ from which Black and Red, respectively, have winning strategies in $\mathcal{G}$). The full algorithm for Theorem 4.1 would be an elaboration of Fig. 4 in which the winning strategy for Black from $U_B$ and the winning strategy for Red from $U_R$ would also be constructed, using the winning strategies calculated from the subgames recursively referred to. There would be two different computations, one for the case $Q_N \neq Q$ and the other for the case $Q_N = Q$. The computation of the strategies in each case from other strategies

previously computed is straightforward. However, the algorithmic notation (i.e., the coding) is involved. Suffice it to say that the most natural data structure for an LVR strategy $f$ would seem to be a two-dimensional array: one dimension indicates a node $q \in Q$, the other dimension indicates an LVR $L$, and the array entry is $f(L, q)$, which is the node to which the player will move the placemarker as directed by the strategy when the LVR at that moment is $L$ and the placemarker is at node $q$.

Function procedure
  [input $\mathcal{G} = (Q, R, B, E, W = \{w_1, \ldots, w_n\}, \Omega)$;
   output functions $U_B(\mathcal{G})$, $U_R(\mathcal{G})$]
begin
if $Q = \emptyset$ then $[U_B := \emptyset; U_R := \emptyset]$ else
$[X_i := FO_{R,0}(\mathcal{G}, \{w_i\})$, $1 \leqslant i \leqslant n$;
 $X_i' := FO_{B,0}(\mathcal{G}, \{w_i\})$, $1 \leqslant i \leqslant n$;
 $N_{i,B} := U_B(SUB(\mathcal{G}, Q - X_i))$, $1 \leqslant i \leqslant n$;
 $N_{i,R} := U_R(SUB(\mathcal{G}, Q - X_i'))$, $1 \leqslant i \leqslant n$;
 $N_B := FO_{B,0}(\mathcal{G}, \bigcup_{i=1}^{n} N_{i,B})$;
 $N_R := FO_{R,0}(\mathcal{G}, \bigcup_{i=1}^{n} N_{i,R})$;
 $Q_N := Q - N_B - N_R$;
 *if* $Q \neq Q_N$ then
   $[U_B := N_B \cup U_B(SUB(\mathcal{G}, Q_N))$;
    $U_R := N_R \cup U_R(SUB(\mathcal{G}, Q_N))]$
 else (/The Lemma 2 case/)
   if $W \in \Omega$ then $[U_B := Q; U_R := \emptyset]$
   else $[U_R := Q; U_B := \emptyset]]$
end

Fig. 4. Recursive solution algorithm (partial).

## 5. The algorithm and its complexity

The solution algorithm is recursive; one solves a game by constructing several smaller games, solving those, and then constructing the solution to the desired game in terms of the solutions of the smaller games. It turns out almost always that the smaller games one constructs are, besides being smaller in the total number of nodes, smaller also in the number of winning-condition nodes. Indeed, the algorithm never calls for the construction of a game with a larger number of winning-condition nodes; and the only case where this number is equal is a case where the constructed game can be directly solved without further recursion steps, as we shall see.

Thus we are able to analyze the algorithm as if it were recursive in terms of the number of winning-condition nodes. Using this analysis we shall demonstrate

that, although the time complexity of the algorithm may be exponential in the number of winning-condition nodes, it is polynomial in the total number of nodes.

Before we can analyze the algorithm we must decide how the information about the game is stored inside a computer. In this section we shall adhere to the convention that $m = |Q|$ and $n = |W|$. We assume that the nodes of the graph are represented as the positive integers $1, \ldots, m$, the first $n$ of which represent the nodes of $W$. There is a one-dimensional array telling whether each node is in $R$ or $B$, and a two-dimensional array telling, for each ordered pair of nodes, whether there is an edge from the first to the second. We assume that all subsets of $Q$ are represented as one-dimensional arrays, and when we speak of constructing a set we mean the construction of such an array.

A subset of $W$ is represented by a number

$$i_1 2^0 + i_2 2^1 + \cdots + i_n 2^{n-1}$$

where $i_j = 1$ if the $j$th node of $W$ is in the subset, and 0 if not. Accordingly, $\Omega$ is represented as a one-dimensional Boolean array of size $2^n$, indexed from 0 to $2^n - 1$. We make no claim that the data structure we have described is the most efficient. Nor do we claim that our time bounds are the best possible. (For example, Richard E. Stearns points out that an alternative to the computational scheme given in the proof of Theorem 5.1, more elaborate but using well known techniques, has a better time bound of $O(m^2)$. This improvement would carry through Theorems 5.2 and 5.4 to yield a slight improvement in Theorem 5.5.)

**Theorem 5.1.** *For the algorithm described in the proof of Theorem* 3.3 (*which finds the set N, or alternatively N′, and the two no-memory strategies from two given subsets D and H of Q in a given game*) *there is a constant* $k_0$ *such that the time of computation is bounded by* $k_0 m^3$.

**Proof.** Referring to the details in the proof of Theorem 3.3 we note that there are constants $c_1$, $c_2$, $c_3$, $c_4$ such that in any game the following are true: (1) For any $X \subseteq Q - H$, the time to compute $\gamma(X)$ is bounded by $c_1 m^2$. (2) The time to compute $N'$ is bounded by $c_2 m^3$. The time to compute the set $N' - N$ after the computation of $N'$ is bounded by $c_3 m^2$. The time to compute $N$ from the sets $N'$ and $N - N'$ is bounded by $c_4 m$.

We further note that the strategy for Red can be obtained as a byproduct of the four steps in the computation of the set $N$ without affecting the statements (1) through (4), although the values of the constants must be modified. As remarked in the last paragraph of the proof of Theorem 3.3, once $N$ is known, Black's no-memory strategy from $Q - H - N$ is easy to compute. Indeed, there is a constant $c_5$ such that the time to compute it is bounded by $c_5 m^2$.

Theorem 5.1 follows from these assertions. □

We modify the substance of the algorithm from what was given in the proof of the Main Theorem only in the case $|W| = 1$.

**Theorem 5.2.** *There is a constant $k$ and an algorithm that can solve every game $\mathcal{G}$ in which $|W| = 1$ with a computation time bounded by $km^3$. Furthermore, the winning strategies of such games can always be no-memory strategies.*

**Proof.** Let $W = \{w\}$. We assume without loss of generality that $W \in \Omega$. Since a game in which $\Omega = 2^W$ is trivial, we further assume that $\emptyset \in 2^W - \Omega$. Then Black wins any play if and only if $w$ is visited infinitely often. We first apply the algorithm of Theorem 3.3, with Black as the player and Red as the opponent, taking $D = \{w\}$ and $H = \emptyset$, getting the set $N$ of all nodes from which Black has a strategy to force the placemarker to $w$ in one or more moves.

*Case* I: $w \in N$. From $N$ Black has a winning strategy; she simply keeps playing into $w$. By Theorem 3.3, Red has a no-memory strategy from $Q - N$ to keep the placemarker in $Q - N$ forever, which is a winning strategy for Red. Thus the game is solved.

*Case* II: $w \notin N$. Red has a winning no-memory strategy from $Q - N$ in $\mathcal{G}$. Furthermore, Red can extend that strategy to a winning no-memory strategy for all of $Q$: he simply makes an arbitrary rule for all nodes of $N$. Since $w \notin N$, if the play goes forever in $N$ then Red wins. But if the placemarker visits $w$ Red would win because the placemarker is now in $Q - N$.

It is clear that everything that is done in this computation can be taken care of by one application of the algorithm in the proof of Theorem 3.3. Thus, for some $k$, the time of computation is bounded by $km^3$. □

We now look at the recursion step in the algorithm covering the cases in which $|W| > 1$. Recall that, for each $i$, $X_i$ is the set of nodes from which Red can force the placemarker to $w_i$ in zero or more moves; and $X_i'$ is the set of nodes from which Black can force the placemarker to $w_i$ in zero or more moves.

From the given game $\mathcal{G}$, at most $2n + 1$ subgames are constructed. These are the subgames determined by the sets $Q - X_i$, for $1 \leq i \leq n$, by the sets $Q - X_i'$, for $1 \leq i \leq n$ and by the set $Q_N$. Let $\mathcal{G}_N = (Q_N, R_N, B_N, E_N, W_N, \Omega_N)$ be the subgame determined by $Q_N$. (Thus, for example, $W_N = W \cap Q_N$).

In all these subgames, the set of winning-condition nodes is less than $n$, except possibly the subgame $\mathcal{G}_N$. So let us examine this subgame more closely. Let $N_{i,B}'$, $N_{i,R}'$ (for appropriate $i$), $N_B'$ and $N_R'$ be defined in $\mathcal{G}_N$ the way $N_{i,B}$, $N_{i,R}$, $N_B$ and $N_R$ are defined in $\mathcal{G}$.

**Theorem 5.3.** *If $W_N = W$ then $N_B' = N_R' = \emptyset$.*

**Proof.** From $W_N = W$ we get $(N_R \cup N_B) \cap W = \emptyset$ (since, by definition, $W_N = W \cap Q_N$ and $Q_N = Q - N_R - B_B$).

*Case* I: $\emptyset \in 2^W - \Omega$. That is, in $\mathcal{G}$ (and hence in $\mathcal{G}_N$) Red wins if the permset is $\emptyset$. We claim that this assumption implies that $N_B = \emptyset$; the proof is as follows.

Assume $N_B \neq \emptyset$. From $N_B$ Black has a strategy to move the placemarker into $\bigcup_{i=1}^{n} N_{i,B}$. In any play in which she plays this strategy the placemarker must sometime arrive at a node $p' \in N_{i,B}$, for some $i$, from which Black has a strategy to play forever in $N_{i,B}$ and win. This implies that some winning-condition node $w_j \in N_{i,B} \subseteq N_B$ since $\emptyset \notin \Omega$, contradicting the fact that $(N_R \cup N_B) \cap W = \emptyset$. Thus $N_B = \emptyset$.

From this result we get $Q_N = Q - N_R$. Note that in $\mathcal{G}$ Red has no move from any $q \in Q_N$ into $N_R$; otherwise $q$ would be in $N_R$ by definition of $N_R$. We are now ready to consider $N'_B$ and $N'_R$.

Suppose that, for some $i$ and $p$, $p \in N'_{i,B}$. By definition, Black would then have, in the game $\mathcal{G}_N$, an LVR strategy from $p$ to keep out of $w_i$ forever and win; if Black plays according to this strategy then the placemarker will remain within $N'_{i,B}$ forever. But then in $\mathcal{G}$ also Black would have an LVR strategy from $p$ to keep out of $w_i$ and win, since $N'_{i,B} \subseteq Q_N$. By definition of $N_{i,B}$, the node $p$ would be in $N_{i,B} \subseteq N_B$, contradicting our result that $N_B = \emptyset$. It follows that $N'_{i,B} = \emptyset$ for all $i$ and thus $N'_B = \emptyset$.

Finally, assume $N'_R \neq \emptyset$. Then, for some $i$, $N'_{i,R} \neq \emptyset$. Red's strategy in $\mathcal{G}_N$ from $N'_{i,R}$ to keep out of $w_i$ forever and win could be extended to the game $\mathcal{G}$ as follows: Any such play, if Black never moved in $N_R$, would continue in $Q_N$ and Red would win. If Black moves into $N_R$ then Red could continue with his winning strategy in $\mathcal{G}$ in the set $N_R$, the play remaining in $N_R$ forever where there are no winning-condition nodes. Thus in this play also Red would win without ever entering $w_i$. We have proved that, in $\mathcal{G}$, Red has a strategy from $N'_{i,R}$ to win without ever entering $w_i$. This implies that $N'_{i,R} \subseteq N_{i,R}$, contradicting the assumption that $N'_{i,R} \subseteq Q_N$. Thus $N'_R = \emptyset$, concluding the proof of our Theorem in Case I.

*Case* II: $\emptyset \in \Omega$. The proof in Case I goes over to this case mutatis mutandis, since nowhere in the above proof have we made use of any special property of $\Omega$ other than the case definition. In particular, we have not made use of the assumption of Section 4 that $W \in \Omega$. □

Because the solution algorithm is a recursive one, our analysis of its complexity should be recursive also. For any $m$ and $n \leq m$, let $F(m, n)$ denote the least upper bound of the time of computation of the solution of any game in which $|Q| \leq m$ and $|W| \leq n$. Note that $F(m, n) \leq F(m', n')$ whenever $m \leq m'$ and $n \leq n'$.

**Theorem 5.4.** *There is a positive constant $k_1$ such that, for all $n > 1$, $m \geq n$ and game $\mathcal{G}$ where $|W| = n$ and $|Q| = m$, the time of computation for solving $\mathcal{G}$ is bounded by $k_1 m^3 n + 4nF(m, n-1)$.*

**Proof.** *Case* I: *the subgame $\mathcal{G}_N$ of $\mathcal{G}$ does not have all the winning-condition nodes of $\mathcal{G}$* (thus $W_N \neq W$). The recursive algorithm involves several steps: (1) The $X_i$'s and $X_{B,i}$'s are constructed. For each the algorithm of Theorem 3.3 is used and the time of computation is bounded by $k_0 m^3$, by Theorem 5.1. (2) The subgame of $\mathcal{G}$ determined by each $X_i$ and $X_{B,i}$ is constructed. The time for this is negligible. (3) The appropriate strategy from the appropriate set of nodes for each such subgame is constructed by recursively calling the algorithm. Each subgame has at most $n-1$ winning-condition nodes, and less than $m$ nodes. (4) If $Q_N \neq \emptyset$, the subgame $\mathcal{G}_N$ is constructed and its sets and strategies are computed, requiring another call of the algorithm. This subgame also has at most $n-1$ winning-condition nodes and less than $m$ nodes.

Thus in this case the time of computation to solve the game $\mathcal{G}$ is bounded by $2k_0 m^3 n + (2n+1)F(m, n-1)$.

*Case* II: *the game $\mathcal{G}_N$ has all the winning-condition nodes of $\mathcal{G}$* ($W_N = W$). Here we must go further with the computation and include the computation of the sets and strategies for $\mathcal{G}_N$. Theorem 5.3 tells us that $N'_B = N'_R = \emptyset$, and so the computation for $\mathcal{G}_N$ is taken from the proof of Lemma 2 in the proof of Theorem 4.1.

The computation of $\mathcal{G}_N$ involves (1) the construction of the sets $(A_N)_i$, $1 \leq i \leq n$, and (2) the solution of the subgames $(\mathcal{G}_N)_i$. (We are adapting the notation in the proof of Lemma 2: $(A_N)_i$ and $(\mathcal{G}_N)_i$ bear the same relations to $\mathcal{G}_N$ that $A_i$ and $\mathcal{G}_i$ bear to $\mathcal{G}$.) Note that the construction of each $(A_N)_i$ involves an application of the algorithm of Theorem 3.3. The time bound for these two steps is therefore $k_0 m^3 n + nF(m, n-1)$, since (as noted parenthetically in the proof of Lemma 2) each of the games $(\mathcal{G}_N)_i$ has at most $n-1$ winning-condition nodes.

Case II is the result of putting together most of what happened in Case I with the above. So the bound in Case II is $3k_0 m^3 n + (3n+1)F(m, n-1)$.

Finally, we take $k_1 = 3k_0$ and, since $n > 1$, our theorem is satisfied in both cases. □

**Theorem 5.5.** *For some constant $c$, $F(m, n) < m^3 (cn)^n$ for all $n \geq 1$ and $m \geq \max(2, n)$.*

**Proof.** Taking $c = \max(k, k_1 + 4)$, Theorems 5.2 and 5.4 imply

$$F(m, 1) < cm^3 \quad \text{and}$$
$$F(m, n) < (c-4)m^3 n + 4nF(m, n-1) \quad \text{for } n \geq 2.$$

From these inequalities, we establish that $F(m, n) < c^n m^3 n!$ by mathematical induction on $n$. The basis, $n = 1$, is clear. As an inductive hypothesis we take

$$F(m, n-1) < c^{n-1} m^3 (n-1)!$$

Since $k_1$ is positive, $c > 4$. We therefore get, for $n \geq 2$:

$$F(m, n) < (c-4)m^3n + 4c^{n-1}m^3n!$$
$$< (c-4)c^{n-1}m^3n! + 4c^{n-1}m^3n! = c^n m^3 n!$$

Since $c^n m^3 n! \leq m^3(cn)^n$, our theorem follows. □

Theorem 5.5 shows that the time complexity of the algorithm, although possibly exponential in $|W|$, is polynomial in $|Q|$. We conjecture that the problem of solving a given game (i.e., the problem of finding the winning strategies and the node sets from which they are applicable) is exponential in $|W|$. We obtain a polynomial-time algorithm if we fix a constant bound on the size of $|W|$.

We are therefore tempted to conclude that the computation required to solve a game is not excessive provided the set $W$ is kept small. But this conclusion is not justified for games that are given as board games. When a board game is transformed into a graph game, as many (or almost as many) nodes are needed in the graph as there are board configurations. As we pointed out in Section 1, any seriously interesting board game will almost certainly have so many configurations that no practical computation on the graph is possible. We should not trumpet too loudly our result that the solution algorithm is polynomial in the total number of nodes when $|W|$ is limited.

## 6. No-memory strategies

If in a game $\mathcal{G}$ Red has a no-memory strategy that he can play from a set $Q' \subseteq Q$ guaranteeing that play will always remain in $Q'$, then that strategy can be given as a function $h : R \cap Q' \to E$ such that, for each $q \in R \cap Q'$, $h(q)$ is an edge leaving $q$. Playing according to that strategy, Red simply moves along the edge $h(q)$ when he is in $q$. Similarly, a no-memory strategy for Black from $Q'' \subseteq Q$ guaranteeing that she will remain in $Q''$ is a function $k : B \cap Q'' \to E$. We have noted that some games have, from every node, a winning no-memory strategy for one of the players, while other games require properly LVR strategies. The former we shall call *no-memory games*. The problem arises, can we quickly tell whether or not a given game is a no-memory game? In this section we shall focus on a somewhat easier, but significantly related, problem.

**Definition.** The pair $(W, \Omega)$ is a *no-memory pair* if $\Omega \subseteq 2^W$, $W$ is finite and all games $\mathcal{G} = (Q, R, B, E, W, \Omega)$ (for suitable $Q, R, B, E$) are no-memory games.

We shall attack the problem of whether a given pair $(W, \Omega)$ is a no-memory pair. It turns out that this question has a rather interesting answer, providing a good possible first step in a method of telling whether or not a given game is a no-memory game.

**Definition.** The pair $(W, \Omega)$ has a *split* if there exist $\alpha, \beta \subseteq W$ such that either $\alpha \cup \beta \in \Omega$, $\alpha \in 2^W - \Omega$ and $\beta \in 2^W - \Omega$ or else $\alpha \cup \beta \in 2^W - \Omega$, $\alpha \in \Omega$ and $\beta \in \Omega$.

The pair $(W, \Omega)$ cannot have a split when $|W| = 1$, but can when $|W| = 2$ or more. For $W = \{w_1, w_2\}$ and $\Omega = \{W\}$, the pair $(W, \Omega)$ has a split with $\alpha = \{w_1\}$ and $\beta = \{w_2\}$.

**Theorem 6.1.** *If $(W, \Omega)$ has no splits and $W \in \Omega$ then, for some $w \in W$,*

$$\{\alpha \in 2^W \mid w \in \alpha\} \subseteq \Omega.$$

**Proof.** Let $\beta$ be a set of maximum size in $2^W - \Omega$. Since $\beta \neq W$, there exists $w \in W$ such that $w \notin \beta$. Now let $\alpha$ be any set such that $w \in \alpha$. The set $\alpha \cup \beta \in \Omega$ since $|\alpha \cup \beta| > |\beta|$. If $\alpha \notin \Omega$ then $(W, \Omega)$ would have a split. It follows that every such $\alpha$ is in $\Omega$. □

The main theorem of this section is

**Theorem 6.2.** *For any $W$ and $\Omega$ where $W$ is finite and $\Omega \subseteq 2^W$, $(W, \Omega)$ is a no-memory pair if and only if it has no splits.*

**Proof.** Assume first that $(W, \Omega)$ has a split of the form $\alpha \cup \beta \in \Omega$, $\alpha \notin \Omega$ and $\beta \notin \Omega$. Let $\alpha \cap \beta = \{u_1, \ldots, u_h\}$, $\alpha - \beta = \{v_1, \ldots, v_i\}$, $\beta - \alpha = \{w_1, \ldots, w_j\}$ and $W - (\alpha \cup \beta) = \{x_1, \ldots, x_k\}$. Then $i > 0$ and $j > 0$. We must construct $Q$, $R$, $B$ and $E$ so that $\mathcal{G} = (Q, R, B, E, W, \Omega)$ is a game but not a no-memory game, i.e., a game in which from some node the player having a winning strategy does not have a no-memory winning strategy.

The required game is given in Fig. 5, which is drawn with the assumption that $h + i$ and $h + j$ are both odd. If $h + i$ $(h + j)$ is even then the upper (lower) node labeled 'if needed' is deleted and the edge from $v_i$ (from $w_j$) is directed to $q$. In every case, all the nodes that are an even distance from the node $b$ are $B$ nodes and all other nodes are $R$ nodes.

Clearly in this game, Black is the only player than ever makes a choice and then only at the node $b$. She wins if and only if she moves from $b$ to $v_1$ infinitely often and from $b$ to $w_1$ infinitely often. Thus she has a winning LVR strategy but not a winning no-memory strategy.

If $(W, \Omega)$ has a split of the form $\alpha \cup \beta \notin \Omega$ but $\alpha \in \Omega$ and $\beta \in \Omega$, a similar game $\mathcal{G}$ can be constructed where, from some node, Red has a winning LVR strategy but no winning no-memory strategy. This concludes the proof of the theorem in one direction.

The proof in the other direction is more involved. We must prove that, in any game $\mathcal{G}$ where $(W, \Omega)$ has no splits, from every node $q \in Q$ one of the two players has a winning no-memory strategy from $q$. Our proof is by mathematical induction on $|W|$.

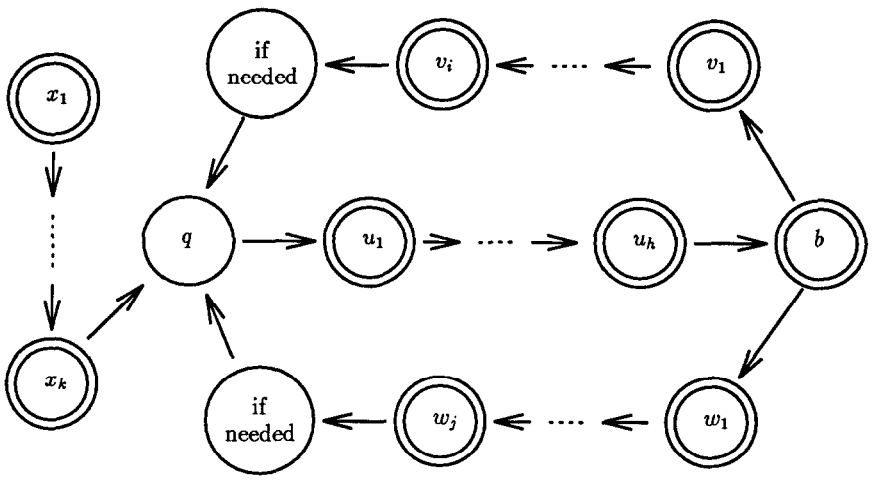


Fig. 5. Split.

The basis of the induction follows from the fact that where $|W| = 1$ the winning strategies can always be no-memory strategies, by Theorem 5.2. As an inductive hypothesis, let us assume for $n \geqslant 2$ that, in all games $\mathcal{G}$ in which $(W, \Omega)$ has no splits and $|W| < n$, all winning strategies can be no-memory strategies. Let $\mathcal{G}$ be a game in which $|W| = n$, and $(W, \Omega)$ has no splits. Again, without loss of generality, we assume that $W \in \Omega$.

By Theorem 6.1, there is a $w \in W$ such that Black wins any play of the game in which $w$ is visited infinitely often. We fix this selection of $w$ for the remainder of the proof. We define $Q'$ to be the set of nodes other than $w$ from which Red has a strategy to keep the placemarker out of $w$ forever; by Theorem 3.3 the strategy is a no-memory strategy.

Let $\mathcal{G}'$ be the pseudogame determined by $Q'$. Red's strategy to keep out of $w$ forever is also a strategy to remain in $Q'$ forever, and so, by Theorem 3.5, $\mathcal{G}'$ is a subgame of $\mathcal{G}$. Let $\mathcal{G}' = (Q', R', B', E', W', \Omega')$. Since $w \notin W'$ $|W'| < n$. Since $(W, \Omega)$ has no splits and since $\Omega' = \Omega \cap 2^{W'}$, $(W', \Omega')$ has no splits either. It follows by the inductive hypothesis that, from every node of $Q'$, one of the two players has a winning no-memory strategy in $\mathcal{G}'$. Let $C'$ $(D')$ be the set of nodes of $Q'$ from which Red (Black) has such a strategy.

The winning no-memory strategy for Red in $\mathcal{G}'$ from $C'$ works also in $\mathcal{G}$ from $C'$, since Black has no move from $Q'$ to $Q - Q'$. But the situation for Black and $D'$ in this regard is more complicated.

*Case* I: Red *is not able in* $\mathcal{G}$ *to force the placemarker from* $w$ *into* $C'$ *in one move.* Then Black can force the placemarker from $w$ into $Q - C'$ in one move. (This inference is valid whether $w$ is an $R$ node or a $B$ node, i.e., whoever has the choice of the move from $w$.)

We can now demonstrate that Black has in $\mathcal{G}$ a winning no-memory strategy

from $Q - C'$. The strategy is as follows: While in $D'$ Black plays her winning no-memory strategy in $\mathcal{G}'$. Suppose that Red plays so as to make the placemarker enter $Q - Q'$. Note that Black has from $Q - Q'$ a no-memory strategy to force the placemarker to $w$, by Theorem 3.3 and the definition of $Q'$. She does this and at $w$ she plays to force it to $Q - C' = (Q - Q') \cup D'$. By Theorem 3.3, this is a no-memory strategy.

However Red plays against this strategy, Black will win: if Red plays infinitely many times into $Q - Q'$, $w$ will be visited infinitely often causing Black to win. On the other hand, if Red at some point refuses to move into $Q - Q'$ any more, then the play will settle down in $D'$, which will also be a win for Black.

Thus, in Case I, Red has a winning no-memory strategy from $C'$ and Black has one from $Q - C'$.

*Case* II: *Red can force the placemarker from w into C' in one move.* The sets $Q'$ and $D'$ are not helpful in Case II. Let $F$ be the set of all nodes $q \in Q - C'$ such that Red has a strategy in $\mathcal{G}$ to force the placemarker in one or more moves into $C'$. Since Red has a winning no-memory strategy from $C'$ in $\mathcal{G}$, Red also has a winning no-memory strategy from $F$ in $\mathcal{G}$. Let $H = Q - C' - F$.

Any node of $Q - C'$ from which Red could, in $\mathcal{G}$, force the placemarker into $C' \cup F$ would itself be in $F$. Hence this could not be true of any node of $H$. It follows that from every node of $H$ there is an edge going back to $H$, and, by Theorem 3.4, that the pseudogame determined by $H$ is a subgame $\mathcal{G}_H$ of $\mathcal{G}$. Let $C_H$ ($D_H$) be the set of all nodes of $H$ from which Red (Black) has a winning no-memory strategy in the subgame $\mathcal{G}_H$. Since $w \notin H$, the inductive hypothesis applies to the game $\mathcal{G}_H$ and thus $H = C_H \cup D_H$.

Since Red has no move from $H$ into $Q - H$, Black's winning strategy in $\mathcal{G}_H$ from $D_H$ is a winning strategy in $\mathcal{G}$. And Red's winning no-memory strategy in $\mathcal{G}_H$ from $C_H$ can be augmented to a winning no-memory strategy in $\mathcal{G}$ because he has a winning no-memory strategy from $Q - H = C' \cup F$.

Thus, in Case II, Red has a winning no-memory strategy from $C' \cup F \cup C_H$, and Black has one from $D_H$. □

The proof of Theorem 6.2 contains an implicit algorithm to find the winning no-memory strategies for the players of a game without splits and the sets from which they are applicable. Certainly this algorithm is no more complex than the general algorithm whose complexity is discussed in Section 5. Investigations have not disclosed any reason to think it can be made substantially simpler.

We close this section by offering a set-theoretic characterization of the no-memory pairs. Where $W = \{w_1, \ldots, w_n\}$, define

$$\Delta_i = \{\alpha \in 2^W \mid w_i \in \alpha \text{ and for all } j < i,\ w_j \notin \alpha\}, \quad \text{for } 1 \leq i \leq n,$$

$$\Delta_{n+1} = \{\emptyset\}.$$

Note that $2^W = \bigcup_{i=1}^{n+1} \Delta_i$.

**Theorem 6.3.** *Where* $W = \{w_1, \ldots, w_n\}$, *let* $S$ *be a subset of the integers between* 1 *and* $n+1$ *inclusive and let* $\Omega = \bigcup_{i \in S} \Delta_i$. *Then* $(W, \Omega)$ *is a no-memory pair.*

**Proof.** Assume the antecedent is true and the consequent false. Then $(W, \Omega)$ has a split, by Theorem 6.2.

*Case* I: $\alpha, \beta \in \Omega$ *but* $\alpha \cup \beta \notin \Omega$. Then $\alpha \in \Delta_i$ and $\beta \in \Delta_j$ for some $i, j \in S$. It follows that $\alpha \cup \beta \in \Delta_{\min(i,j)} \subseteq \Omega$, a contradiction.

*Case* II: $\alpha, \beta \notin \Omega$ *but* $\alpha \cup \beta \in \Omega$. Then $\alpha \cup \beta \in \Delta_i$, for some $i \in S$. It follows that either $\alpha \in \Delta_i$ or $\beta \in \Delta_i \subseteq \Omega$, also a contradiction. □

**Theorem 6.4.** *Let* $W = \{u_1, \ldots, u_n\}$ *and let* $(W, \Omega)$ *be a no-memory pair. Then, for some permutation* $\pi$ *of the set* $\{1, \ldots, n\}$, *taking, for each* $i$, $w_i = u_{\pi(i)}$, *there exists a subset* $S$ *of the integers between* 1 *and* $n+1$ *such that*

$$\Omega = \bigcup_{i \in S} \Delta_i.$$

**Proof.** If $W \in \Omega$ then by Theorem 6.1 there is a $u_j$ such that $\{\alpha \in 2^W \mid u_j \in \alpha\} \subseteq \Omega$. If $W \in 2^W - \Omega$ then (by the same principle), for some $u_j$, $\{\alpha \in 2^W \mid u_j \in \alpha\} \subseteq 2^W - \Omega$. In either case take $\pi(1) = j$, and we have either $\Delta_1 \subseteq \Omega$ or $\Delta_1 \subseteq 2^W - \Omega$, respectively.

As an inductive hypothesis assume that $\pi(1), \pi(2), \ldots, \pi(i)$ have been assigned thus defining $w_1, w_2, \ldots, w_i$ and for each $j \leq i$ either $\Delta_j \subseteq \Omega$ or $\Delta_j \subseteq 2^W - \Omega$. We set $W' = \{u_j \mid j \neq \pi(j') \text{ for any } j' \leq i\}$ and $\Omega' = \{\alpha \mid \alpha \in 2^{W'} \text{ and } \alpha \in \Omega\}$. Since $(W', \Omega')$ is a no-memory pair, there exists $u_j \in W'$ such that either $\{\alpha \mid u_j \in \alpha \in 2^{W'}\} \subseteq \Omega'$ or it is $\subseteq 2^{W'} - \Omega'$. We set $\pi(i+1) = j$ and $w_{i+1} = u_j$, and we have $\Delta_{i+1} \subseteq \Omega$ or $\Delta_{i+1} \subseteq 2^W - \Omega$.

It is clear that at the conclusion of this construction, when all $w_i$'s have been defined, there will be a set $S$ such that

$$\Omega = \bigcup_{i \in S} \Delta_i$$

which is our desired result. □

## 7. Declaring the winner in finite time

If infinite games are to be actually played as games then it must be possible in some way to judge what would probably happen in an infinite play, as indicated somehow by what has actually happened after some finite amount of time. We must be able to declare at a certain point in the play of the game that one of the two players seems to be winning and that the other player may as well concede.

At the same time, if infinite games are to be stimulating to the players playing them then they must be sufficiently complex that either no one is able to figure

out the winning strategy or else no one finds it worth the trouble. Indeed, we might expect a game to be a board game whose graph would be too enormous to construct, like the finite-duration games of *checkers*, *chess* and *go*, which we discussed in Section 1.

Our task in this section is quite difficult: to find, for any given infinite game whose winning strategies we cannot work out, a procedure that will enable us to justifiably declare one of the two players the winner after some finite amount of time.

One obvious suggestion is to wait for the play to settle down in some $\alpha \subseteq W$. When we are convinced that the placemarker has been visiting all the nodes of $\alpha$ repeatedly during a sufficiently long interval when no winning-condition nodes outside $\alpha$ have been visited then we can declare Black or Red the winner according to whether $\alpha \in \Omega$ or $\alpha \in 2^W - \Omega$.

There is something to be said for this suggestion, but there is a complication in the fact that the permset of an infinite play cannot be determnined from any finite interval even if we know that one of the players has, and is using, a winning strategy. Recall the example of Fig. 1 of Section 2 in which $W = \{b_1, r_1\}$ and $\Omega = \{\emptyset, W\}$. Black has the winning strategy from all nodes. But as Black plays this strategy guaranteeing herself a win, it is Red who can determine whether the permset is $\emptyset$ or $W$. Red could consistently and repeatedly move into $b_1$ for an arbitrarily long interval causing Black to respond by consistently moving into $r_1$, from which we would infer that the permset will be $W$. But then Red could suddenly decide to play into $b_2$ and do that forever; Black would forever respond by moving into $r_2$, and the permset would turn out to be $\emptyset$. Our prediction that the permset would be $W$, based on apparently good evidence, would turn out to be false.

We must therefore analyze the matter in greater depth. We let the postive integers designate the times at which moves are made in a play of the game, and we let $node(t)$ be the node of the graph that has the placemarker at time $t$. For $\alpha \subseteq W$, we define $begin(\alpha, t)$ to be the smallest positive integer $t' \leqslant t$ such that

$$\{node(x) \cap W \mid t' \leqslant x \leqslant t\} \subseteq \alpha.$$

Thus $begin(W, t) = 1$; and if $begin(\alpha, t) > 1$ then $node(begin(\alpha, t) - 1) \in W - \alpha$. The function $begin(\alpha, t)$ is undefined if $node(t) \in W - \alpha$.

For $node(t) \notin W - \alpha$ and $\alpha \neq \emptyset$, $score(\alpha, t)$ (the *score* for $\alpha$ at time $t$) is defined to be the maximum nonnegative integer $h$ such that the interval between time $begin(\alpha, t)$ and time $t$ inclusive can be decomposed into $h$ disjoint intervals $I_1, \ldots, I_h$ ($I_i < I_j$ for $i < j$) such that $node(I_i) \cap W = \alpha$ for each $i$; for $node(t) \in W - \alpha$, $score(\alpha, t) = 0$; $score(\emptyset, t)$ is undefined. (By *interval* we mean a set $\{t \mid t_1 \leqslant t \leqslant t_2\}$ where $t_1$ and $t_2$ are the *beginning* and the *end* of the interval. We write $node(I)$ to mean $\{node(t) \mid t \in I\}$.)

Note that if $node(t) \notin W - \alpha$ and $score(\alpha, t) = 0$ then for some $p \in \alpha$, either $p$ was not visited at all by the placemarker at time $t$ or before, or else the

placemarker visited some $q \in W - \alpha$ since the last time it visited $p$. Also note that, for any $\alpha$ and $t$, if $node(t+1) \notin W - \alpha$ then either $score(\alpha, t+1) = score(\alpha, t)$ or $score(\alpha, t+1) = score(\alpha, t) + 1$.

In any infinite play and for any value of $h$, however large, for some $t \geqslant 1$ and $\alpha \in 2^W$, $score(\alpha, t) = h$. In particular, this will be true when $\alpha$ is the permset of the play. But it is also possible for this to be true when $\alpha$ is not the permset.

The question is, under what conditions and for which values of $h$, where $\alpha \in \Omega$ ($\alpha \in 2^W - \Omega$), are we justified in declaring Black (Red) the winner at time $t$ when $score(\alpha, t) = h$? Without loss of generality, let us assume for the remainder of this section that $\alpha \in 2^W - \Omega$. We assume that, because $Q$ is very large, we cannot ascertain precisely the winning strategies for the game, or the nodes from which each player has a winning strategy. On the other hand, we shall assume that the winning strategies are in a certain class of strategies, the two major classes being the class of LVR strategies and the class of no-memory strategies. It is reasonble to expect that there might be an interesting class of strategies between these two. One of these will be suggested at the end of this section, but will not be investigated beyond the mere suggestion.

Let $U_B$ be the set of all nodes from which Black has a winning strategy in $\mathcal{G}$. We assume that from every $p \in Q$, one of the two players has a winning strategy from $p$. Our principle can now be stated:

> Given that Red and Black are playing a game $\mathcal{G}$ for which we know that the winning strategies are in a class $C$, we are not justified in awarding the win to Red at time $t$ on the basis of $score(\alpha, t)$ for $\alpha \in 2^W - \Omega$ unless we can demonstrate the following: Either $\{node(x) \mid begin(\alpha, t) \leqslant x \leqslant t\} \cap U_B = \emptyset$ or, for every winning strategy $f$ of class $C$ for Black from $U_B$, there exists a time $t'$, $begin(\alpha, t) \leqslant t' \leqslant t$, such that Black has not moved according to $f$ at time $t'$ although $node(t') \in U_B$.

For the sake of our discussion we put $h = score(\alpha, t)$. The justification for our principle is simply this: if at some time during the interval the placemarker is in $U_B$ and if Black plays her winning strategy at that time and forever after, then she would win the game and not Red. Therefore, we should not award the game to Red unless $h$ is large enough so that we can be sure that either (1) during the interval Black never had a winning strategy, or (2) whatever winning strategy in the class $C$ she might have played, there was at least one time when she did not play it. In the latter case, there may be several different winning strategies in the class $C$: the value of $h$ must be large enough to show that Black's moves are not consistent with any one of them.

This principle is a necessary condition for awarding the game to Red, not a sufficient condition. There may be other things we should consider. For example, we may insist that $t$ itself be sufficiently large to give both players a chance to learn the game to some degree. Secondly, we may wish to insist that $\alpha$ occur more times than the value of $h$ needed to satisfy the stated principle, so as to test

further that Red knows what he is doing. We may wish to give Black more of a chance to spoil things for Red, etc. We shall not attempt precise formulations of these other considerations.

At any rate, our proposed general method for awarding the game to one of the two players after some finite amount of time involves keeping score for all $\alpha \in 2^W$. If (say) $\alpha \in 2^W - \Omega$ and $score(\alpha, t)$ is sufficiently large then the play is stopped and the win is awarded to Red. The award does not involve a judgment that Red is capable of winning however Black plays in the future. Rather the judgment is that Red would win if Red plays in the future at least as well as he has been playing up to time $t$ and Black plays at least as poorly.

The judgment at time $t$ may have gone against Black because she is not in a position to win and Red is playing sufficiently well to maintain his situation against the way Black is playing. It may be, however, that Black has been in a winning position at times when she played in a faulty manner; the judgment against her in this case reflects her poor playing.

We believe that our proposed general method satisfies the objective formulated at the beginning of this section, even though we have not worked out the details completely. The virtue of the method is that it can be applied without having to compute the winning strategies of the game or anything else requiring computational resources of comparable magnitude. Perhaps we have no right to claim that the method will enable us to pick a probable winner; indeed we shall excuse ourselves for not attempting to define what that means. But we shall be able to claim that if one player (Red or Black) is at any time declared to be the winner of any play of the game then the other player either never was in a position to win or else failed to play strategically at least once when he or she was in a winning position.

We now prove some theorems that demonstrate that our method is sound.

**Theorem 7.1.** *For* $\alpha, \beta \subseteq W$, *if* $score(\alpha, t) > score(\alpha, t-1) > 0$ *and* $score(\beta, t) > score(\beta, t-1) > 0$, *for any* $t \geq 2$, *then* $\alpha = \beta$.

**Proof.** For the proof by contradiction, assume the antecedent and $\alpha \neq \beta$. Without loss of generality, assume also that $w \in \beta - \alpha$. Then $w$ was last visited before time $begin(\alpha, t)$. There are at least two intervals, $I_1, I_2 \subseteq \{x \mid begin(\alpha, t) \leq x \leq t\}$ for which $I_1 < I_2$ and $node(I_1) \cap W = node(I_2) \cap W = \alpha$. Let $t'$ be any time for which $node(\{x \mid t' \leq x \leq t\}) \cap W = \beta$, and let $I_3 = \{x \mid t' \leq x \leq t\}$. Since $w \in \beta - \alpha$, it must be that $t' < begin(\alpha, t)$. Let $I_3' = \{x \mid t' \leq x \leq t-1\}$.

If $node(t) \in W$ then $node(t) \in \alpha$; in that case $node(t)$ must have been visited during $I_1$ which is wholly inside interval $I_3'$. Thus if $I_3$ is used in the set of intervals in the definition of $score(\beta, t)$, $I_3'$ could be used in its place. The same is true if $node(t) \notin W$. We would then have a set of intervals of equal cardinality that could be used for $score(\beta, t-1)$ as well.

It follows that $score(\beta, t-1)$ can be no less than $score(\beta, t)$, a contradiction. □

This theorem shows that whatever specific policy we institute following our general method for declaring a winner, there will never be a time when we would be obliged to declare both players to be the winner, since the first time any set $\alpha \subseteq W$ reaches its desired score cannot be the first time any other set $\beta \subseteq W$ reaches its desired score.

**Theorem 7.2.** *Let $\mathcal{G}$ be a no-memory game. Suppose that $t' > t$ and, in some play of $\mathcal{G}$, $node(t') = node(t)$ and $\{node(t'') \mid t' \leqslant t'' \leqslant t\} \cap W = \alpha \in 2^W - \Omega$. Then either no node of $U_B$ is visited between $t'$ and $t$ or else, for every winning no-memory strategy $f$ for Black, at least one node of $U_B \cap B$ is visited some time when Black did not play according to $f$.*

**Proof.** Assume that Black plays correctly whenever the placemarker is in $U_B \cap B$, and that, for some $t''$, $t' \leqslant t'' \leqslant t$, $node(t'') \in U_B$. Now as long as Black keeps playing her winning strategy at time $t''$ and afterward the placemarker will remain in $U_B$, by Theorem 3.2. Thus $node(t) \in U_B$. But, since $node(t') = node(t)$, it follows by the same reasoning that, for all $t''$ such that $t' \leqslant t'' \leqslant t$, $node(t'') \in U_B$.

Put $t_0 = t'$, $t_1 = t$, $d = t_1 - t_0$, and, for all $h \geqslant 2$, $t_h = t_1 + (h-1)d$. If Black and Red were to play forever after $t_1$, playing between $t_{h-1}$ and $t_h$ (for each $h \geqslant 2$) the way they played between $t_0$ and $t_1$, then the permset would be $\alpha$ and Red would win. But in this play the placemarker is always in $U_B$ and Black is always playing her winning no-memory strategy, a contradiction. □

We note that, if $\emptyset \neq \alpha \in 2^W - \Omega$, $node(t) \in \alpha$ and $score(\alpha, t) = 2$ in a no-memory game, then the antecedent, and hence the consequent, of Theorem 7.2 are satisfied.

**Theorem 7.3.** *Let $\mathcal{G}$ be a game, and let $\emptyset \neq \alpha \in 2^W - \Omega$, $|\alpha| = k$. If $score(\alpha, t) = k! + 1$ then either* (a), *for all $t'$, $begin(\alpha, t) \leqslant t' \leqslant t$ implies $node(t') \notin U_B$, or* (b), *for any winning LVR strategy $f$ for Black, there exists a $t'$, $begin(\alpha, t) \leqslant t' < t$, such that $node(t') \in U_B \cap B$ and Black did not play according to $f$ at time $t'$.*

**Proof.** Assume that (a) and (b) are both false. Then we can prove as in the proof of Theorem 7.2 that $node(t') \in U_B$ for all $t'$ such that $begin(\alpha, t) \leqslant t' \leqslant t$. From $score(\alpha, t) = k! + 1$, it follows that, for some $w \in \alpha$, there must be times $t_1 < t_2 < \cdots < t_{k!+1}$ such that $begin(\alpha, t) \leqslant t_1$, $t_{k!+1} \leqslant t$ and $node(t_h) = w$ for $1 \leqslant h \leqslant k! + 1$. There must be $h_1$ and $h_2$, $0 \leqslant h_1 < h_2 \leqslant k!$, such that $LVR(t_{h_1}) = LVR(t_{h_2})$. Put $j_0 = t_{h_1}$, $j_1 = t_{h_2}$, $d = j_1 - j_0$, and, for $x \geqslant 2$, $j_x = j_0 + xd$. If Red and Black were to play forever, after time $j_1$, so that for each $i \geqslant 1$ they play the same

between times $j_i$ and $j_{i+1}$ as they did between times $j_0$ and $j_1$, then the permset would be $\alpha$. Red would win in spite of the fact that the play is in $U_B$ and Black is playing a winning strategy, a contradiction. □

**Theorem 7.4.** *Let $\mathcal{G}$ be a game, and let $\emptyset \in 2^W - \Omega$. If $t' < t$ and, during a play of $\mathcal{G}$, $node(t') = node(t)$ and $\{node(t'') \mid t' \leq t'' \leq t\} \cap W = \emptyset$ then either no node of $U_B$ is visited between $t'$ and $t$ or else, for every winning LVR strategy $f$ for Black, at least one node of $U_B \cap B$ is visited some time when Black did not play according to $f$.*

The proof is the same, word for word, as the proof of Theorem 7.2.

Comparing Theorem 7.2 about games that have no-memory strategies and Theorem 7.3 about the more general class of games, we see that we must wait much longer to declare a winner in the general game than in a game that has winning no-memory strategies. The question then is, can we find a subfamily of games for which we can get an improvement over our result of Theorem 7.3 but which still encompasses substantially more than the subfamily of no-memory games? Does there exist a class of strategies that is properly between the class of no-memory strategies and the class of LVR strategies that would determine such a family of games?

Our research has not established any such family of games, but we have an idea, and a hope that it might someday prove fruitful. The idea is based on the observation that among winning LVR strategies some are more efficient than others.

For example, consider the game of Fig. 6, in which $W = \{r_1, r_2, r_3, r_4\}$ and $\Omega = \{W\}$. Red has no choices in this game, which is therefore a game of solitaire for Black. Nevertheless, it is a good example to illustrate efficient and inefficient LVR strategies. An efficient LVR strategy for Black would be as follows

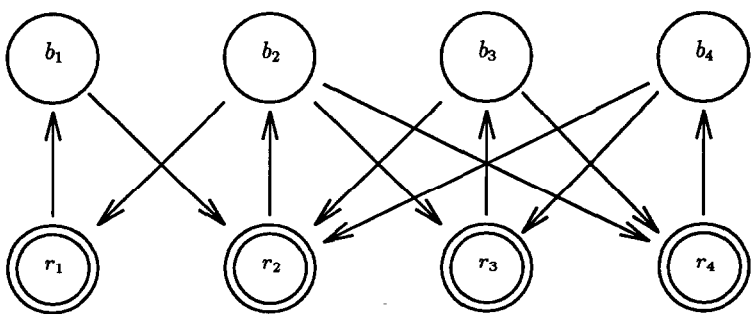


Fig. 6. Inefficient winning strategy.

(omitting $b_1$, from which Black has no choice):

From $b_2$,
  if the LVR is, or ends in, $r_1$, $r_2$, then move to $r_3$;
  otherwise move to $r_1$.
From $b_3$ move to $r_4$.
From $b_4$ move to $r_2$.

This strategy is efficient because it enables Black to cycle through her only winning set as quickly as possible, namely, in a total of ten moves (five for each player).

An inefficient winning LVR strategy for Black in this game is as follows:

From $b_2$,
  if the LVR is, or ends in, $r_3$, $r_4$, $r_2$, then move to $r_4$;
  if the LVR is, or ends in, $r_4$, $r_3$, $r_2$, then move to $r_1$;
  otherwise move to $r_3$.
From $b_3$,
  if the LVR is, or ends in, $r_2$, $r_4$, $r_3$, then move to $r_2$;
  otherwise move to $r_4$.
From $b_4$,
  if the LVR is, or ends in, $r_3$, $r_2$, $r_4$, then move to $r_3$;
  otherwise move to $r_2$.

Let us trace the path of the placemarker assuming that Black uses this strategy and play begins at $r_1$. Noting for each $i$ that $b_i$ always follows $r_i$, and that $r_i$ never follows $b_i$, we abbreviate $r_1$, $b_1$, $b_2$, $r_3$, $b_3$, ... as $1, 2, 3, \ldots$. Thus the trace of the placemarker is 1, 2, 3, 4, 2, 4, 3, 2, 1, 2, 3, 4, 2, 4, 3, 2, 1, .... It thus takes 16 moves to cycle through Black's winning set, compared to 10 moves with the efficient strategy. This play shows a score of 2 for Red's winning set $\{r_2, r_3, r_4\}$ at certain times. When Black played her more efficient LVR strategy, the play showed at any time at most a score of 1 for any of Red's winning sets.

The question is, could a class $C$ of efficient LVR strategies be defined so that a modified Theorem 4.1 and an improved Theorem 7.3 are valid for this smaller class? The improvement in Theorem 7.3 would be a reduction in the quantity $k! + 1$. We conjecture that if such an improvement is possible it would be valid only for some restricted class of games.

## 8. Abstract infinite games

The abstraction here is achieved simply by removing the graph from the infinite game. Each player moves by simply announcing the selection of a letter from an alphabet. We stipulate that Red moves first, and thereafter the players move in turn. In any infinite play of the game an infinite word (i.e., an omega word) over

the alphabet is thereby determined. Winning and losing depends on that infinite word.

In any game we stipulate a finite alphabet $\Sigma$. At any time in the play of the game a (finite) word $u$ has been spelled out by the sequence of the players' selections. To make the game sufficiently general we also stipulate restrictions on the choice of letter each player is allowed to make at various times: we stipulate a function $\phi: \Sigma^* \to 2^\Sigma - \{\emptyset\}$ such that, where $u$ is the word spelled out up to a certain time, $\phi(u)$ is a nonempty subset of $\Sigma$ that tells the next-moving player to chose a letter from the subset $\phi(u)$. Finally, we stipulate a set $\Psi$ of infinite words over $\Sigma$; Black wins if and only if the infinite word determined by the play is in $\Psi$.

Thus we formally define an abstract infinite game $\mathscr{A}$ as an ordered triple $(\Sigma, \phi, \Psi)$ where $\Sigma$, $\phi$ and $\Psi$ are as described above.

It is not difficult to see that infinite graph games can be represented as abstract infinite games. Recall our remark at the beginning of Section 2 that our concept of game is really a set of games, since games are usually conceived as having an initial configuration, i.e., an initial node on the graph where every play is to begin. Given an infinite graph game $\mathscr{G} = (Q, R, B, E, W, \Omega)$ with a designated initial node $q_0 \in R$ we construct $\mathscr{A} = (\Sigma, \phi, \Psi)$ as follows.

Let $d$ be the maximum outdegree of the nodes of $Q$ in $\mathscr{G}$. Let $\Sigma = \{a_1, \ldots, a_d\}$. We label the edges of $E$ as follows: For each $q \in Q$ suppose there are $i$ edges leaving $q$, $i \leqslant d$. Arbitrarily label these edges $a_1, \ldots, a_i$.

We now have a way for the players of $\mathscr{G}$ to move simply by announcing a letter of $\Sigma$. The function $\phi$ of $\mathscr{A}$ is easily defined: For each $u \in \Sigma^*$ let $q$ be the node reached by the walk in the graph of $\mathscr{G}$ that spells out $u$. If $i$ is the outdegree of $q$ then $\phi(u) = \{a_1, \ldots, a_i\}$.

We see that, for every infinite play of the game $\mathscr{G}$ beginning at $q_0$, there is an infinite word over $\Sigma$ that uniquely represents that play. We thus define $\Psi$ in $\mathscr{A}$ to be the set of all infinite words over $\Sigma$ that represent plays whose permsets are in the set $\Omega$.

However, there are abstract infinite games that are not meaningfully represented as infinite graph games. Consider, for example, the game $\mathscr{A} = (\Sigma, \phi, \Psi)$ where $\Sigma = \{0, 1\}$, $\phi(u) = \Sigma$ for all $u \in \Sigma^*$, and $\Psi$ is the set of all infinite words having, for each $i$, both $0^i$ and $1^i$ as subwords. Although the result is not surprising to many readers, we sketch a proof that no infinite graph game corresponds to this abstract infinite game.

Let $\mathscr{G}$ be an infinite graph game in which $|Q| = m$, and, for all $q \in Q$, there are exactly two edges of $E$ leaving $q$, one labeled 0 and the other labeled 1. With these labels, every (finite) walk in the graph spells out a word $u \in \Sigma^*$. Similarly, for every $u \in \Sigma^*$ and $q \in Q$, there is exactly one such walk from $q$ spelling out $u$. Let us say that two finite words $u_1$ and $u_2$ are *equivalent from* $q \in Q$ if the walks from $q$ spelling out $u_1$ and $u_2$ (1) end at the same node and (2) visit exactly the same set of nodes in their duration. Let us say that two infinite words $z_1$ and $z_2$ are *equivalent from* $q$ if the walks from $q$ spelling out $z_1$ and $z_2$ have the same

permset. We state two elementary propositions, leaving the proof in each case to the reader:

**Proposition 1.** *For each $k \geq 2m$ and $q \in Q$ there exists $j$, $m \leq j < 2m$, such that $1^j$ and $1^k$ are equivalent from $q$.*

**Proposition 2.** *For each $q$, if* (a) $z = u_1u_2u_3 \cdots$ *is an infinite word,* (b) *each $u_i$ is a finite word,* (c) *for each $i \geq 1$, the walk from $q_0$ spelling out $u_1 \cdots u_i$ ends at $q_i$, and* (d) *for each $i \geq 1$, $u_i'$ and $u_i$ are equivalent from $q_{i-1}$, then the infinite words $z$ and $z' = u_1'u_2'u_3' \cdots$ are equivalent from $q_0$.*

Now consider the infinite word $z = 010011 \cdots 0^i1^i0^{i+1}1^{i+1} \cdots$. From Propositions 1 and 2 it follows that for some infinite sequence of integers $j_{2m}, j_{2m+1}, \ldots$, where $m \leq j_x < 2m$ for each $x \geq 2m$, the infinite words $z$ and

$$z' = 010011 \cdots 0^{2m-1}1^{2m-1}0^{2m}1^{j_{2m}}0^{2m+1}1^{j_{2m+1}}0^{2m+2}1^{j_{2m+2}} \cdots$$

are equivalent from $q_0$. Now $z$ represents a play of the game $\mathscr{A}$ which is a win for Black, whereas $z'$ represents a play that is a win for Red. But in $\mathscr{G}$, these two words represent wins for the same player. Thus $\mathscr{A}$ is not represented by $\mathscr{G}$. Our proof shows that no infinite graph game represents $\mathscr{A}$.

Abstract infinite games were introduced in 1953 by Gale and Stewart [4]. In their paper they focused on the question, for which sets $\Psi$ of infinite words is the game *determined,* viz., for which such sets does one of the two players have a winning strategy? They gave a rather simple proof of the existence of an undetermined abstract infinite game with $\Sigma = \{0, 1\}$, using the axiom of choice via the theorem that all sets can be well ordered, and well known facts about cardinals and ordinals. Although their proof was simple, their result was remarkable. Keep in mind that an undetermined game is not merely one where neither player has a computable winning strategy; it is one where no mathematical function is a winning strategy, not even a noncomputable one.

Morton Davis continued the investigation of Gale and Stewart in 1964 [3]. Like them he was interested in characterizing those games $(\Sigma, \phi, \Psi)$ that are determined, by focusing on the topological properties of the set $\Psi$ in the infinite tree determined by $\Sigma$. For this it was convenient, and caused no loss of theoretical generality, to restrict the investigation to those in which $\Sigma = \{0, 1\}$ and, for all $u \in \Sigma^*$, $\phi(u) = \Sigma$. The set of plays of such a game is simply the set of infinite words over $\{0, 1\}$.

The topology on this space of infinite words comes about from the following concept of limit point: the infinite word $a_1a_2a_3 \cdots$ is a *limit point* of the set of infinite words $\Gamma$ if, for each $n \geq 1$, there is an infinite word $g_1g_2g_3 \cdots \in \Gamma$ such that, for all $i \leq n$, $a_i = g_i$. The first level of the Borel hierarchy is the union of the class of denumerable unions of closed sets and the class of denumerable intersections of open sets. (Some papers in the literature refer to this as the

second level.) The second level is the union of the class of the denumerable unions of sets in the first level and the class of denumerable intersections of such sets.

Gale and Stewart proved that all games for which $\Psi$ is in the first level of the Borel hierarchy are determined. Morton Davis extended this result to all games for which $\Psi$ is in the second level of the Borel hierarchy.

We have observed above that every infinite graph game with a designated start node can be converted into an equivalent abstract infinite game. It turns out that the $\Psi$ of the latter is in the second level of the Borel hierarchy. (This is implied by [11, Theorem 5.2, p. 154]; that $\Psi$ is a 'regular $\omega$-language' is clear.) Thus we have another proof that, for each node of an infinite graph game, one player has a winning strategy when play begins at that node. However, the proof that we have presented in Section 4 proves more than that since it establishes that every strategy is computable and (as we shall show in the next section) is executable by a finite automaton.

In [14] Zeitman develops a concept of graph game that is a generalization of both abstract infinite games and the infinite graph games of this paper. If her graph is finite then her game is one of ours. If her graph is an infinite rooted tree then her game is, in effect, an abstract infinite game. It is interesting how she handles the general case, since it includes games that are in neither class. Of course, each of her games can be shown to be equivalent to an abstract infinite game by a construction similar to our construction above for graph games whose graphs are finite.

The papers [6], [12] and [13] describe their games in the abstract-infinite-game format, focusing on the infinite tree of moves. Yet they are concerned predominantly with games that are equivalent to the infinite graph games of this paper; thus the problem of characterizing those abstract games that are equivalent to graph games in terms of the proprties of the infinite tree of moves is an important one for the authors of those papers. The paper [13], points out that the graph game is simply 'the transition table of a game automaton', which is (as the next section will try to make clear) a finite automaton. Roughly speaking, a game automaton is a device that is capable of moving along the path of the infinite tree and telling which player is the winner of the play of the game that the path represents.

## 9. Finite automata

The concept of infinite graph game is one of several outgrowths from Richard Büchi's study of a certain formal logical system known as the sequential calculus [1]. That system had two types of variables standing for nonnegative integers and classes of nonnegative integers. As such it was an example of a monadic second-order theory (see [5]). However, it was weaker than most formal systems

of arithmetic in that addition and subtraction could not be defined in it. Its only primitive notions were the successor function and set membership.

We shall not describe the sequential calculus in any detail; we mention it only for historical reference. (The interested reader would do well to read from Sections 5 and 6 of [9].) Büchi proved that every formula of the sequential calculus could be interpreted as saying something about the infinite history of some nondeterministic finite automaton. One of the three open problems at the end of Büchi's paper was the following: given a sequential-calculus formula that makes an assertion about the infinite history of the inputs and outputs of a finite automaton, does there exist a nondeterministic finite automaton that always satisfies that assertion in its every possible infinite history?

Research on this problem of Büchi's naturally led to the concept of infinite graph game (in a somewhat different guise). Landweber's doctoral dissertation [7], written under Büchi's direction, had the first correct proof that it is decidable which of the two players has a winning strategy given (in our terms) an infinite graph game and a designated start node. He used this result to establish the affirmative answer to Büchi's open problem (see also [8] and [2]).

For us, an interesting feature of Landweber's work is that the winning player in an infinite graph game can execute his winning strategy by a finite automaton whose size depends on the size of the game. We explain this in terms introduced in the present paper. The winning strategy mentioned in our main theorem (Section 4) directs the winning player to move in a way that depends on the location in the graph and on the LVR. We construct a finite automaton capable of sensing the location of the placemarker at any time: that is to say, the identity of the node that is the placemarker's present location is an input to the automaton. If the strategy to be played is what we have called a no-memory strategy then the automaton would not need any memory since the output (i.e., the move) would depend only on the node being visited.

On the other hand, if the strategy is an LVR strategy other than a no-memory strategy, then the automaton would have to use the LVR as well as the identity of the node currently visited to determine its move. It would therefore have to remember something about its past. But a complete memory of the past is not needed, just the LVR.

In any game there are only finitely many LVR's, namely, $\Sigma_{i=0}^{n} i!$ where $n$ is the number of winning-condition nodes. Furthermore, as we observed in Section 2, the LVR at any moment of time is determined by the LVR of the immediately previous moment and the new node. Accordingly, an LVR strategy for the player of a game can be taken as a finite automaton, identifying states with the LVR's, the input conditions with the nodes, and the outputs with the moves that the player may make.

**Acknowledgment**

I am grateful to Richard E. Stearns for a critical reading of an early draft of this paper, and for numerous helpful suggestions.

## References

[1] J.R. Büchi, On a decision method in restricted second-order arithmetic, in: Proc. Int. Congr. Logic, Method, and Phil. Sci. of 1960 (Stanford Univ. Press, Stanford, CA, 1962) 1–11. Reprinted in [9].

[2] J.R. Büchi and L.H. Landweber, Solving sequential conditions by finite-state strategies, Trans. Am. Math. Soc. 138 (1969) 295–311. Reprinted in [9].

[3] M. Davis, Infinite games of perfect information, in: Ann. Math. Studies 52—Advances in Game Theory (Princeton Univ. Press, Princeton, NJ, 1964) 85–101.

[4] D. Gale and F.M. Stewart, Infinite games with perfect information, in: Ann. Math. Studies 28—Contributions to the Theory of Games (Princeton Univ. Press, Princeton, NJ, 1953) 245–266.

[5] Y. Gurevich, Monadic second-order theories, in: J. Barwise and S. Feferman, eds., Model-theoretic Logics (Springer, Berlin, 1985).

[6] Y. Gurevich and L. Harrington, Trees, automata and games, in: Fourteenth Sym. on Theory of Computation (Assoc. Comput. Mach., New York, 1982) 60–65.

[7] L.H. Landweber, A design algorithm for sequential machines and definability in monadic second-order arithmetic, PhD dissertation, Purdue Univ. (1967).

[8] L.H. Landweber, Finite-state games—a solvability algorithm for restricted second-order arithmetic, Notices Am. Math. Soc. 14 (1967) 129–130.

[9] S. Mac Lane and D. Siefkes, eds., The Collected Works of J. Richard Büchi (Springer, Berlin, 1990).

[10] A. Nerode, A. Yakhnis and V. Yakhnis, Concurrent programs as strategies in games, Technical Report '90–78, Mathematical Sciences Inst., Cornell Univ. (1990).

[11] W. Thomas, Automata on infinite objects, in: J. van Leeuwen, ed., Handbook of Theoretical Computer Science, B (Elsevier, Amsterdam and MIT Press, Cambridge, MA, 1990) 135–191.

[12] A. Yakhnis and V. Yakhnis, Extension of Gurevich–Harrington's restricted memory determinacy theorem: a criterion for the winning player and an explicit class of winning strategies, Ann. Pure Appl. Logic 48 (1990) 277–297.

[13] A. Yakhnis and V. Yakhnis, Gurevich–Harrington's games defined by finite automata, Ann. Pure Appl. Logic 62 (1993) 265–294.

[14] S. Zeitman, Unforgettable forgetful determinacy, Logic and Computation, to appear.