RUTCOR RESEARCH REPORT

Games of no return [a]

Jack Edmonds [b]   Vladimir Gurvich[c]

RRR 4-2010, February 2010

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]The second author gratefully acknowledges partial support by DIMACS, Center for Discrete Mathematics and Theoretical Computer Science; by Graduate School of Information Science and Technology, University of Tokyo; by the Aarhus University Research Foundation and Center of Algorithmic Game Theory; by University of Pierre and Marie Curie, Paris 6; by Max Planck Institute at Saarbruecken.

[b]Equipe combinatoire & optimization, Universite Pierre and Marie Curie, Paris 6; e-mail: jackedmonds@roger.com

[c]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ, 08854; e-mail: gurvich@rutcor.rutgers.edu

RUTCOR RESEARCH REPORT
RRR 4-2010, FEBRUARY 2010

GAMES OF NO RETURN [1]

Jack Edmonds and Vladimir Gurvich

Abstract. Let $D = (V, A)$ be a finite directed graph (digraph) each vertex $v \in V$ of which is interpreted as a position and each arc $a = (v, v') \in A$ as a possible move from position v to v'. Two players, 1 and 2, take turns moving a token from a given initial position v_0. The game ends as soon as the token returns to a position, where it has already been. By definition, the player who made the last move loses, while the opponent wins (in the standard version, and vice versa in the misere version).
The defined *games of no return* generalize classical combinatorial (NIM-type) games. Indeed, every such game (D, v_0) on an acyclic digraph D turns into a game of no return (D', v_0) after adding a loop to every terminal vertex of D. In this case, D has a unique kernel $K \subseteq V$ and player 2 wins iff $v_0 \in K$. The kernel can be obtained in linear time by the well-known von Neumann algorithm.
Interestingly, a no return games (D, v_0) can be solved in polynomial time also in case when digraph $D = (V, A)$ is *symmetric*, that is, $a = (v, v') \in A$ iff $a' = (v', v) \in A$. Let us replace every such pair of arcs by a non-directed edge $e = (v, v')$ and denote by G the obtained non-directed graph. We show that player 1 wins in the symmetric game (D, v_0) iff every maximum matching of G contains v_0. Moreover, a move (v_0, v) is winning iff edge (v_0, v) belongs to such a maximum matching. Both conditions can be verified in polynomial time, due to the polynomial algorithm for finding a maximum matching of a graph given by the first author in 1965.
We suggest a polynomial algorithm for solving games for a slightly larger class of digraphs, which contains both acyclic and symmetric ones. Let us note that these games (as well as the symmetric and combinatorial games) can be solved in the *positional* strategies, that is, an optimal move (v, v') in any position v depends only on v, not on the preceding positions and moves. Yet, for general digraphs, it is not always the case and a winning strategy might require an exponential description.
We show that problem Q of solving a no return game is both NP- and coNP-hard. Moreover, $Q \in NP$ iff $Q \in coNP$. Thus, it is unlikely that Q belongs to NP or coNP. Indeed, then Q would be both NP- and coNP-complete, implying $NP = coNP$.
In fact, the same construction proves NP-hardness of the four types of games, in which a position or move should or should not be repeated; the games of (no) return and of (no) repetition. Yet, for the acyclic and symmetric digraphs, problem Q is polynomial in all four cases.
Keywords: games of no return, combinatorial games, positional strategy, symmetric digraph, kernel, Sprague-Grundy function, maximum matching, NP, $coNP$.

1 Introduction

1.1 General definitions and results

Let $D = (V, A)$ be a finite directed graph (digraph). Its vertex $v \in V$ and arc $a = (v, v') \in A$ will be interpreted as a position and possible move from it, respectively.

Loops and oppositely directed arcs, $a = (v, v'), a' = (v', v)$ are allowed but for convenience we will not allow parallel arcs, that is, no two distinct arcs can lead from v to v'.

Two players, 1 and 2, take turns moving a token along the arcs of A; player 1 begins from a given initial position v_0. The game ends as soon as the token returns to a position in which it has already been before. By definition, the player who made the last move loses, while the opponent wins; in the standard version, and vice versa in the *misere* version, which we will logically call the *game of return*. The standard version will be considered unless it is explicitly said otherwise.

The obtained trajectory of the token in D will be called a *lasso* and a simple directed path beginning from v_0 (that is a lasso without several last moves) will be called a *debut*.

Let us notice that the number of lassos and debuts might be exponential in $n = |V|$.

The pair (D, v_0) is called a *game of no return* in positional form.

A strategy of player 1 (respectively 2) is a mapping that assigns a move $a = (v, v') \in A$ to every debut that ends in v and consists of even (respectively, odd) number of moves.

A strategy is called *positional* if this move $a = (v, v') \in A$ depends only on v, that is, it does not depend on the preceding positions or moves of the debut.

To specify a positional strategy it would suffice to fix at most n arcs: one in each vertex. In contrast, for a general (not necessarily positional) strategy an arc should be chosen for each odd, or respectively even, debut. Both these sets of debuts might be exponential in n. Thus, an exponential input might be required to fix a general strategy.

Respectively, the number of general strategies might be doubly exponential, while the number of positional strategies is at most exponential in $n = |V|$.

Example 1 *Consider game of (no) return (D, v_0) with a bipartite digraph D in Figure 1. It is easy to verify that player 2 wins both the standard and misere versions. Let us notice that in each case the winning strategy is unique and not positional. Indeed, it is easy to see that in the most right position v the winning move of player 2 depends on the debut; it also depends on the the played version, the standard or misere one.*

For a similar example, see construction in the proof of Theorem 1 in Section 3.1.

Being finite games with perfect information, games of no return can be solved in general (pure) strategies; in other words, either ($Q1$) player 1 has a winning strategy, or ($Q2$) player 2 has such a strategy; see Section 2.4 for more details.

Yet, what is the complexity of the corresponding two decision problems? Obviously, $Q1$ holds if $Q2$ fails and vice versa. It is also clear that problems $Q1$ and $Q2$ are polynomially equivalent. Indeed, in order to reduce $Q1$ to $Q2$, or vice versa, it is enough to add to the digraph D one new arc leading from a new vertex v_{00} to v_0. Obviously, players 1 ad 2 swaps in the two games and 1 wins in one of them iff 2 wins in the other.

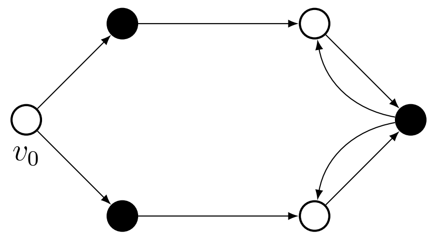


Figure 1: Player 2 wins in four games: of (no) return and (no) repetition, both standard and misere versions, but none of them (s)he can win in positional strategies.

Theorem 1 *Problem $Q1$ is NP-hard.*

The proof will be given in Section 2. This claim can be immediately extended to the statement that both $Q1$ and $Q2$ are both NP- and coNP-hard. Thus, it is unlikely that $Q1$ or $Q2$ is in NP or coNP, that is, $\{Q_1, Q_2\} \cap (NP \cup coNP) = \emptyset$. Indeed, otherwise both Q_1 and Q_2 would be both NP- and $coNP$-complete, and hence, $NP = coNP$.

1.2 Polynomially solvable cases

Yet, there are two important types of digraphs for which problems $Q1$ and $Q2$ are polynomial. In both cases the corresponding games will be solved in positional strategies.

1.2.1 Combinatorial games on acyclic digraphs

First, let us demonstrate that the classical combinatorial games are a special case of games of no return. Given a finite acyclic digraph $D = (V, A)$ and initial position $v_0 \in V$, the combinatorial game (D, v_0) is defied by the same rules as the corresponding game of no return, with one natural correction. Since no position of the digraph D can appear in a play twice, due to the acyclicity of D, the game is over as soon as a player has no move. By definition, this player loses, while the opponent wins; in the standard version, and vice versa in the misere version.

Combinatorial games are well known. In particular, digraph D has a unique kernel $K \subseteq V$ and player 2 wins iff $v_0 \in K$; the kernel can be found in linear time by von Neumann's algorithm; see Section 2.1 for the precise definitions, more details, and references.

Let us add a loop to each terminal vertex of D and denote the obtained digraph D'. The original combinatorial game (D, v_0) and obtained game of no return (D', v_0) are equivalent, since there is an obvious one-to-one correspondence between the lassos of D' and plays (directed paths from v_0 to a terminal) of D.

1.2.2 Games of no return on symmetric digraphs

Interestingly, a game of no return (D, v_0) can be solved in polynomial time also in case when digraph $D = (V, A)$ is *symmetric*, that is, $a = (v, v') \in A$ iff $a' = (v', v) \in A$.

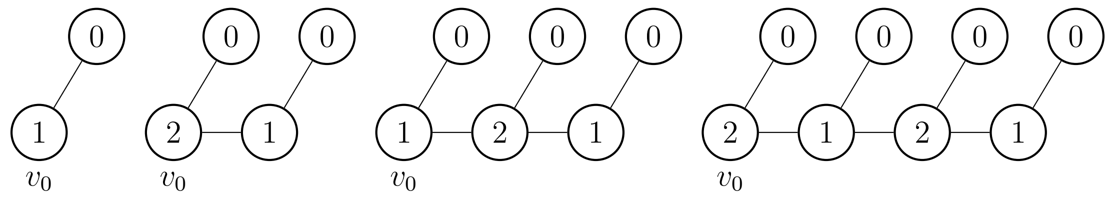


Figure 2: EGGS function $g(v_0)$ taking values 1 or 2

Let us replace every such two arcs by a non-directed edge $e = (v, v')$ and denote the obtained non-directed graph by G.

Theorem 2 *In a symmetric game of no return (D, v_0) player 1 wins iff v_0 belongs to every maximum matching of G. Moreover, a move (v_0, v_1) of player 1 is winning iff edge (v_0, v_1) belongs to such a maximum matching.*

A simple and constructive proof will be given in Section 3.3.

Now, let us notice that Theorem 2 allows us in polynomial time to decide whether player 1 wins in game (D, v_0) and to find all winning moves when the answer is positive.

This is due to the polynomial time algorithm of finding a maximum matching in a graph developed by the first author in 1965 [6]. Indeed, making use of this algorithm, let us find maximum matchings M in G and M' in $G[V \setminus \{v_0\}]$. Obviously, player 2 wins when $|M| = |M'|$ and player 1 wins if $|M| > |M'|$. In the last case, we can find all winning moves in v_0 as follows. For every move (v_0, v_1) let us find a maximum matching M_1 in $G[V \setminus \{v_0, v_1\}]$. Clearly, the move (v_0, v_1) is winning when $|M| = |M_1| + 1$ and it is losing if $|M| > |M_1| + 1$. For example, one can easily verify the above conditions for the games in Figure 2, in which one move from v_0 is winning, while the other one is losing.

Let us also note that, by Theorem 2, if if a player wins then (s)he has a *positional* winning strategy that, as we already mentioned, can be determined in polynomial time.

Remark 1 *Let us refer to [14, 15, 16, 4] for another problem of positional game theory that seems too difficult in general but can be efficiently solved for symmetric digraphs.*

Finally, let us remark that the misere version of a symmetric game, or in other words, a symmetric game of return, is trivial, since player 2 obviously wins.

1.2.3 Games of no return on symacyclic digraphs

The results of the above two subsections can be easily combined as follows.

A digraph $D = (V, A)$ will be called *symacyclic* if it can be partitioned in symmetric components such that every directed cycle of D belongs to one of them, or in other words, if an acyclic digraph appears after contraction of all these components.

Obviously, both symmetric and acyclic digraphs are symacyclic. They correspond to the cases of one and $n = |V|$ trivial components, respectively.

Theorem 3 *A game of no return (D, v_0) with a symacyclic digraph D has a solution in positional strategies, which can be found in polynomial time.*

The proof will be given in Section 3.4.

1.3 Games of no repetition

Let us slightly modify the rules and assume that the game is over as soon as an arc (rather than a vertex) appears in the play the second time. By definition, the player who made the last, "repeated", move loses; in the standard version, and (s)he wins in the misere version. The latter will be consistently called a *game of repetition*.

The games of (no) repetition differ substantially from the games of (no) return, yet, can be polynomially reduced to them by the following simple transformation.

Given a game (D, v_0) of no repetition, let us extend digraph D by one vertex v_{00} and one arc $a_0 = (v_{00}, v_0)$ and denote the obtained extended digraph by D'. Furthermore, let $L(D')$ be the line digraph of D', that is, a vertex w of D' is assigned to each arc (v_1, v_2) of D and D' contains an arc (w, w') iff two corresponding arcs of D are series: $(v_1, v_2), (v_1', v_2')$, and $v_2 = v_1'$. The initial position v_0 is assigned to a_0 by convention.

Proposition 1 *The original game of (no) repetition (D, v_0) and the obtained game $(L(D'), a_0)$ of (no) return are equivalent.*

Proof . Indeed, there is an obvious one-to-one correspondence between the repetition-free debuts of the first game and the (return-free) debuts of the second one. In particular, there is a one-to-one correspondence between the plays of the two games and it is immediate to verify that the results of the corresponding plays are always the same. □

For example, if (D, v_0) is an arborescence T with the root v_0 and a loop added to each terminal of T then we obtain $L(D')$ from D by inserting one extra arc before each loop.

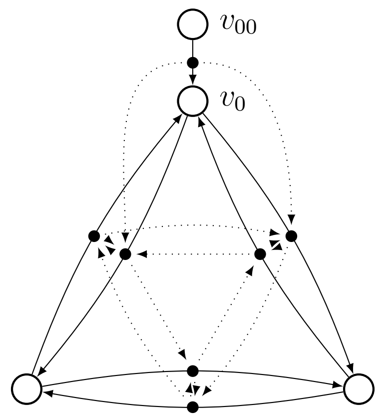
A more complicated, example is given in Figure 3.

Thus, combinatorial games provide four polynomially solvable cases: games of (no) return and (no) repetition, in the standard or misere version.

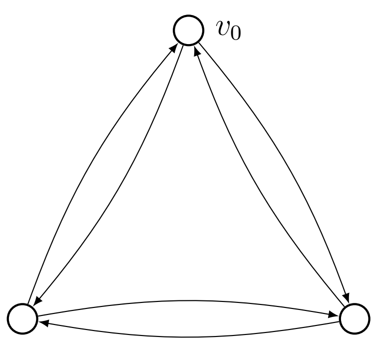
However, for games of no return there are other interesting polynomial cases: symmetric and symacyclic games. It would be interesting to find some polynomial cases for the games of no repetitions, too. In general, the problem is hard.

Theorem 4 *To solve a game of (no) repetition is NP-hard.*

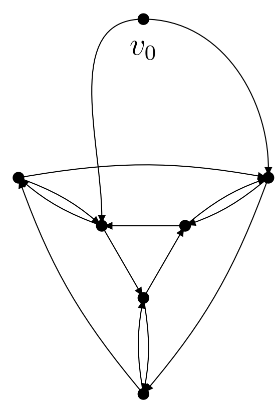
The problem is reduced from SAT by the same gadget as for Theorem 1, see Section 3.1.



(a)



(b)



(c)

Figure 3: A game of no repetition and the equivalent game of no return.

2 Reducing games of no return to combinatorial games

2.1 Finite acyclic combinatorial games

Given a finite acyclic digraph $D = (V, A)$ and initial position $v_0 \in V$, two players 1 and 2 take turns moving a token along the arcs of A; player 1 begins in v_0. Since D is finite any play will terminate. By definition, the player who made the last move wins (in the standard version, and the opponent wins in the so-called misere version).

A *kernel* is a set of positions $K \subseteq V$ which is

(i) independent, i.e., there is no move (v, v') such that both $v, v' \in K$ and

(ii) absorbing, that is, for each $v \notin K$ there is a move (v, v') such that $v' \in K$.

Every acyclic digraph has a unique kernel K [21]. By (ii), each final position belongs to K. It is easy to show that a player who came to a position $v \in K$, say player 2, can win. Indeed, by (i), player 1 must leave K, then by (ii), player 2 can reenter K, then 1 must leave K again, and 2 can again reenter, etc. Note that by (ii), player 1 cannot enter a terminal position, so player 2, sooner or later, will enter it, since digraph D is acyclic and finite.

The kernel in D can be found by in linear time as follows [21]. For $i = 0, \ldots, n-1$ in iterations i find in the current digraph the set of all terminals V_i and then the set V_i' of all vertices from which V_i can be reached in one move. Delete $V_i \cup V_i'$ from i and repeat; $\bigcup_i V_i$ is the (unique) kernel in D. Starting in V_i' one can win in $2i + 1$ moves; beginning in V_i one cannot win but can resist $2i$ moves.

The theory of combinatorial games was started by Charles Bouton in 1901 by the paper "Nim, a game with a complete mathematical theory" [5]. Let us notice that in fact already in this paper the concept of kernel and the above linear time algorithm for its computing appears, yet, only for the digraph of NIM. In [21] these results were just extended to general digraphs. It is also worth mentioningg that in [5] both the standard and misere versions of NIM were studied. Now, the standard and misere combinatorial games have an extensive literature; see, e.g., [1, 3, 5, 3, 7, 8, 9, 10, 11, 12, 13, 19, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32].

2.2 Sum of combinatorial games and Sprague-Grundy functions

Given two combinatorial games $\mathcal{G}_1 = (D_1, v_0^1)$ and $\mathcal{G}_2 = (D_2, v_0^2)$, their sum $\mathcal{G}_1 + \mathcal{G}_2$ is defined as follows. By each move, a player chooses either $\mathcal{G}_1$ or $\mathcal{G}_2$ and moves the token in it leaving the token in the other game in its place. The game $\mathcal{G}_1 + \mathcal{G}_2$ is over as soon as terminal positions are reached in both G_1 and G_2. The player who cannot move loses. Respectively, in the misere version this player wins. Obviously, the operation $\mathcal{G}_1 + \mathcal{G}_2$ is associative, hence the sum $\mathcal{G}_1 + \ldots + \mathcal{G}_m$ is well defined. How to play the sum? Even if we know both kernels K_1 and K_2 we still do not know the kernel K of $\mathcal{G}_1 + \mathcal{G}_2$. Indeed,

if $v_1 \in K_1$ and $v_2 \in K_2$ then $(v_1, v_2) \in K$,

if $v_1 \in K_1$ and $v_2 \notin K_2$ then $(v_1, v_2) \notin K$,

if $v_1 \notin K_1$ and $v_2 \in K_2$ then $(v_1, v_2) \notin K$, and yet, if $v_1 \notin K_1$ and $v_2 \notin K_2$ then the status of (v_1, v_2) is unknown, it can be in K or not.

Theory of playing the sums was developed by Sprague, 1936 [30], and Grundy, 1939 [17].

They generalized the concept of kernel as follows. Given $S \subseteq \mathbb{Z}_+ = \{0, 1, \ldots\}$, the *minimum excluded value* of S is defined as $mexS = min(\mathbb{Z}_+ \setminus S)$. Given a digraph $D = (V, A)$, its *Sprague-Grundy function* (SG-function) g is defined recursively as folllows: $g(v) = 0$ if v is a terminal position of D and $g(v) = mex\{g(w) | (v, w) \in A\}$, that is, the SG-value of v is the smallest non-negative integer which does not appear among SGF's values of the immediate successors of v.

Lemma 1 *The SG-function $g(v)$ is strictly less than the out-degree of v. In particular, the SG-function of a digraph is strictly bounded by its maximum out-degree.*

Proof . It follows immediately from the above recursive definition of the SG-function. □

The main two results of the Sprague-Grundy theory are given by the next two claims.

Theorem 5 *The kernel of a digraph is the set of zeros of its Sprague-Grundy function.* □

Yet, not only zeros are important. Given $a, b \in \mathbb{Z}_+$, let us write them as binary numbers and add them bitwise mod 2. The obtained number $c = a \oplus b$ is called the *Nim-sum* of a and b. For example, $1 \oplus 2 = 01_2 + 10_2 = 11_2 = 3$, $2 \oplus 3 = 10_2 + 11_2 = 01_2 = 1$, $3 \oplus 1 = 11_2 + 01_2 = 10_2 = 2$, $1 \oplus 2 \oplus 3 = 01_2 + 10_2 + 11_2 = 00_2 = 0$.

Since the NIM-sum is associative, it is well defined for m numbers.

Theorem 6 *The Sprague-Grundy function of the sum of m games $\mathcal{G}_1 + \ldots + \mathcal{G}_m$ is the NIM-sum of their Sprague-Grundy functions, $g(v_1, \ldots, v_m) = g_1(v_1) \oplus \ldots \oplus g_n(v_m)$.* □

For example, NIM with m piles of beans is the sum of m games with one pile each. Of course, one-pile NIM is trivial: the first player can win immediately. Yet, the one-pile NIM SGf is $g(x_i) = x_i$ and hence, the SGf of the m pile NIM is $g(x_1, \ldots, x_m) = g_1(x_1) \oplus \ldots \oplus g_m(x_m)$. The zeros of this function are exactly the positions of NIM in which the beginning player cannot win (in other words, they form the kernel). These arguments lead to the solution of NIM obtained by Bouton [5].

In general, given m combinatorial games $\mathcal{G}_1, \ldots, \mathcal{G}_m$ whose SG-functions $g_1, \ldots, g_m$ are known, the Sprague-Grundy theory enables one to play the sum $\mathcal{G} = \mathcal{G}_1 + \ldots + \mathcal{G}_m$. It is not more difficult than playing NIM with m piles. To find an optimal move in a position $v = (v_1, \ldots, v_m)$ of $\mathcal{G}$, one should just compute the NIM-sum $g(v_1, \ldots, v_m) = g_1(v_1) \oplus \ldots \oplus g_m(v_m)$, by Theorem 6. The zeros of g form the kernel of $\mathcal{G}$, by Theorem 5.

2.3 Games of no return in extensive form

Every game of no return (D, v_0) can be equivalently represented as a combinatorial game (T, w_0) on an arborescence (directed tree) T. Yet, such a transformation is exponential.

To every debut (that is, a simple directed path p in D beginning in v_0) let us assign a vertex $w = w(p)$ and include an edge (w, w') from $w = w(p)$ to $w' = w(p')$ in T iff p' is a one move extension of p. The initial position w_0 in T is defined by the 0-path, which contains only v_0, in D. By construction, $w = w(p)$ is a terminal position of T iff every one-move extension p' of p is a lasso, or in other words, if each move from w returns to p.

Proposition 2 *The original game of no return (D, v_0) and obtained combinatorial game (T, w_0) are equivalent.* □

Proof . It follows immediately from the above construction and definitions.

Let us also remark that for each vertex w of T its in-degree is 1, while its out-degree is at most the out-degree of the end-vertex v of the path $p(v)$ in D; this is also a strict upper bound for the SG-value in w.

2.4 Sums of games of no return and EGGS functions

In particular, the SG-function g can be translated from T to D. However, in D it will be a function of the debuts, not positions. Still, g is well defined in the initial position v_0.

Following its recursive definition, one can compute in linear time the SG-function in T. Yet, the size of T might be exponential in the size of D, and the complexity of computing $g(v_0)$ remains an open problem. We will call $g(v_0)$ EGGS-function.

These functions are instrumental in solving the sums of games of no return. Given m such games $\mathcal{G}_i = (D_i, v_0^i); i \in [m] = \{1, \dots m\}$. The sum $\mathcal{G} = \oplus_{i=1}^m \mathcal{G}_i = \mathcal{G}_1 \oplus \dots \oplus \mathcal{G}_m$ is defined, similarly to the sum of combinatorial games, as follows. By each move, a player chooses an $i \in [m]$ and moves the token in the game $\mathcal{G}_i$ leaving $m-1$ tokens of the remaining games in their places. The sum $\mathcal{G}$ is over as soon as one token returns to a position in which it has already been before. The player who made the last move loses; respectively, in the misere version this player wins.

An efficient algorithm, or oracle, that outputs the EGGS-functions $g_i = g_i(v_0^i)$ for $i \in [m]$ would allow us to find an optimal move in the initial position $v^0 = \{v_0^i; i \in [m]\}$ of the sum $\mathcal{G}$ as follows. Let us compute all m EGGS-functions and their NIM-sum $g_0(v_0) = \oplus_{i=1}^m g_i(v_0^i)$. If $g_0(v^0) = 0$ then, by Theorem 5, in v^0 there are no winning moves. Yet otherwise, if $g_0(v^0) > 0$, by Theorem 6, there is a $j \in [m]$ such that in game $\mathcal{G}_j = (D_j, v_0^j)$ there is a move (v_0^j, v_1^j) reducing the EGGS-function $g_j = g_j(v_0^j)$ by a positive integer ℓ so that the NIM-sum becomes 0. Such $j \in [m]$ and ℓ can be found, in the same standard way as in the classical game of NIM [30, 17].

Then, we can find the winning move in $\mathcal{G}$ as follows. Let us consider one by one all first moves (v_0^j, v_1^j) in the game G_j; in each case eliminate v_0^j from G_j and make use of the oracle to compute the EGGS-function $g_j^1(v_1^j)$ for the obtained subgame $\mathcal{G}_1^j = (D_j[V_j \setminus \{v_j^0\}], v_1^j)$.

By Theorem 6, there is a move (v_0^j, v_1^j) such that $g_j(v_0^j)$ is reduced by ℓ and, by this, the NIM-sum of the EGGS-functions is reduced to 0. This is the winning move in $\mathcal{G}$.

Unfortunately, it is difficult to compute the EGGS-functions, in general. Given a game of no return (D, v_0) and a positive integer threshold t, let us consider the following decision problem Q_t: whether $g(v_0) \leq t$ or $g(v_0 > t$? By Theorem 1, the problem is NP-hard, in general. Yet, by Theorem 2, problem Q_0 is polynomial in case of the symmetric (and even symacyclic) digraphs. Still, problem Q_1 remains open, even for the symmetric digraphs whose degree is bounded by a constant c. By Lemma 1, in this case the EGGS function is bounded by $c - 1$. Examples on the Figure 2 show that even in this case the length of the debuts that should be analyzed might be unbounded.

In any case, a polynomial algorithm or oracle computing the EGGS-function for a family F of games of no return (for example, for symmetric games) would allow us to solve the sums of games from F in polynomial time.

3 Proofs

3.1 Arbitrary digraphs; proof of Theorems 1 and 4

Let us show that $Q1$ is NP-hard, i.e., this is NP-hard to decide whether player 1 wins in the game (D, v_0). We will reduce $Q1$ from the classical NP-complete problem SAT: satisfiability of a CNF $C = \bigwedge_{j=1}^{m} C_j$ of n variables and m clauses.

First, to every variable x_i of C (for $i = 1, \dots, n$) let us assign a digraph defined by the four edges $(u_i, x_i), (x_i, w_i), (u_i, \overline{x}_i), (\overline{x}_i, w_i)$ and connect the obtained n digraphs successively identifying w_i and u_{i+1} for $i = 1, \dots, n-1$. We obtain a series-parallel bipartite digraph with $2n + (n + 1)$ vertices and $4n$ arcs; see the example in Figure 4. There are 2^n directed pathes through this digraph from $v_0 = u_1$ to w_n. They are in a one-to-one correspondence with 2^n assignments of variables $x_1, \dots, x_n$. Player 1 is in full control and can chose any one of these 2^n assignments-pathes. Then let us add $m + 1$ new arcs: (w_n, c) and (c, c_j) for $j = 1, \dots, m$. Let us assign to each position c_j the j-th clause C_j of the CNF C and add arcs leading from c_j to the vertices corresponding to all literals of C_j; see example in Figure 4, where

$$C = C_1 \wedge C_2 \wedge C_3 = (x_1 \vee \overline{x}_2 \vee x_3)(\overline{x}_1 \vee x_2 \vee \overline{x}_3)(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3).$$

It is easy to verify that the obtained digraph is bipartite. Hence, for each position v it is well defined which player makes a move in v. In fact, player 2 makes a non-trivial decision only in position c, that is, she chooses a clause of CNF C, while player 1 makes all other decisions, that is, he chooses an assignment and a variable in every clause. It is easy to see that he wins whenever there is a satisfying assignment for C. To do so, he should choose a satisfying literal for every clause and then the unique directed path from $u_1 = v_0$ to w_n avoiding all chosen literals; see example on Figure 3, where the literals x_3, x_2, and $\overline{x}_1$ satisfy the clauses C_1, C_2, and C_3, respectively.

Let us remark that the above winning strategy of player 1 is positional, by construction.

Conversely, player 2 wins whenever CNF C is not satisfiable. In this case, for every directed path p from u_1 to w_n chosen by player 1 there is a vertex c_j such that every arc from c_j terminates in p. Hence, (c, c_j) is a winning move for player 2.

Let us notice, however, that c_j (typically) depends on p. Hence, the winning strategy of player 2 may be not positional. □

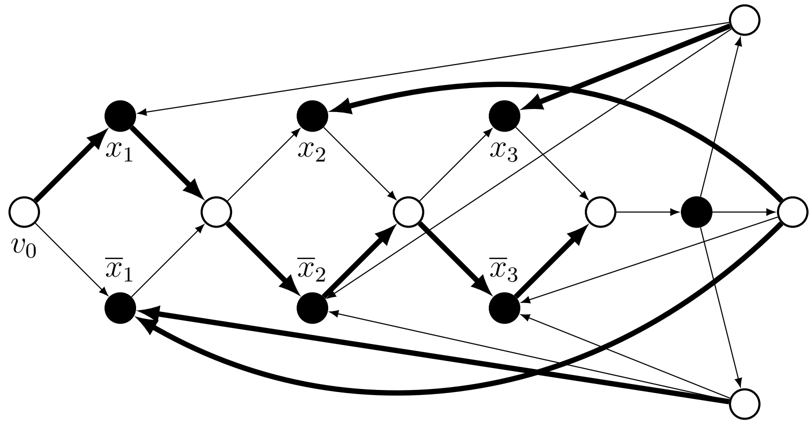


Figure 4: Reduction from SAT to games of no return; player 1 wins iff the corresponding CNF is satisfiable. The same gadget works for both the standard and misere versions of games of (no) return and (no) repetition.

The same gadget proves Theorem 4 as well. However, let us note that player 1 wins in the game of no return iff player 2 wins in the game of no repetition.

3.2 Games of return and of repetition are NP-hard to solve, too

The misere version for games of no return (respectively, of no repetition) is defined by the same rules as the standard version, yet, the result of each play is opposite, that is, the player repeating a position (respectively, a move) wins rather than loses. Thus, it would be natural to call these games the *games of return* (respectively, *of repetition*).

Let us note that making use of the gadget from the previous subsection we can prove NP-hardness for both, the standard and misere versions. The only minor difference is that in the latter case the directed path p is chosen in accordance with the assignment, while in the former case p is complementary.

Let us notice that in the bipartite digraph in Figure 1 player 2 can win all four games: of (no) return and of (no) repetition, that is, both the standard and misere versions. Also, let us note that misere version is trivial for the symmetric games, since player 2 always wins.

The misere combinatorial games have an extensive literature; see, for example, [1, 3, 5, 3, 7, 8, 19, 28, 29, 31, 32].

3.3 Symmetric digraphs; proof of Theorem 2

First, let us show that player 2 wins in game (D, v_0) whenever there is a maximum matching M in G avoiding $v0$. In this case, a winning (**positional**) strategy of player 2 can be defined as follows: if vertex v belongs to M (i.e., in G there is an edge $(v, v') \in M$) then player 2 should chose the corresponding move (v, v') in D; otherwise an arbitrary move. The point is that "otherwise" will never happen. Indeed, for any move (v_0, v_1) of player 1, vertex v_1 must belong to M, since M is maximum. Indeed, otherwise M could be extended by edge (v_0, v_1), since none of its two vertices belongs to M. Then, according to the chosen strategy, player 2 makes the (unique) move (v_1, v_2) in D such that $(v_1, v_2) \in M$ in G. And again, for every move (v_2, v_3) of player 1, vertex v_3 must belong to M, since M is maximum. Otherwise, M could be enlarged by replacing the edge (v_1, v_2) by two edges (v_0, v_1) and (v_2, v_3), since v_0 and v_3 do not belong to M, etc.

In general, for every step $i = 1, 2, \dots$, the $2i$-th move (v_{2i-1}, v_{2i}) (of player 2) is uniquely defined by M, in accordance with the chosen strategy; while for every $(2i-1)$-st move (v_{2i}, v_{2i+1}) (of player 1) vertex $v_{2i+1})$ must belong to M, since M is maximum. Indeed, otherwise M could be enlarged by replacing $\{(v_1, v_2), \dots, (v_{2i-1}, v_{2i}\}$ by $\{(v_0, v_1), \dots, (v_{2i}, v_{2i+1}\}$, since v_0 and v_{2i+1} do not belong to M.

Thus, player 1 always makes a move leading from one edge of M to another, while player 2 always makes a move (v_{2i-1}, v_{2i}) corresponding to an edge of M, in accordance with the chosen strategy. Let us show that the obtained position v_{2i} could not appear earlier in the play, which means that player 2 cannot lose. Indeed, $v_{2i} \neq v_0$, since v_0 does not belong to M, while v_{2i} does. Then, who could come to v_{2i} before? If player 1 then the move (v_{2i}, v_{2i-1}) (of player 2) would follow immediately after; if player 2 then only from v_{2i-1}; in both cases position v_{2i-1} would appear in the play twice, earlier than v_{2i} did. □

Now, let us prove that player 1 wins in game (D, v_0) if v_0 belongs to all maximum matchings of G. To do so, let us fix such a matching M and show that the following (**positional**) strategy of player 1 is winning : if vertex v belongs to M, i.e., in G there is an edge $(v, v') \in M$, then player 1 should choose the corresponding move (v, v') in D and an arbitrary move otherwise.

Let us notice that this strategy is defined by exactly the same rule as the above winning strategy of player 2 and again we will show that "otherwise" never happens.

Let M contain (v_0, v_1). This will be the starting move of player 1. Let us show that for any move (v_1, v_2) of player 2 vertex v_2 belongs to M. Indeed, otherwise one can replace (v_0, v_1) by (v_1, v_2) and get a maximum matching M' avoiding v_0, a contradiction. Then, player 1 chooses a unique move $(v2, v_3)$, in accordance with M. And again for every move (v_3, v_4) of player 2, vertex v_4 must belong to M. Indeed, otherwise one can replace $\{(v_0, v_1), (v_2, v_3)\}$ by $\{(v_1, v_2), (v_3, v_4)\}$ and get a maximum matching M' avoiding v_0, a contradiction.

In general, for every step $i = 1, 2, \dots$, the $2i-1$-st move (v_{2i-2}, v_{2i-1}) (of player 1) is uniquely defined by M, in accordance with the chosen strategy; while for every $2i$-th move (v_{2i-1}, v_{2i}) (of player 2) vertex $v_{2i})$ must belong to M, since otherwise one can replace $\{(v_0, v_1), \dots, (v_{2i-2}, v_{2i-1})\}$ by $\{(v_1, v_2), \dots, (v_{2i-1}, v_{2i})\}$ and get a maximum matching M' avoiding v_0, a contradiction. To show that player 1 cannot lose, we just repeat the last part of the previous proof replacing players 1 and 2. □

Assuming that v_0 belongs to every maximum matching in G, we wish to prove that a move (v_0, v_1) of player 1 is winning in game (D, v_0) iff (v_0, v_1) belongs to a maximum matching of G. We have just finished with the "if part". To proceed with the "only if one", let us consider the partition $N = N^+ \cup N^-$, where $N = N(v_0)$ is the set of all vertices adjacent to v_0 in G, while $N^+ = N^+(v_0)$ (respectively, $N^- = N^-(v_0)$) consists of all vertices v_1 such that (v_0, v_1) is (respectively, is not) in a maximum matching of G. We have already proven that move (v_0, v_1) is winning when $v_1 \in N^+$ and now want to show that otherwise this move is losing. It results from the following claim.

Lemma 2 *Every vertex of N^- belongs to each maximum matching of the induced subgraph $G[V \setminus \{v_0\}]$.*

Proof . Let us assume indirectly that a vertex $v_1 \in N^-$ does not belong to a maximum matching M of $G[V \setminus \{v_0\}]$. Then, adding (v_0, v_1) to M, we obtain a maximum matching in G that contains (v_0, v_1), in contradiction with $v_1 \in N^-$. □

Now, the desired statement follows from the first part of Theorem 2, proven above. □

Remark 2 *An algorithm that finds a maximum matching in a bipartite graph was suggested by Claude Berge in 1957 [2]. This algorithm and the above proof are both based on the method of augmenting paths.*

3.4 Symacyclic digraphs; proof of Theorem 3

By definition, a symacyclic digraph $D = (V, A)$ is partitioned into $k \geq 1$ symmetric subgraphs $D_i = (V_i, A_i),\ i \in [k] = \{1, \ldots, k\}$ that form an acyclic digraph Γ; in other words, $V = \cup_{i=1}^{k} V_i$ is a partition of V and every directed cycle of D is contained in D_i for some $i \in [k]$.

First, let us solve (in positional strategies) all symmetric games (D_t, v), for each symmetric digraph $D_t = (V_t, A_t)$, corresponding to a terminal vertex of Γ, and each initial positions $v \in V_t$. Due to acyclicity of Γ, the obtained solutions of the subgame (D_t, v) coincide with solution of the original game (D, v) for every $v \in V_t$.

Then, let us choose in Γ a non-terminal vertex such that every its successor is a terminal and consider the corresponding symmetric digraph $D_m = (Vm, A_m)$. Let $(v_m, v_t) \in A$ be an arc from $v_m \in V_m$ to $v_t \in V_t$, where $D_t = (V_t, A_t)$ is a terminal component. The game (D_t, v_t) is already solved. If the player, who begins in this game from v_t, loses (respectively, wins) then let us extend D_m by one new vertex v' and two opposite arcs $(v, v'), (v', v)$ (respectively, by two new vertices v', v'' and two pairs of opposite arcs $(v, v'), (v', v), (v', v''), (v'', v')$).

Let us perform the same procedure for every arc $(v_m, v_t) \in A$ leading from $D_m = (V_m, A_m)$ to a terminal component and denote $D'_m = (V'_m, A'_m)$ the obtained extended symmetric digraph. Then, let us solve symmetric game (D'_m, v) for each possible initial position $v \in V_m$. It is easy to verify that, for every $v \in V_m$, the obtained solutions (in positional strategies) of the symmetric game (D'_m, v) gives a solution of the original symacyclic game (D, v).

Thus, we can proceed with the standard backward induction to solve game (D, v) in positional strategies for all $v \in V$. Since a symmetric game can be solved in polynomial time, the same holds for the symacyclic games too. □

4 Conclusions and open problems

We considered games of (no) return and of (no) repetition, the standard and misere versions. Each of these four types of games is NP-hard to solve, for arbitrary digraphs; see Theorem 1 and its corollaries. Moreover, none of the corresponding four problems is in $NP \cup coNP$ unless $NP = coNP$.

However, some special cases are polynomially solvable, most notably symmetric (and symaciclic) games of no return; see Theorems 2, 3, and 4. The solution is based on the polynomial algorithm finding maximum matching in a on-directed graph [6]. Let us remark that in the obtained solution the winning strategy in positional.

Also, it would be interesting to find more non-trivial polynomially solvable cases.

It is an open question, whether the sum of symmetric games of no return is polynomially solvable. A positive answer would be provided by a polynomial algorithm for computing the EGGS function $g(v_0)$ of a symmetric game (D, v_0). Yet, the following decision problem is open: given a symmetric games of no return and positive integer threshold t, whether $g(v_0) \leq t$ or $g(v_0) > t$? By Theorem 2, the problem is polynomial for $t = 0$. Is it still true for $t = 1$? or larger t? if not in general then, perhaps, for the digraph whose out-degree is bounded by a constant c ? By Lemma 1, then the EGGS function is bounded by $c - 1$.

Acknowledgements We are thankful to Endre Boros who suggested the concept of symacyclic digraphs and to Vladimir Oudalov who kindly prepared the figures.

References

[1] A.J. Banerji and C.A. Dunning, On misere games. Cybernetics, and Systems 23 (1992), 221-228

[2] C. Berge, Two Theorems in Graph Theory, Proceedings of National Academy of Sciences USA 43 (1957) 842-844.

[3] E.R. Berlekamp, J.H. Conway, and R.K. Guy, Winning ways for your mathematical plays, vol.1-4, second edition, A.K. Peters, Natick, MA, 2001 - 2004.

[4] E. Boros, V. Gurvich, K. Makino, and Wei Shao, Nash-solvable bidirected cyclic two-person game forms, RUTCOR Research Report, RRR-30-2007 and DIMACS Technical Report, DTR-2008-13, Rutgers University.

[5] C.L. Bouton, Nim, a game with a complete mathematical theory, Ann. of Math., 2-nd Ser., 3 (1901-1902), 35-39.

[6] Jack Edmonds, Paths, trees, and flowers, Canad. J. Math. 17 (1965) 449–467.

[7] T.S. Fergusson, On sums of graph games with last player loosing, Int. J. of Game Theory, 3 (1974), 159-167.

[8] T.S. Fergusson, Misere annihilation games, J. of Combinatorial Theory, Ser.A (1984), 205-230.

[9] A.S. Fraenkel and I. Borosh, A generalization of Wythoff's game, *J. Comb. Theory* **15** (1973) 135–143.

[10] A.S. Fraenkel, How to beat your Wythoff games' opponent on three fronts *Amer. Math. Monthly* **89** (1982) 353–361.

[11] A.S. Fraenkel, Wythoff games, continued fractions, cedar trees and Fibonacci searches, *Theoretical Computer Science* **29** (1984) 49–73.

[12] Fraenkel, Euclid and Wythoff Games. Technical Report MCS05-02, Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, http://wisdomarchive.wisdom.weizmann.ac.il: 81/archive/00000370/01/05-02.pdf

[13] J. W. Grossman, Problem # 1537, Math. Mag. 70 (1997) 382.

[14] A.I. Gol'berg and V.A. Gurvich, Tight cyclic game forms, Russian Mathematical Surveys, 46:2 (1991) 241-243.

[15] A.I. Gol'berg and V.A. Gurvich, Some properties of tight cyclic game forms, Russian Acad. Sci. Dokl. Math., 43:3 (1991) 898-903.

[16] A.I. Gol'berg and V.A. Gurvich, A tightness criterion for reciprocal bipartite cyclic game forms, Russian Acad. Sci. Dokl. Math., 45 (2) (1992) 348-354.

[17] P.M. Grundy, Mathematics of games, Eureka, 2 (1939), 6-8.

[18] P.M. Grundy and C.A.B. Smith, Disjunctive games with the last player loosing, Proc. Cambridge Philos. Soc., 52 (1956) 527-523.

[19] V. Gurvich, On the misere version of game Euclid and miserable games, Discrete Math. 307 (2007), 1199-1204.

[20] L. Lovsz and M.D. Plummer, Matching Theory, North-Holland, 1986.

[21] J. von Neumann and O. Morgenstern, Theory of games and economic behavior, Princeton University Press, Princeton, NJ, 1944.

[22] G. Nivasch, The Sprague-Grundy function of the game Euclid, The Sprague-Grundy function of the game Euclid, Discrete Mathematics, 306 (2006) 27982800.

[23] G. Nivasch, The Sprague-Grundy function for Wythoff's game: On the location of the g-values, M.Sc. Thesis, 2004, Weizmann Institute of Sciences, Rehovot 76100, Israel, http://www.wisdom.weizmann.ac.il/ gabrieln.

[24] G. Nivasch, More on the Sprague-Grundy function for Wythoff's game, in Games of No Chance 3 (M. H. Albert and R. J. Nowakowski, editors), MSRI Publications 56, 377410, Cambridge University Press, 2009.

[25] R.J. Nowakowski ed., Games of No Chance, Mathematical Sciences Research Institute Publications, 29, Cambridge Univ. Press, 1998.

[26] R.J. Nowakowski ed., More games of No Chance 3, Mathematical Sciences Research Institute Publications, 42, Cambridge Univ. Press, 2002.

[27] M.H. Albert and R.J. Nowakowski eds., Games of No Chance 3, Mathematical Sciences Research Institute Publications, 56, Cambridge Univ. Press, 2009.

[28] T.E. Plambeck, Pretending in misere combinatorial games, Manuscript, 2003, http://www.plambeck.org/oldhtml/mathematics/games/misere

[29] T.E. Plambeck, Taming the wild in impartial combinatorial games, Manuscript, 2005, http://www.plambeck.org/oldhtml/mathematics/games/misere

[30] R. Sprague, Über mathematische Kampfspiele, Tohoku Math. J., 41 (1936), 438-444.

[31] W.A. Wythoff, A modification of the game of Nim, Niew Archief voor Wiskunde, 7 (1907), 199-202.

[32] Y. Yamasaki, On misere Nim-type games, J. Math. Soc. Japan, 32 (1980) 461-475.