JOURNAL OF COMBINATORIAL THEORY, Series A 46, 21–38 (1987)

# PSPACE-Hardness of Some Combinatorial Games

AVIEZRI S. FRAENKEL AND ELISHEVA GOLDSCHMIDT

*Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, Israel 76100*

*Communicated by the Managing Editors*

Received May 19, 1986

PSPACE-hardness of four families of win–lose–draw games played on a digraph with blocking, capture, or annihilation rules is proved. Special cases of three of the families are shown to be PSPACE-complete. Further, the PSPACE-completeness of six families of win–lose games in which two players mark or remove nodes of digraphs according to given rules is proved. The first two families contain partizan games, the last eight impartial games. All the games have been defined previously in the literature or are slight variations of such games. 

## 1. INTRODUCTION

In this work we study the complexity of two types of games: The first type are games with blocking, capture, and annihilation rules as introduced by Conway, Fraenkel, and Yesha. The second type includes games in which the players mark or remove nodes according to given rules. We show that these two types of games are PSPACE-hard and PSPACE-complete, respectively.

All games we introduce here belong to the class of two-player perfect-information games without chance moves as discussed in [1, 2, 4, 9]. Those of the first type are win–lose–draw games, those of the second type win–lose games. In all games to be proved PSPACE-complete or PSPACE-hard below, we assume that the first player unable to move is the loser; his opponent the winner. If there is no last move, the outcome is defined to be a draw.

First, we list some PSPACE-complete problems. In Section 2 we prove a lemma from which we can easily deduce that some of the games given here are in PSPACE. In Section 3 we prove PSPACE-hardness of four families of naturally defined and interesting two-player perfect-information board games of the first type. Special cases of three of these families are shown to be PSPACE-complete. We mention that PSPACE-hardness of some combinatorial games were treated previously, for example, in [3, 5, 8, 9]. In



0097-3165/87 $3.00

[6], games of the type considered here or very similar ones were proved to be NP-hard.

In the final Section 4 we prove PSPACE-completeness of six families of naturally defined, interesting two-player perfect-information graph games, i.e., games played on a directed graph by marking nodes or removing them from the graph according to specified rules. Again we mention that the complexities of some similar games were treated previously, for example, in [8] and [11].

Our polynomial transformations will be made from the following three decision problems, which are known to be PSPACE-complete.

(1) QBF [10]. Input is a pair $(A, \xi)$, where $A$ is a Boolean conjunctive-normal-form (CNF) formula and $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ is a list of distinct Boolean variables, including all those occurring in $A$. Two players move alternately. Move $i$ consists of assigning to $\xi_i$ a value of 0 (false) or 1 (true). After $n$ moves, the first player wins if and only if the assignment which has been produced makes $A$ true.

(2) $G$(POS CNF) [9]. Input is a positive CNF formula $A$ (i.e., a CNF formula in which $\bar{x}$ does not occur). A move consists of choosing some variable of $A$ which has not yet been chosen. The game ends after all variables of $A$ have been chosen. The first player wins if and only if $A$ is true when all variables chosen by the first player are set to 1 and all variables chosen by the second player are set to 0.

We define here a similar game $G'$(POS CNF). It is also played on a positive CNF formula. Two players alternate choosing a variable which has not been chosen before and setting it to true or false. Either player can also pass (i.e., do nothing) at any move. The game ends after all variables have been chosen and set, and the first player wins if the formula is true. It is obvious that $G'$(POS CNF) is also PSPACE-complete.

(3) $G$(POS DNF 2) [9]. Input is a positive disjunctive-normal-form (DNF) formula $A$ with at most two variables in any disjunct. A move consists of choosing any variable of $A$ that has not yet been chosen. The loser is the first player after whose move $A$ is true, when the variables which have been played so far are set to 1 and all other variables are set to 0.

*Notation.* (1) All digraphs $G = (V(G), E(G))$ are finite. They may contain directed cycles unless otherwise specified. For $u \in V(G)$, let

$$F_{(G)}(u) = F(u) = \{v \in V(G): (u, v) \in E(G)\}.$$

(2) Throughout, the number of variables $x_i$ of a CNF or DNF formula is denoted by $n$; the number of clauses $c_j$ is denoted by $m$.

Note that adjoining $(x_{n+1} \vee \bar{x}_{n+1})$ to a QBF formula or adding a clause $(x_{n+1} \vee x_{n+2} \vee x_{n+3})$ to $G$(POS CNF) or $G'$(POS CNF) or adding a

clause $x_{n+1}$ to $G$(POS DNF 2) does not affect the outcome of any of the above games. Therefore we may assume that the number $n$ of variables of a Boolean formula used for a polynomial reduction is odd or even at will.

## 2. Games in PSPACE

A *position* $y = (\mathbf{u}, i)$ in the games given here, consists of a subset $\mathbf{u} \subset V(G)$ called a *configuration*, and an integer $i \in \{1, 2\}$ specifying the player whose turn it is to move from the configuration $\mathbf{u}$. In the games given in Section 3 the subset $\mathbf{u}$ consists of nodes on which the tokens reside. In the games given in Section 4 the subset $\mathbf{u}$ denotes the nodes which are marked or which are left in the graph. Most of our games are *impartial*, that is, the set of configurations reachable in one move from $(\mathbf{u}, 1)$ and $(\mathbf{u}, 2)$ are identical for all configurations $\mathbf{u}$. In this case a position is completely specified by the configuration, and the labels 1, 2 can be disposed of. Only our first two games are *partizan*, that is, not necessarily impartial.

The players play alternately. It is known [4, 9] that under these conditions the set of the game's positions can be partitioned uniquely into three mutually disjoint subsets N, P, and D. They have the following properties: $y \in \mathrm{N}$ if the Next player, that is, the player moving from position $y$, can win independently of his opponent's moves; $y \in \mathrm{P}$ if the Previous player, that is the opponent of the next player, can win independently of his opponent's moves; and $y \in \mathrm{D}$ if no player can force a win from $y$ and therefore both can Draw and avoid losing.

We now prove a general result on membership of games in PSPACE. From it we can deduce easily that the games in 3.1, 3.3, and 3.4, when restricted to directed acyclic graphs, are in PSPACE, as well as the games of Section 4.

Lemma 1. *Let $E$ be a perfect-information two-player partizan game with outcomes restricted to lose, win, or draw. Assume there is a rule whereby for each terminal position $t$ the outcome for each player can be determined by scanning $t$ (so $t$ does not have to be looked up in a table). Let $I$ be a position of $E$ with $|I| = n$. Suppose that every position reachable from $I$ has at most $f(n)$ followers where every follower can be constructed from the previous follower in polynomial space, and that every game starting from $I$ lasts at most $g(n)$ distinct moves. If $g$ is polynomial and $f$ is bounded by $c^{p(n)}$ ($c > 1$ a constant, $p$ a polynomial), then $\Pi(I) \in$ PSPACE, where $\Pi(I)$ is the problem: determine whether $I$ is a win, lose, or draw for the first player.*

*Proof.* Let $R$ be the "game-tree" of $I$. This is a digraph whose nodes map into $I$ and the positions reachable from $I$ (the game starting at $I$),

where $(u, v)$ is a directed edge from $u$ to $v$ if and only if there is a move from position $u$ to position $v$. The root of the tree is $I$ and its levels are labeled alternately by *first* and *second*, according to whether the first or the second player moves from the level. (Some of the nodes of $R$ may correspond to the same game position, arrived at by different sequences of moves.) Every node of $R$ thus has out-degree $\leqslant f(n)$, and $R$ has *depth* (= number of levels) at most $g(n)$. Hence there are at most $f(n)^{g(n)}$ paths from the root to the *leaves* (= terminal positions). Each such path can be described by a sequence $a_1 a_2 \cdots a_{g(n)}$, $a_i \in \{1, 2, \ldots, f(n)\}$ $(1 \leqslant i \leqslant g(n))$. Such a sequence can be encoded by the binary representation of the value $a_i$ and of the index $i$, requiring at most $\lceil p(n) \log c \rceil$ and $\lceil \log g(n) \rceil$ bits, respectively. Hence the entire path encoding has length $O(p(n)\, g(n) \log g(n))$, so the path encoding is in PSPACE.

Labeling the nodes of the graph by N, P, or D can be done in a bottom–up (endorder) traversal. In this way, the label of each node is decided after all the labels of its followers have already been decided, and the memory requirement is only the stack size $O(p(n)\, g(n) \log g(n))$. The label value N, P, or D, is decided by methods of classical game theory (see, e.g., [1, 2, 4]). ∎

## 3. Board Games with Tokens

The games given here are played on a directed graph in the following manner. At the start of the game, a finite number of tokens is placed on the nodes of the graph. A move consists of selecting exactly one token and moving it from its node along one of the directed edges to a new node, according to the rules of the specific game. Players alternate moves, and play continues until one of the players is unable to make a move. The player first unable to move is the loser, his opponent the winner. If there is no last move, the outcome is defined to be a draw.

The first two games of this type are partizan, and the last two are impartial. The games in 3.1, 3.3, and 3.4 are proved to be PSPACE-hard even when restricted to directed acyclic graphs. With this restriction they are, in fact, PSPACE-complete.

### 3.1. *Edge-Blocking*

Tokens are placed on distinct nodes of a digraph $G = (V, E)$, whose edges are distributed into two sets $E_1$ and $E_2$ (not necessarily disjoint), such that $E_1 \cup E_2 = E$. The tokens are of one type only. A move of player $i$

consists of selecting a token from some $u \in V$ and moving it to some *unoccupied* $v \in F_i(u)$ $(i \in \{1, 2\})$. Here and below we adopt the notation:

$$F_i(u) = \{v: (u, v) \in E_i\} \qquad (i \in \{1, 2\}).$$

A configuration of the game is given by $\mathbf{u}$, the subset of $V$ on which tokens reside. The game is partizan when $E_1 \neq E_2$. The decision problem EDGE-BLOCKING is whether $(\mathbf{u}, i) \in \mathrm{N}$ for an arbitrary position $(\mathbf{u}, i)$ $(i \in \{1, 2\})$. The SIMPLIFIED-EDGE-BLOCKING decision problem is defined by restricting $G$ to be a directed acyclic graph (DAG).

THEOREM 1. *The problem* $G$(POS CNF) *is polynomial reducible to* SIMPLIFIED-EDGE-BLOCKING. *Hence the latter is* PSPACE-*complete and consequently* EDGE-BLOCKING *is* PSPACE-*hard.*

*Proof.* Given an instance of the $G$(POS CNF) problem with an even number of variables. We construct a DAG $G = (V, E)$ with subsets $E_1, E_2$ of $E$ satisfying $E_1 \cup E_2 = E$, and a configuration $\mathbf{u}$ in the game played on $G$, such that $(\mathbf{u}, 1) \in \mathrm{N}$ if and only if the first player wins in $G$(POS CNF),

$$V = \bigcup_{i=1}^{n} \{x_i, a_i, \bar{a}_i\} \cup \{b_1, b_2, b_3\} \cup \bigcup_{j=1}^{m} \left( \{c_j\} \cup \bigcup_{x_i \text{ is in } c_j} \{d_{ij}\} \right),$$

$$F_1(x_i) = \{a_i\} \cup \{b_1\}, \qquad F_2(x_i) = \{\bar{a}_i\} \cup \{b_1\} \quad (1 \leqslant i \leqslant n),$$

$$F_1(b_1) = \{b_2\}, \qquad F_2(b_1) = \{b_3\}, \qquad F_2(b_2) = \bigcup_{j=1}^{m} \{c_j\}, \qquad F_1(b_3) = \{b_2\},$$

$$F_1(c_j) = \bigcup_{x_i \text{ is in } c_j} \{d_{ij}\} \qquad (1 \leqslant j \leqslant m),$$

$$F_2(d_{ij}) = \{a_i\} \qquad (x_i \text{ is in } c_j,\ 1 \leqslant j \leqslant m),$$

$$\mathbf{u} = \{b_1\} \cup \{x_1, x_2, \ldots, x_n\}.$$

In Fig. 1 and below edges of type 1 are represented by solid lines, those of type 2 by dotted lines. Tokens are represented by solid circles in Fig. 1. Also, in most of the following proofs, the construction will be indicated only by an example, skipping the formal definition of the constructed graph, as it is very easy to reconstruct the formal definition from the example.

Each player in turn chooses a variable. In normal play, the first player always gives a "true" assignment to the variable he chooses (by moving the token from $x_i$ to $a_i$), and the second player always gives a "false" assignment to the variable he chooses (by moving from $x_i$ to $\bar{a}_i$). After all variables have been chosen, the first player moves the token from node $b_1$ to $b_2$. Then the second player moves the token from $b_2$ to one of the clauses, say $c_j$. If $c_j$ is satisfied, say by $x_i$, the first player moves the token from $c_j$ to $d_{ij}$ and wins. If the clause in not satisfied, regardless of which $d_{ij}$

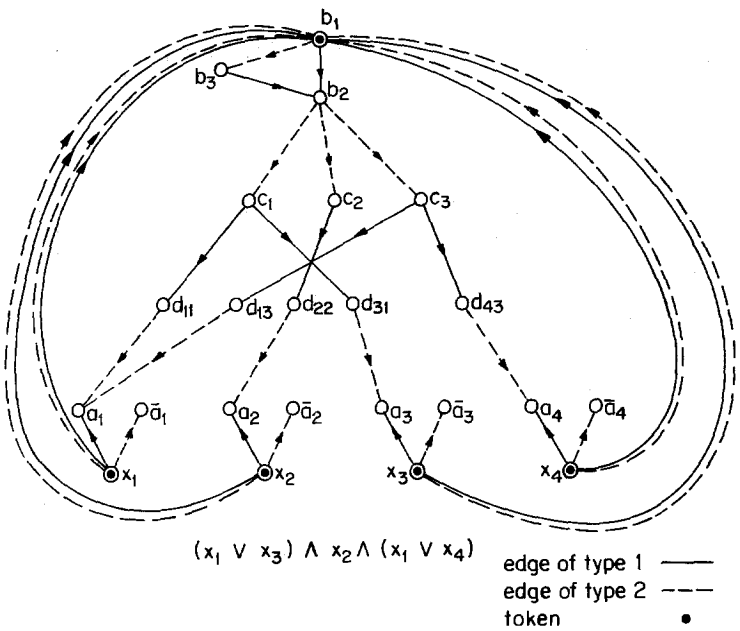


FIG. 1. SIMPLIFIED-EDGE-BLOCKING is PSPACE-complete.

the first player decides to move the token to, the second player will move the token from that $d_{ij}$ to $a_i$ and win. See Fig. 1.

Assume now that the first player moves the token from $b_1$ to $b_2$ before all variables have been chosen. The second player will reply by moving one of the tokens from the $x$'s to $b_1$. Then the first player will be able to reply only by moving tokens from the $x$'s to the $a$'s. Also the second player will move tokens from the $x$'s to the $a$'s as long as possible. When all the tokens from the $x$'s will have been moved, the second player will win by moving the token from $b_1$ to $b_3$. Assume now that the second player moves the token from $b_1$ to $b_3$ before all the tokens from the $x$'s have been removed. The first player will reply by moving one of the tokens from the $x$'s to $b_1$. Then the second player will be able to reply only by moving tokens from the $x$'s to the $a$'s. The first player will do the same. When all the tokens from the $x$'s will have been moved the first player wins.

Finally, every position of EDGE-BLOCKING played on $G = (V, E)$ has at most $|V|^2$ options. If $G$ is a DAG, every EDGE-BLOCKING game played on it lasts at most $|V|^2$ moves. Therefore SIMPLIFIED EDGE-BLOCKING $\in$ PSPACE by Lemma 1. ∎

The last argument of the proof holds also for the games in 3.3 and 3.4, and will therefore not be repeated there.

### 3.2. *Node-Blocking*

Tokens of two types are placed on distinct nodes of a digraph $G=(V, E)$. The edges are of one type only. A move of player $i$ consists of selecting a token of type $i$ and moving it to some *unoccupied* $v \in V$. The game was introduced in [6] as BLOCKING, and proved NP-hard there. A configuration of this (partizan) game is a vector $\mathbf{u}=(\bar{u}_1, \bar{u}_2)$, where $\bar{u}_i$ is the subset of nodes occupied by tokens of type $i$, and $\bar{u}_1 \cap \bar{u}_2 = \varnothing$. The decision problem NODE-BLOCKING is whether $(\mathbf{u}, i) \in \mathrm{N}$ for an arbitrary position $(\mathbf{u}, i)$ $(i \in \{1, 2\})$.

THEOREM 2. *The problem* QBF *is polynomial reducible to* NODE-BLOCKING. *Hence the latter is* PSPACE-*hard.*

*Proof.* Given an instance of the QBF problem with an even number of variables. We construct a digraph $G=(V, E)$ and a configuration $\mathbf{u}$ in the game played on $G$, such that $(\mathbf{u}, 1) \in \mathrm{N}$ if and only if the first player can

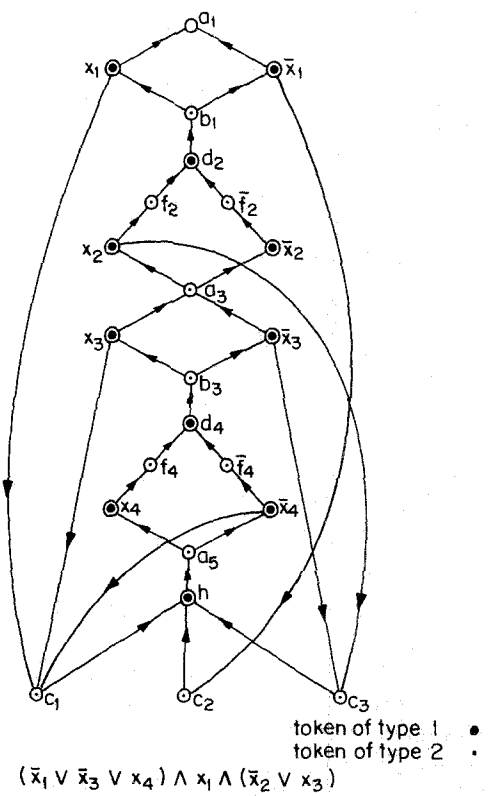


FIG. 2. NODE-BLOCKING is PSPACE-hard.

win in QBF. The construction is obvious from Fig. 2, in which tokens of type 1 are represented by solid circles, those of type 2 by dots. The same convention for tokens is adopted in the sequel. The configuration is defined by $\mathbf{u} = (\bar{u}_1, \bar{u}_2)$, where

$$\bar{u}_1 = \bigcup_{i=1}^{n} \{x_i, \bar{x}_i\} \cup \bigcup_{2i=2}^{n} \{d_{2i}\} \cup \{h\},$$

$$\bar{u}_2 = \bigcup_{2i+1=3}^{n+1} \{a_{2i+1}\} \cup \bigcup_{2i+1=1}^{n-1} \{b_{2i+1}\} \cup \bigcup_{2i=2}^{n} \{f_{2i}, \bar{f}_{2i}\} \cup \bigcup_{j=1}^{m} \{c_j\}.$$

Note the difference in graph construction between even and odd numbered literals. Also $(x_i, c_j) \in E$ if and only if $\bar{x}_i$ is in $c_j$ and vice versa.

The only unoccupied node is $a_1$ (see Fig. 2). The first player moves the token either from $x_1$ or $\bar{x}_1$ to $a_1$, then the second player moves the token from $b_1$ to the vacancy, the first player moves the token from $d_2$ to $b_1$ and then the second player moves the token either from $f_2$ or $\bar{f}_2$ to $d_2$ and so on. In general, if the first player plans to make true $c_j$ containing $x_i$ ($\bar{x}_i$), he will move the token from $x_i$ ($\bar{x}_i$) to $a_i$. If the second player plans to make false some clause $c_j$ containing $x_i$ ($\bar{x}_i$), he will move the token from $\bar{f}_i$ ($f_i$) to $d_i$. This phase ends when the second player moves the token on $a_{n+1}$ to the only unoccupied node (either $x_n$ or $\bar{x}_n$). Then the first player moves the token from $h$ to $a_{n+1}$ and then the second player chooses a clause $c_j$ by moving the token from node $c_j$ to $h$. If the clause has not been satisfied then the first player cannot move because all the tokens on nodes dominating this clause are of type 2, and so the first player loses. If the clause has been satisfied, then there is at least one token of type 1 on one of the $x_i$'s or $\bar{x}_i$'s which dominates the unoccupied $c_j$, so the first player moves that token from that $x_i$ ($\bar{x}_i$). The $a_i$ or $b_i$ dominating the unoccupied $x_i$ has a token of type 1, so the second player cannot move and the first player wins. ∎

### 3.3. *Hit*

Tokens of $r$ different types are placed on distinct nodes of a digraph $G = (V, E)$, whose edges are distributed into $r$ sets $E_1, \ldots, E_r$ (not necessarily disjoint) such that $\bigcup_{i=1}^{r} E_i = E$. A move consists of selecting a token of type $i$ on some $u \in V$ and moving it to some $v \in F_i(u)$ if $v$ is not occupied by a token of type $i$. If $v$ is occupied by a token of type other than $i$, say $j$, then the token of type $j$ is removed. A configuration of the game is represented by a vector $\mathbf{u} = (\bar{u}_1, \ldots, \bar{u}_r)$, where $\bar{u}_i$ is the subset of nodes occupied by tokens of type $i$ $(1 \leqslant i \leqslant r)$, and $\bar{u}_i \cap \bar{u}_j = \varnothing$, $(1 \leqslant i < j \leqslant r)$.

The game is impartial. A position is therefore completely specified by the configuration. The decision problem HIT is whether $\mathbf{u} \in \mathrm{N}$ for an arbitrary configuration $\mathbf{u}$. We define the SIMPLIFIED-HIT decision problem by restricting $G$ to be a DAG, $r = 2$, and $E_1 \cap E_2 = \varnothing$.

THEOREM 3. *The problem* QBF *is polynomial reducible to* SIMPLIFIED-HIT. *Hence the latter is* PSPACE*-complete and consequently* HIT *is* PSPACE*-hard.*

*Proof.* Given an instance of the QBF problem with an even number of variables, we construct a DAG $G=(V, E)$ with subsets $E_1, E_2$ of $E$ satisfying $E_1 \cup E_2 = E$, $E_1 \cap E_2 = \emptyset$, and a configuration $\mathbf{u}$ in the game played on $E$, such that $\mathbf{u} \in \mathrm{N}$ if and only if the first player wins in QBF

$$\bar{u}_1 = \{a_1\},$$
$$\bar{u}_2 = \{x_1, \bar{x}_1, ..., x_n, \bar{x}_n\} \cup \{c_1, c_2, ..., c_m\} \cup \{b_n\}, \qquad \mathbf{u} = (\bar{u}_1, \bar{u}_2).$$

Note that in the first $3n-1$ steps the only way to play is by moving the token of type 1 until it reaches $b_n$, and note also that the steps from $a_i$ (choosing the truth assignment for $x_i$) are taken alternately between the two players (see Fig. 3).

The first player attempts to move the token of type 1 so that it will capture a token of type 2 on some $x_i$ in $c_j$ or $\bar{x}_i$ in $c_j$ for every $j=1,..., m$; the second player attempts to move it such that it does not capture any token on $x_i$ in $c_j$ and $\bar{x}_i$ in $c_j$ for some $j$. Eventually the first player reaches $b_n$ and then the second player chooses a clause $c_j$ and captures the token of type 1 with the token on $c_j$. Then the first player can win if and only if one of the

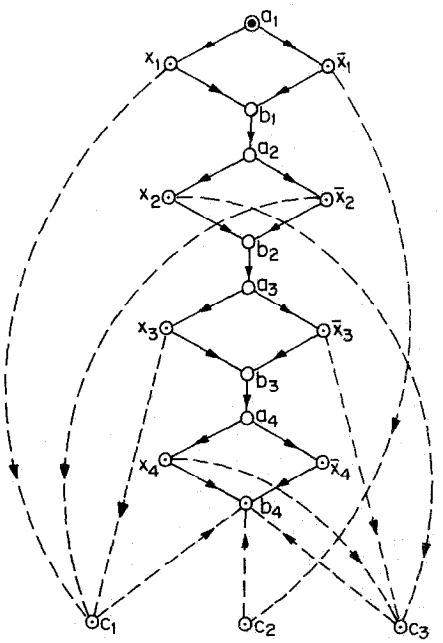


FIG. 3. SIMPLIFIED-HIT is PSPACE-complete.

tokens on the $x$'s and $\bar{x}$'s dominating $c_j$ has not been removed. This means that the clause has been satisfied. If the clause has not been satisfied, then all the tokens on $x$'s and $\bar{x}$'s dominating that $c_j$ have been removed and the second player wins. ∎

### 3.4. *Annihilation*

Tokens of $r$ different types are placed on distinct nodes of a digraph $G=(V, E)$, whose edges are distributed into $r$ sets $E_1,..., E_r$ (not necessarily disjoint), such that $\bigcup_{i=1}^{r} E_i = E$. A move consists of selecting a token of type $i$ on some $u \in V$ and moving it to some $v \in F_i(u)$. If $v$ is occupied by a token of any type, both tokens are annihilated and removed from the game.

A configuration of the game is represented by a vector $\mathbf{u}=(\bar{u}_1,..., \bar{u}_r)$ where $\bar{u}_i$ is the subset of nodes occupied by tokens of type $i$ $(1 \leqslant i \leqslant r)$, and $\bar{u}_i \cap \bar{u}_j = \varnothing$ $(1 \leqslant i < j \leqslant r)$. The game is impartial. A position is therefore completely specified by the configuration. The decision problem ANNIHILATION is whether $\mathbf{u} \in \mathrm{N}$ for any given configuration $\mathbf{u}$. We define the SIMPLIFIED-ANNIHILATION decision problem by restricting $G$ to be a DAG, $r=2$, and $E_1 \cap E_2 = \varnothing$. SIMPLIFIED-ANNIHILATION was proved NP-hard in [6]. The case $r=1$ was proved to be polynomial ($O(n^6)$) in [7], and provided the motivation for both [6] and the present paper.

THEOREM 4. *The problem* $G'$(POS CNF) *is polynomial reducible to* SIMPLIFIED-ANNIHILATION. *Hence the latter is* PSPACE-*complete and consequently* ANNIHILATION *is* PSPACE-*hard.*

*Proof.* Given an instance of the $G'$(POS CNF) problem with an even number of variables, we construct a DAG $G=(V, E)$ with subsets $E_1$, $E_2$ of $E$ satisfying $E_1 \cup E_2 = E$, $E_1 \cap E_2 = \varnothing$, and a configuration $\mathbf{u}$ in the game played on $G$, such that $\mathbf{u} \in \mathrm{N}$ if and only if the first player wins in $G'$(POS CNF). See Fig. 4.

The nodes $\{d_1,..., d_{n+1}\}$ are called the "chain." Say that the first player has a winning strategy for $G'$(POS CNF). The first player plays this strategy using type 1 tokens. Moving a token from $a_i$ to $x_i$ ($\bar{x}_i$) corresponds to setting $x_i$ to true (false). If the second player advances the type 2 token along the chain, the first player views this as though the second player passed in the game $G'$(POS CNF), and responds according to his assumed winning strategy. Consider the first time in SIMPLIFIED-ANNIHILATION when the second player has just moved and all type 1 tokens have been moved. At this time the type 2 token is on a chain node, and

the distance from the type 2 token to $h$ is odd. (*)

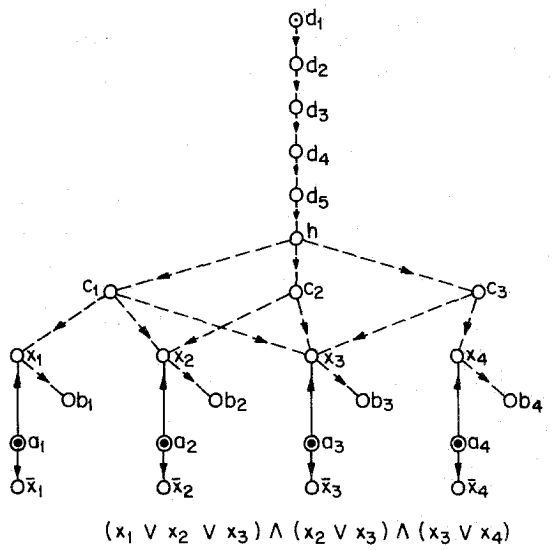


FIG. 4. SIMPLIFIED-ANNIHILATION is PSPACE-complete.

The latter fact is true because the total number of moves made in the game up to this point is even, and an even number of this total, moved type 1 tokens, so an even number of this total advanced the type 2 token along the chain. From this point on, the players must take turns advancing the type 2 token along the chain. The first player moves it from $d_{n+1}$ to $h$ (because of (*)), and the second player moves it from $h$ to some $c_j$. If the clause $c_j$ is satisfied, the first player moves the type 2 token to some node $x_i$ containing a type 1 token, both tokens are annihilated, and the first player wins. If the clause is not satisfied, then the first player moves the token to any unoccupied $x_i$ in clause $c_j$ and the second player wins by moving this token to $b_i$.

The case where the second player wins $G'$(POS CNF) is the same. In this case it does not help the first player to advance along the chain prior to setting all the truth values. ∎

## 4. GAMES OF MARKING OR REMOVING NODES

The games given here are also played on directed graphs. Players alternate moves, and the play continues until one of the players is unable to make a move. The player first unable to move loses.

We compare here three different types of games, each with three versions of rules. The game are: Geography, Geodesic, and Úlehla-Geodesic, each with some restriction on the length of the path(s) which are being marked

or removed in each step. The games here are all impartial, so a position of a game is completely specified by the configuration.

### 4.1. *Geographic Games*

UNBOUNDED-GEOGRAPHY is a game played on a directed graph. The first player chooses an unmarked node and marks it. Each player in turn chooses an unmarked node which is accessible by an unmarked (directed) path from the last node which was chosen and marks it and all nodes on the shortest unmarked path(s) from the node the previous player chose. A configuration of the game is represented by $\mathbf{u} = (\bar{u}, v)$, where $\bar{u}$ is the subset of nodes already marked and $v$ is the node which was chosen last.

FIXED-GEOGRAPHY is the same game as UNBOUNDED-GEOGRAPHY with the restriction that only moves which cause a nonempty set of paths of exactly length $k$ to be marked are permissible. FIXED-GEOGRAPHY for $k=1$ when the initial position consists of exactly one marked node has been known as GENERALIZED-GEOGRAPHY and was proved to be PSPACE-complete in [9].

LOWER-BOUND-GEOGRAPHY is again the same game as UNBOUNDED-GEOGRAPHY, with the restriction that only moves which cause a nonempty set of paths of length at least $k$ to be marked are permissible.

#### 4.1.1. *Unbounded-Geography*

The decision problem UNBOUNDED-GEOGRAPHY is whether $\mathbf{u} \in \mathrm{N}$ for any given configuration $\mathbf{u}$.

THEOREM 5. *The problem* $G$(POS DNF 2) *is polynomial reducible to* UNBOUNDED-GEOGRAPHY. *Hence the latter is* PSPACE-*complete.*

*Proof.* Given an instance of the $G$(POS DNF 2) problem where all the conjuncts contain exactly two variables (each conjunct which has one variable $x_i$ can be extended to $x_i \wedge x_i$), we construct a digraph $G = (V, E)$ such that the first player can win UNBOUNDED-GEOGRAPHY on this graph if and only if he can win $G$(POS DNF 2),

$$V = \bigcup_{i=1}^{n} \{x_i\} \cup \bigcup_{j=1}^{m} \{c_j\} \cup \bigcup_{x_i \text{ is in } c_j} \{e_{ij}\} \cup \bigcup_{x_i \wedge x_i \text{ is clause } c_j} \{e'_{ij}\},$$

$$F(x_i) = \bigcup_{j \neq i} \{x_j\} \cup \bigcup_{x_i \text{ is in } c_j} \{e_{ij}, e'_{ij}\} \cup \bigcup_{x_i \text{ is not in } c_j} \{c_j\} \qquad (1 \leqslant i \leqslant n),$$

$$F(c_j) = \bigcup_{x_i \text{ is in } c_j} \{x_i\} \qquad (1 \leqslant j \leqslant m),$$

$$F(e_{ij}) = F(e'_{ij}) = \{c_j\} \qquad (x_i \text{ is in } c_j).$$

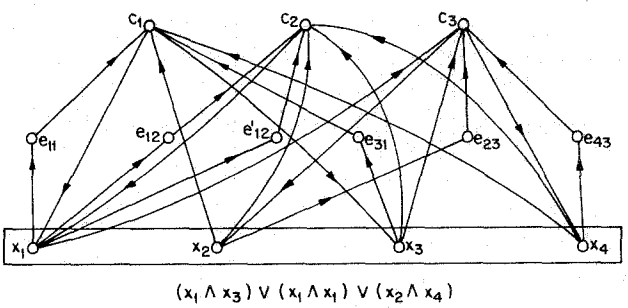


FIG. 5. UNBOUNDED-GEOGRAPHY is PSPACE-complete.

In Fig. 5, a curve enclosing a set of nodes represents a directed complete subgraph of these nodes (for every $u$ and $v$ in the subgraph there is an edge $(u, v)$ and $(v, u)$).

In normal play, the players alternately choose a variable which has not yet been chosen. After one of the players causes the formula to be true (i.e., there is a conjunct $c_j$ all of whose variables have been chosen), his opponent chooses $c_j$ and wins (see Fig. 5). We will discuss here all possibilities of not playing according to normal play:

(1) One of the players chooses $c_j$ when the clause $c_j$ is not yet satisfied (say $x_i$ which is in $c_j$ has not yet been chosen). His opponent will choose $e_{ij}$. The shortest path from $c_j$ to this $e_{ij}$ is through $x_i$. The only follower of $e_{ij}$ is $c_j$ which is already marked, and the opponent wins.

(2) One of the players chooses $e_{ij}$ or $e'_{ij}$. If after this move there is still some $x_k$ which is in $c_j$ that has not yet been marked, his opponent will choose $e_{kj}$ (or $e'_{kj}$). Note that there is always such $e_{kj}$ or $e'_{kj}$ even if $k = i$ and $e_{ij}$ is the first node marked. Since the shortest directed path from $e_{ij}$ to $e_{kj}$ passes through $c_j$ and since the only follower of $e_{kj}$ (or $e'_{kj}$) is $c_j$, the opponent wins. If all $x_k$ which are in $c_j$ are already marked, his opponent will choose $c_j$ and win.

Finally, every node of UNBOUNDED-GEOGRAPHY played on $G = (V, E)$ has at most $|V|$ children and every such game lasts at most $|V|$ moves. Therefore UNBOUNDED-GEOGRAPHY is in PSPACE by Lemma 1. ∎

The last argument of the proof holds also for the games proved PSPACE-hard in the rest of this section, and is therefore omitted.

COROLLARY 1. LOWER-BOUND-GEOGRAPHY *is* PSPACE-*complete.*

*Proof.* LOWER-BOUND-GEOGRAPHY when restricting the paths to be marked in each step to be at least of length 1, is the same game as UNBOUNDED-GEOGRAPHY. Since the latter has been proved to be PSPACE-complete, so is LOWER-BOUND-GEOGRAPHY. ▮

### 4.2. *Geodetic Games*

In geodetic games played on directed graphs, each player in turn chooses an unmarked node $u$ and marks $u$ and all nodes on all the shortest directed paths between $u$ and the already marked nodes, where now a *directed path* between nodes $a$ and $b$ is any path directed either from $a$ to $b$ or from $b$ to $a$. A configuration of the game is represented by $\mathbf{u}$, where $\mathbf{u}$ is the subset of nodes already marked.

#### 4.1.2. *Fixed-Geodesic*

Given a directed graph $G=(V, E)$ with a distinct node $h$ which is marked. A move consists of choosing an unmarked node $u$ such that the shortest directed path between $u$ and the already marked nodes is exactly of length $k$, and then in marking all nodes on all directed paths of length $k$ between $u$ and the previously marked nodes. The decision problem FIXED-GEODESIC is whether $\mathbf{u} \in \mathrm{N}$ for any given configuration $\mathbf{u}$.

We define the SIMPLIFIED-FIXED-GEODESIC decision problem by restricting the graph $G$ to be without directed cycles (DAG).

THEOREM 6. *The problem* QBF *is polynomial reducible to* SIMPLIFIED-FIXED-GEODESIC. *Hence the latter and consequently also* FIXED-GEODESIC *are* PSPACE-*complete.*

*Proof.* Given an instance of the QBF problem with an odd number of variables. We construct a DAG $G=(V, E)$ with a distinct node $h$ which is marked, such that the first player can win in SIMPLIFIED-FIXED-GEODESIC with $k=2$ if and only if he can win in QBF

$$V=\bigcup_{i=1}^{n}\{x_i, \bar{x}_i\} \cup \bigcup_{i=0}^{n}\{a_i\} \cup \bigcup_{j=1}^{m}\{c_j, c_j'\} \cup \{h\}$$

$$\cup \bigcup_{x_{2i+1} \text{ is in } c_j}\{d_{2i+1,j}, e_{2i+1,j}, f_{2i+1,j}\} \cup \bigcup_{\bar{x}_{2i} \text{ is in } c_j}\{g_{2i,j}\}$$

$$\cup \bigcup_{\bar{x}_{2i+1} \text{ is in } c_j}\{\bar{d}_{2i+1,j}, \bar{e}_{2i+1,j}, \bar{f}_{2i+1,j}\} \cup \bigcup_{x_{2i} \text{ is in } c_j}\{\bar{g}_{2i,j}\},$$

$$F(h)=\{a_0\},$$

$$F(x_{2i})=\{a_{2i}, a_{2i-1}\} \cup \bigcup_{\bar{x}_{2i} \text{ is in } c_j}\{g_{2i,j}\},$$

$$F(\bar{x}_{2i}) = \{a_{2i}, a_{2i-1}\} \cup \bigcup_{x_{2i} \text{ is in } c_j} \{\bar{g}_{2i,j}\} \qquad (1 \leqslant 2i \leqslant n-1),$$

$$F(a_{2i}) = \{x_{2i+1}, \bar{x}_{2i+1}\} \qquad (0 \leqslant 2i \leqslant n-1),$$

$$F(a_{2i+1}) = \{x_{2i+1}, \bar{x}_{2i+1}\} \qquad (1 \leqslant 2i+1 \leqslant n),$$

$$F(c_j) = \{a_n, c'_j\} \qquad (1 \leqslant j \leqslant m),$$

$$F(c'_j) = \bigcup_{x_{2i+1} \text{ is in } c_j} \{d_{2i+1,j}\} \cup \bigcup_{\bar{x}_{2i+1} \text{ is in } c_j} \{\bar{d}_{2i+1,j}\}$$

$$\cup \bigcup_{x_{2i} \text{ is in } c_j} \{\bar{g}_{2i,j}\} \cup \bigcup_{\bar{x}_{2i} \text{ is in } c_j} \{g_{2i,j}\} \qquad (1 \leqslant j \leqslant m),$$

$$F(e_{2i+1,j}) = \{d_{2i+1,j}\} \qquad (x_{2i+1} \text{ is in } c_j),$$

$$F(\bar{e}_{2i+1,j}) = \{\bar{d}_{2i+1,j}\} \qquad (\bar{x}_{2i+1} \text{ is in } c_j),$$

$$F(f_{2i+1,j}) = \{e_{2i+1,j}\} \cup \{x_{2i+1}\} \qquad (x_{2i+1} \text{ is in } c_j),$$

$$F(\bar{f}_{2i+1,j}) = \{\bar{e}_{2i+1,j}\} \cup \{\bar{x}_{2i+1}\} \qquad (\bar{x}_{2i+1} \text{ is in } c_j).$$

The first player can only mark either $x_1$ or $\bar{x}_1$. Then the second player can only choose either $x_2$ or $\bar{x}_2$ and so on. In general, if the first player plans to make true a clause $c_j$ containing $x_i$ ($\bar{x}_i$), he will mark $x_i$ ($\bar{x}_i$). If the second player plans to make false some clause $c_j$ containing $x_i$ ($\bar{x}_i$), he will mark $\bar{x}_i$ ($x_i$). After the first player selected $x_n$ or $\bar{x}_n$, the second player can only choose a clause, say $c_j$. If $c_j$ is satisfied by some $x_i$ ($\bar{x}_i$) when $i$ is even, the first player can play on $\bar{g}_{ij}$ ($g_{ij}$) since $\bar{x}_i$ ($x_i$) is not marked, so the shortest path to the marked nodes is to $c_j$. This is a last move, so the first player wins. If this clause is satisfied by some $x_i$ ($\bar{x}_i$) when $i$ is odd, the first player can play on $d_{ij}$ ($\bar{d}_{ij}$) and win, since $x_i$ ($\bar{x}_i$) is marked so the second player cannot play on $f_{ij}$ ($\bar{f}_{ij}$) (see Fig. 6).

Assume now that the clause $c_j$ is not satisfied. The first player cannot play on any $g_{ij}$ or $\bar{g}_{ij}$, because then the shortest path between that node to the already marked nodes is of length 1 (the path to $x_i$ or $\bar{x}_i$). If the first player plays on some $d_{ij}$ or $\bar{d}_{ij}$ (when $x_i$ or $\bar{x}_i$ is in $c_j$), then the second player will win by marking $f_{ij}$ or $\bar{f}_{ij}$. ∎

4.2.2. *Lower-Bound-Geodesic*

Given a directed graph $G = (V, E)$ with a distinct node $h$ which is marked. A move consists of choosing an unmarked node $u$ such that the shortest directed path between $u$ and the already marked nodes is at least of length $k$, and then in marking all nodes on all directed paths of length at least $k$ between $u$ and the previously marked nodes. The decision problem LOWER-BOUND-GEODESIC is whether $\mathbf{u} \in \mathbf{N}$ for any given con-

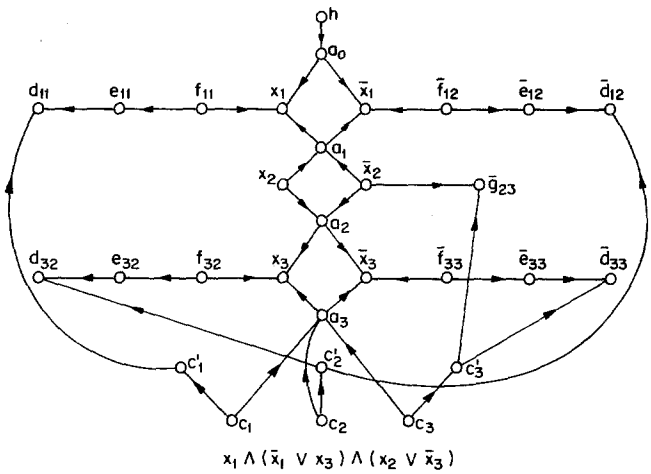


FIG. 6. SIMPLIFIED-FIXED-GEODESIC is PSPACE-complete.

figuration **u**. We define the SIMPLIFIED-LOWER-BOUND-GEODESIC decision problem by restricting the graph $G$ to be a DAG.

COROLLARY 2. *The problem* QBF *is polynomial reducible to* SIMPLIFIED-LOWER-BOUND-GEODESIC. *Hence the latter and consequently also* LOWER-BOUND-GEODESIC *are* PSPACE-*complete.*

*Proof.* The reduction is the same as that of Theorem 6. Note that in the construction of the graph there are no directed paths of length longer than two. ∎

### 4.3. *Úlehla-Geodesic*

In [11] a proof is given that the strategy for von Neumann's Hackendot can be calculated in polynomial time. We define here a similar game which is played on graphs, not only on forests.

ÚLEHLA-GEODESIC is a game which is played on a directed graph with the property that at least one of its nodes has 0-in-degree (called "root"). Each player in turn chooses a node of the graph and deletes all nodes on all shortest paths from all the roots to this node. A configuration of the game is represented by **u**, where **u** is the subset of nodes which are still in the graph.

#### 4.3.1. *Fixed-Úlehla-Geodesic*

FIXED-ÚLEHLA-GEODESIC is the same game as ÚLEHLA-GEODESIC with the restriction that only those moves such that all

shortest paths from all current roots have length exactly $k$ are permissible. The decision problem FIXED-ÚLEHLA-GEODESIC is whether $\mathbf{u} \in \mathrm{N}$.

We define the SIMPLIFIED-DIXED-ÚLEHLA-GEODESIC decision problem by restricting the graph $G$ to be a DAG.

THEOREM 7. *The problem* $G$(POS DNF 2) *is polynomial reducible to* SIMPLIFIED-FIXED-ÚLEHLA-GEODESIC. *Hence the latter and also* FIXED-ÚLEHLA-GEODESIC *are* PSPACE-*complete.*

*Proof.* Given an instance of the $G$(POS DNF 2) problem, we construct a DAG $G=(V, E)$ such that the first player can win in FIXED-ÚLEHLA-GEODESIC on this graph with $k=2$ if and only if he can win in $G$(POS DNF 2).

The roots of the graph are the $x$'s and $h_1$. While none of the conjuncts has yet been satisfied, the players can only play on the $b$'s and remove the nodes $x_i$, $a_i$, and $b_i$ each (see Fig. 7). After the formula has become true (one of the conjuncts, say $c_j$, has been satisfied) the next player will play this $c_j$ and by this move he will remove from the graph this $c_j$, all nodes $a_i$ and $x_i$ for all the $x_i$'s which have not yet been chosen, if any, as well as $h_1$ and $h_2$. If all the $x_i$'s have already been chosen, this move is still legal because of the presence of $h_1$ and $h_2$. All paths in the remaining graph contain at most one node $\{b_k\}$ and $\{c_k\}$, and hence the player who played this $c_j$ wins. ∎

4.3.2. *Lower-Bound-Úlehla-Geodesic*

A LOWER-BOUND-ÚLEHLA-GEODESIC game is the same game as ÚLEHLA-GEODESIC with the restriction that only moves which cause paths of at least $k$ nodes to be removed from the graph are permissible. A SIMPLIFIED-LOWER-BOUND-ÚLEHLA-GEODESIC game is the same game restricted to an initial graph (and hence all intermediate graphs) to be a DAG.

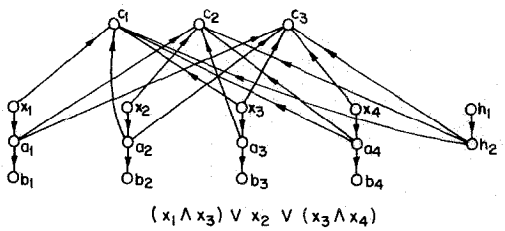


FIG. 7. SIMPLIFIED-FIXED-ÚLEHLA-GEODESIC is PSPACE-complete.

COROLLARY 3. *G*(POS DNF 2) *is polynomial reducible to* SIMPLIFIED-LOWER-BOUND-ÚLEHLA-GEODESIC. *Hence the latter and also* LOWER-BOUND-ÚLEHLA-GEODESIC *are* PSPACE-*complete.*

*Proof.* The reduction is like the previous one. Note that in the construction of the graph there are no paths of length more than two. ∎

We summarize this section with the following table of games known to be PSPACE-complete. The rules of the games are briefly indicated: If the length of the path(s) which are allowed to be marked (deleted) is bounded, we denote it by $k$ and the special $k$ which we did the reduction with is written in the table. If no such restriction exists we say that the restriction is $n$, the number of nodes.

| Description | Geography | Geodesic | Úlehla-Geodesic |
|---|---|---|---|
| exactly $k$ | $k=1$[a] | $k=2$ | $k=2$ |
| at least $k$ | $k=1$ | $k=2$ | $k=2$ |
| up to $n$ | proved | unknown | unknown |

[a] Proved in [9].

## REFERENCES


1. E. R. BERLEKAMP, J. H. CONWAY, AND R. K. GUY, "Winning Ways," Academic Press, London, 1982.
2. J. H. CONWAY, "On Numbers and Games," Academic Press, London, (1976).
3. S. EVEN AND R. E. TARJAN, A combinatorial problem which is complete in polynomial space, *J. Assoc. Comput. Mach.* **23** (1976), 710–719.
4. A. S. FRAENKEL, From Nim to Go, Annals of Discrete Mathematics, Vol. 6, *in* "Proc. Symp. on Combinatorial Math., Optimal Designs and their Applications," Colorado State University, Fort Collins, CO, 1978 (J. Srivastava, Ed.), pp. 137–156, 1980.
5. A. S. FRAENKEL, M. R. GAREY, D. S. JOHNSON, T. SCHAEFER, AND Y. YESHA, The Complexity of checkers on an $N \times N$ board—preliminary report, *in* "Proc., Nineteenth Annual Symp. Foundations of Computer Science," Ann Arbor, MI, October 1978, pp. 55–64, IEEE Computer Soc., Long Beach, CA, 1978.
6. A. S. FRAENKEL AND Y. YESHA, Complexity of problems in games, graphs and algebraic equations, *Discrete Appl. Math.* **1** (1979), 15–30.
7. A. S. FRAENKEL AND Y. YESHA, Theory of annihilation games—I, *J. Combin. Theory Ser. B* **33** (1982), 60–86.
8. D. LICHTENSTEIN AND M. SIPSER, Go is polynomial-space-hard, *J. Assoc. Comput. Mach.* **27** (1980), 393–401.
9. T. J. SCHAEFER, On the complexity of some two-person perfect-information games. *J. Comput. System Sci.* **16** (1978), 185–225.
10. L. J., STOCKMEYER AND A. R. MEYER, Word problems requiring exponential time; preliminary report, *in* "Fifth Annual ACM Symposium on Theory of Computing," pp. 1–9, Association for Computing Machinery, New York, 1973.
11. J. ÚLEHLA, A complete analysis of Von Neumann's Hackendot, *Internat. J. Game Theory* **9** (1980), 107–113.