Theory of Combinatorial Games

Aviezri S. Fraenkel Robert A. Hearn Aaron N. Siegel

February 12, 2014

fraenkel@wisdom.weizmann.ac.il
http://www.wisdom.weizmann.ac.il/~fraenkel
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot 76100, Israel

bob@hearn.to
H3 Labs LLC, Palo Alto, CA, USA

aaron.n.siegel@gmail.com
Twitter, San Francisco, CA, USA

Aim: To present a systematic development of the theory of combinatorial games from the ground up. Approach: Computational complexity. Combinatorial games are completely determined; the questions of interest are efficiencies of strategies. Methodology: Divide and conquer. Ascend from Nim to Chess and Go in small strides at a gradient that's not too steep. Presentation: Mostly informal; examples of combinatorial games sampled from various strategic viewing points along scenic mountain trails illustrate the theory. Add-on: A taste of *constraint logic*, a new tool to prove intractabilities of games.

1 Motivation and an Ancient Roman War-Game Strategy

The current mainstream of the family of combinatorial games consists of two-person games with perfect information (unlike some card games where information is hidden), without chance moves (no dice), and **outcome** restricted to (lose, win), (tie, tie) and (draw, draw) for the two players who move alternately (no passing).

Instead of the long terminology "combinatorial game(s)", we shall usually simply write "game(s)". In *normal* play, to *win a game* means to make the last move in it. This is the main concern of game theory, covered in sections 2-5. But in section 6, we expose the modern theory of *misère* play, where the player making the last move loses. A tie is an end position with no winner and no loser, as may occur in tic-tac-toe, for example. A draw is a "dynamic tie", i.e., a non-end position such that neither player can force a win, but each can find a next non-losing move. (In "noncombinatorial" game theory, each player receives a **payoff** at the end of the game. For combinatorial games it is natural to assign a payoff of $+1$ to the winner, -1 to the loser and 0 for tying or drawing: once play is in a draw cycle it is abrogated. Our games are **zero-sum** games in this sense.)

The modern theory of combinatorial games is portrayed in the groundbreaking work of Conway [Con01], the encyclopedic compilation of Berlekamp, Conway and Guy [BCG04], the attractive textbook by Albert, Nowakowski and Wolfe [ANW07], and the authoritative graduate-level book of Siegel [Sie13] that studies the modern theory of partizan games and misère play.

The primeval and simplest combinatorial game is Nim: Given m piles of finitely many tokens, a move consists of selecting a single non-empty pile and removing from it a positive number of tokens, that is, at least one, and up to and including the entire pile. The player first unable to move loses, the opponent wins (**normal** play). For $m = 1$, player I can win if the pile is nonempty, simply by removing it entirely. For $m = 2$, player I can win if the piles are of unequal size, by a move that equalizes their size, followed by imitating on one pile what player II does on the other. For $m > 2$, the winning strategy, first given in [Bou02], is quite surprising, yet simple: compute the XOR (eXclusive OR) of the binary representation of the pile sizes. If the resulting binary **nim-sum** is non-zero, the *next* player (player I) has a move making it zero (a winning move). If it is zero, *every* move will make it non-zero (a losing move). This is shown in section 2 in the more general setting of "NIM-type" games. Thus for $m = 3$ and pile sizes $1, 2, 3$, a simple case analysis shows that the *previous* player (player II) can win. Indeed, the nim-sum $1 \oplus 2 \oplus 3$ is 0.

As an exercise, can you win by beginning to play in a game of NIM with 4 piles of sizes $2, 3, 5, 7$? If so, do you have a unique winning strategy?

The family of combinatorial games contains simple games such as NIM, as well as seemingly complex games such as CHECKERS, CHESS and GO. The fundamental question that arises naturally is why some games, such as NIM, are easy to solve, whereas others in the family, such as GO, seem so complex? The quest for answers to this problem motivates this survey.

For throwing some light on the question, a Roman Cæsars' motto is adopted:

DIVIDE AND CONQUER .

There are several mathematical differences between NIM-type and CHESS-type games. After identifying them, a concentrated attack is launched on each of them separately, which seems to have a better chance of success than trying in vain to scale the sheer cliff separating NIM from CHESS. Thus, we ascend from NIM towards CHESS and GO at a moderate gradient, by gradually introducing into NIM more and more complications in a natural order of increasing complexity. The adventures occurring on the way comprise the story of this chapter.

In section 2 we review the classical theory of acyclic games, **sum** of games and the Sprague-Grundy function, which is the main tool for solving acyclic games. We also show that complexities of games are normally much higher than those encountered in optimization problems such as the Traveling Salesperson Problem.

An "apparent" difference between NIM and CHESS is the **board** which exists for the latter but not for the former. However, Figure 1 shows that also NIM can be considered as a board game: a_i indicates a nim-heap of size i, and the directed edges indicate the permissible moves. Thus placing a token on each of the vertices a_1, a_2 and a_3 and moving them along directed edges, where any number of tokens may reside on any vertex, is isomorphic to NIM with pile sizes 1, 2, 3. Conclusion: this "apparent" difference is not really a mathematical difference.

Here are some more substantive differences:

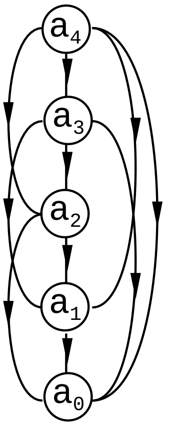


Figure 1: NIM as a board game.

• **Cycles.** NIM-type games are finite and "acyclic", i.e., there is an underlying "well-ordering principle" which guarantees that no position is assumed twice. This is not the case for CHESS-type games. Applying the Divide And Conquer Principle, we deal with such "cyclic" games separately in section 3, where it is shown that cycles indeed destroy the classical theory. A generalized theory is developed there which recovers a polynomial strategy for cyclic games.

• **Token Interactions.** Another difference is that in NIM-type games, considered as board games, tokens coexist peacefully on the same vertex (board square), whereas they interact in various ways such as jumping, deflecting, capturing, etc., in CHESS-type games. Many of these interactions cause the games to become PSPACE-hard (notion explained near the end of section 2) even in simplified form, e.g., when played on planar or acyclic or bipartite graphs. However, if both tokens disappear on impact, a "just barely polynomial" strategy can be given for general cyclic digraphs (directed graphs). This topic is studied in section 4.

• **Partizanship.** A game is **impartial** if the set of **options** (positions reached in a single move) of every position is the same for the two players. If this doesn't necessarily hold, the game is **partizan**. NIM-type games are impartial, whereas CHESS-type games are partizan (the "black" player cannot move a white piece and vice versa). Note that the set of impartial games is a subset of the set of partizan games. It turns out that partizan games, taken up in section 5, are in general PSPACE-hard even on acyclic digraphs;

see Yedwab [Yed85], Morris [Mor81]. See also Pultr and Morris [PM84].

- **Termination Set.** Another difference concerns the conventions for ending the play of the game, i.e., the termination set τ. Roughly, the complexity of the strategy seems to increase with the size $|\tau|$ of τ. The simplest games are those played on a digraph G, where τ is the set of *leaves* of G (vertices of outdegree 0), followed by those in which τ consists of all positions whose only options are leaves – such as in **misère** play: the player making the last move loses – to cases where τ is even larger, such as in Chess and Go. A theory for general τ has yet to be developed, but we treat misère play in section 6.

As we progress from the easy games to the more complex ones, we will develop some understanding for the *poset* of tractabilities and efficiencies of game strategies: in the realm of existential questions, tractabilities and efficiencies are, by and large, linearly ordered, from polynomial to exponential. However, as explained near the end of section 2, game problems are formulated by an – often unbounded – number of alternating quantifiers. For such problems the notion of a "tractable", "polynomial" or "efficient" computation – defined formally in Definition 1, section 2 – is much more complex.(Which is more tractable: a game that ends after four moves, but it's undecidable who wins [Rab57], or a game requiring an Ackermann function of moves to finish but the winner can play randomly, having to pay attention only near the end [FLN88], [FN85] ?) Since we are concerned with game complexities, we present, in section 7, a modern tool for proving game intractabilities conveniently and efficiently. In section 8, the Conclusion, we briefly illuminate our ascent from Nim to Chess and Go, and indicate possible further directions of combinatorial game theory.

2 The Classical Theory, Sum of Games, Complexity

In this section we will see how to play "arbitrary" finite acyclic games such as Beat Doug (Figure 2). (Doug – "DAG", Directed Acyclic Graph.)

Place one token on each of the four starred vertices. A move consists of selecting a token and moving it, along a directed edge, to a neighboring vertex on this acyclic digraph. As usual we consider normal play, so the player making the last move wins. Tokens can coexist peacefully on the

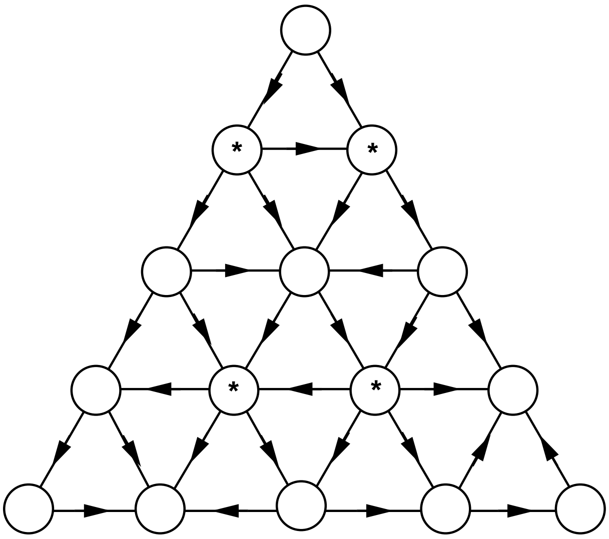


Figure 2: Beat DOUG on this DAG (directed acyclic graph).

same vertex. For the given position, how much time does it take to:

(a) compute who can win;

(b) compute an optimal next move;

(c) consummate the win, that is, actually make the last move?

Denote by $\mathbb{N}$ and $\mathbb{N}^+$ the set of all nonnegative integers and the set of all positive integers respectively. Following the divide and conquer methodology, let's begin with a more structured digraph, rather than solving immediately the "arbitrary" Beat Doug. Given $n \in \mathbb{N}^+$ (the initial score) and $t \in \mathbb{N}^+$ (the maximal step size), a move in the game Scoring consists of selecting $i \in \{1, \ldots, t\}$ and subtracting i from the current score, initially n, to generate the new score. Play ends when the score 0 is reached. The player reaching 0 wins (normal play). Notice that Nim is the special case $t = \infty$ of Scoring.

The digraph $G = (V, E)$ for Scoring is shown in Figure 3 for $n = 8$ and $t = 3$: it is an acyclic digraph, where V is the set of game positions, and $(u, v) \in E$ if and only if there is a move from u to v (then v is an **option** of u). A position (vertex) $u \in V$ is labeled N (for Next player win) if the player moving from u can win; otherwise it's a P-position (Previous player win). Denote by $\mathcal{P}$ the set of all P-positions, by $\mathcal{N}$ the set of all N-positions, and by $F(u)$ the set of all options of any vertex u. For any acyclic game, the partition of the vertex-set into , $\mathcal{N}$ exists uniquely and satisfies,

$$u \in \mathcal{P} \quad \text{if and only if} \quad F(u) \subseteq \mathcal{N} \ , \tag{1}$$

$$u \in \mathcal{N} \quad \text{if and only if} \quad F(u) \cap \mathcal{P} \neq \emptyset \ . \tag{2}$$

In words: u is a P-position if and only if all its options (direct followers) are N-positions; and u is an N-position if and only if it has an option in $\mathcal{P}$.

As suggested by Figure 3, we have $\mathcal{P} = \{k(t+1) \colon k \in \mathbb{N}^{+}\}$, so $\mathcal{N} = \{\{0, \ldots, n\} \setminus \mathcal{P}\}$. The winning strategy consists of dividing n by $t+1$. Then $n \in \mathcal{P}$ if and only if the remainder r is zero. If $r > 0$, the unique winning move is from n to $n - r$.

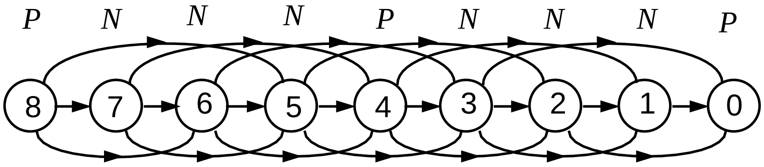


Figure 3: The digraph for SCORING, with initial score $n = 8$ and maximal step $t = 3$. Positions marked N are wins and P are losses for the player moving from those positions.

Is this a tractable strategy? ("Tractable" – see Definition 1.)

INPUT SIZE: $\Theta(\log n)$ (**succinct** input).
STRATEGY COMPUTATION: $O(\log n)$ (division of n by t).
LENGTH OF PLAY: $\lceil n/(t+1) \rceil$.

Thus the computation time is linear in the input size, but the length of play is exponential!

To the "run-of-the-mill-algorithmicians" the latter fact dooms the game as intractable. It may be quite a surprise to them that it does not prevent the strategy from being tractable: whereas we dislike computing in more than polynomial time, we observe that at least some members of the human race relish to see some of its members being tormented for an exponential length of time, from before the era of the Spanish matadors and inquisition, through soccer and tennis, to CHESS and GO! But there are other requirements for making a strategy polynomial as we will see presently, so at present let's say that the strategy is *tractable*.

Recapping our story up to now, we have made some progress: we got a tractable strategy for winning in SCORING. But what about the case when we have k scores $n_1, \ldots, n_k \in \mathbb{N}^+$ and $t \in \mathbb{N}^+$? A move consists of selecting one of the current scores and subtracting from it some $i \in \{1, \ldots, t\}$. Play ends when all the scores are zero. Figure 4 shows an example ($k = 4$). This is a **sum** of SCORING games, itself also a SCORING game. The notion of sum often permits us to simplify the strategy analysis, if the components of the game are disjoint. For example, NIM is the sum of its piles. It's easy to see that the game of Figure 4 is equivalent to the game played on the digraph of Figure 5, with tokens on vertices 5, 6, 7 and 8. A move consists of selecting a token and moving it right by not more than $t = 3$ places. Tokens can coexist on the same vertex. Play ends when all tokens reside on 0. What's a winning strategy?

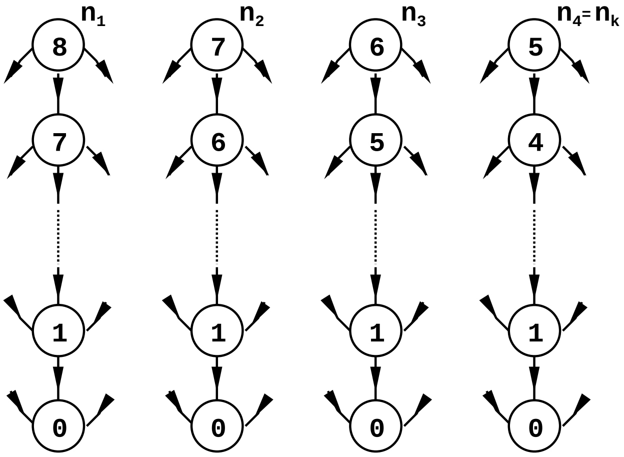


Figure 4: A SCORING game consisting of a sum of four SCORING games. Here $k = 4$, $n_1 = 8$, $n_2 = 7$, $n_3 = 6$, $n_4 = 5$, and $t = 3$.

We hit two snags when trying to answer this question:

(i) Though the sum of P-positions is in $\mathcal{P}$, the sum of N-positions is in $\mathcal{P} \cup \mathcal{N}$. Thus a game of two tokens, one on each of 5 and 7, is seen to be an N-position (the move $7 \to 5$ clearly results in a P-position), whereas the sum of a token on 3 and 7 is seen, by inspection, to be a P-position. So the simple P-, N-strategy breaks down for sums, which arise frequently in combinatorial game theory.

(ii) The **game-graph** has exponential size in the input size $\Omega(\Sigma_{i=1}^{k} \log n_i)$ of the "regular" digraph $G = (V, E)$ (with $|V| = n+1$, where $n = \max_i n_i$) on which the game is played with k tokens (Figure 5 in our case). However, G is not the game-graph of the game: each tuple of k tokens on G corresponds to a single vertex of the game-graph, whose vertex-set thus has size $\binom{k+n}{n}$ — the number of k-combinations of $n+1$ distinct objects with at least k repetitions. For $k = n$ this gives $\binom{2n}{n} = \Theta(4^n/\sqrt{n})$, which is doubly exponential in the input size!

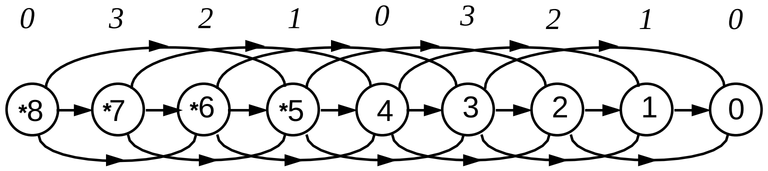


Figure 5: A game on a graph, but not a game-graph.

The main contribution of the classical theory is to provide a polynomial strategy for sums despite the exponential size of the game-graph. On G, label each vertex u with the least nonnegative integer not among the labels of the options of u (see top of Figure 5). These labels are called the **Sprague–Grundy** function values of the game on G, or the g-function for short [Spr36], [Gru39]. It is a function from the vertices of a digraph into the nonnegative integers, defined recursively by

$$g(u) = \operatorname{mex} g(F(u)),$$

where for any subset $S \subsetneq \mathbb{N}$,

$$\operatorname{mex} S = \min \mathbb{N} \setminus S$$

is the least nonnegative integer not in S. Notice that g of the empty set is 0. The function g exists uniquely on every finite acyclic digraph.

For $u = (u_1, \ldots, u_k)$, a vertex of the game-graph (whose very construction entails exponential effort), we have

$$g(u) = g(u_1) \oplus \cdots \oplus g(u_k)\ , \qquad \mathcal{P} = \{u \colon g(u) = 0\}\ , \qquad \mathcal{N} = \{u \colon g(u) > 0\}\ ,$$

where $\oplus$ denotes **nim-sum** (summation over GF(2), also known as *exclusive or*, which we already met in section 1). To compute a winning move from an

N-position, note that there is some i for which $g(u_i)$ has a 1-bit at the binary position where $g(u)$ has its leftmost 1-bit. Reducing $g(u_i)$ appropriately makes the NIM-sum 0, and there is a corresponding move with the i-th token. For the example of Figure 5 we have

$$g(5) \oplus g(6) \oplus g(7) \oplus g(8) = 1 \oplus 2 \oplus 3 \oplus 0 = 0 \ ,$$

a P-position, so every move from this position is losing.

Is this Sprague–Grundy strategy polynomial? For SCORING, the remainders $r_1, \ldots, r_k$ of dividing $n_1, \ldots, n_k$ by $t+1$ are the g-values, as suggested by Figure 5. The computation of each r_j has size $O(\log n)$, where $n = \max n_i$. Since $k \log n < (k + \log n)^2$, the strategy computation (items (a) and (b) at the beginning of this section) is polynomial in the input size (k is a constant). The length of play remains exponential.

Since the strategy for SCORING is tractable for a single game as well as for a sum, we may say that SCORING has a **polynomial** strategy (see Definition 1 below).

Now consider a general nonsuccinct acyclic digraph $G = (V, E)$, that is, the input size is not logarithmic: If the graph has $|V| = n$ vertices and $|E| = m$ edges, the input size is $\Theta((m + n) \log n)$ (each vertex is represented by its index of size $\log n$, and each edge by a pair of indices), and g can be computed in $O((m+n) \log n)$ steps (by a "depth-first" search; each g-value is at most n, of size at most $\log n$). For a sum of k tokens on the input digraph, the input size is $\Theta((k + m + n) \log n)$, and the strategy computation for the sum can be carried out in $O((k+m+n) \log n)$ steps (nim-adding k summands of g-values). Note also that for a nonsuccinct digraph the length of play is only linear rather than exponential, in contrast to a succinct (logarithmic input size) digraph.

Our original BEAT DOUG problem is now also solved with a polynomial strategy. Figure 6 depicts the original digraph of Figure 2 with the g-values added in (we'll see later how to compute g). Since $2 \oplus 3 \oplus 3 \oplus 4 = 6$, the given position is in $\mathcal{N}$. Moving $4 \to 2$ is a unique winning move. The winner can consummate a win in polynomial time. Also notice that the strategy for NIM is a special case of the Sprague-Grundy strategy.

However, the strategy of classical games is not very robust: slight perturbations in various directions can make the analysis considerably more difficult. Thus the theory for WELTER, which amounts to NIM with all piles of *distinct* size, is rather complicated [Con01] (ch. 13).

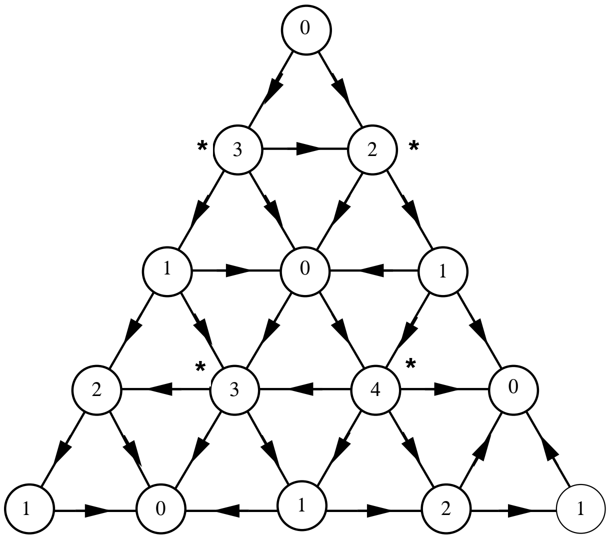


Figure 6: The beaten Doug.

We point out that there is an important difference between the strategies of Beat Doug and Scoring. In both, the g-function plays a key role. But for the latter, some further property is needed to yield a strategy that's polynomial, since the input graph is (logarithmically) succinct. In this case the extra ingredient is the periodicity modulo $(t+1)$ of g, which was easy to establish. For other succinct games, it may be harder to prove polynomiality, such as for general *octal games* [BCG04], Vol. 1.

2.1 Complexity, Hardness and Completeness

What, then, are tractable, polynomial and efficient games? We abstract some of the properties of Nim, since it has a simple strategy, and it is the sum of its piles.

Definition 1. Let $c > 1$ denote arbitrary constants and denote by n the size of a sufficiently succinct encoding of a digraph $G = (V, E)$. A subset T of combinatorial games with a **polynomial** strategy has the following properties. For normal play of every $G = (V, E) \in T$, and every position u of G:

(a) The P-, N-, D- and tie-label of u can be computed in time $O(n^c)$ (polynomial time; D denotes draw – see next section).

(b) An optimal next move from any N- to a P-position and from any D- to a D-position and from any non-end tie- to a tie-position can be computed in time $O(n^c)$ (polynomial time).

(c) The winner can consummate a win in at most $O(c^n)$ moves (exponential time).

(d) The subset T is closed under summation, i.e., $G_1, G_2 \in T$ implies $G_1 + G_2 \in T$. (Thus (a), (b), (c) hold for $G_1 + G_2$ for every independently chosen position of G_1 and for every independently chosen position of G_2.)

A subset $T_1 \subseteq T$ for which (a)–(d) hold also for misère play – the player making the last move loses – is a subset of games with an **efficient** strategy.

A superset $T^1 \supseteq T$ for which (a)–(c) hold is a superset of games with a **tractable** strategy.

A game in some such T or T_1 or T^1 is called **polynomial** or **efficient** or **tractable**, respectively.

A decidable game[1] which has no polynomial (tractable) strategy is called **nonpolynomial** (**intractable**).

Strictly speaking, in view of (c), the terminology "polynomial" ought to be replaced by something else, such as "adequate". But "polynomial" is so universally used for problems that are computationally reasonable, that "polynomial" is preferred. Ramifications in several directions of Definition 1 are considered in [Fra04].

To prove that a problem is tractable, polynomial or efficient, the normal procedure is to construct an algorithm that has those properties. But how do we show that, no matter how hard we try, a problem doesn't have a good solution? We explain briefly a next best way to do something in this direction.

Roughly, NP consists of all problems whose solution can be *verified* – not necessarily *found*, only verified – using an amount of time that's polynomial in a succinct input size of the problem. It's NP-complete if it's among the hardest problems in NP. It's NP-hard if it's NP-complete, except that it needs *at least* a polynomial amount of time. PSPACE consists of all problems

[1] A problem is decidable if there exists an algorithm to solve all its instances. Otherwise it is undecidable.

that can be solved using a polynomial amount of space (hence of time), and EXPTIME – all problems that can be solved in an exponential amount of time. Hardness and completeness are defined analogously to the respective definitions of NP. NP-complete problems share the following idiosyncrasies:

- If any NP-complete problem will be shown to have a polynomial-time algorithm, then all of them are polynomial, and if any is shown to have a lower non-polynomial bound, then all of them are non-polynomial.

- It is widely believed that NP-complete problems are non-polynomial.

- Completeness results are asymptotic. With any NP-complete problem there is associated some parameter n, and the result holds for large n. For games, n is typically the size of a side of the board.

Analogous results hold a-fortiori for PSPACE-complete problems. But EXPTIME-completeness is an unconditional *provable* intractability: any EXPTIME-complete problem has a lower exponential time bound for its solution, asymptotically.

Optimization problems, such as TSP (Traveling Salesperson Problem) are typically NP-complete, since there is a single existential quantifier (does there exist a tour of cost $< C$?). In a two-person game, the question whether player I can win involves an alternating number of existential and universal quantifiers: does player I have a move such that for every move of player II there exists a move of player I $\cdots$ such that player I wins? If the number of alternating quantifiers is bounded, the game tends to be PSPACE-complete, such as Hex [Rei81]; if their number is unbounded, it is typically EXPTIME-complete, such as Chess [FL81].

We do not know of any PSPACE-complete or EXPTIME-complete game problem that has a known polynomial solution for finite boards as encountered in practice, such as 8 × 8 or 19 × 19. Thus, though completeness and hardness are asymptotic properties, in practice they seem to say something also about actual games.

3 Introducing Draws

In this section we learn how to beat Craig (**C**yclic d**IGRA**ph) efficiently. The four starred vertices in Figure 7 contain one token each. The moves are identical to those of Beat Doug; tokens can coexist peacefully on any vertex. The only difference is that now the digraph $G = (V, E)$ may have cycles and loops (the latter correspond to passing a move), or may be infinite. In addition to the P- and N-positions, which satisfy (1) and (2), we now may have also Draw-positions, D.

Definition 2. Given a game Γ, with game-graph $G = (V, E)$, where G may be finite or infinite, acyclic or cyclic. Denote by $\mathcal{O}$ the set of all nonnegative ordinals not exceeding $|V|$. By recursion on $n \in \mathcal{O}$ define the multisets,

$$P_n = \{u \in V,\ n = \min\ m : F(u) \subseteq \bigcup_{i<m} N_i\},$$

$$N_n = \{u \in V,\ n = \min\ m : F(u) \cap \left(\bigcup_{i<m} P_i\right) \neq \emptyset\}.$$

Finally, let
$\mathcal{P} = \bigcup_{n\in\mathcal{O}} P_n, \quad \mathcal{N} = \bigcup_{n\in\mathcal{O}} N_n, \quad \mathcal{D} = V \setminus (\mathcal{P} \cup \mathcal{N}),$
where $\mathcal{D}$ is the set of all D-positions.

The definition implies

$$u \in \mathcal{D} \quad \text{if and only if} \quad F(u) \cap \mathcal{P} = \emptyset \text{ and } F(u) \cap \mathcal{D} \neq \emptyset\ .$$

Introducing cycles causes several problems:

- Moving a token from an N-position such as vertex 4 in Figure 8 to a P-position such as vertex 5 is a nonlosing move, but doesn't necessarily lead to a win. A win is achieved only if the token is moved to the *leaf* 3. The digraph might be embedded inside a large digraph, and it may not be clear to which P-option to move in order to realize a win.

- The partition of V into $\mathcal{P}, \mathcal{N}$ and $\mathcal{D}$ is not unique, as it is for $\mathcal{P}$ and $\mathcal{N}$ in the classical case. For example, vertices 1 and 2 in Figure 8, if labeled P and N, would still satisfy (1) and (2), and likewise for vertices 8 and 9 (either can be labeled P and the other N).

Both of these shortcomings can be remedied by introducing a suitable counter function J – see [FY86].

For handling sums, we would like to use the g-function (Sprague-Grundy function), but there are two problems:

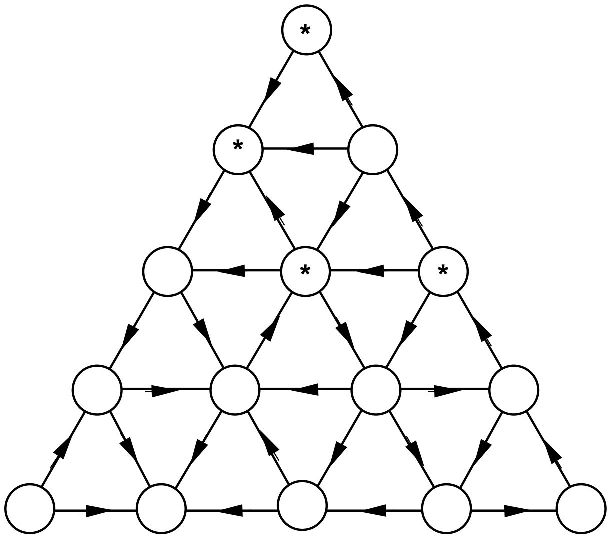


Figure 7: Beat CRAIG in this Cyclic dIGRAph.

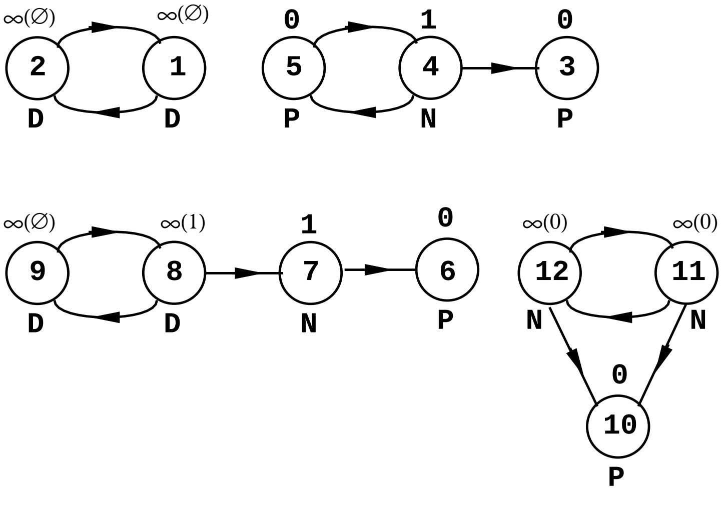


Figure 8: P-, N-, D- and γ-values for simple digraphs.

- The question of the existence of g on a digraph G with cycles or loops is NP-complete, even if G is planar and its degrees are ≤ 3, with each indegree ≤ 2 and each outdegree ≤ 2 [Fra81]. (NP-completeness without these restrictions, or less restrictions, has been proved in [Chv73], [vL76], [FY79].)

- The strategy of a cyclic game isn't always determined by the g-function, even if it exists.

This is one of those rare cases where two failures are better than one! The second failure opens up the possibility that perhaps there's another tool that always works, and if we are optimistic, we might even hope that it is also polynomial. There is indeed such a generalized g-function γ. It was introduced by Smith [Smi66]; an improved version was (re)discovered in [FP75]; see also [Con01] (ch. 11), [FY86].

The γ-function is defined the same way as the g-function, except that it can assume not only values in $\mathbb{N}$, but in $\mathbb{N} \cup \{\infty\}$, where the symbol ∞ denotes a value bigger than every natural number. We also use the notation $\gamma(u) = \infty(K)$, where K is the set of finite γ-values of the options of u. We have $\gamma(u) = \infty(K)$, if there is $v \in F(u)$ with $\gamma(v) = \infty$, and v has no option w with $\gamma(w)$ equal to the least nonnegative integer not in K. The formal definition is given in [FY86]. Figure 8 depicts γ-values for some simple digraphs. Every finite digraph with n vertices and m edges has a unique γ-function that can be computed in $O(mn \log n)$ steps. This is a polynomial-time computation, though bigger than the g-values computation.

To get a strategy for sums, define the **generalized nim-sum** as the ordinary nim-sum augmented by:

$$a \oplus \infty(L) = \infty(L) \oplus a = \infty(L \oplus a) = \infty(a \oplus L), \qquad \infty(K) \oplus \infty(L) = \infty(\emptyset),$$

where $a \in \mathbb{N}$ and $L \oplus a = \{l \oplus a \colon l \in L\}$. For a sum of k tokens on a digraph $G = (V, E)$, let $u = (u_1, \ldots, u_k)$. We then have $\gamma(u) = \gamma(u_1) \oplus \cdots \oplus \gamma(u_k)$, and

$$\begin{aligned} \mathcal{P} &= \{u \colon \gamma(u) = 0\}, \\ \mathcal{N} &= \{u \colon 0 < \gamma(u) < \infty\} \cup \{u \colon \gamma(u) = \infty(K) \text{ and } 0 \in K\}, \\ \mathcal{D} &= \{u \colon \gamma(u) = \infty(K) \text{ and } 0 \notin K\}. \end{aligned} \tag{3}$$

Thus a sum consisting of a token on vertex 4 and one on 8 in Figure 8 has γ-value $1 \oplus \infty(1) = \infty(1 \oplus 1) = \infty(0)$, which is an N-position (the move $8 \to 7$ results in a P-position). Also one token on 11 or else on 12 is an N-position. But a token on both 11 and 12; or on 8 and 12 is a D-position of their sum, with γ-value $\infty(\emptyset)$. Also a token on 7 and 12 is a D-position, since $\infty(0) \oplus 1 = \infty(0 \oplus 1) = \infty(1)$. A token on 4 and 7 is a P-position of the sum.

With k tokens on a digraph, the strategy for the sum can be computed in $O((k+mn)\log n)$ steps. It is polynomial in the input size $\Theta((k+m+n)\log n)$, since $k + mn \le (k + m + n)^2$. Also, for certain succinct "linear" graphs, γ provides a polynomial strategy. See [FT75].

Beat Craig is now also solved with a polynomial strategy. From the γ-values of Figure 9 we see that the position given in Figure 7 has γ-value $0 \oplus 1 \oplus 2 \oplus \infty(2,3) = 3 \oplus \infty(2,3) = \infty(1,0)$, so by (3) it's an N-position, and the unique winning move is $\infty(2,3) \to 3$. Again the winner can force a win in polynomial time.

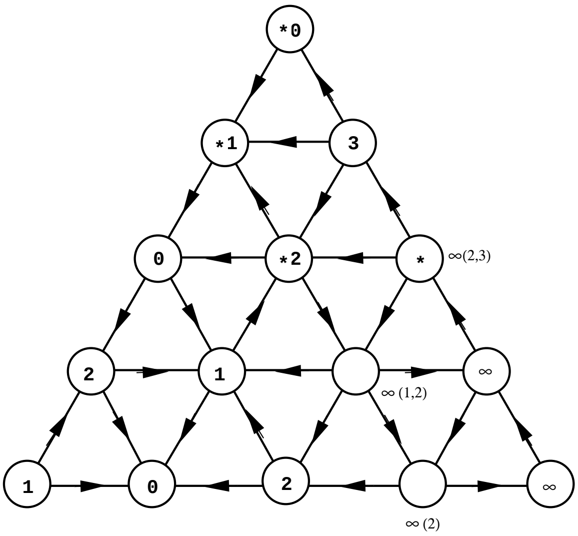


Figure 9: Craig has also been beaten.

As an exercise, beat an even bigger Craig: compute the labels P, N, D for the digraph of Figure 10 with tokens placed on vertices $A - E$, or for various other initial token placements.

We end this section with the *Fundamental Theorem of Combinatorial Game Theory* for impartial games which may be cyclic.

Theorem 1. *Let Γ be a two-person cyclic game with perfect information whose game-graph may be infinite, without chance moves and without ties. Then for every position of Γ there either exists a winning move for precisely one of the two players, or else, both players can maintain a draw.*

Proof. Every position has at least one label from among $\{\mathcal{P}, \mathcal{N}, \mathcal{D}\}$. Indeed, for any position u which is neither in $\mathcal{P}$ nor in $\mathcal{N}$, Definition 2 implies $u \in \mathcal{D}$.

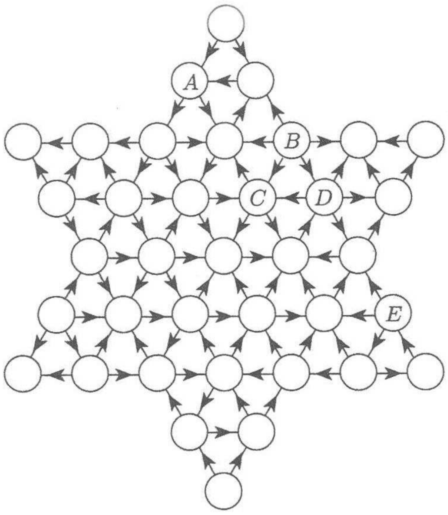


Figure 10: Beat an even bigger Craig.

So suppose that there exists $u_0 \in (\mathcal{P} \cap \mathcal{N})$. Then $u_0 \in (P_{m_0} \cap N_{k_0})$ for some ordinals $k_0, m_0 \in \mathcal{O}$. It then follows that $F(u_0) \subseteq \mathcal{N}$, $F(u_0) \cap \mathcal{P} \neq \emptyset$. By Definition 2, there thus is $u_1 \in F(u_0)$ with $u_1 \in (P_{m_1} \cap N_{k_1})$, where $k_1 < m_0$, $m_1 < k_0$. Hence $F(u_1) \subseteq \mathcal{N}$, $F(u_1) \cap \mathcal{P} \neq \emptyset$. Thus there is $u_2 \in F(u_1)$ with $u_2 \in (P_{m_2} \cap N_{k_2})$, where $k_2 < m_1$, $m_2 < k_1$. This leads to two infinite sequences $k_0 > m_1 > k_2 > m_3 > \ldots$ and $m_0 > k_1 > m_2 > k_3 > \ldots$, such that $u_i \in (P_{m_i} \cap N_{k_i})$ for all $i \in \mathbb{N}$. This contradicts the well-ordering of the ordinals. Hence $(\mathcal{P} \cap \mathcal{N}) = \emptyset$.

By the definition of $\mathcal{D}$ in Definition 2, $\mathcal{N} \cap \mathcal{D} = \mathcal{P} \cap \mathcal{D} = \emptyset$. We have shown that every position of Γ gets a unique label from among $\{P, N, D\}$. □

4 Adding Interactions between Tokens

Here we learn how to beat Anne (**Ann**ihilation). On the five-component digraph depicted in Figure 11, place tokens at arbitrary locations, but at most one token per vertex. A move is defined as in the previous games, but if a token is moved onto an occupied vertex, both tokens are annihilated (removed). The digraph has cycles, and could also have loops (passing positions). Note that the three components with z-vertices are identical, as are the two y-components. The only difference between a z- and a y- component is in the orientation of the top horizontal edge. With tokens on the twelve starred vertices, can the first player win or at least draw, and if so, what's an optimal move? How "good" is the strategy? The indicated position may be a bit complicated as a starter. So consider first a position consisting of four tokens only: one on z_0 and the other on z_2 in two of the z-components. Secondly, consider the position also consisting of four tokens: a single token on each of y_0 and y_2 in each y-component. It's clear that in both of these games player II can at least draw, simply by imitating on one component what player I does on the other. Can player II actually win in one or both of these games?

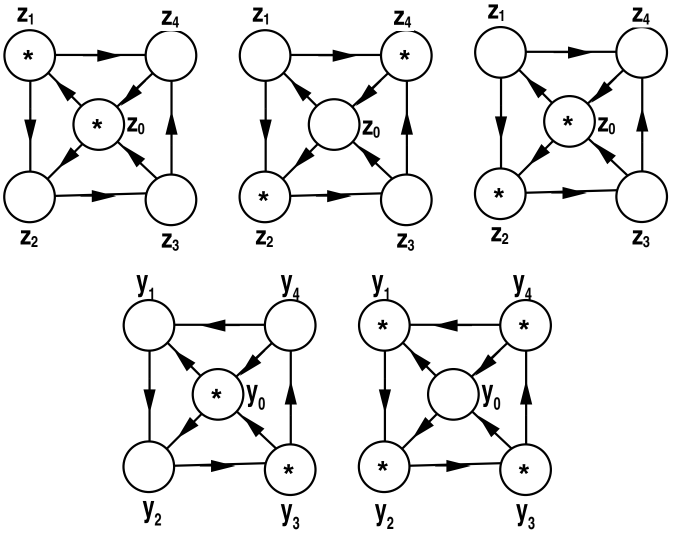


Figure 11: Beat Anne in this ANNihilation game.

Annihilation games were proposed by John Conway. It's easy to see that on a finite *acyclic* digraph, annihilation can affect the length of play, but the strategy is the same as for the classical games: Since $g(u) \oplus g(u) = 0$, the winner doesn't need to use annihilation, and the loser cannot be helped by it. But the situation is quite different in the presence of cycles. In Figure 12(a), a token on each of the vertices z_1 and z_3 is clearly a D-position for the nonannihilation case, but it's a P-position when played with annihilation (the second move is a winning annihilation move). In Figure 12(b), with annihilation, a token on each of z_1 and z_2 is an N-position, whereas a token on each of z_1 and z_3 is a D-position. The theory of annihilation games is discussed in depth in [FY82]; see also [Fra74], [FY76], [FY79], [FTY78]. Misère annihilation play was analyzed by Ferguson [Fer84].

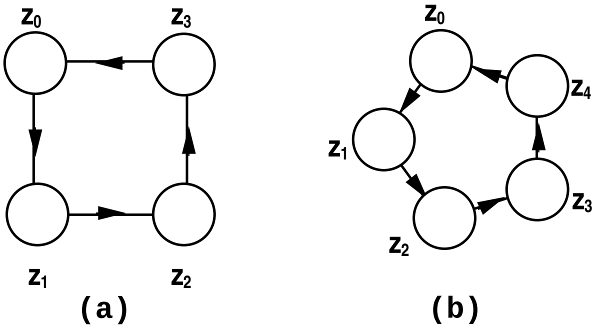


Figure 12: Annihilation on simple cyclic digraphs.

The **annihilation graph** is a certain game-graph of an annihilation game. The annihilation graph of the annihilation game played on the digraph of Figure 12(a) consists of two components. One is depicted in Figure 13(b), namely, the component $G^0 = (V^0, E^0)$ with 8 vertices and an even number of tokens. The "odd" component G^1 also has 8 vertices. In general, a digraph $G = (V, E)$ with $|V| = n$ vertices has an annihilation graph $G = (V, E)$ with $|V| = 2^n$ vertices, namely all n-dimensional binary vectors. The γ-function on G determines whether any given position is in $\mathcal{P}$, N or $\mathcal{D}$, according to (3); and γ, together with its associated counter function, determines an optimal next move from an N- or D-position.

The only problem is the exponential size of G. We can recover an $O(n^6)$ strategy by computing an **extended γ-function** σ on an induced subgraph of G of size $O(n^4)$, namely, on all vectors of weight ≤ 4 (at most four 1-bits). In Figure 14, the numbers inside the vertices are the σ-values, computed by Gaussian elimination over GF(2) of an $n \times O(n^4)$ matrix. This computation

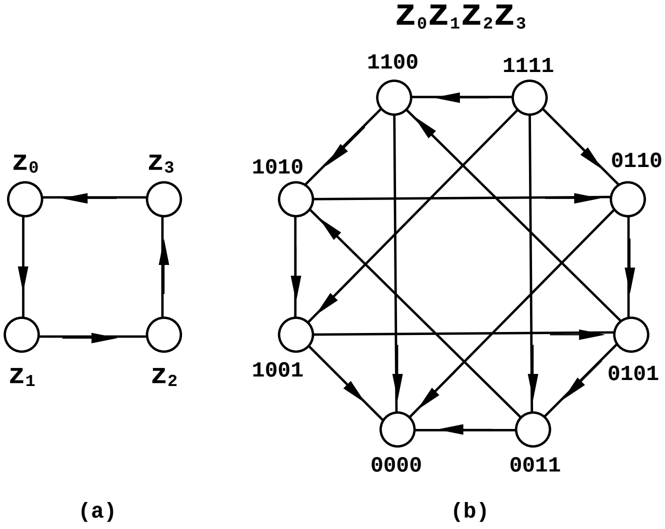


Figure 13: (b) depicts the "even" component G^0 of the annihilation graph G of the digraph (a).

also yields the values $t = 2$ for Figure 14(a) and $t = 1$ for Figure 14(b): If $\sigma(u) \geq 2^t$, then $\gamma(u) = \infty$, whereas $\sigma(u) < 2^t$ implies $\gamma(u) = \sigma(u)$.

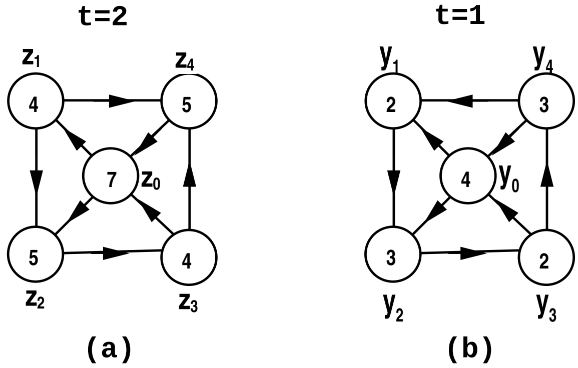


Figure 14: The σ-function.

Thus for Figure 14(a) we have $\sigma(z_0, z_2) = 5 \oplus 7 = 2 < 4$, so $\gamma(z_0, z_2) = 2$. Hence two such copies constitute a P-position ($2 \oplus 2 = 0$). (How can player II consummate a win?) In Figure 14(b) we have $\sigma(y_0, y_2) = 3 \oplus 4 = 7 > 2$, so $\gamma(y_0, y_2) = \infty$, in fact, $\infty(0, 1)$, so two such copies constitute a D-position.

(How can the two players maintain the draw?) We have thus answered the two questions posed in the second paragraph of the present section.

The position given in Figure 11 is repeated in Figure 15, together with the σ-values. From left to right we have: for the z-components, $\gamma = 3 \oplus 0 \oplus 2 = 1$; and for the y-components, $\infty(0,1) \oplus 0 = \infty(0,1)$, so the γ-value is $\infty(0,1) \oplus 1 = \infty(0,1)$. Hence the position is an N-position by (3). There is, in fact, a unique winning move, namely $y_0 \to y_2$ in the first component from the left. Any other move leads to drawing or losing. We have learned how to beat Anne.

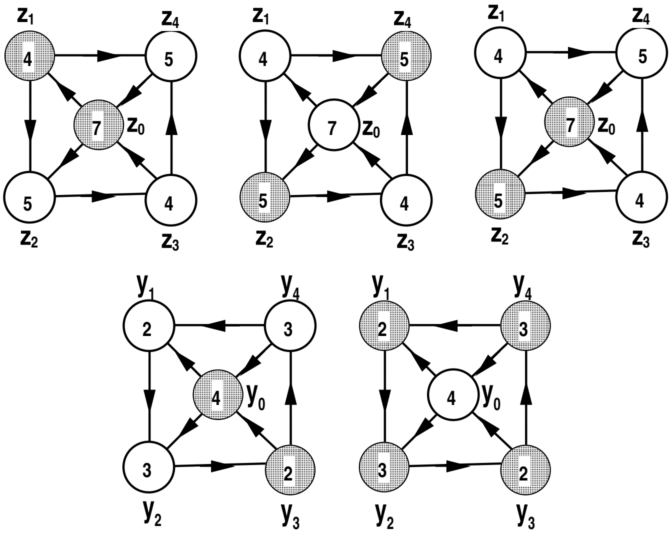


Figure 15: Poor beaten Anne. (Gray circles show initial token positions.)

For small digraphs, a counter function c is not necessary, but for larger ones it is needed for consummating a win. There is a problem in computing c: our polynomial algorithm produces γ and c only for an $O(n^4)$ portion of G. Whereas γ can then be extended easily to all of G, this does not seem to be the case for c. There is a way out involving a **broad** strategy.

A strategy is **narrow** if it uses only the present position u for deciding whether u is a P-, N-, or D-position, and for computing a next optimal move. It is **broad** [Fra91] if the computation involves any of the possible predecessors of u, whether actually encountered or not. It is **wide** if it uses any ancestor that was actually encountered in the play of the game. Wide strategies were defined by Kalmár [Kal28] and Smith [Smi66], but then both authors immediately reverted back to narrow strategies, since both authors remarked that the former do not seem to have any advantage over the latter. Yet for annihilation games, only a broad strategy was found that is polynomial. For details see [FY82].

For certain (Chinese) variations of Go, for Chess and some other games, there are rules that forbid certain repetitions of positions, or modify the outcome in the presence of such repetitions. Now if all the history is included in the definition of a move, then every strategy is narrow. But the way [Kal28] and [Smi66] defined a move — much the same as the intuitive meaning — there is a difference between a narrow and wide strategy for these games.

As an exercise, compute the label $\in \{P, N, D\}$ of the stellar configuration marked by letters in "Interstellar encounter with Jupiter" (Figure 16), where J is Jupiter, the other letters are various fragments of the Shoemaker–Levy comet, and all the vertices are "space-stations". A move consists of selecting Jupiter or a fragment, and moving it to a neighboring space-station along a directed trajectory. Any two bodies colliding on a space-station explode and vanish in a cloud of interstellar dust. Whereas in "Beat Anne" there is no leaf, here there are six "black holes", where a body is absorbed and cannot escape. Both players are viciously bent on making the final move to destroy this solar subsystem. Is the given position a win for player I or for player II? Or is it a draw, so that a part of this subsystem will exist forever? And if so, can it be arranged for Jupiter to survive as well? (An encounter of the Shoemaker–Levy comet with Jupiter took place in mid-July, 1994.)

Various impartial and partizan variations of annihilation games were shown to be NP-hard, PSPACE-complete or EXPTIME-complete [GR95], [FG87], [GR95]. We mention here only briefly an interaction related to annihilation. Electrons and positrons are positioned on vertices of the game Matter and Antimatter (Figure 17). A move consists of moving a particle along a directed trajectory to an adjacent station — if not occupied by a particle of the same kind, since two electrons (and two positrons) repel each other. If there is a resident particle, and the incoming particle is of the opposite type, they annihilate each other, and both disappear from the play. It is not very hard to determine the label of any position on the given digraph. But what can be said about a general digraph? About succinct digraphs? Note that the special case where all the particles are of the same type, is the generalization of Welter played on the given digraph. Welter is Nim with the restriction that no two piles have the same size. It has a polynomial strategy, but its validity proof is rather intricate [Con01] (ch. 13).

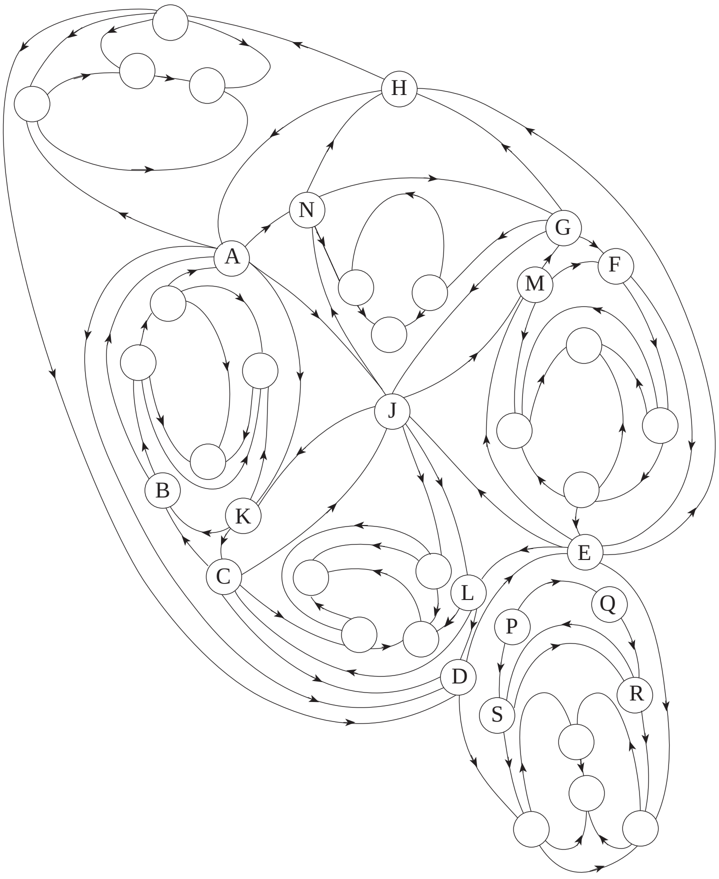


Figure 16: Interstellar encounter with Jupiter.

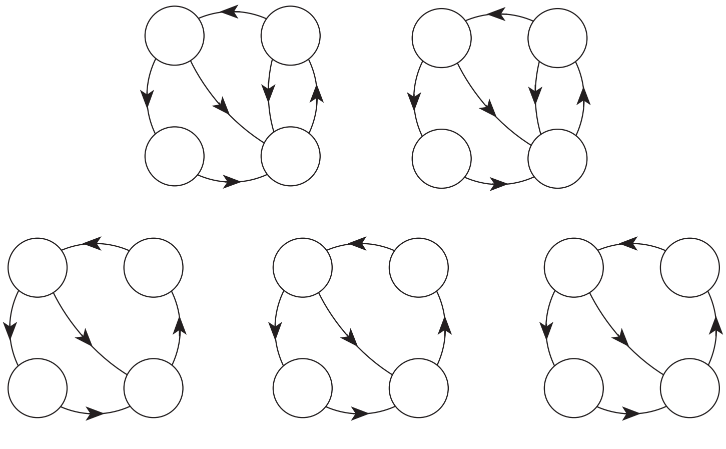


Figure 17: Matter and antimatter.

5 Partizan Games

In a **partizan combinatorial game** there are two players, Left and Right, who have distinct sets of moves available from each position. A game G is **short** if it meets both of the following conditions:

- G is **finite**: it has just finitely many distinct subpositions; and
- G is **acyclic**: there is no infinite sequence of moves proceeding from G.

Formally, a short partizan game G can be represented as an ordered pair $(\mathscr{G}^L, \mathscr{G}^R)$, where $\mathscr{G}^L$ and $\mathscr{G}^R$ are sets of "simpler" games (that is, games with strictly fewer subpositions). Elements of $\mathscr{G}^L$ (respectively $\mathscr{G}^R$) are called **Left** (respectively **Right**) **options** of G. We'll sometimes write

$$G = \{\mathscr{G}^L \mid \mathscr{G}^R\},$$

though we'll usually list the options of G explicitly:

$$G = \{G_1^L, G_2^L, \ldots, G_m^L \mid G_1^R, G_2^R, \ldots, G_n^R\}$$

or abuse notation and write simply

$$G = \{G^L \mid G^R\}$$

to indicate that G^L and G^R range over all the Left and Right options of G. The simplest game is the **empty game** 0, from which there are no options for either player:

$$0 = \{\ \mid\ \}.$$

Then we define the set of short games $\tilde{\mathbb{G}}$ by

$$\tilde{\mathbb{G}}_0 = \{0\}; \qquad \tilde{\mathbb{G}}_{n+1} = \Big\{\{\mathscr{G}^L \mid \mathscr{G}^R\} : \mathscr{G}^L, \mathscr{G}^R \subset \tilde{\mathbb{G}}_n\Big\}; \qquad \tilde{\mathbb{G}} = \bigcup_{n \geq 0} \tilde{\mathbb{G}}_n.$$

The theory of partizan games was introduced by Berlekamp, Conway and Guy in the 1970s and early 1980s. The classical texts *Winning Ways for Your Mathematical Plays* [BCG04] and *On Numbers and Games* [Con01] remain excellent introductions.

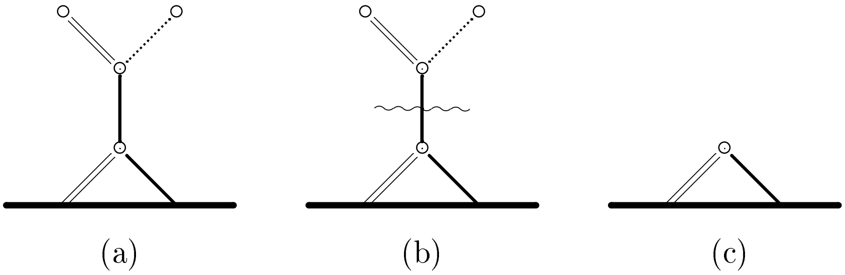


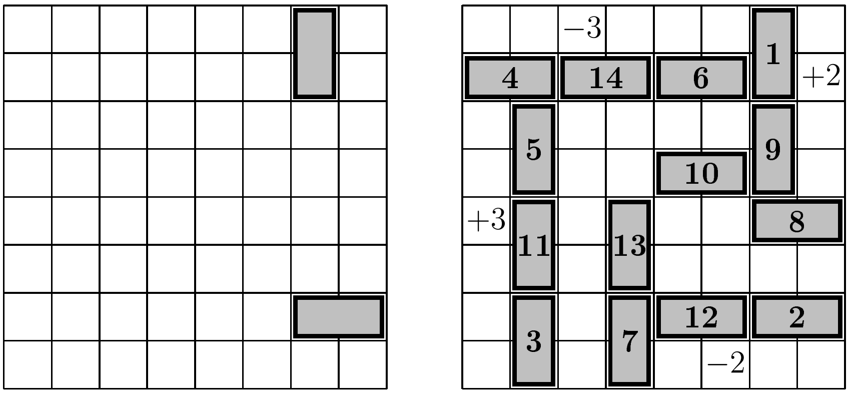
Figure 18: (a) A Hackenbush position; (b) A typical opening move for Left; (c) The resulting position after Left's move.

5.1 Two Examples: Hackenbush and Domineering

Hackenbush is played on a finite undirected graph with colored edges, such as the one in Figure 18(a). The solid horizontal line in Figure 18(a) represents a single vertex of the graph, the **ground**. On her turn, Left may remove any bLue (soLid) edge; Right may remove any Red (paRallel) one. GrEen (dottEd) edges may be removed by either player. After each move, any edges no longer connected to the ground are also removed from play. Hackenbush follows the same normal-play convention as Nim: whoever makes the last move wins.

Domineering is played on an $m \times n$ checkboard, typically 8×8. Left and Right alternately place dominoes on the board. Each domino must cover exactly two adjacent squares, and dominoes may never overlap. Moreover, Left must place verticaLly-oriented dominoes, and Right must place hoRizontally-oriented ones. Eventually, the players will run out of moves (since the board will fill up with dominoes), and whoever makes the last move wins. (Notice that making the last move coincides with placing the most dominoes, with ties broken in favor of the second player.)

Figure 19(a) shows a typical position after each player has made one move: Left made an opening move in the northeast corner of the board, and Right responded in the southeast. Figure 19(b) shows the first fourteen moves of a game played between David Wolfe and Dan Calistrate, in the finals of the first (and last) World Domineering Championship. Left 13 was a fatal mistake, and after Right 14 Calistrate went on to win the match and the tournament. A description of the Wolfe–Calistrate Domineering match can be found in [Wes96].



(a) (b)

Figure 19: (a) A typical DOMINEERING opening; (b) The first fourteen moves of Wolfe–Calistrate 1994, Round 3.

Notice that the position in Figure 19(b) can be subdivided into six separate territories, and no single move can affect more than one such component. Subsequent play on the four components labelled +3, +2, −2 and −3 is entirely predictable: Left will place exactly n dominoes on each $+n$ component, and Right will place n dominoes on each $-n$ component. The remaining two regions are more exciting; their resolutions depend on who plays first on which territory. Assigning meaningful mathematical values to such components, and describing their combinatorial interactions, is a central goal of the partizan theory.

5.2 Outcomes and Sums

If G is a short partizan game, then G belongs to one of four **outcome classes**:

$\mathscr{N}$ first player (the Next player) can force a win;

$\mathscr{P}$ second player (the Previous player) can force a win.

$\mathscr{L}$ Left can force a win, no matter who moves first;

$\mathscr{R}$ Right can force a win, no matter who moves first;

The proof that every game belongs to one of these four classes is a trivial generalization of Theorem 1, the Fundamental Theorem for impartial games.

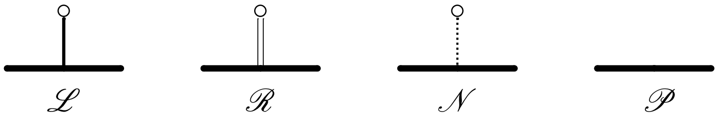


Figure 20: Four HACKENBUSH positions with distinct outcome classes.

We denote by $o(G)$ the outcome class of G. Figure 20 gives examples of HACKENBUSH positions representing all four classes.

The **disjunctive sum** $G + H$ is formed as follows: Place copies of G and H side-by-side; on her turn, a player must move in *exactly one* of the two components. Formally, we may write

$$G + H = \left\{G^L + H,\ G + H^L \,\middle|\, G^R + H,\ G + H^R\right\}. \tag{†}$$

Here G^L ranges over all Left options of G, and H^L ranges over all Left options of H, so that the Left options of $G + H$ are given by the union

$$\left\{X + H : X \in \mathscr{G}^L\right\} \cup \left\{G + Y : Y \in \mathscr{H}^L\right\}. \tag{‡}$$

The notation in the equation marked (†) is generally clearer and more succinct than set notation (‡), and we'll use it throughout this article without further comment.

Each game G also has a **negative** $-G$, obtained by interchanging the roles of Left and Right:

$$-G = \left\{-G^R \,\middle|\, -G^L\right\}$$

We write $G - H$ as shorthand for $G + (-H)$.

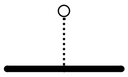
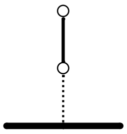
The definition of disjunctive sum is motivated by examples such as DOMINEERING, in which endgame positions decompose naturally into sums. The position in Figure 19(b), for example, can be written as the sum of six independent territories. Likewise, positions in NIM and KAYLES can be written as the disjunctive sum of single piles.

This modularity is central to combinatorial game theory. Given a sum of games

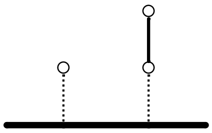
$$G = G_1 + G_2 + \cdots + G_k,$$

it is often impractical to undertake a brute-force analysis of G itself. Instead, we study the components G_i individually, and attempt to extract information that can be pieced back together to determine $o(G)$. In Section 2, this "information" took the form of **nim values**; in the context of partizan games, a more general notion of **game value** is needed.

Observe that it's not always sufficient to know the outcomes of each component. For example, let G and H be the following simple HACKENBUSH positions:

$G =$  $H =$ 

Then $o(G) = o(H) = \mathscr{N}$: either player can win immediately (on either game, played in isolation) by chopping the unique green edge, moving to 0. Also $o(G + G) = \mathscr{P}$, by the obvious symmetry argument. However on the sum

$G + H =$ 

Left can win no matter who moves first, since she can arrange that *Right* is always first to chop a green edge. So $o(G + H) = \mathscr{L}$, and this shows that G and H have unequal values.

5.3 Values

If G and H are partizan games, then we write

$$G = H \quad \text{if} \quad o(G + X) = o(H + X) \text{ for all } X.$$

Here X ranges over *all* short partizan games (that is, all elements of $\tilde{\mathbb{G}}$). In particular, suppose G and H are HACKENBUSH positions. Then X ranges over all HACKENBUSH positions, but *also* over games that are not necessarily representable in HACKENBUSH. This is deliberate: the universal quantifier is essential in order to get a good theory, and as we'll see in a moment it provides a common language for identifying shared structure in combinatorial games.

The **game value** of G is its equivalence class modulo equality. The idea is that the given an arbitrary sum

$$G = G_1 + G_2 + \cdots + G_k,$$

the value, and hence the outcome, of G can be computed from the values of each G_i. The set of game values is denoted by $\mathbb{G}$.

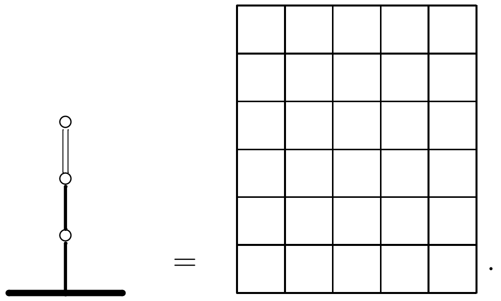
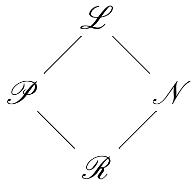


Figure 21: A nontrivial identity between HACKENBUSH and DOMINEERING.

Figure 21 gives a nontrivial example of two games with the same value.

The outcome classes are naturally partially-ordered by *favorability to Left*:



This induces a partial order of $\mathbb{G}$:

$$G \geq H \quad \text{if} \quad o(G+X) \geq o(H+X) \text{ for every all } X.$$

If $G \geq H$, then Left will be satisfied to replace the component H with G, in any conceivable sum of games. The basic theorems are as follows:

Theorem 2. *$o(G) \geq \mathscr{P}$ if and only if $G \geq 0$, for all short games G.*

Theorem 3. *$\mathbb{G}$ is a partially-ordered Abelian group under disjunctive sum, with identity* 0.

Note that $o(G) \geq \mathscr{P}$ if and only if Left can force a win on G as second player. So Theorem 2 implies that *every second-player win is equal to* 0. This directly generalizes the impartial theory, in which every second-player win has nim value 0.

We'll also write $G \cong H$ to mean that G and H are identical (**isomorphic**) games. Certainly $G \cong H$ implies $G = H$, but $G = H$ does *not* imply $G \cong H$ (since in particular, if G is *any* second-player win, then $G = 0$).

5.4 Simplest Forms

The central result of the partizan theory is the *Simplest Form Theorem*: every game value has a unique simplest representative. The Simplest Form Theorem is obtained through the following explicit construction.

For a given G, we identify several types of "extraneous" options:

- A Left option G^{L_1} is **dominated** (by G^{L_2}) if $G^{L_2} \geq G^{L_1}$ for some other Left option G^{L_2}.
- A Right option G^{R_1} is **dominated** (by G^{R_2}) if $G^{R_2} \leq G^{R_1}$ for some other Right option G^{R_2}.
- A Left option G^{L_1} is **reversible** (through $G^{L_1R_1}$) if $G^{L_1R_1} \leq G$ for some Right option $G^{L_1R_1}$.
- A Right option G^{R_1} is **reversible** (through $G^{R_1L_1}$) if $G^{R_1L_1} \geq G$ for some Left option $G^{R_1L_1}$.

Dominated options can be removed from G without affecting its value: in any sum $G + X$ from which Left would like to move to $G^{L_1} + X$ (with G^{L_1} dominated by G^{L_2}), she is equally satisfied to play $G^{L_2} + X$ instead.

Reversible options are a bit more subtle. If G^{L_1} is reversible through $G^{L_1R_1}$, then G^{L_1} can be replaced with the set of all $G^{L_1R_1L}$, without affecting the value of G. Symbolically:

$$G = \left\{ G^{L_1R_1L}, G^{L'} \mid G^R \right\},$$

with $G^{L'}$ ranging over all Left options of G *except* G^{L_1}. This operation is known as **bypassing** the reversible move G^{L_1} (through $G^{L_1R_1}$).

Any game G can be simplified by repeatedly eliminating dominated options and bypassing reversible ones. Each such operation strictly reduces the number of edges in the game tree of G, so this process necessarily produces a game K with no dominated or reversible options, and such that $K = G$. Such K is called the **canonical form** or **simplest form** of G, and the following theorem shows that it is unique.

Theorem 4 (Simplest Form Theorem). *Suppose that $G = H$, and neither G nor H has any dominated or reversible options. Then $G \cong H$.*

The Simplest Form Theorem follows immediately by inductive application of the following lemma:

Lemma 1. *Suppose that $G = H$, and neither G nor H has any dominated or reversible options. Then for every H^L, there is a G^L such that $G^L = H^L$, and vice versa; and likewise for Right options.*

Proof. Consider a Left option H^L. Since $G - H \geq 0$, Left must have a winning response to Right's opening move $G - H^L$. In particular, either $G^L - H^L \geq 0$ for some G^L, or else $G - H^{LR} \geq 0$ for some H^{LR}. But the latter would imply

$$H = G \geq H^{LR},$$

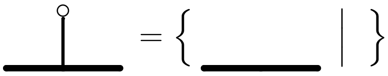
contradicting the assumption that H has no reversible options. So necessarily $G^L \geq H^L$ for some G^L. An identical argument now shows that $H^{L'} \geq G^L$ for some $H^{L'}$, so that

$$H^{L'} \geq G^L \geq H^L.$$

But H has no dominated options, so none of the inequalities can be strict, and in particular $G^L = H^L$. Proofs of the other cases are the same. □

5.5 Numbers

Consider a single blue Hackenbush stalk, from which Left can move to 0, and Right has no move at all:

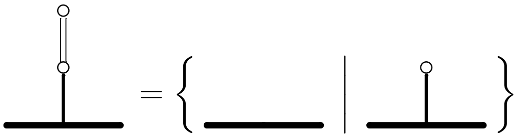
 $= \{0 \mid \;\}$

This game is denoted by 1, since it behaves like one spare move for Left. Since $1 > 0$, it generates a subgroup of $\mathbb{G}$ isomorphic to $\mathbb{Z}$, and it is customary to identify this subgroup with $\mathbb{Z}$. In particular we have

$$2 = 1 + 1 = \{1 \mid \;\}, \quad 3 = 2 + 1 = \{2 \mid \;\}, \quad \ldots$$

and in general $n + 1 = \{n \mid \;\}$, and $-(n+1) = \{\; \mid -n\}$.

In Figure 22 we see various other **numbers**, for example

$\frac{1}{2} =$  $= \{0 \mid 1\}.$

The identity $\frac{1}{2} + \frac{1}{2} = 1$ is easily verified by showing that the difference game

$$\tfrac{1}{2} + \tfrac{1}{2} - 1$$

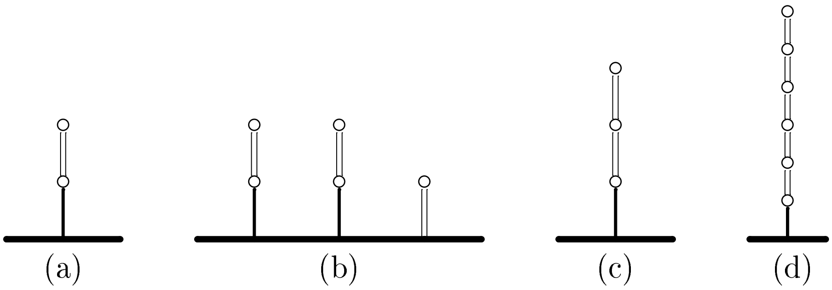


Figure 22: HACKENBUSH positions: (a) $\frac{1}{2}$; (b) $\frac{1}{2} + \frac{1}{2} - 1$; (c) $\frac{1}{4}$; (d) $\frac{1}{32}$

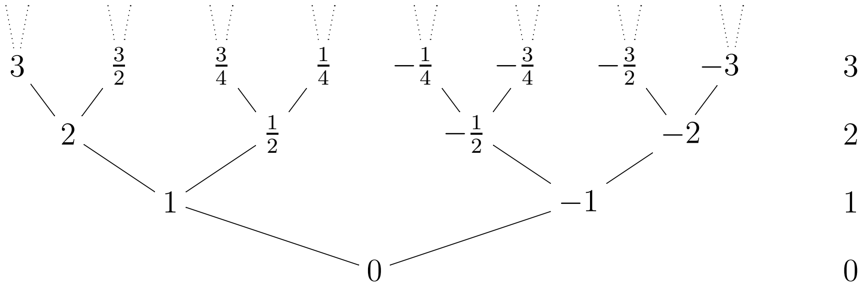


Figure 23: The Number Tree (with birthdays labeled on the right).

is a second-player win. Larger denominators can be similarly constructed:

$$\frac{1}{2^{n+1}} = \left\{ 0 \;\middle|\; \frac{1}{2^n} \right\}$$

and such numbers generate a subgroup of $\mathbb{G}$ isomorphic to $\mathbb{D}$, the group of **dyadic rationals**:

$$\mathbb{D} = \big\{ q \in \mathbb{Q} : 2^n q \in \mathbb{Z} \text{ for some } n \geq 0 \big\}.$$

The canonical form of $m/2^n$ (in lowest terms) is given by

$$\frac{m}{2^n} = \left\{ \frac{m-1}{2^n} \;\middle|\; \frac{m+1}{2^n} \right\}.$$

The inductive structure of numbers is neatly visualized in Figure 23. For each $n \geq 0$, there are 2^n numbers with **birthday** exactly n.

Now if x is a number, then it is a disadvantage to move on x, in the sense that

$$x^L < x < x^R$$

for every x^L and x^R. Remarkably, this criterion characterizes the dyadic rationals.

Theorem 5. *Let x be a short game, and suppose that $y^L < y < y^R$ for every subposition y of x and every y^L and y^R. Then $x \in \mathbb{D}$.*

This observation has several fundamental consequences.

Theorem 6 (Number Avoidance Theorem)**.** *Suppose that x is equal to a number but G is not. If Left (resp. Right) has a winning move on $G + x$, then she can win by playing on G.*

Theorem 7 (Number Translation Theorem)**.** *Suppose that x is equal to a number but G is not. Then*

$$G + x = \left\{G^L + x \;\middle|\; G^R + x\right\}.$$

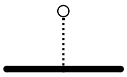
5.6 Infinitesimals

Numbers provide a natural metric against which other games can be calibrated. In particular, there is a vast hierarchy of games that are **infinitesimal** in the sense that

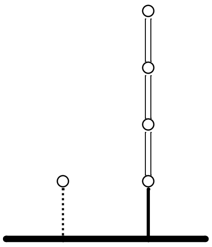
$$x > G > -x$$

for all positive numbers x.

The simplest nonzero infinitesimal is the game $*$ (pronounced “star”), from which either player can move to 0:

$$* = \{0 \mid 0\} =$$ 

It’s easily checked that $*$ is an infinitesimal, since on the sum

$$* + \frac{1}{2^n} \quad =$$ 

Left can win easily by playing preferentially on $*$, independent of the value of n.

Note that $*$ is isomorphic to a nim-heap of size 1. In the partizan context, a nim-heap of size m is denoted by $*m$ (pronounced "star m"). Symbolically:

$$*m = \{0, *, *2, \ldots, *(m{-}1) \mid 0, *, *2, \ldots, *(m{-}1)\}.$$

Each $*m$ (for $m \geq 1$) is a first-player win, and so is confused with 0. The simplest *signed* infinitesimals are

$$\uparrow = \{0 \mid *\} \quad (\text{"up"}) \qquad \text{and} \qquad \downarrow = -\uparrow = \{* \mid 0\} \quad (\text{"down"})$$

Certainly $\uparrow > 0$, since Left can win no matter who moves first. But $\uparrow$ is infinitesimal, by the same argument used for $*$: on $\downarrow + 2^{-n}$ (say), Left can win by playing preferentially on $\downarrow$.

5.7 Stops and the Mean Value

If G is not a number, then its **confusion interval** is given by

$$\mathcal{C}(G) = \{x \in \mathbb{D} : G \not\gtrless x\}.$$

The reader is invited to check the following examples:

- $\mathcal{C}(*) = \{0\}$, a singleton.
- $\mathcal{C}(\uparrow) = \emptyset$.
- $\mathcal{C}(\{3 \mid -3\})$ is the closed interval $[-3, 3]$.
- $\mathcal{C}(\{3 + * \mid -3\})$ is the half-open interval $[-3, 3)$.

The endpoints of $\mathcal{C}(G)$ are fundamental invariants of G, known as the **Left stop** $L(G)$ and **Right stop** $R(G)$ of G. Between them lies a third invariant, the **mean value** $m(G)$, which has the following remarkable properties:

$$m(G + H) = m(G) + m(H) \quad \text{for all } G \text{ and } H;$$

and for all G, the difference

$$\big(n \cdot G\big) - \big(n \cdot m(G)\big)$$

is bounded by a constant independent of n. Therefore $m(G)$ is a number that closely approximates the limiting behavior of many copies of G.

One can think of G as vibrating between its Left and Right stops in such a way that its "center of gravity" lies at $m(G)$.

6 Misère Play

We now return to the subject of impartial games, but considered under the **misère play convention**, in which the player who makes the last move *loses.* The misère theory was introduced by Plambeck and Siegel [Pla05, PS08]; see [Sie13] for a concise overview.

The Fundamental Theorem works in misère play too, with the same proof, so that every impartial game G has a **misère outcome** ($\mathscr{N}$ or $\mathscr{P}$) in addition to its normal outcome. The misère outcome of G is denoted by $o^-(G)$.

The motivating question in misère impartial games is this: What is the misère analogue of the Sprague–Grundy Theory? There are several reasonable answers to this question, each relevant in a different set of circumstances.

6.1 Misère Nim Value

Let G be a Nim position, with heaps of sizes $a_1, \ldots, a_k$. Recall that $o(G) = \mathscr{P}$ if and only if $a_1 \oplus \cdots \oplus a_k = 0$. A similar rule works in misère play, but it is slightly more complicated.

Theorem 8 (Bouton)**.** *The* Nim *position G with heaps $a_1, \ldots, a_k$ is a misère $\mathscr{P}$-position if and only if*

$$a_1 \oplus \cdots \oplus a_k = 0,$$

unless *every $a_i = 0$ or 1. In that case, G is a $\mathscr{P}$-position if and only if*

$$a_1 \oplus \cdots \oplus a_k = 1.$$

In particular, note that $*$ is a misère $\mathscr{P}$-position, but $*m$ is an $\mathscr{N}$-position for all $m \neq 1$. This motivates the following misère analogue of nim values. Recall that the *(normal) nim value* of G is given recursively by

$$\mathscr{G}(G) = \begin{cases} 0 & \text{if } G \cong 0; \\ \operatorname*{mex}_{G' \in G} \mathscr{G}(G') & \text{otherwise.} \end{cases}$$

The **misère nim value** is similarly defined, but with a different base case:

$$\mathscr{G}^-(G) = \begin{cases} 1 & \text{if } G \cong 0; \\ \operatorname*{mex}_{G' \in G} \mathscr{G}(G') & \text{otherwise.} \end{cases}$$

The misère nim value of G determines its outcome. In fact we can say something slightly stronger:

Theorem 9. *$\mathscr{G}^-(G)$ is the unique value of m such that $o^-(G + *m) = \mathscr{P}$. In particular, G is a misère $\mathscr{P}$-position if and only if $\mathscr{G}^-(G) = 0$.*

The problem with misère nim values is that they're not well-behaved in sums. For example, let $G = *$ and $H = *2 + *2$. Then G and H are both $\mathscr{P}$-positions (by Theorem 8), so

$$\mathscr{G}^-(G) = \mathscr{G}^-(H) = 0.$$

However it's not hard to show (using Theorem 9, say) that

$$\mathscr{G}^-(G + *2) = 3, \quad \text{but} \quad \mathscr{G}^-(H + *2) = 2.$$

So the misère nim value of a sum of games can't be determined from the nim values of its components.

6.2 Genus Theory

The **genus** of G (plural **genera**), denoted by $\mathscr{G}^{\pm}(G)$, is obtained by conjoining its normal and misère nim values:

$$\mathscr{G}^{\pm}(G) = \big(\mathscr{G}(G), \mathscr{G}^-(G)\big).$$

For brevity it's customary to write $\mathscr{G}^{\pm}(G) = a^b$ in place of $\mathscr{G}^{\pm}(G) = (a, b)$.

Remarkably, genus values *are* well-behaved in sums, but *only* for a particular class of games known as **tame games**. First note that one can easily classify all the genera that arise in misère Nim:

- If G has no heaps of size ≥ 2, then $\mathscr{G}^{\pm}(G) = 0^1$ or 1^0, depending on the parity of the number of heaps of size 1.
- Otherwise, $\mathscr{G}^{\pm}(G) = a^a$, where $a = \mathscr{G}(G)$. (This follows from Theorems 8 and 9.)

So the only genera in misère Nim are 0^1, 1^0, and those of the form a^a for some $a \geq 0$. An arbitrary game G is **tame** if all its subpositions have genus values drawn from this ensemble.

The nim-addition operator $\oplus$ extends to tame genera according to the following addition table:

$$\begin{array}{ll}
0^1 \oplus 0^1 = 0^1 & \qquad a^a \oplus 0^1 = a^a \\
0^1 \oplus 1^0 = 1^0 & \qquad a^a \oplus 1^0 = (a \oplus 1)^{a \oplus 1} \\
1^0 \oplus 1^0 = 0^1 & \qquad a^a \oplus b^b = (a \oplus b)^{a \oplus b}
\end{array}$$

The main theorem is the following:

Theorem 10 (Conway)**.** *If G and H are tame, then so is $G + H$, and moreover*

$$\mathscr{G}^{\pm}(G + H) = \mathscr{G}^{\pm}(G) \oplus \mathscr{G}^{\pm}(H).$$

This provides a reasonably straightforward extension of the theory of misère Nim to arbitrary tame games. In particular, any tame game can be treated as a Nim position *in sums involving other tame games.*

For example, let $G = *$ and $H = *2 + *2$. We noted above that $\mathscr{G}^{-}(G) = \mathscr{G}^{-}(H) = 0$, but $G + *2$ and $H + *2$ have distinct misère nim values. This is explained by the fact that $\mathscr{G}^{\pm}(G) = 1^0$, but $\mathscr{G}^{\pm}(H) = 0^0$. There are two fundamentally different types of tame games with $\mathscr{G}^{-}$-value 0, corresponding to the two cases in the statement of Theorem 8.

Likewise, consider $J = *2$ and $K = *2 + *2 + *2$. Here we have $\mathscr{G}^{\pm}(J) = \mathscr{G}^{\pm}(K) = 2^2$. Since J and K have the same genus, Theorem 10 implies that $o^{-}(J + X) = o^{-}(K + X)$ for any *tame* X. *However*, consider the game

$$X = \{0,\ *2 + *3\}$$

whose options are 0 and $*2 + *3$. X is not tame (since its genus is 2^0), and indeed it's not hard to check that

$$o^{-}(J + X) = \mathscr{N}, \quad \text{whereas} \quad o^{-}(K + X) = \mathscr{P}.$$

So even though J and K are both tame and have the same genus, they nonetheless behave differently in sums with a suitable *wild* game. The question of how best to extend the genus theory to wild games is an ongoing research problem; the rest of this section will describe the (considerable) advances that have been made in this direction.

6.3 Misère Canonical Form

The most straightforward idea is simply to define *misère equality* for impartial games, the same way we defined equality for partizan games in Section 5:

$$G = H \quad \text{if} \quad o^{-}(G + X) = o^{-}(H + X) \text{ for all } X,$$

with X ranging over all impartial games. Then the **misère game value** of G is its equivalence class modulo misère equality. This obviously works, in the sense that misère game value is automatically well-behaved in sums. The central problem with misère nim values (and genus values for wild games) is therefore definitionally circumvented.

But misère game values suffer from a different problem, which is that there are rather a lot of them. If G is an impartial game, then an option $G' \in G$ is said to be **(misère) reversible** if there is some $G'' \in G'$ such that $G'' = G$. Obviously if G is misère reversible, then it is equal to a simpler game, namely G'', so this is a sort of analogue of partizan reversible moves from Section 5. The following theorem of Conway is one of the crowning results of the misère theory.

Theorem 11 (Conway)**.** *Suppose that $G = H$, and neither G nor H has any reversible moves. Then $G \cong H$.*

Theorem 11 says that reversible moves are *only* type of reduction available for impartial games. This is true in both normal and misère play: the "=" sign in Theorem 11 can be interpreted to mean *either* normal *or* misère equality (provided the corresponding notion of "reversible" is also used). In normal play, it's essentially a restatement of the Sprague–Grundy Theorem, so here we have a quite clear analogue of the normal-play theory.

Sadly, reversible moves in misère play are exceedingly rare. Consider the set of game values with birthday ≤ 6. In normal play, there are just seven of them:

$$0, *, *2, \ldots, *6.$$

Conversely, in misère play Conway has shown that there are more than $2^{4171779}$. In this sense misère game values spectacularly fail to yield a coherent theory.

6.4 Misère Quotients

The above results suggest that genus values preserve too little information, whereas misère game values preserve too much. The theory of *misère quotients* offers a third approach: rather than aim for a single, fully general extension of the Sprague–Grundy theory, we instead accept a multiplicity of *local* analogues.

Recall the definition of misère equality:

$$G = H \quad \text{if} \quad o^-(G + X) = o^-(H + X) \text{ for all } X.$$

In defining misère game values, we allowed X to range over all impartial games. If instead G, H and X are restricted to range over *tame* games, then the resulting equivalence classes correspond one-to-one with genus values (and in fact this is just a restatement of the genus theory). So genus values can be viewed as the structure obtained when misère equivalence is localized to the set of tame games.

Along these lines, let $\mathscr{A}$ be any nonempty set of impartial games that is **closed** in the following sense:

- If $G, H \in \mathscr{A}$, then $G + H \in \mathscr{A}$ (additive closure); and
- If $G \in \mathscr{A}$ and $G' \in G$, then $G' \in \mathscr{A}$ (hereditary closure).

Then define

$$G \equiv H \pmod{\mathscr{A}} \quad \text{if} \quad o^-(G+X) = o^-(H+X) \text{ for all } X \in \mathscr{A}.$$

Let $\mathcal{Q}$ be the corresponding set of equivalence classes. The closure assumptions on $\mathscr{A}$ imply that $\mathcal{Q}$ is a commutative monoid, and there is a surjective homomorphism

$$\Phi : \mathscr{A} \to \mathcal{Q}.$$

Denote by $\mathcal{P} \subset \mathcal{Q}$ the subset corresponding to $\mathscr{P}$-positions from $\mathscr{A}$:

$$\mathcal{P} = \{\Phi(G) : G \in \mathscr{A},\ o^-(G) = \mathscr{P}\}.$$

The structure $(\mathcal{Q}, \mathcal{P})$ is the **misère quotient** of $\mathscr{A}$, and is denoted by $\mathcal{Q}(\mathscr{A})$. It serves as a localized analogue of the Sprague–Grundy theory, in the following sense. Suppose that we wish to study a game $G \in \mathscr{A}$ that decomposes in $\mathscr{A}$:

$$G = G_1 + G_2 + \cdots + G_k, \quad \text{each } G_i \in \mathscr{A}.$$

Given the Φ-values of each G_i, say $x_i = \Phi(G_i)$, then we can multiply them out in the arithmetic of $\mathcal{Q}$ to determine $\Phi(G)$:

$$\Phi(G) = x = x_1 x_2 \cdots x_k$$

and then check whether $x \in \mathcal{P}$. So far we haven't said anything terribly profound. What's surprising (and what makes misère quotients so powerful) is that the monoid $\mathcal{Q}$ often turns out to be finite, even when $\mathscr{A}$ is infinite, and even when $\mathscr{A}$ contains some wild games. In such cases, the problem

$$\begin{aligned}\mathscr{Q} \cong \langle a,b,c,d,e,f,g \mid {} & a^2 = 1,\ b^3 = b,\ bc^2 = b,\ c^3 = c,\ bd = bc,\\ & cd = b^2,\ d^3 = d,\ be = bc,\ ce = b^2,\\ & e^2 = de,\ bf = ab,\ cf = ab^2c,\ d^2f = f,\\ & f^2 = b^2,\ b^2g = g,\ c^2g = g,\ dg = cg,\\ & eg = cg,\ fg = ag,\ g^2 = b^2\rangle\end{aligned}$$

$$\mathcal{P} = \{a, b^2, ac, ac^2, d, ad^2, e, ade, adf\}$$

Figure 24: The misère quotient of Kayles.

of determining the outcome of the sum G reduces to a small number of operations on the finite multiplication table $\mathscr{Q}$.

For a simple example, let $\mathscr{A}$ consist of all sums involving $*$ and $*2$. Then every element of $\mathscr{A}$ is tame, so the elements of $\mathscr{Q}$ correspond to genera of games in $\mathscr{A}$, which are restricted to the six possibilities

$$0^1,\ 1^0,\ 0^0,\ 1^1,\ 2^2,\ 3^3.$$

The structure of the corresponding monoid follows directly from the addition table for genus values:

$$\mathscr{Q} \cong \langle a, b : a^2 = 1,\ b^3 = b\rangle,$$

with $\mathcal{P} = \{a, b^2\}$, corresponding to genera 1^0 and 0^0.

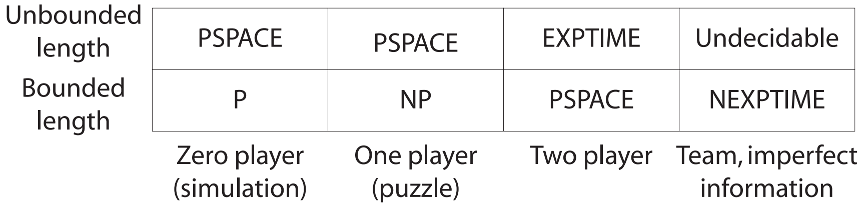
A fairly typical misère quotient is shown in Figure 24. It's the quotient of the set of positions in the game Kayles, and therefore succinctly describes the winning strategy for misère Kayles. It's worth noting that the original solution to misère Kayles ran forty-three pages long. A streamlined proof in *Winning Ways* reduced this to "just" five pages. That the entire proof can be encoded by the succinct monoid presentation in Figure 24 nicely illustrates the power of the quotient theory.

7 Constraint Logic

While combinatorial game theory seeks efficient algorithms for games, often no efficient algorithm exists. Then, we seek instead to show hardness. In recent years a new tool has emerged for proving hardness of games: **constraint logic** [DH08, HD09]. With constraint logic, the games we consider are both more specialized and more general than what is traditionally addressed by classical game theory. More specialized, because we are concerned only with determining the winner of a game, and not with other issues such as maximizing payoff, cooperative strategies, etc. More general, because classical game theory is concerned only with the interactions of two or more players, whereas constraint logic addresses, in addition, games with only one player (puzzles) and even with no players at all (simulations). Constraint logic offers, for a variety of types of game, a simple path to hardness reductions; generally a small number of constraint logic "gadgets" must be built out of components of the target game.

The starting point of constraint logic is the perspective that *games model computation*. Different types of game model different types of computation. For example, the idea of nondeterministic computation nicely matches the feature of puzzles that a player must choose a sequence of moves or piece placements to satisfy some global property. Thus, puzzles are often NP-complete (see section 2.1). Even more striking is the correspondence between alternation, the natural extension to nondeterminism, and two-player games. Constraint logic is a family of games (played on directed graphs) which model computation ranging from that of monotone Boolean circuits (P-complete) all the way to unrestricted Turing machines (undecidable). For any game to be analyzed, the category of game will suggest a potential complexity, which may be proved by a reduction from the corresponding type of constraint logic. The entire range of constraint-logic games and complexities is shown in Table 1.

The chief advantage in showing a game hard by a reduction from constraint logic, rather than from a standard problem such as SAT or QBF, is that constraint logic is very similar in nature to many actual games, often making reductions extremely simple. For example, essentially the entire proof that sliding-block puzzles are PSPACE-complete is contained in Figure 30 [HD05]. This problem, originally posed by Martin Gardner [Gar64], had



	Zero player (simulation)	One player (puzzle)	Two player	Team, imperfect information
Unbounded length	PSPACE	PSPACE	EXPTIME	Undecidable
Bounded length	P	NP	PSPACE	NEXPTIME

Table 1: Game categories and their natural complexities. Constraint Logic is complete in each class.

been open for nearly 40 years. Other games and puzzles shown hard via constraint logic include TipOver [Hea06a], sliding tokens (a dynamic version of Independent Set) [HD05], River Crossing [Hea04], Triangular Rush Hour [HD09], Push-2-F [DHH02], Amazons [Hea09], Konane [Hea09], Cross Purposes [Hea09], Hitori [HD09], and Wriggle puzzles [Max07]. Some games and puzzles with existing hardness proofs have also been shown hard via constraint logic, with simpler constructions (in some cases, also strengthening the existing results), including Sokoban [HD05], Rush Hour [HD05], and the Warehouseman's Problem [HD05]. Finally, constraint logic has also been applied to several problems outside the domain of games proper, including showing undecidability of some decision problems for multi-port finite-state machines [Hie10].

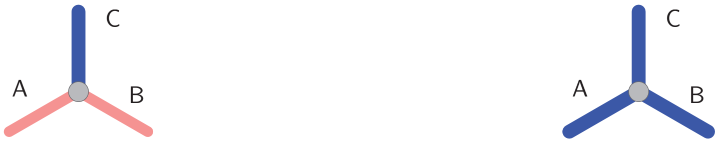
7.1 The Constraint-Logic Framework

The general model of games we develop is based on the idea of a **constraint graph**; the rules defining legal moves on such graphs are called **constraint logic**. In later sections the graphs and the rules will be specialized to produce one-player, two-player, etc. games.[2] A game played on a constraint graph is a computation of a sort, and simultaneously serves as a useful problem to reduce to other games to show their hardness.

A **constraint graph** is a directed graph with edge weights among $\{1, 2\}$. An edge is then called **red** or **blue**, respectively. The **inflow** at each vertex is the sum of the weights on inward-directed edges. Each vertex has a nonnegative **minimum inflow**. A **legal configuration** of a constraint graph has an inflow of at least the minimum inflow at each vertex; these are the **constraints**. A **legal move** on a constraint graph is the reversal of the direction of a single edge that results in a legal configuration. Generally, in any game, the goal will be to reverse a given edge by executing a sequence of (legal) moves. In multiplayer games, each edge is controlled by an individual player, and each player has his own goal edge. In deterministic games, a unique sequence of moves is forced. For the bounded games, each edge may only reverse once.

It is natural to view a game played on a constraint graph as a computation. Depending on the nature of the game, it can be a deterministic

[2] In the interest of space, we omit some of the definitions—and all discussion of zero-player games (Deterministic Constraint Logic)— and refer the reader to [Hea06b], [DH08], or [HD09].



(a) AND vertex. Edge C may be directed outward if and only if edges A and B are both directed inward.

(b) OR vertex. Edge C may be directed outward if and only if either edge A or edge B is directed inward.

Figure 25: AND and OR vertices. Red (light gray, thinner) edges have weight 1, blue (dark gray, thicker) edges have weight 2, and vertices have a minimum in-flow constraint of 2.

computation, or a nondeterministic computation, or an alternating computation, etc. The constraint graph then **accepts** the computation just when the game can be won.

AND/OR Constraint Graphs; Planarity. Certain vertex configurations in constraint graphs are of particular interest. An **AND vertex** (Figure 25(a)) has minimum inflow constraint 2 and incident edge weights of 1, 1, and 2. It behaves as a logical AND in the following sense: the weight-2 (blue) edge may be directed outward if and only if both weight-1 (red) edges are directed inward. Otherwise, the minimum inflow constraint of 2 would not be met. An **OR vertex** (Figure 25(b)) has minimum inflow constraint 2 and incident edge weights of 2, 2, and 2. It behaves as a logical OR: a given edge may be directed outward if and only if at least one of the other two edges is directed inward.

It turns out that for all the game categories, it will suffice to consider constraint graphs containing only AND and OR vertices. For some of the game categories, there can be many subtypes of AND and OR vertex, because each edge may have a distinguishing initial orientation (in the case of bounded games), and a distinct controlling player (when there is more than one player). In some cases there are alternate vertex "basis sets" that enable simpler reductions to other problems than do the complete set of ANDs and ORs.

For all but the bounded zero-player case, it also suffices to only consider planar constraint graphs. In practice this makes for much easier hardness reductions; often, crossover gadgets are the most difficult pieces of a reduction to construct. With constraint logic, we get them for free. The most common problem used to show NP-hardness is 3SAT, but in many instances this planarity property makes constraint logic reductions simpler.

Directionality; Fanout. As implied above, although it is natural to think of AND and OR vertices as having inputs and outputs, there is nothing enforcing this interpretation. A sequence of edge reversals could first direct both red edges into an AND vertex, and then direct its blue edge outward; in this case, we could say that its "inputs" have "activated", enabling its "output" to "activate". But the reverse sequence could equally well occur. In this case we could view the AND vertex as a splitter, or **FANOUT** gate: directing the blue edge inward allows both red edges to be directed outward, effectively splitting a signal.

In the case of OR vertices, again, we can speak of an active input enabling an output to activate. However, here the choice of input and output is entirely arbitrary, because OR vertices are symmetric.

7.2 One-Player Games

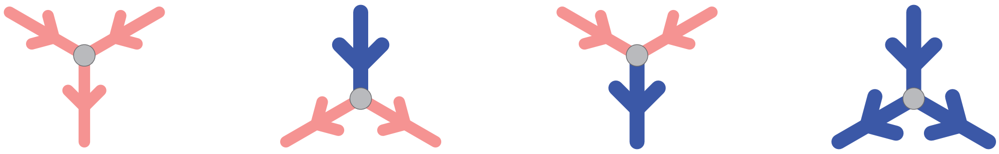
The one-player version of constraint logic is called Nondeterministic Constraint Logic (NCL). The rules are simply that on a turn the player reverses a single edge that results in a legal configuration. The goal is to reverse a particular edge.

7.2.1 Bounded Games

Bounded Nondeterministic Constraint Logic (Bounded NCL) is formally defined as follows:

BOUNDED NONDETERMINISTIC CONSTRAINT LOGIC (BOUNDED NCL)
INSTANCE: Constraint graph G, edge e in G.
QUESTION: Is there a sequence of moves on G that eventually reverses e, such that each edge is reversed at most once?

Bounded NCL is NP-complete (reduction from 3SAT). It remains NP-complete when the graph G is required to be a planar graph which uses



(a) CHOICE (b) AND (c) FANOUT (d) OR

Figure 26: Basis vertices for Bounded NCL.

only the vertex types shown in Figure 26.[3] It also turns out to be useful to reduce from graphs that have the property that only a single edge can initially reverse; this problem is also NP-complete.

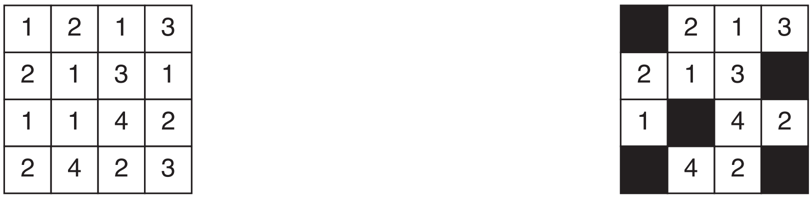
A related problem is Constraint Graph Satisfiability:

CONSTRAINT GRAPH SATISFIABILITY
INSTANCE: Unoriented planar constraint graph G using only AND and OR vertices.
QUESTION: Does G have a configuration that satisfies all the constraints?

Properly, this problem is not a constraint-logic game, because the moves (assignments of edge orientations) are not reversals from one legal configuration to another. But it is similar in spirit, and can prove useful for reductions. Constraint Graph Satisfiability is NP-complete (reduction from 3SAT). Note that for Constraint Graph Satisfiability, unlike proper Bounded NCL, only two types of vertex are needed.

Sample Application: Hitori. Hitori was popularized by Japanese publisher Nikoli, along with its more-famous sibling Sudoku, and several other "pencil-and-paper" puzzles. In Hitori, we are given a grid with each square labeled with an integer, and the goal is to paint a subset of the squares so that (1) no row or column has a repeated unpainted label (similar to Sudoku), (2) painted squares are never adjacent, and (3) the unpainted squares are all connected. A simple Hitori puzzle and its solution are shown in Figure 27. We give a reduction from Constraint Graph Satisfiability

[3] Here we show the initial, "inactivated" orientation of the edges. In an AND, the blue edge may reverse if the red edges first reverse; in a FANOUT, the red edges may reverse if the blue edge first reverses.



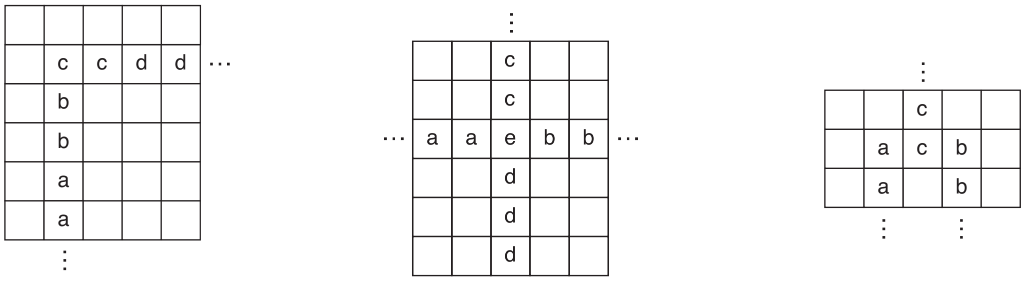
1	2	1	3
2	1	3	1
1	1	4	2
2	4	2	3

(a) Puzzle

■	2	1	3
2	1	3	■
1	■	4	2
■	4	2	■

(b) Solution

Figure 27: A simple Hitori puzzle and its solution.



	c	c	d	d	⋯
	b				
	b				
	a				
	a				
	⋮				

(a) Wire, turn

			⋮			
			c			
			c			
⋯	a	a	e	b	b	⋯
			d			
			d			
			d			

(b) OR, parity

		⋮		
		c		
	a	c	b	
	a		b	
	⋮		⋮	

(c) AND

Figure 28: Hitori gadgets.

(Section 7.2) showing that it is NP-complete to determine whether a given $n \times n$ Hitori puzzle has a solution [HD09].

Wiring. We represent graph edge orientation with wires, or strings of adjacent squares, consisting of integers $x_1, x_1, x_2, x_2, ..., x_{n-1}, x_{n-1}, x_n, x_n$, where the x_i are distinct. If the first x_1 is unpainted, then the next must be painted (by rule 1 above), forcing the first x_2 to be unpainted (by rule 2), etc.; thus the last x_n must be painted. If the first x_1 is painted, the last x_n may be painted or unpainted: we could (for example) have the second x_1 and the first x_2 both unpainted without violating the rules.

Wires may be turned, as in Figure 28(a): if the bottom a is unpainted, then the right d must be painted. (We assume that the unlabeled squares all contain distinct integers not otherwise used in the gadgets.)

OR Vertex / Parity Gadget. In Figure 28(b), first consider the ds. At most one can be unpainted, but no two adjacent may be painted. Therefore, both the lower and the upper one must be painted, and e must be unpainted.

If both the left a and the right b are unpainted, then the right a and the left b must be painted. As an unpainted square, e must be connected to the other unpainted squares (rule 3); the lower c is the only way out. Therefore, the lower c is unpainted, and the upper one painted. But if either the left a or the right b is painted, then the other a or b will be unpainted, allowing another way out for e. Then the lower c may be painted, and the upper c unpainted. These are the same constraints an OR vertex has, again with an unpainted "port" square (left a, right b, top c) corresponding to an outward-directed edge, and a painted port square corresponding to an inward-directed edge.

This gadget can also serve to alter the positional parity in wiring, so that the various gadgets can be connected arbitrarily, by using only one input, and blocking the other one (for example, by adding another b to the right of the right one).

AND Vertex. Similar but simpler reasoning as above shows that the gadget in Figure 28(c) satisfies the same constraints as an AND vertex, with the lower a and b ("inputs") corresponding to the red edges, and the upper c ("output") to the blue edge: the output square may be unpainted if and only if both input squares are painted.

Assembly. Given a planar AND / OR constraint graph, we construct a HITORI puzzle by connecting together AND and OR vertex gadgets with wires, adjusting positional parity as needed. If the graph has a legal configuration, then every wire can be painted so as to satisfy all the HITORI constraints, as described. Similarly, if the HITORI puzzle can be solved, then a legal graph configuration can be read off the wires.

7.2.2 Unbounded Games

NONDETERMINISTIC CONSTRAINT LOGIC (NCL) is the canonical form of constraint logic:

NONDETERMINISTIC CONSTRAINT LOGIC (NCL)
INSTANCE: Constraint graph G, edge e in G.
QUESTION: Is there a sequence of moves on G that eventually reverses e?

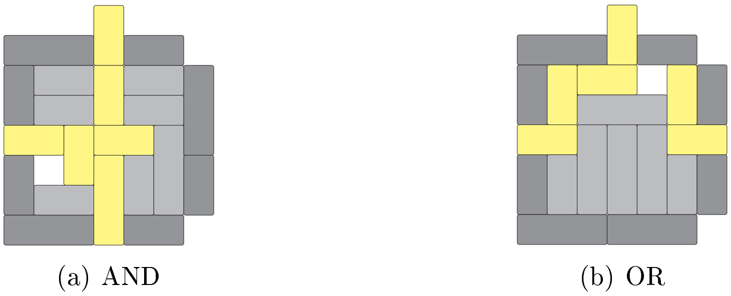


Figure 30: Constraint-logic gadgets showing PSPACE-completeness of Sliding Blocks.

NCL is PSPACE-complete (reduction from QBF), and remains PSPACE-complete when the graph G is required to be a planar graph which uses only AND and OR vertices (Figure 25). NCL reductions are often very straightforward, for two reasons. First, only two gadgets must be constructed. Second, one-player games (puzzles) are generally easier to reduce to than multi-player games. For these reasons, and because there is a large supply of candidate puzzles to analyze, NCL reductions form the largest set of existing constraint-logic reductions.

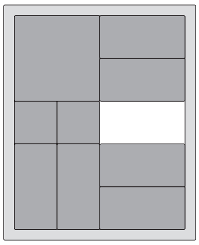


Figure 29: Dad's Puzzle.

Sample Application: Sliding Blocks. In the usual kind of sliding-block puzzle, one is given a box containing a set of rectangular pieces, and the goal is to slide the blocks around so that a particular piece winds up in a particular place. A popular example is Dad's Puzzle, shown in Figure 29; it takes 59 moves to slide the large square to the bottom left. We outline a reduction from Nondeterministic Constraint Logic (Section 7.2) showing that it is PSPACE-complete to determine whether a given sliding-block puzzle in an $n \times n$ box has a solution. For a formal proof (which is also stronger, using only 1×2 blocks), see [HD05] or [HD09].

AND Vertex. The gadget shown in Figure 30(a) satisfies the same constraints as an AND vertex. Assume that the outer, dark-colored "wall" blocks are fixed. Then, the only way the top "signal" (light-colored) block may slide

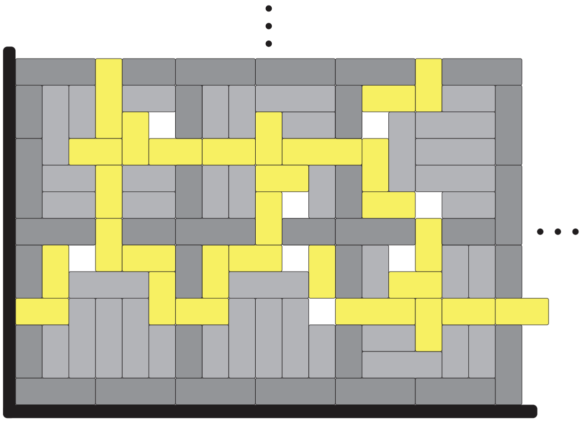


Figure 31: Sliding-block gadget assembly.

down is if the left signal block first slides left, and bottom signal block slides down. This allows the other signal blocks to move out of the way.

OR Vertex. Similarly, the gadget shown in Figure 30(b) satisfies the same constraints as an OR vertex—the top signal block may slide down if and only if either the left or the right signal block first slides out.

Assembly. To use these gadgets to represent arbitrary planar AND / OR constraint graphs, we assemble them as shown in Figure 31. The wall blocks are shared between adjacent vertices, as are the signal blocks that act as graph edges. We put a grid of the gadgets inside a box. This keeps the wall blocks from moving, as required. The goal is to slide the particular signal block that corresponds to the target edge in the input constraint graph. Then, the puzzle can be solved just when the constraint-logic problem is solvable.

7.3 Two-Player Games

The two-player version of constraint logic, Two-Player Constraint Logic (2CL), is defined as follows. To create different moves for the two players, Black and White, we label each constraint graph edge as either Black or White. (This is independent of the red/blue coloration, which is simply a shorthand for edge weight.) Black (White) is allowed to reverse only Black

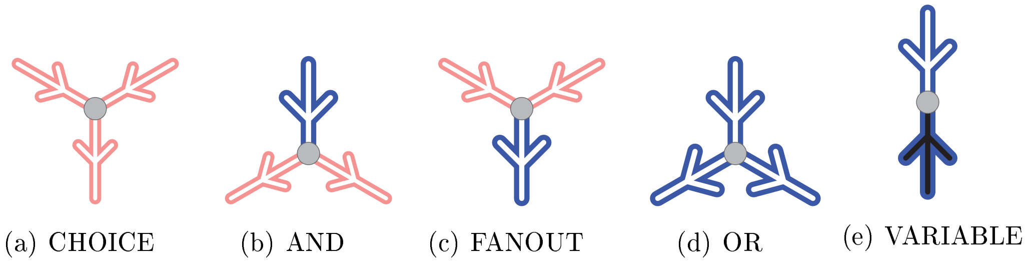

(a) CHOICE (b) AND (c) FANOUT (d) OR (e) VARIABLE

Figure 32: Basis vertices for BOUNDED 2CL.

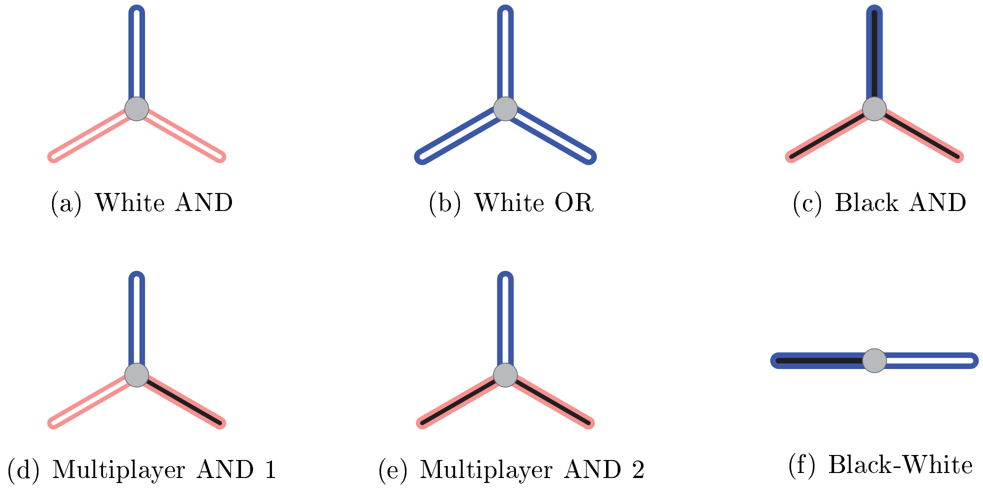
(White) edges. As before, a move must reverse exactly one edge and result in a valid configuration. Each player has a target edge he is trying to reverse.[4] (We omit the formal definitions here.)

Bounded games. The bounded case permits each edge to reverse at most once. BOUNDED 2CL is PSPACE-complete (reduction from G_{pos}(POS CNF), a variant of QBF [Sch78]). It remains PSPACE-complete when the constraint graph is a planar graph using only the vertex types shown in Figure 32. Indeed, the actual reduction showing BOUNDED 2CL PSPACE-complete is almost trivial, and the main benefit of using BOUNDED 2CL for game reductions, rather than simply using one of the many QBF variants, is that when reducing from BOUNDED 2CL one does not have to build a crossover gadget. The complexity of AMAZONS remained open for several years, despite some effort by the game-complexity community; its constraint-logic reduction showing PSPACE-completeness is straightforward [HD09].

The vertex set in Figure 32 is actually almost the same as that for BOUNDED NCL (Figure 26); the only addition is a single vertex type allowing for player interaction. Most of the gadgets can be single-player constructions.

Unbounded games. The unbounded case simply removes the restriction of edges reversing at most once. 2CL is EXPTIME-complete (reduction from G_6, one of the several Boolean formula games shown EXPTIME-complete by

[4] In combinatorial game theory, it is normal to define the loser as the first player unable to move. This definition would work perfectly well for 2CL, rather than using target edges to determine the winner; the hardness reduction would not be substantially altered. However, the given definition is more consistent with the other varieties of constraint logic: always, the goal is to reverse a given edge.



(a) White AND

(b) White OR

(c) Black AND

(d) Multiplayer AND 1

(e) Multiplayer AND 2

(f) Black-White

Figure 33: Basis vertices for 2CL.

[SC79]). 2CL remains EXPTIME-complete when the graph is a planar graph using only the vertices shown in Figure 33. In principle, this should enable much simpler reductions to actual games than the standard reductions from Boolean formula games. The existing CHESS [FL81], CHECKERS [Rob84b], and GO [Rob83] hardness results are all quite complicated; there could be simpler reductions from 2CL. However, such reductions have not yet been found. Enforcing the necessary constraints in a two-player game gadget is much more difficult than in a one-player game.

7.4 Team Games

The natural team private-information constraint logic (TPCL) assigns to each player a set of edges he can reverse, and a set of edges whose orientation he can see, in addition to the target edge he aims to reverse. There are two teams, Black and White; a team wins when a player on that team reverses his target edge. (We omit the formal definitions here.)

Bounded games. As usual for bounded games, with BOUNDED TEAM PRIVATE CONSTRAINT LOGIC we allow each edge to reverse at most once. BOUNDED TPCL is NEXPTIME-complete (reduction from the DEPENDENCY QBF problem introduced in [PR79]). It remains NEXPTIME-complete, even for planar graphs which use only AND and OR vertices, and when there is only one Black player and two White players. (Unlike other forms of constraint logic, here we don't enumerate a specific smaller set of basis vertices; note that there are several different types of AND and OR vertices, depending on controlling player, initial edge orientation, and edge visibility.)

Unbounded games. To enable a simpler reduction to an unbounded form of team constraint logic, we allow each player to reverse up to some given constant k edges on his turn, rather than just one, and leave the case of $k = 1$ as an open problem. TPCL is undecidable (shown by a series of reductions beginning with acceptance of a Turing machine on an empty input). It remains undecidable even for planar graphs which use only AND and OR vertices. As with BOUNDED TPCL, several different AND- and OR-subtypes are used in the reduction, which we do not enumerate.

The undecidability here is rather striking, given that this is a game played with a finite number of positions! Essentially, this means that the games with a bounded amount of state can simulate any unbounded Turing computation.

The ability for a player to reverse multiple edges on a turn, and the lack of a small set of basis vertices, would seem to make TPCL a challenging problem to reduce from to show other problems undecidable. However, TPCL has already been applied to show some decision problems for multi-port finite-state machines undecidable [Hie10].

8 Conclusion

We have now forged a trail from NIM to CHESS and GO. In section 2 we dealt with classical acyclic impartial games. We then presented a polynomial theory of cyclic games in section 3. In section 4 we added interactions between tokens. Next we tackled partizan games in section 5, and misère play in section 6. All of these – cycles, token interactions, misère play – are absent from NIM but very much present in CHESS and GO. This trail is still rather thin: token interaction was restricted to annihilation. Most of the other interactions lead to intractable games. Similarly for partizan games. Misère play was portrayed in the Introduction as a special case of "Termination Set", general cases of which occur in CHESS and GO. In this sense we listed misère play as a road step towards CHESS and GO.

Game intractability results, besides their intrinsic value, serve as trail guides: They indicate the boundary beyond which polynomial strategies are unlikely to exist, where we have to resort to analysis of restricted or special cases. Section 7 provides a modern convenient tool for proving intractability, though it doesn't seem to work for all cases. Intractable games, in the technical meaning of intractability, though it's only an asymptotic result – for $n \times n$ boards as n goes to infinity – are rather unlikely to have a tractable strategy for a finite actual board. In fact, we do not know of any such case. Therefore only special cases are likely to be analysable. For misère play this was done by restricting the universe of the games. For Chess and Go it was done by treating endgames.

Thus we have arrived at Chess and Go from two directions: The former has been showed to be EXPTIME-complete [FL81], which is a provable intractability, and the latter even EXPSPACE-complete, under certain game-rules of Go [Rob84a]; and there are some constructive results for their endgames. Elkies [Elk96], [Elk06] has some results about Chess endgames, and Berlekamp and his students have some spectacular results about Go endgames [Ber91], [BW94]. Go play tends to break up into almost independent subgames at the end, so the strong tool of game-sums can be unleashed to attack them. This is not quite the case for Chess, partly because some of the pieces are so strong that they dominate much of the entire board, rather than only locally.

How can we broaden this still rather thin trail? One direction could be to extend the misère play theory to more general termination sets, as exist for Chess and Go. Another is to broaden the fledgling theory of scoring games, where scores are accumulated during play. These were and are independently studied by John Milnor, Mark Ettinger, Fraser Stewart, Will Johnson and Carlos Santos. Related avenues include competitive auctions, incentives (Elwyn Berlekamp), bidding games (Sam Payne) and, more generally, connections between combinatorial games and classical games with applications to economics. In quite a different direction, the yearned for emergence of quantum and biological computing are potential brute force tools to bridge the complexity gap between polynomial and non-polynomial games.

Acknowledgment. We thank an anonymous referee whose remarks made us explain better concepts from Combinatorial Game Theory to Classical Game Theorists, and improved the exposition in various ways.

References

[ANW07] M. Albert, R. J. Nowakowski, and D. Wolfe. *Lessons in Play: An Introduction to Combinatorial Game Theory.* A K Peters, 2007.

[BCG04] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning ways for your mathematical plays.* A K Peters Ltd., Wellesley, MA, second edition, 2001-2004. Vol. 1: 2001; Vols. 2 and 3: 2003; Vol. 4: 2004.

[Ber91] Elwyn Berlekamp. Introductory overview of mathematical Go endgames. In *Combinatorial games (Columbus, OH, 1990)*, volume 43 of *Proc. Sympos. Appl. Math.*, pages 73–100. Amer. Math. Soc., Providence, RI, 1991.

[Bou02] Charles L. Bouton. Nim, a game with a complete mathematical theory. *Ann. of Math. (2)*, 3(1-4):35–39, 1901/02.

[BW94] Elwyn Berlekamp and David Wolfe. *Mathematical Go.* A K Peters Ltd., Wellesley, MA, 1994. Chilling gets the last point, With a foreword by James Davies.

[Chv73] V. Chvátal. On the computational complexity of finding a kernel. Report no. crm-300, Centre de Recherches Mathématiques, Université de Montréal, 1973.

[Con01] J. H. Conway. *On numbers and games.* A K Peters Ltd., Natick, MA, second edition, 2001.

[DH08] Erik D. Demaine and Robert A. Hearn. Constraint logic: A uniform framework for modeling computation as games. *Computational Complexity, Annual IEEE Conference on*, 0:149–162, 2008.

[DHH02] Erik D. Demaine, Robert A. Hearn, and Michael Hoffmann. Push-2-F is PSPACE-complete. In *Proceedings of the 14th Canadian Conference on Computational Geometry (CCCG 2002)*, pages 31–35, Lethbridge, Alberta, Canada, August 12–14 2002.

[Elk06] Noam D. Elkies. New directions in enumerative chess problems. *Electron. J. Combin.*, 11(2):Article 4, 14 pp. (electronic), 2004/06.

[Elk96] Noam D. Elkies. On numbers and endgames: combinatorial game theory in chess endgames. In *Games of no chance (Berkeley, CA, 1994)*, volume 29 of *Math. Sci. Res. Inst. Publ.*, pages 135–150. Cambridge Univ. Press, Cambridge, 1996.

[Fer84] Thomas S. Ferguson. Misère annihilation games. *J. Combin. Theory Ser. A*, 37(3):205–230, 1984.

[FG87] Aviezri S. Fraenkel and Elisheva Goldschmidt. PSPACE-hardness of some combinatorial games. *J. Combin. Theory Ser. A*, 46(1):21–38, 1987.

[FL81] Aviezri S. Fraenkel and David Lichtenstein. Computing a perfect strategy for $n \times n$ chess requires time exponential in n. *J. Combin. Theory Ser. A*, 31(2):199–214, 1981.

[FLN88] Aviezri S. Fraenkel, Martin Loebl, and Jaroslav Nešetřil. Epidemiography. II. Games with a dozing yet winning player. *J. Combin. Theory Ser. A*, 49(1):129–144, 1988.

[FN85] Aviezri S. Fraenkel and Jaroslav Nešetřil. Epidemiography. *Pacific J. Math.*, 118(2):369–381, 1985.

[FP75] Aviezri S. Fraenkel and Yehoshua Perl. Constructions in combinatorial games with cycles. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, pages 667–699. Colloq. Math. Soc. János Bolyai, Vol. 10. North-Holland, Amsterdam, 1975.

[Fra74] Aviezri S. Fraenkel. Combinatorial games with an annihilation rule. In *The influence of computing on mathematical research and education (Proc. Sympos. Appl. Math., Vol. 20, Univ. Montana, Missoula, Mont., 1973)*, pages 87–91. Amer. Math. Soc., Providence, R.I., 1974.

[Fra81] Aviezri S. Fraenkel. Planar kernel and Grundy with $d \leq 3$, $d_{\text{out}} \leq 2$, $d_{\text{in}} \leq 2$ are NP-complete. *Discrete Appl. Math.*, 3(4):257–262, 1981.

[Fra91] Aviezri S. Fraenkel. Complexity of games. In *Combinatorial games (Columbus, OH, 1990)*, volume 43 of *Proc. Sympos. Appl. Math.*, pages 111–153. Amer. Math. Soc., Providence, RI, 1991.

[Fra04] Aviezri S. Fraenkel. Complexity, appeal and challenges of combinatorial games. *Theoret. Comput. Sci.*, 313(3):393–415, 2004. Algorithmic combinatorial game theory.

[FT75] Aviezri S. Fraenkel and Uzi Tassa. Strategy for a class of games with dynamic ties. *Comput. Math. Appl.*, 1(no.2):237–254, 1975.

[FTY78] Aviezri S. Fraenkel, Uzi Tassa, and Yaacov Yesha. Three annihilation games. *Math. Mag.*, 51(1):13–17, 1978.

[FY76] Aviezri S. Fraenkel and Yaacov Yesha. Theory of annihilation games. *Bull. Amer. Math. Soc.*, 82(5):775–777, 1976.

[FY79] Aviezri S. Fraenkel and Yaacov Yesha. Complexity of problems in games, graphs and algebraic equations. *Discrete Appl. Math.*, 1(1-2):15–30, 1979.

[FY82] Aviezri S. Fraenkel and Yaacov Yesha. Theory of annihilation games. I. *J. Combin. Theory Ser. B*, 33(1):60–86, 1982.

[FY86] Aviezri S. Fraenkel and Yaacov Yesha. The generalized Sprague-Grundy function and its invariance under certain mappings. *J. Combin. Theory Ser. A*, 43(2):165–177, 1986.

[Gar64] Martin Gardner. The hypnotic fascination of sliding-block puzzles. *Scientific American*, 210:122–130, 1964.

[GR95] Arthur S. Goldstein and Edward M. Reingold. The complexity of pursuit on a graph. *Theoret. Comput. Sci.*, 143(1):93–112, 1995.

[Gru39] P. M. Grundy. Mathematics and games. *Eureka*, 2, 1939. reprinted in *Eureka* **27** (1964), 9–11.

[HD05] Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343(1-2):72–96, October 2005. Special issue “Game Theory Meets Theoretical Computer Science”.

[HD09] Robert A. Hearn and Erik D. Demaine. *Games, puzzles and computation*. A K Peters, 2009.

[Hea04] Robert A. Hearn. The complexity of sliding block puzzles and plank puzzles. In *Tribute to a Mathemagician*, pages 173–183. A K Peters, 2004.

[Hea06a] Robert Hearn. TipOver is NP-complete. *Mathematical Intelligencer*, 28(3):10–14, 2006.

[Hea06b] Robert A. Hearn. *Games, Puzzles, and Computation*. PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, May 2006. `http://www.swiss.ai.mit.edu/~bob/hearn-thesis-final.pdf`.

[Hea09] Robert A. Hearn. Amazons, Konane, and Cross Purposes are PSPACE-complete. In Michael H. Albert and Richard J. Nowakowski, editors, *Games of No Chance 3*, number 56 in MSRI Publications, Cambridge, 2009. Cambridge University Press.

[Hie10] R M Hierons. Reaching and distinguishing states of distributed systems. *SIAM Journal on Computing*, 39(8):3480–3500, 2010.

[Kal28] L. Kalmár. Zur theorie der abstrakten spiele. *Acta Sci. Math. Univ. Szeged*, 4:65–85, 1928.

[Max07] Oriel Maxime. Wriggler puzzles are PSPACE-complete. Manuscript, August 2007.

[Mor81] F. L. Morris. Playing disjunctive sums is polynomial space complete. *Internat. J. Game Theory*, 10:195–205, 1981.

[Pla05] Thane E. Plambeck. Taming the wild in impartial combinatorial games. *INTEGERS: The Electr. J. Combin. Number Thy.*, 5(1):#G05, 2005.

[PM84] A. Pultr and F. L. Morris. Prohibiting repetitions makes playing games substantially harder. *Internat. J. Game Theory*, 13(1):27–40, 1984.

[PR79] Gary L. Peterson and John H. Reif. Multiple-person alternation. In *FOCS*, pages 348–363. IEEE, 1979.

[PS08] Thane E. Plambeck and Aaron N. Siegel. Misère quotients for impartial games. *J. Combin. Thy. Ser. A*, 115(4):593–622, May 2008.

[Rab57] Michael O. Rabin. Effective computability of winning strategies. In *Contributions to the theory of games, vol. 3*, Annals of Mathematics Studies, no. 39, pages 147–157. Princeton University Press, Princeton, N. J., 1957.

[Rei81] Stefan Reisch. Hex ist PSPACE-vollständig. *Acta Inform.*, 15(2):167–191, 1981.

[Rob83] J. M. Robson. The complexity of Go. In *Proceedings of the IFIP 9th World Computer Congress on Information Processing*, pages 413–417, 1983.

[Rob84a] J. M. Robson. Combinatorial games with exponential space complete decision problems. In *Mathematical foundations of computer science, 1984 (Prague, 1984)*, volume 176 of *Lecture Notes in Comput. Sci.*, pages 498–506. Springer, Berlin, 1984.

[Rob84b] J. M. Robson. N by N Checkers is EXPTIME complete. *SIAM Journal on Computing*, 13(2):252–267, May 1984.

[SC79] Larry J. Stockmeyer and Ashok K. Chandra. Provably difficult combinatorial games. *SIAM Journal on Computing*, 8(2):151–174, 1979.

[Sch78] Thomas J. Schaefer. On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16:185–225, 1978.

[Sie13] Aaron N. Siegel. *Combinatorial Game Theory.* Graduate Studies in Mathematics, Vol. 146. Amer. Math. Soc., 2013.

[Smi66] Cedric A. B. Smith. Graphs and composite games. *J. Combinatorial Theory*, 1:51–81, 1966.

[Spr36] R. Sprague. Über mathematische Kampfspiele. *Tôhoku Math. J.*, 41:438–444, 1935–36.

[vL76] J. van Leeuwen. Having a Grundy-numbering is NP-complete. Report no. 207, Computer Science Dept., Pennsylvania State Univ., University Park, PA, 1976.

[Wes96] Julian West. Championship-level play of Domineering. In Richard J. Nowakowski, editor, *Games of No Chance*, number 29 in MSRI Publications, pages 85–91. Cambridge University Press, Cambridge, 1996.

[Yed85] L. J. Yedwab. On playing well in a sum of games. Master's thesis, MIT, 1985. MIT/LCS/TR-348.