

The Avoider-Enforcer Game in Hypergraphs of Rank 3

Florian Galliot^a Valentin Gledel^b Aline Parreau^c

Submitted: Mar 27, 2025; Accepted: Feb 20, 2026; Published: May 22, 2026

© The authors. Released under the CC BY-ND license (International 4.0).

Abstract

In the Avoider-Enforcer convention of positional games, two players—Avoider and Enforcer—take turns selecting vertices from a hypergraph H . Enforcer wins if, by the time all vertices of H have been selected, Avoider has completely filled an edge of H with her vertices; otherwise, Avoider wins. In this paper, we first give some general results, in particular regarding the outcome of the game and disjoint unions of hypergraphs. We then determine which player has a winning strategy for all hypergraphs of rank 2, and for linear hypergraphs of rank 3 when Avoider plays the last move. The structural characterisations we obtain yield polynomial-time algorithms.

Mathematics Subject Classifications: 05C65, 68W99, 91A24, 91A43

1 Introduction

Positional games have been introduced by Hales and Jewett [15] and popularised by Erdős and Selfridge [8]. The game board is a hypergraph H with vertex set $V(H)$ and edge set $E(H)$. Two players take turns picking a previously unpicked vertex of the hypergraph. There are four main *conventions* for determining the winner. In *strong games*, both players have the same objective: in the *Maker-Maker* convention, whoever fills an edge (i.e. picks all its vertices) wins the game, whereas in the *Avoider-Avoider* convention, whoever first fills an edge loses the game. In *weak games*, the players' objectives are complementary: in the *Maker-Breaker* convention, Maker aims at filling an edge while Breaker aims at preventing her from doing so, whereas in the *Avoider-Enforcer* convention, Avoider aims at avoiding filling an edge while Enforcer aims at forcing her to do so. The Maker-Maker

^aAix-Marseille Université, CNRS, I2M, UMR 7373, 13453 Marseille, France
(florian.galliot@univ-amu.fr).

^bUniversité Savoie Mont Blanc, CNRS UMR5127, LAMA, Chambéry, F-73000, France
(valentin.gledel@univ-smb.fr).

^cUniv Lyon, CNRS, INSA Lyon, UCBL, Centrale Lyon, Univ Lyon 2, LIRIS, UMR5205, F-69622
Villeurbanne, France (aline.parreau@univ-lyon1.fr).

and Maker-Breaker conventions characterise *achievement games*, whereas the Avoider-Avoider and Avoider-Enforcer conventions characterise *avoidance games*. Since all these conventions define finite two-player games with perfect information, either there exists a winning strategy for one of the players, or optimal play leads to a draw (this can only happen in strong games).

The first convention to be introduced was the Maker-Maker convention [15], which is the most natural one since it models the famous game of Tic-Tac-Toe. It can be noted, using a strategy-stealing argument [15], that there are no situations where the players would rather pass their turn, and therefore either the first player has a winning strategy or optimal play leads to a draw. Apart from that, few general results are known, as this convention is hard to tackle. There is no monotonicity (adding edges may harm the first player), and it is not known how to compute a winning strategy for the first player on a disjoint union of two games given a winning strategy on each of them. These difficulties are mainly due to the fact that the players pursue two goals at once: filling an edge, and preventing the opponent to fill one first.

This is the reason why the Maker-Breaker convention was introduced by Chvátal and Erdős [6]. Constructive results are generally easier to achieve compared to the Maker-Maker convention: for instance, it is very easy to design optimal strategies on a disjoint union of two games from optimal strategies on each separate game. These games are monotonous, since adding an edge can clearly only benefit Maker. Some other general results are known, such as the “Erdős-Selfridge criterion” or “pairing strategies” which give sufficient conditions for Breaker to have a winning strategy. The Maker-Breaker convention is actually by far the most studied (see for example the books [2, 17]). Maker-Breaker games on hypergraphs of small rank have been much investigated. Hypergraphs of rank at most 2 are trivially solved. Hypergraphs of rank 3 were first studied in the linear case (i.e. when any two edges share at most one vertex) by Kutz, who obtained a polynomial-time algorithm to decide which player has a winning strategy [20]. This result has been recently extended to all hypergraphs of rank 3 by Galliot, Gravier and Sivignon [11]. On the other hand, it has been known since 1978 that deciding the winner of a Maker-Breaker game is PSPACE-complete even if the edges have size 11 [25]. This result has since then been improved to edges of size 6 [24] and very recently to edges of size 5 [19].

The other two conventions, which revolve around avoiding filling an edge, have received less attention but are now being increasingly studied [3, 12, 22, 26]. The way in which avoidance games are played is quite different. Indeed, in these conventions, players would often rather pass than pick a vertex. As a consequence, there are cases where the second player has a winning strategy. All in all, methods used for achievement games mostly do not transfer to avoidance games. As an illustration, it took until 2023 to prove that deciding which player has a winning strategy is PSPACE-complete for the Avoider-Enforcer convention [12], over 40 years later than for the Maker-Breaker convention. Among the two avoidance conventions, Avoider-Avoider games, being strong games, present a similar predicament as Maker-Maker games. Deciding which player (if any) has a winning strategy on a hypergraph of rank 2 is already PSPACE-complete in this convention [9].

This paper focuses on the Avoider-Enforcer convention, which owes its first general formulation to Lu [21]. Since this is a weak convention i.e. the two players have complementary goals, one can hope that these games satisfy some convenient general properties, similar to those of Maker-Breaker games (despite the differences in the way these games are played). For example, it is already known that the Erdős-Selfridge criterion can be adapted to the Avoider-Enforcer convention [18]. Furthermore, pairing strategies have been used to prove the PSPACE-completeness of Avoider-Enforcer games [12, Lemma 6]. That last paper illustrates how complexity proofs for Maker-Breaker games can often be adapted to Avoider-Enforcer games. Beside these works, the Avoider-Enforcer convention seems to have been studied mostly for specific games [1, 3, 7, 13] and not for general hypergraphs, except those with pairwise-disjoint edges which are entirely solved [10].

Our contribution. The aim of this paper is to give general properties of Avoider-Enforcer games and study this convention in hypergraphs of small rank. An important property that we establish is that, in contrast with the Maker-Breaker convention, the relevant distinction to make when studying Avoider-Enforcer games is based on who plays the last move of the game (when all vertices are picked) rather than who plays the first move. Indeed, any player that has a winning strategy playing last also has a winning strategy playing second-to-last. This result leads us to define the *outcome* of a hypergraph as the identity of the player that has a winning strategy in each case (Avoider playing last or Enforcer playing last). We then show that the outcome of a disjoint union of games is uniquely determined by the outcomes of its components. We then consider hypergraphs of small rank. For hypergraphs of rank 2, we give a structural characterisation of the outcome. As for hypergraphs of rank 3, we initiate their study by investigating the linear case, which was the first case to be solved in the Maker-Breaker convention [20]. We completely solve the subcase where Avoider plays last, by providing a structural characterisation of hypergraphs on which Avoider has a winning strategy playing last. All our structural characterisations can straightforwardly be recognised in polynomial time. Definitions and general results are given in Section 2. Section 3 is dedicated to hypergraphs of rank 2, and Section 4 to linear hypergraphs of rank 3 when Avoider is the last player.

2 General results on the Avoider-Enforcer game

2.1 Definitions and notations

Hypergraphs. Let H be a hypergraph. We denote its vertex set by $V(H)$ and its edge set by $E(H) \subseteq 2^{V(H)}$. We say that H is *even* (resp. *odd*) if it has an even (resp. odd) number of vertices. A *neighbour* of a vertex x in H is a vertex $y \neq x$ such that some edge of H contains both x and y . The *degree* of a vertex in H is the number of edges it belongs to. A vertex is called *isolated* if it has degree 0. A *leaf-edge* of H is an edge with at most one vertex of degree greater than 1. A *subhypergraph* of H is a hypergraph H' such that $V(H') \subseteq V(H)$ and $E(H') \subseteq E(H)$. A *transversal* of H is a set of vertices that intersects

every edge of H .

Let $k \in \mathbb{N}$. An edge is a k -edge if it contains exactly k vertices. A hypergraph has *rank* k if all its edges have at most k vertices, and it is k -uniform if every edge contains exactly k vertices.

A hypergraph is *linear* if any two distinct edges share at most one vertex. A *linear tree* (following the definition from [14]) is recursively defined as a hypergraph that either is an isolated vertex or can be obtained from a linear tree T by adding an edge that shares exactly one vertex with T .

A *walk* in a hypergraph H is a sequence (e_1, \dots, e_ℓ) of edges of H such that $e_i \cap e_{i+1} \neq \emptyset$ for all $1 \leq i \leq \ell - 1$. We say a walk (e_1, \dots, e_ℓ) is *simple* if $e_i \cap e_j = \emptyset$ for all $1 \leq i, j \leq \ell$ such that $|i - j| \geq 2$. We say a hypergraph H is *connected* if, for all distinct $a, b \in V(H)$, there exists a walk (e_1, \dots, e_ℓ) in H such that $a \in e_1$ and $b \in e_\ell$. The *connected components* of H are the maximal connected subhypergraphs of H . If H is connected, then a vertex $y \in V(H)$ is called a *cut vertex* of H if the hypergraph induced by $V(H) \setminus \{y\}$ is disconnected or a *non-cut vertex* otherwise.

The *union* of two hypergraphs H_1 and H_2 , denoted by $H_1 \cup H_2$, is the hypergraph with vertex set $V(H_1) \cup V(H_2)$ and edge set $E(H_1) \cup E(H_2)$. This is a *disjoint union* if $V(H_1)$ and $V(H_2)$ are disjoint.

For further definitions and notations on hypergraphs, we refer the reader to the reference books [4] and [5].

Avoider-Enforcer Games. Let H be a hypergraph. An Avoider-Enforcer game on H is played by two players, Avoider and Enforcer. In turn, the players pick a vertex of H that has not been picked yet. The game ends when all the vertices have been picked. Enforcer has won if Avoider has *filled* an edge of H (i.e. Avoider has picked all the vertices of some edge), otherwise Avoider has won. By convention, if H contains the empty edge, then Enforcer wins. A *round* is a sequence of two moves, one for the first player and one for the second player. We say that *Avoider wins on H as first* (resp. *second*, resp. *last*, resp. *second-to-last*) *player* if Avoider has a winning strategy for the Avoider-Enforcer game played on H when Avoider is the first (resp. second, resp. last, resp. second-to-last) player. We introduce analogous definitions for Enforcer. Being the last player means playing the last move of the game: in other words, the last player is the first player if H is odd or the second player if H is even.

It can be noticed that the opposite of Avoider filling an edge is Enforcer picking at least one vertex in each edge i.e. filling a transversal. Therefore, Avoider is an “enforcer of transversals”, while Enforcer is an “avoider of transversals”. This role-reversal argument yields the following observation.

Observation 1. *Let H be a hypergraph, and denote by H^T the hypergraph with same vertex set as H whose edges are the transversals of H . Avoider wins on H as last player (resp. second-to-last player) if and only if Enforcer wins on H^T as last player (resp. second-to-last player).*

A convenient property of weak positional games is that there is never a need to consider hypergraphs where vertices have already been picked, as this case reduces to the case where

the game has not started yet. Indeed, let us introduce the following notations, where H is any hypergraph:

- For all $x \in V(H)$, we denote by H^{+x} the hypergraph defined by $V(H^{+x}) = V(H) \setminus \{x\}$ and $E(H^{+x}) = \{e \setminus \{x\} \mid e \in E(H)\}$.
- For all $y \in V(H)$, we denote by H^{-y} the hypergraph defined by $V(H^{-y}) = V(H) \setminus \{y\}$ and $E(H^{-y}) = \{e \in E(H) \mid y \notin e\}$.

The operators $^{+x}$ and $^{-y}$ can be used to update the hypergraph during the game, so that the edges of the updated hypergraph represent what remains of the initial edges. Whenever Enforcer picks a vertex y , any edge containing y is “killed” since it will never be filled by Avoider. Using these notations, after one round of play of the Avoider-Enforcer game played on H with Avoider picking x and Enforcer picking y , it is as if a new game starts on the updated hypergraph H^{+x-y} . Therefore, all results on hypergraph classes which are stable under the operators $^{+x}$ and $^{-y}$ also hold for hypergraphs on which some moves have already been made, since we can always reduce to the case of a “fresh” hypergraph using the above updating method. The winner of the game can be defined by induction. For example, when Enforcer is the first player: if $|V(H)| \leq 1$ then Enforcer wins on H if and only if $\emptyset \in E(H)$, otherwise Enforcer wins on H if and only if there exists $y \in V(H)$ (Enforcer’s pick) such that for all $x \in V(H) \setminus \{y\}$ (Avoider’s answer) Enforcer wins on H^{-y+x} .

2.2 Outcome and importance of the last move

In Maker-Breaker games, it is well known (see for example [17]) that any player who has a winning strategy as second player also has a winning strategy as first player. This leads to only three possible outcomes: Maker wins as first or second player, Breaker wins as first or second player, or the first player wins. On the contrary, in Avoider-Enforcer games, winning as second player is not a guarantee that one can win as first player, nor the other way around. As illustrated in Figure 1, all four outcomes exist when it comes to who plays first.

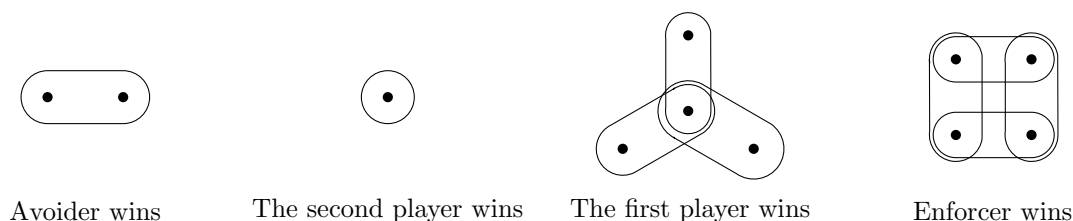


Figure 1: Examples of hypergraphs for each possible outcome.

However, a result similar to that of Maker-Breaker games can still be obtained, by considering the last move rather than the first move:

Theorem 2. *Let H be a hypergraph. If Avoider (resp. Enforcer) wins on H as last player, then Avoider (resp. Enforcer) also wins on H as second-to-last player.*

Proof. By Observation 1, it suffices to consider Enforcer’s side of things. Suppose Enforcer wins on H as last player. We will use strategy stealing to show that Enforcer can basically apply the same strategy when he is the second-to-last player. The proof will distinguish between two cases depending on the parity of H .

First assume that H is even, meaning Enforcer has a winning strategy Σ as second (and last) player. When playing first, Enforcer can pick an arbitrary vertex u , “forget” this move, and then act as second player applying the strategy Σ . In other words, Enforcer imagines a “fictitious game” where he is the second player and u has not been played, and where he applies Σ . Enforcer copies each Avoider move into the fictitious game, and then copies the answer given by Σ in the real game. The only exception is whenever Σ requires that Enforcer picks the forgotten vertex, in which case he picks another arbitrary vertex instead in the real game, which becomes the new forgotten vertex. The fact that Σ is a winning strategy for Enforcer as second player ensures that Avoider fills an edge at some point in the fictitious game. Since the set of vertices picked by Avoider is the same in both games, Avoider also fills an edge in the real game, which concludes.

Assume now that H is odd, meaning Enforcer has a winning strategy Σ as first (and last) player. When playing second, Enforcer will “forget” the vertex u that Avoider picked as her first move, and then act as first player applying the strategy Σ . Same as before, whenever Σ requires that Enforcer picks the forgotten vertex, he picks another arbitrary vertex instead in the real game, which becomes the new forgotten vertex. The fact that Σ is a winning strategy for Enforcer as first player ensures that Avoider fills an edge at some point in the fictitious game. Since the set of vertices picked by Avoider in the real game is a superset of what it is in the fictitious game (it is the same plus the vertex u), Avoider also fills an edge in the real game, which concludes. \square

Theorem 2 invites us to consider outcomes in terms of the player playing last, rather than the player playing first as is done for achievement games. In the following, we will denote by $o(H)$ the outcome of the game played on a hypergraph H . Since any player who has a winning strategy as last player also has a winning strategy as second-to-last player, there are only three possible values for $o(H)$: either Avoider wins in all cases (denoted by $o(H) = \mathcal{A}$), or Enforcer wins in all cases (denoted by $o(H) = \mathcal{E}$), or the player playing second-to-last wins (denoted by $o(H) = \mathcal{SL}$). Note that one can recover the traditional outcomes by considering the parity of H : if $o(H) = \mathcal{SL}$, then the first player has a winning strategy if H is even, and the second player has a winning strategy if H is odd.

In the Maker-Breaker convention, it is possible to reduce to the case where, say, Maker starts. Indeed, if Breaker starts then it suffices to consider all possibilities for Breaker’s first move y , after which a new game begins on the updated hypergraph H^{-y} with Maker as first player. Similarly, in the Avoider-Enforcer convention, where the key information is who plays last rather than who plays first, we would like to be able to choose the last player and always reduce to that case. For this, an idea would be to consider all possibilities, not for the first move (chosen by the first player), but for the last move (forced by the second-to-last player). Indeed, the second-to-last player has the luxury of avoiding any particular vertex x of their choice: by never picking x , they will force

the last player to eventually pick x . Therefore, an intuition for a reduction would be the following: if x is the “worst” vertex to pick, then the second-to-last player could just as well give x to their opponent before the game starts, and then play as last player on what remains. One direction of this reasoning does hold:

Proposition 3. *Let H be a hypergraph.*

- *If there exists $y \in V(H)$ such that Avoider wins on H^{-y} as last player, then Avoider wins on H as second-to-last player.*
- *If there exists $x \in V(H)$ such that Enforcer wins on H^{+x} as last player, then Enforcer wins on H as second-to-last player.*

Proof. We only prove the first assertion, as the second is proved analogously. We show the contrapositive. Suppose that Enforcer has a winning strategy Σ on H as last player. Let $y \in V(H)$: we want to show that Enforcer wins on H^{-y} as second-to-last player. Note that the first player is the same in both games, since the last player and the parity are different. Initially, Enforcer follows the strategy Σ in the fictitious game played on H , and copies the moves in the real game played on H^{-y} . In the fictitious game, being the last player, Enforcer will inevitably pick y at some point. At that moment, the updated hypergraph H' is the same in both games. By optimality of Σ , Enforcer wins on H' as last player, therefore Enforcer also wins on H' as second-to-last player by Theorem 2. In conclusion, Enforcer wins the real game as a whole. \square

Unfortunately, neither converse holds in Proposition 3 (see Figure 2 for counterexamples). This is not surprising: the “worst vertex” might change depending on what happens during the game, so it might not be a good idea for the second-to-last player to identify one permanently before the game begins.



Figure 2: Counterexamples to the converse of the first (left) and the second (right) assertions of Proposition 3. As in all figures from now, we represent edges of size 2 with a straight line and edges of size 3 with a “claw” shape.

All in all, we do not know of an easy way to reduce to the case where Avoider or Enforcer is the last player, so we will have to consider both possibilities in our studies. However, we do have a simplification when a 1-edge is involved.

Proposition 4. *Let H be a hypergraph containing a 1-edge $\{y\}$. Then:*

- *Avoider loses on H as last player.*
- *Avoider wins on H as second-to-last player if and only if Avoider wins on H^{-y} as last player.*

Proof. The first assertion is obvious: since Enforcer is the second-to-last player, he can force Avoider to pick y eventually (this can also be seen as an application of Proposition 3, indeed Enforcer trivially wins on H^{+y} which contains the empty edge). Let us now address the second assertion. The “if” direction is simply Proposition 3, so assume Avoider has a winning strategy Σ on H as second-to-last player. We want to show that Avoider wins on H^{-y} as last player. Note that the first player is the same in both games, since the last player and the parity are different. Therefore, and since Σ obviously never has Avoider picking y , Avoider can simply apply Σ playing on H^{-y} as last player. By optimality of Σ , Avoider never fills an edge of H , so she never fills an edge of the subhypergraph H^{-y} either. \square

2.3 Outcome of a disjoint union

In Maker-Breaker games, it is easy to deduce the outcome of a disjoint union from the outcome of its components. For instance, if Maker wins as first or second player on H_1 , then the same holds on any disjoint union $H_1 \cup H_2$, since Maker may disregard H_2 and play all her moves inside H_1 . In this example, it is key that playing twice in a row in some component cannot harm Maker. Such arguments do not hold in Avoider-Enforcer games, which makes the study of disjoint unions less straightforward. However, we now show that the Avoider-Enforcer convention also has the convenient property that the outcome of a disjoint union is uniquely determined by the outcome of its components. We start by proving a useful lemma.

Lemma 5. *Let H be a hypergraph.*

- *If a player (Avoider or Enforcer) wins on H as first player, then that same player would have a winning strategy if their opponent was allowed to start by making any even number of moves.*
- *If a player (Avoider or Enforcer) wins on H as second player, then that same player would have a winning strategy if their opponent was allowed to start by making any odd number of moves.*

Proof. By Observation 1, it suffices to address Enforcer’s side of things.

- Suppose that Enforcer has a winning strategy Σ on H as first player. Now consider the game played on H where Avoider starts by playing on a subset X of $V(H)$ with $|X|$ even. Enforcer first arbitrarily partitions X into pairs $\{u_i, v_i\}$. Then, he “forgets” that these vertices have been played, and he applies the strategy Σ . If, at some point during the game, Σ requires Enforcer to play inside X (say, some u_i), then he imagines that he picks u_i and that Avoider answers by picking v_i . Since the set of vertices actually picked by Avoider is a superset of the imaginary one, which contains an edge since Σ is a winning strategy, it also contains an edge.
- Suppose that Enforcer wins on H as second player: this means that, whatever Avoider’s first move x , Enforcer wins on H^{+x} as first player. Now, if Avoider makes an even number of moves additionally to x (i.e. an odd total number of moves at the start including x), Enforcer still wins according to the even case which we have already addressed. \square

Theorem 6. *Let H_1 and H_2 be two hypergraphs with disjoint vertex sets. The outcome of $H_1 \cup H_2$ is obtained from the outcomes of H_1 and H_2 according to the following table:*

	H_1	\mathcal{A}	\mathcal{SL}	\mathcal{E}
H_2	\mathcal{A}	\mathcal{A}	\mathcal{SL}	\mathcal{E}
\mathcal{SL}	\mathcal{SL}	\mathcal{E}	\mathcal{E}	\mathcal{E}
\mathcal{E}	\mathcal{E}	\mathcal{E}	\mathcal{E}	\mathcal{E}

Proof. The proof will distinguish cases depending on the outcomes of H_1 and H_2 as well as their parity.

1. $o(H_1) = \mathcal{E}$ (symmetric to the case $o(H_2) = \mathcal{E}$).

Suppose that Enforcer is the last player on $H_1 \cup H_2$, and consider the following strategy. Enforcer plays arbitrary moves, all of them inside H_2 . We can assume that, by the point all vertices of H_2 have been picked, Avoider has not yet filled an edge. Note that, since $o(H_1) = \mathcal{E}$, Enforcer has a winning strategy on H_1 as both first or second player. Therefore, according to Lemma 5, he has a winning strategy on what remains of H_1 , no matter the parity of the number of moves that Avoider has played in it. Enforcer can now follow this strategy to win the game.

2. $o(H_1) = o(H_2) = \mathcal{SL}$.

Suppose that Enforcer is the last player. We will give different strategies for Enforcer depending on the possible parities of H_1 and H_2 :

- If H_1 and H_2 are both odd, then $H_1 \cup H_2$ is even, so Enforcer is the second player. Moreover, Enforcer has winning strategies Σ_1 and Σ_2 on H_1 and H_2 respectively as second player. Therefore, a winning strategy for Enforcer in $H_1 \cup H_2$ as second player is to play each move in the same hypergraph Avoider has just played in, according to Σ_1 or Σ_2 respectively. Avoider will fill an edge in one of the two hypergraphs: at the latest, this happens as she picks the last remaining vertex of H_1 or H_2 .
- If H_1 is even and H_2 is odd, then $H_1 \cup H_2$ is odd, so Enforcer is the first player. Moreover, Enforcer has winning strategies Σ_1 on H_1 as first player and Σ_2 on H_2 as second player. Therefore, if Enforcer plays his first move in H_1 according to Σ_1 and then plays each subsequent move in the same hypergraph Avoider has just played in according to Σ_1 or Σ_2 respectively, Enforcer will win similarly to the previous case.
The case where H_1 is odd and H_2 is even is symmetric.
- If H_1 and H_2 are both even, then $H_1 \cup H_2$ is even, so Enforcer is the second player. Moreover, Enforcer has winning strategies Σ_1 and Σ_2 on H_1 and H_2 respectively as first player. Consider the following strategy for Enforcer. Without loss of generality, assume that Avoider starts in H_1 . Enforcer then

plays his first move in H_2 , according to Σ_2 . We know from Lemma 5 that, if Avoider was the next player to pick a vertex in H_1 , then Enforcer would have a winning strategy on the remainder of H_1 . Therefore, Enforcer now plays each subsequent move in the same hypergraph Avoider has just played in, according to that strategy or to Σ_2 respectively. Avoider will fill an edge in one of the two hypergraphs: at the latest, this happens as she picks the last remaining vertex of H_1 or H_2 .

In all cases, Enforcer wins on $H_1 \cup H_2$ as last player, $o(H_1 \cup H_2) = \mathcal{E}$.

3. $o(H_1) = o(H_2) = \mathcal{A}$.

Suppose that Avoider is the last player. We will give different strategies for Avoider depending on the possible parities of H_1 and H_2 :

- If H_1 and H_2 are both even, then $H_1 \cup H_2$ is even, so Avoider is the second player. Moreover, Avoider has winning strategies Σ_1 and Σ_2 on H_1 and H_2 respectively as second player. Therefore, when playing on $H_1 \cup H_2$, Avoider can simply play each move in the same hypergraph Enforcer has just played in, according to Σ_1 or Σ_2 respectively. Since both hypergraphs are even, Avoider will play the last move in both of them, which ensures that no one plays twice in a row in H_1 or H_2 . Therefore, for each $i \in \{1, 2\}$, the game restricted to H_i is a normal game where Avoider manages to avoid filling an edge.
- If H_1 is even and H_2 is odd, then $H_1 \cup H_2$ is odd, so Avoider is the first player. Moreover, Avoider has winning strategies Σ_1 on H_1 as second player and Σ_2 on H_2 as first player. Therefore, when playing on $H_1 \cup H_2$, Avoider can simply play her first move in H_2 according to Σ_2 , and then play each subsequent move in the same hypergraph Enforcer has just played in, according to Σ_1 or Σ_2 respectively. Similarly to the previous case, she will avoid filling an edge in both hypergraphs and thus win the game.
The case where H_1 is odd and H_2 is even is symmetric.
- If H_1 and H_2 are both odd, then $H_1 \cup H_2$ is even, so Avoider is the second player. Moreover, Avoider has winning strategies Σ_1 and Σ_2 on H_1 and H_2 respectively as first player. Without loss of generality, assume Enforcer starts in H_1 . Avoider plays her first move in H_2 according to Σ_2 , and then plays each subsequent move in the same hypergraph Enforcer has just played in, using Σ_2 in H_2 and Lemma 5 in H_1 . Indeed, since the number of remaining vertices after the first round is even in both hypergraphs, Avoider will play the last move in both, and the next move made in H_1 will be by Enforcer. After two Enforcer moves have been made in H_1 , Avoider still has a winning strategy on what remains of H_1 by Lemma 5. In conclusion, except for the first two moves made in H_1 which that are taken care of by Lemma 5, the order of moves in both hypergraphs will be normal and Avoider will avoid filling an edge in either.

In all cases, Avoider wins on $H_1 \cup H_2$ as last player, $o(H_1 \cup H_2) = \mathcal{A}$.

4. $o(H_1) = \mathcal{S}\mathcal{L}$ and $o(H_2) = \mathcal{A}$.

First of all, we exhibit a winning strategy for Enforcer on $H_1 \cup H_2$ as second-to-last player. Enforcer plays arbitrary moves, all of them inside H_2 . We can assume that, by the point all vertices of H_2 have been picked, Avoider has not yet filled an edge. Now, consider Enforcer's first turn after all vertices of H_2 are played: he has not played a single move in H_1 yet, and the number of remaining vertices in H_1 is even since Enforcer is the second-to-last player. If H_1 is even, then Enforcer has a winning strategy on H_1 as first player and Avoider has played an even number of moves in H_1 , so Enforcer wins according to Lemma 5. If H_1 is odd, then Enforcer has a winning strategy on H_1 as second player and Avoider has played an odd number of moves in H_1 , so Enforcer also wins according to Lemma 5.

Finally, it remains to show that Avoider wins on $H_1 \cup H_2$ as second-to-last player. To do so, we will distinguish between the possible parities of H_1 and H_2 :

- If H_1 is odd and H_2 is even, then $H_1 \cup H_2$ is odd, so Avoider is the second player. Moreover, Avoider has winning strategies Σ_1 and Σ_2 on H_1 and H_2 respectively as second player. When playing on $H_1 \cup H_2$, Avoider can, as long as possible, play each move in the same hypergraph Enforcer has just played in, according to Σ_1 or Σ_2 respectively. First suppose that all vertices in H_2 are picked before all vertices in H_1 are: in this case, parity ensures that Avoider can perform this strategy until the end, thus avoiding filling an edge in either hypergraph and winning the game. Now, suppose that Enforcer picks the last remaining vertex in H_1 before all vertices in H_2 have been picked. In particular, Avoider has successfully avoided filling an edge so far, and it is now her turn. The remainder of H_2 at this point is even, and Avoider has a winning strategy on it as second player, so she also has a winning strategy on it as first player by Theorem 2. Therefore, Avoider can follow that strategy to win the game.
- If H_1 and H_2 are both even, then $H_1 \cup H_2$ is even, so Avoider is the first player. Since $o(H_1) = \mathcal{S}\mathcal{L}$ and H_1 is even, there exists $x \in V(H_1)$ such that Avoider has a winning strategy as second player on H_1^{+x} . Note that Avoider does not have a winning strategy as first player on H_1^{+x} by Lemma 5, so $o(H_1^{+x}) = \mathcal{S}\mathcal{L}$. Since H_1^{+x} is odd and H_2 is even, the odd/even case which we have already addressed yields $o(H_1^{+x} \cup H_2) = \mathcal{S}\mathcal{L}$, so Avoider has a winning strategy on $H_1 \cup H_2$ as second-to-last player where she picks x as her first move.
- If H_1 and H_2 are both odd, then $H_1 \cup H_2$ is even, so Avoider is the first player. Since $o(H_2) = \mathcal{A}$ and H_2 is odd, there exists $x \in V(H_2)$ such that $o(H_2^{+x}) = \mathcal{A}$. Since H_1 is odd and H_2^{+x} is even, the odd/even case which we have already addressed yields $o(H_1 \cup H_2^{+x}) = \mathcal{S}\mathcal{L}$, so Avoider has a winning strategy on $H_1 \cup H_2$ as second-to-last player where she picks x as her first move.
- If H_1 is even and H_2 is odd, then $H_1 \cup H_2$ is odd, so Avoider is the second player. As his first move, Enforcer picks $y \in V(H_i)$ for some $i \in \{1, 2\}$. Note that,

since Avoider has a winning strategy on H_i as first player, she has a winning strategy on H_i^{-y} as second player by Lemma 5, so $o(H_1^{-y}) = \mathcal{SL}$ if $i = 1$ or $o(H_2^{-y}) = \mathcal{A}$ if $i = 2$. Now, since Avoider has a winning strategy on H_{3-i} as first player, there exists $x \in V(H_{3-i})$ such that Avoider has a winning strategy on H_{3-i}^{+x} as second player. This means $o(H_1^{+x}) = \mathcal{SL}$ if $i = 2$ or $o(H_2^{+x}) = \mathcal{A}$ if $i = 1$. All in all, the odd/even case which we have already addressed yields $o(H_i^{-y} \cup H_{3-i}^{+x}) = \mathcal{SL}$, so Avoider has a winning strategy after picking x . \square

Remark 7. When $o(H_1) = o(H_2) = \mathcal{E}$, not only does Enforcer win on $H_1 \cup H_2$ as last player, but he can actually force Avoider to fill two edges, one in each hypergraph. Indeed, we have $o(H_1^T) = o(H_2^T) = \mathcal{A}$ by Observation 1, hence $o(H_1^T \cup H_2^T) = \mathcal{A}$ by Theorem 6, so Enforcer as last player can avoid filling a transversal in H_1 and H_2 at once, meaning Avoider fills an edge in both.

Corollary 8. *Adding or removing any number of isolated vertices in a hypergraph does not modify the outcome.*

Proof. An isolated vertex obviously has outcome \mathcal{A} , which we know is neutral for the disjoint union by Theorem 6. \square

2.4 Some basic strategic principles

A crucial property of the Maker-Breaker convention is subhypergraph monotonicity: if Maker wins on some subhypergraph of H , then Maker also wins on H . A similar result holds for Enforcer in the Avoider-Enforcer convention. Let us specify that we will never consider induced subhypergraphs in this paper: a subhypergraph of H is anything that can be obtained from H by removing vertices and/or edges.

Proposition 9 (Monotonicity Principle). *Let H be a hypergraph, and let H' be a subhypergraph of H . If Enforcer wins on H' as last player (resp. as second-to-last player), then Enforcer wins on H as last player (resp. as second-to-last player).*

Proof. It is possible to build H from the subhypergraph H' in two steps, by adding the missing vertices first and then the missing edges. The first step is outcome-neutral by Corollary 8. The second step cannot harm Enforcer: obviously, a winning Enforcer strategy on a given hypergraph also wins on any hypergraph with the same vertex set and additional edges. \square

From Avoider's side of things, pairing strategies are a simple but powerful tool [16]. A *pairing* in a hypergraph H is a set Π of pairwise disjoint pairs in $V(H)$ such that, for all $e \in E(H)$, there exists $\pi \in \Pi$ satisfying $\pi \subseteq e$.

Proposition 10 (Pairing Principle). *Let H be a hypergraph. If H admits a pairing, then $o(H) = \mathcal{A}$.*

Proof. Let Π be a pairing of H . It suffices to address the case where Avoider is the last player. Her strategy is as follows: whenever Enforcer has just picked some $y \in \{x, y\} \in \Pi$, she answers by picking x , otherwise she picks an arbitrary vertex outside Π . Since Π involves an even number of vertices and Avoider is the last player, she can always follow this strategy: indeed, before Enforcer's turn, there is always an even number of unpicked vertices both inside Π and outside Π . Enforcer is thus forced to play first inside each pair from Π . At the end of the game, Enforcer has picked at least one (actually: exactly one) vertex in each pair from Π , which ensures that he has filled a transversal of H by definition of a pairing. Therefore, Avoider wins. \square

This principle may also be applied locally, for instance if we identify a pair of vertices that are equivalent in the following sense:

Definition 11. Let H be a hypergraph, and let $x, y \in V(H)$ be distinct. We say x and y are *indistinguishable* in H if the following two conditions are both satisfied:

- For all $e \in E(H)$ such that $x \in e$ and $y \notin e$, we have $(e \setminus \{x\}) \cup \{y\} \in E(H)$.
- For all $e \in E(H)$ such that $y \in e$ and $x \notin e$, we have $(e \setminus \{y\}) \cup \{x\} \in E(H)$.

If two vertices are indistinguishable, we can assume that the two players will each pick one of the vertices. This is the principle of the so-called ‘‘Super Lemma’’, proved in Oijid’s PhD thesis [23]. For the sake of completeness, we give a proof here. Note that if x and y are indistinguishable vertices of an hypergraph H , then $H^{+x-y} = H^{+y-x}$. This hypergraph corresponds to the hypergraph obtained from H as follows: x and y are removed, any edge containing both x and y is removed, and any edge e containing exactly one of x or y is replaced by $e \setminus \{x, y\}$.

Proposition 12 (Super Lemma [23]). *Let H be a hypergraph, and let $x, y \in V(H)$ be indistinguishable. Then $o(H) = o(H^{+x-y})$.*

Proof. Let $H' = H^{+x-y} = H^{+y-x}$. Fix an order (i.e. choose the last player), and let P be the player who has a winning strategy Σ on H' for that order. We want to show that P also has a winning strategy on H for the same order. Note that the first player is the same in both games since H and H' have the same parity. Therefore, P can play a fictitious game on H' following the strategy Σ and copy the moves in the real game played on H , except when the opponent picks x or y in which case P picks the other. If x and y are the only vertices remaining in the real game and P is second-to-last, then P picks one arbitrarily. In all cases, the vertices x and y end up being shared between both players in the real game: by definition of H' , this implies that Avoider fills an edge of H in the real game if and only if Avoider fills an edge of H' in the fictitious game. By optimality of Σ , this ensures that P wins. \square

Thanks to Proposition 12, hypergraphs can always be assumed to have no pairs of indistinguishable vertices, up to some polynomial-time preprocessing.

3 Hypergraphs of rank 2

In this section, we solve the game on hypergraphs of rank 2. Note that, by Proposition 4, the case of hypergraphs of rank 2 reduces to the case of 2-uniform hypergraphs i.e. graphs. Therefore, we will only consider graphs in this section.

3.1 Some elementary graphs and their outcome

We recall some vocabulary from the literature which will be useful, along with some notations. For any graph G , we denote by $2G$ the graph formed by two vertex-disjoint copies of G . We denote by P_n and C_n the *path* and the *cycle* on n vertices respectively. If x_1, \dots, x_n are consecutive vertices along a path P_n (resp. a cycle C_n), then that path (resp. cycle) may be denoted as $x_1x_2 \cdots x_n$. The *bull* is the graph obtained by adding a pendent edge to two vertices of C_3 (see Figure 3, left). The *3-sunlet* is the graph obtained by adding a pendent edge to all vertices of C_3 (see Figure 3, middle). A *pseudo-star* is any connected graph G such that $|V(G)| \geq 3$ and containing a vertex y such that all connected components of G^{-y} are of size 1 or 2. An example pseudo-star is given on the right of Figure 3. Note that C_3 , P_3 , P_4 and P_5 are pseudo-stars.

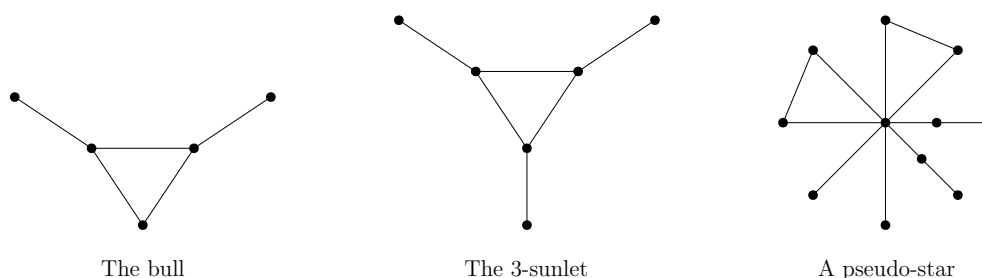


Figure 3: Some elementary graphs.

Proposition 13. P_1 and P_2 have outcome \mathcal{A} . The bull, C_5 and pseudo-stars have outcome \mathcal{SL} . The 3-sunlet, C_4 and $2P_3$ have outcome \mathcal{E} .

Proof. Clearly, $o(P_1) = o(P_2) = \mathcal{A}$. It is also straightforward that $o(P_3) = \mathcal{SL}$: indeed, Avoider wins as second-to-last player because she only picks one vertex in total, and Enforcer wins as second-to-last player by picking a vertex of degree 1 whatever Avoider's first pick is. As a consequence, $o(2P_3) = \mathcal{E}$ by Theorem 6. Since C_4 , C_5 , the bull, the 3-sunlet and pseudo-stars all contain P_3 as a subgraph, Enforcer wins on all these graphs as second-to-last player by Corollary 9 (Monotonicity Principle), so assume from now that Avoider is the second-to-last player.

- Let $xyzt$ be a C_4 . Since C_4 is even, Avoider plays first. By symmetry, we can assume that Avoider starts by picking x . Enforcer then picks z , and Avoider loses since her next pick will fill the edge xy or the edge xt .

- Let $xyztu$ be a C_5 . Since C_5 is odd, Enforcer plays first. By symmetry, we can assume that Enforcer starts by picking x . Avoider then picks y , and wins by picking either t or u in the second round.
- Consider a bull B formed by a C_3 xyz plus edges xu and yv . Since B is odd, Enforcer plays first. After Enforcer's first pick, x and u are both free, or y and v are both free. By symmetry, assume the former. Avoider then picks u , and wins by not picking x in the second round.
- Consider a 3-sunlet S formed by a C_3 xyz plus edges xu , yv and zw . Since S is even, Avoider plays first. If Avoider plays inside the cycle before Enforcer does, then she loses, since Enforcer will then force her to pick one of the other two vertices of the cycle eventually. Therefore, by symmetry, assume Avoider starts by picking u . Enforcer then picks v , and Avoider picks w . Enforcer finally picks y , and Avoider loses since her next pick will fill the edge xu or the edge zw .
- Lastly, consider a pseudo-star G . By definition, there exists a vertex y such that each component of G^{-y} is a P_1 or a P_2 . Therefore, the edges of G^{-y} form a pairing of G^{-y} , so $o(G^{-y}) = \mathcal{A}$ by Proposition 10 (Pairing Principle). Proposition 3 thus ensures that Avoider wins on G as second-to-last player. \square

3.2 Complete structural characterisation of the outcome

Using the previous elementary examples, we are going to establish the following result, which solves the game for general graphs.

Theorem 14. *Let G be a graph.*

- *If G does not contain P_3 as a subgraph, then $o(G) = \mathcal{A}$;*
- *If G contains $2P_3$, C_4 or the 3-sunlet as a subgraph, then $o(G) = \mathcal{E}$;*
- *In all other cases, $o(G) = \mathcal{SL}$.*

Corollary 15. *The outcome of the Avoider-Enforcer game can be computed in polynomial time on hypergraphs of rank 2.*

To prove Theorem 14, we discuss depending on who is the last player. For both cases, we give two equivalent characterisations of the winner: one that describes the connected components (extensional) and one by forbidden subgraphs (intensional). Let us start with the case where Avoider is the last player, which is straightforward.

Proposition 16. *Let G be a graph. The following three assertions are equivalent:*

- (1) *Avoider wins on G as last player.*
- (2) *Every connected component of G is P_1 or P_2 .*
- (3) *G does not contain P_3 as a subgraph.*

Proof. It is clear that (3) \implies (2). Since Avoider loses on P_3 as last player by Proposition 13, we also have (1) \implies (3) by Proposition 9 (Monotonicity Principle). Finally, if each connected component of G is P_1 or P_2 , then the edges of G form a pairing of G hence $o(G) = \mathcal{A}$ by Proposition 10 (Pairing Principle), so (2) \implies (1). \square

The case where Enforcer is the last player is not difficult either, though slightly more involved. The key is the following lemma, which is purely a graph theory result.

Lemma 17. *The connected graphs that do not contain $2P_3$, C_4 or the 3-sunlet as a subgraph are exactly the following:*

1. P_1 ;
2. P_2 ;
3. C_5 ;
4. the bull;
5. pseudo-stars.

Proof. Let G be a connected graph that does not contain $2P_3$, C_4 or the 3-sunlet as a subgraph. We distinguish between three cases:

- Firstly, suppose G is a path P_n . Since G contains no $2P_3$, we have $n \leq 5$. If $n = 1$ (resp. $n = 2$), then we are in Case 1 (resp. Case 2). If $n \in \{3, 4, 5\}$, then we are in Case 5.
- Secondly, suppose G is a cycle C_n . Since G contains no $2P_3$, we have $n \leq 5$. Moreover, since G contains no C_4 , we have $n \neq 4$. If $n = 3$, then we are in Case 5. If $n = 5$, then we are in Case 3.
- Lastly, suppose G is neither a path nor a cycle. Since G is connected, this means G contains a vertex x of degree at least 3. Let y, z, t be three distinct neighbours of x . If G^{-x} contains no P_3 , then we are in Case 5. Therefore, we now assume that G^{-x} contains a path P on three vertices. Define $m = |\{y, z, t\} \cap V(P)|$.
 - It is impossible that $m = 3$. Indeed, if we had $P = yzt$ for instance, then $xyzt$ would be a C_4 in G .
 - It is impossible that $m \leq 1$. Indeed, if we had $y, z \notin V(P)$ for instance, then $yxz \cup P$ would be a $2P_3$ in G .
 - In conclusion, we have $m = 2$. Without loss of generality, assume that $V(P) = \{u, y, z\}$ for some $u \notin \{x, y, z, t\}$. It is impossible that $P = yuz$, because $xyuz$ would be a C_4 in G . Therefore, we have $P = uyz$ or $P = yzu$: by symmetry, assume $P = uyz$. We get a bull B in G formed by the C_3 xyz and the pendant edges xt and yu . Finally, it is impossible that G contains a sixth edge intersecting $V(B)$, because a $2P_3$, C_4 or 3-sunlet would appear in all cases: see Figure 4. Since G is connected, this means $G = B$, so we are in Case 4.

Conversely, it is clear that P_1 , P_2 , C_5 , the bull and pseudo-stars are connected graphs which do not contain $2P_3$, C_4 or the 3-sunlet as a subgraph. This ends the proof. \square

Proposition 18. *Let G be a graph. The following three assertions are equivalent:*

- (1) *Avoider wins on G as second-to-last player.*
- (2) *Every connected component of G is P_1 or P_2 , except for at most one which is either: C_5 , the bull or a pseudo-star.*

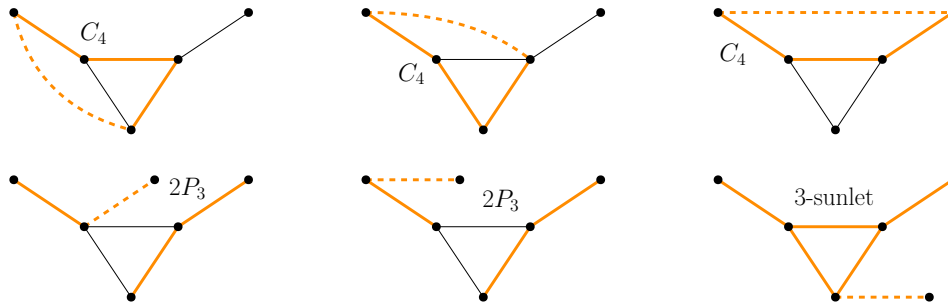


Figure 4: All possibilities of adding an edge (dashed line) connected to a bull, up to symmetries. The forbidden subgraph thus created is highlighted.

(3) G does not contain $2P_3$, C_4 or the 3-sunlet as a subgraph.

Proof. Since Avoider loses on $2P_3$, C_4 and the 3-sunlet as second-to-last player by Proposition 13, we have (1) \implies (3) by Proposition 9 (Monotonicity Principle). If G does not contain $2P_3$, C_4 or the 3-sunlet as a subgraph, then each connected component of G is a P_1 , P_2 , C_5 , bull or pseudo-star by Lemma 17, moreover at most one is a C_5 , bull or pseudo-star otherwise we would get a $2P_3$, so (3) \implies (2). Finally, recall that $o(P_1) = o(P_2) = \mathcal{A}$ and $o(C_5) = o(\text{bull}) = o(\text{pseudo-star}) = \mathcal{SL}$ by Proposition 13. Therefore, using Theorem 6, a disjoint union of copies of P_1 and P_2 has outcome \mathcal{A} , and adding one component which is a C_5 , bull or pseudo-star makes the outcome \mathcal{SL} . Since \mathcal{A} and \mathcal{SL} are the two outcomes where Avoider wins as second-to-last player, we get (2) \implies (1). \square

Proof of Theorem 14 and Corollary 15. Theorem 14 clearly follows from Propositions 16 and 18. Since testing whether a graph contains a fixed given subgraph can be done in polynomial time, Corollary 15 ensues. \square

4 Linear hypergraphs of rank 3

After settling hypergraphs of rank 2, a natural first step is to address linear hypergraphs of rank 3, similarly to what was done for the Maker-Breaker convention [20]. In this section, we solve the game on linear hypergraphs of rank 3 when Avoider is the last player.

4.1 Leaf-edges and reduced hypergraphs

The most basic linear hypergraphs are linear trees. Let us start by observing some of their properties.

Proposition 19. *Let T be a linear tree.*

- *If T has no leaf-edge, then T is an isolated vertex.*
- *If T has exactly one leaf-edge, then T has a single edge.*
- *If T has exactly two leaf-edges e and e' , then T forms a linear simple walk (e, \dots, e') .*

Proof. The first two assertions are straightforward. Let us prove the third one by induction on the number of edges. Clearly, if a linear tree has exactly two edges, then both are leaf-edges and they form a linear simple walk. Now, let $m \geq 2$ such that the property holds, and let T be a linear tree with exactly two leaf-edges, obtained from an isolated vertex by adding edges e_1, \dots, e_{m+1} successively. Denote by T_i the linear tree with edges e_1, \dots, e_i . Since T_2 and $T_{m+1} = T$ have exactly two leaf-edges, and the number of leaf-edges never decreases during the process of adding edges, T_m must also have exactly two leaf-edges e_r and e_s for some $1 \leq r, s \leq m$. Now, by the induction hypothesis, T_m forms a linear simple walk $(e_r \dots, e_s)$. As e_{m+1} is a leaf-edge of T_{m+1} , one of e_r and e_s is not a leaf-edge of T_{m+1} . By symmetry, assume e_s is not a leaf-edge of T_{m+1} : we have $e_{m+1} \cap e_s = \{x\}$ where x is of degree 1 in T_m . Since $|e_{m+1} \cap V(T_m)| = 1$, this implies that e_s is the only edge of T_m that intersects e_{m+1} , so $(e_r \dots, e_s, e_{m+1})$ is a linear simple walk. \square

Note that two vertices of degree 1 in the same leaf-edge of a hypergraph are indistinguishable, so Proposition 12 allows us to remove these two vertices and that leaf-edge without altering the outcome. Therefore, we introduce the following property, which we will freely assume throughout our study of the game on linear hypergraphs of rank 3.

Definition 20. A hypergraph of rank 3 is *reduced* if it contains no leaf-edge of size 3. The *reduced hypergraph* of a hypergraph H is obtained from H as follows: remove every leaf-edge $e \in E(H)$ of size 3 as well as two vertices of degree 1 of e , then repeat the process on the remaining hypergraph until there is no leaf-edge of size 3 anymore.

Note that the definition of a linear tree yields the following property.

Observation 21. *Let H be a 3-uniform hypergraph. The reduced hypergraph of H is an isolated vertex if and only if H is a linear tree.*

4.2 Some elementary linear hypergraphs of rank 3

Definition 22. We define the following linear hypergraphs of rank 3:

- For distinct vertices a and b , an *ab-chain* is a hypergraph P which forms a linear simple walk (e_1, \dots, e_ℓ) in which $|e_i| = 3$ for all $1 \leq i \leq \ell$ and e_1 (resp. e_ℓ) is the only edge containing a (resp. b). Such a walk is said to *represent* P . We say $\ell = |E(P)| \geq 1$ is the *length* of P . A *chain* is an *ab-chain* for some vertices a and b .
- A *nunchaku* [11] is a hypergraph N formed by the edges of a linear simple walk (e_1, \dots, e_ℓ) in which $|e_i| = 3$ for all $2 \leq i \leq \ell - 1$ and $|e_1| = |e_\ell| = 2$. We say $\ell = |E(N)| \geq 2$ is the *length* of N .
- A *cycle* is a hypergraph C obtained from an *ab-chain* P of length at least 2 by adding a 3-edge $\{a, b, c\}$ where $c \notin V(P)$. We say $\ell = |E(C)| \geq 3$ is the *length* of C .
- The *prism* is the hypergraph H obtained from a cycle C of length 3 by adding a 3-edge containing the vertices that are of degree 1 in C .

See Figure 5 for some illustrations.

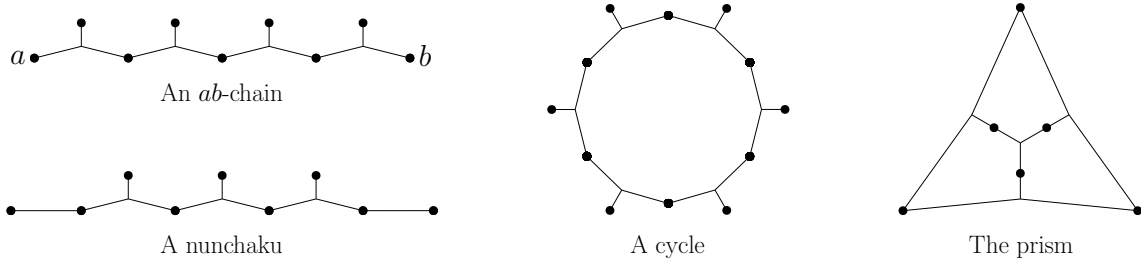


Figure 5: Some elementary linear hypergraphs of rank 3.

Given a connected 3-uniform linear hypergraph H and two distinct vertices $a, b \in V(H)$, we denote by $\text{dist}_H(a, b)$ the length of a shortest ab -chain in H . The following lemma ensures in particular that dist_H is a metric on $V(H)$.

Lemma 23. *Let H be a linear hypergraph, and let $a, b, c \in V(H)$ be pairwise distinct. Let P be an ab -chain and let P' be a bc -chain. Then $P \cup P'$ contains an ac -chain.*

Proof. Let (e_1, \dots, e_ℓ) and (e'_1, \dots, e'_m) be linear simple walks representing P and P' respectively, with $a \in e_1$, $b \in e_\ell \cap e'_1$ and $c \in e'_m$. If $c \in V(P)$, then we can define $1 \leq i \leq \ell$ as the minimum index such that $c \in e_i$, and the sequence (e_1, \dots, e_i) clearly is a linear simple walk which represents an ac -chain. A similar argument concludes if $a \in V(P')$. Therefore, we may assume that $a \notin V(P')$ and $c \notin V(P)$. Since $b \in V(P')$, we can define $1 \leq i \leq \ell$ as the minimum index such that $e_i \cap V(P') \neq \emptyset$, and $1 \leq j \leq m$ as the maximum index such that $e_i \cap e'_j \neq \emptyset$. The definitions of i and j ensure that the sequence $(e_1, \dots, e_i, e'_j, \dots, e'_m)$ is a simple walk, moreover it is linear since H is. Finally, since $a \notin V(P')$ and $c \notin V(P)$, the only edge of this walk containing a is e_1 and the only edge containing c is e'_m , therefore it represents an ac -chain. \square

We will see that nunchakus, prisms and cycles play a major role for the Avoider-Enforcer game on hypergraphs of rank 3. We now determine their outcome.

Proposition 24. *Nunchakus have outcome \mathcal{SL} . The prism and cycles have outcome \mathcal{A} .*

Proof. We first prove the result on nunchakus when Avoider is the last player (note that Avoider is also the first player since nunchakus are odd). We proceed by induction on length. A nunchaku of length 2 is simply a P_3 graph, on which we already know Avoider loses as last player. Now, let N be a nunchaku of length $\ell \geq 3$, and assume that Avoider loses as last player on all nunchakus of length at most $\ell - 1$. Let a and b be the vertices of degree 1 in the 2-edges of N . If Avoider's first pick is in one of the 2-edges, then she loses since this creates a 1-edge. Otherwise, Avoider's first pick x creates two new nunchakus N_1 and N_2 , both shorter than N , one of which does not contain a and the other does not contain b (see Figure 6). Enforcer can respond by picking $y \in \{a, b\}$, so that N_i is a subhypergraph of the updated hypergraph N^{+x-y} for some $i \in \{1, 2\}$. The induction hypothesis ensures that Enforcer wins on N_i as second-to-last player, so Enforcer wins on N^{+x-y} as second-to-last player by Proposition 9 (Monotonicity Principle), therefore Enforcer wins on N as second-to-last player.

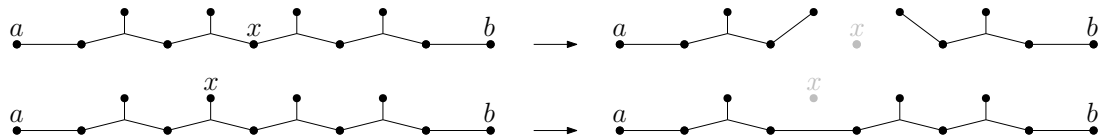


Figure 6: Top: if x is of degree 2 in N (left) then N^{+x} contains two disjoint nunchakus (right). Bottom: if x is of degree 1 in N (left) then N^{+x} contains two nunchakus sharing a 2-edge (right).

To show that Avoider wins on any nunchaku N as second-to-last player, it suffices to show that there exists $y \in V(N)$ such that Avoider wins on N^{-y} as last player, according to Proposition 3. Let y be a vertex of degree 1 in a 2-edge of N : since N^{-y} admits a pairing, as illustrated in Figure 7, we know $o(N^{-y}) = \mathcal{A}$ by Proposition 10 (Pairing Principle).

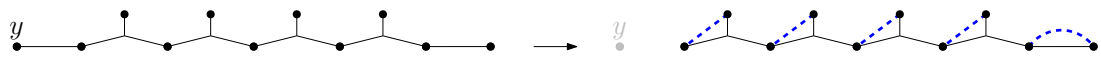


Figure 7: N is on the left, N^{-y} is on the right with a pairing being represented.

It is clear that a cycle admits a pairing, where each vertex of degree 1 is paired with a neighbour of degree 2, so Avoider likewise wins on any cycle as last player. Finally, let H be a prism, and suppose Avoider is the last player: since H is even, this means Enforcer is the first player. Enforcer picks some vertex y , and Avoider picks some $x \in N(y)$ (the set of neighbours of y). Since x is of degree 2 in H (as is any vertex in a prism), this choice ensures that the updated hypergraph H^{-y+x} contains only one 2-edge e . There are only two rounds of play left, so Avoider simply has to pick some $z \notin e$ in the next round to win. \square

4.3 Structural characterisation of the outcome when Avoider is the last player

We are going to establish the following result.

Theorem 25. *Let H be a reduced linear hypergraph of rank 3. The following two assertions are equivalent:*

- (1) *Avoider wins on H as last player.*
- (2) *Each connected component of H is a cycle, a prism, an isolated 2-edge or an isolated vertex.*

Corollary 26. *Deciding whether a linear hypergraph of rank 3 has outcome \mathcal{A} can be done in polynomial time.*

First of all, notice that addressing the connected case is sufficient: indeed, by Theorem 6, Avoider wins as last player if and only if Avoider wins on each connected component as last player. Also note that the result is trivial if there exists a 0-edge or a 1-edge, since Avoider then loses as last player.

Enforcer's main idea is to rely on Avoider creating a nunchaku: since a nunchaku has outcome \mathcal{SL} according to Proposition 24, this would ensure Enforcer's win as second-to-last player. This happens more easily if Avoider is the first player. Therefore, we distinguish two cases depending on parity.

4.3.1 Odd case: Avoider is first and last

We can first address the case where there exists a 2-edge.

Proposition 27. *Avoider loses as last player on any odd connected linear hypergraph of rank 3 which contains a 2-edge.*

Proof. If Avoider plays her first move inside a 2-edge, then this creates a 1-edge. Otherwise, linearity and connectedness ensure that there exists a chain between the vertex that Avoider picks and a 2-edge, so Avoider's first move creates a nunchaku. In all cases, we get a subhypergraph that has outcome \mathcal{SL} , so Enforcer wins as second-to-last player by Proposition 9 (Monotonicity Principle). \square

In the 3-uniform case, Avoider needs to play twice before a nunchaku can appear. After Avoider plays her first move, Enforcer wants to pick a non-cut vertex of the starting hypergraph, so that Avoider's second pick will necessarily be connected to the first by a chain. For this, the following lemma (which is not game-related in itself) is key.

Lemma 28. *Any connected reduced linear 3-uniform hypergraph which is not an isolated vertex has at least two non-cut vertices.*

Lemma 28 is an immediate consequence of the following result, which details how to find a non-cut vertex.

Lemma 29. *Let H be a connected reduced linear 3-uniform hypergraph which is not an isolated vertex, and let x be an arbitrary vertex of H . Out of all vertices that maximise $\text{dist}_H(x, \cdot)$, let u be one with minimum degree. Then u is a non-cut vertex of H .*

Proof. Suppose for a contradiction that u is a cut vertex of H . First notice that all vertices which are not in the same connected component as x in H^{-u} are neighbours of u . More specifically:

Claim 30. *Let y be a vertex that is not in the same connected component as x in H^{-u} . Then $\text{dist}_H(x, y) = \text{dist}_H(x, u)$, and there exists an edge $e_y \supseteq \{u, y\}$ such that, for all shortest xy -chain in H , e_y is the only edge of the chain containing u .*

Proof of Claim 30. Let y be a vertex that is not in the same connected component as x in H^{-u} : in other words, all xy -chains in H contain u . Let P be a shortest xy -chain in H , and let e_y be its edge containing y . By definition of u , we have $\text{dist}_H(x, y) \leq \text{dist}_H(x, u)$, so u cannot appear in any edge of P other than e_y , otherwise we would get an xu -chain shorter than P (see Figure 8). By linearity of H , that edge e_y must belong to all shortest xy -chains in H . \square



Figure 8: Left: the actual position of u inside P . Right: any other position for u yields an xu -chain shorter than P .

Let y be a vertex that is not in the same connected component as x in H^{-u} , which exists since u is a cut vertex of H . Let e_y be given by Claim 30. Note that, by linearity of H , all neighbours of y not in e_y are in the same connected component as y in H^{-u} , so we can also apply Claim 30 to them.

Claim 31. *There cannot exist an edge $\{u, z_1, z_2\}$ where $z_1, z_2 \in N(y) \setminus e_y$.*

Proof of Claim 31. Suppose for a contradiction that such an edge $e = \{u, z_1, z_2\}$ exists, and let P be a shortest xz_1 -chain in H . We deduce from Claim 30 that the only edge of P containing u is the one containing z_1 . By linearity of H , that edge is necessarily e , so we get an xz_2 -chain shorter than P in H , contradicting the fact that $\text{dist}_H(x, z_2) = \text{dist}_H(x, u) = \text{dist}_H(x, z_1)$ by Claim 30. \square

Putting Claims 30 and 31 together, we see that all elements of $N(y) \setminus e_y$ are neighbours of u , but no two of them can be in the same edge alongside u . Therefore:

$$d(u) \geq |N(y) \setminus e_y| + |\{e_y\}| = |N(y)| - 1 = 2d(y) - 1.$$

However, since $\text{dist}_H(x, y) = \text{dist}_H(x, u)$, we must have $d(u) \leq d(y)$ according to our choice of u . The only possibility is that $d(u) = d(y) = 1$, which means that e_y is a leaf-edge, a contradiction to the fact that H is reduced. \square

Proof of Lemma 28. It suffices to consider any vertex x , apply Lemma 29 on x to get a non-cut vertex u , and then apply Lemma 29 again on u to get a second one. \square

Corollary 32. *Let H be an odd connected reduced linear 3-uniform hypergraph. Avoider wins on H as last player if and only if H is an isolated vertex. Without the “reduced” assumption, the same holds when replacing “isolated vertex” by “linear tree”.*

Proof. Suppose H is not an isolated vertex. Since H is odd, Avoider plays first and picks some vertex x . Enforcer then picks a non-cut vertex $y \neq x$, which exists by Lemma 28, and then Avoider picks some vertex z . Since H^{-y} is connected, there exists an xz -chain in H that does not contain y . In the updated hypergraph, that chain is now a nunchaku. Since a nunchaku has outcome \mathcal{SL} , Enforcer wins as second-to-last player by Proposition 9 (Monotonicity Principle). As for the last assertion of this corollary, it follows from Observation 21. \square

4.3.2 Even case: Avoider is second and last

Again, we can first address the case where there exists a 2-edge:

Proposition 33. *Let H be an even connected reduced linear hypergraph of rank 3 which contains a 2-edge. Avoider wins on H as last player if and only if H is an isolated 2-edge.*

Proof. The “if” direction is trivial, so let us show the “only if” direction. Suppose Avoider wins on H as last player. In particular, H contains no 0-edge, no 1-edge, and only one 2-edge $e = \{a, b\}$, otherwise we would get a nunchaku in H . The idea is to transform into the odd 3-uniform case which we have already addressed. For this, define the hypergraph H_0 obtained from H by adding a new isolated vertex v : by Corollary 8, the outcome is unchanged. Then, define the hypergraph H_1 obtained from H_0 by replacing the 2-edge $e = \{a, b\}$ with the superset 3-edge $e_1 = \{a, b, v\}$: Avoider obviously wins on H_1 as last player, with the same strategy as on H_0 . Note that H_1 is an odd connected linear 3-uniform hypergraph. Therefore, by Corollary 32, H_1 is a linear tree, so H is a linear tree as well. Since H is reduced, e is the only edge of H , which concludes. \square

In the 3-uniform case, we are once again going to need a structural lemma on hypergraphs which is not game-related.

Lemma 34. *Let H be a connected reduced linear 3-uniform hypergraph which is neither an isolated vertex, a cycle nor a prism. Then there exists a vertex $y \in V(H)$ such that no connected component of H^{-y} is a linear tree.*

Proof. Let H be a connected reduced linear 3-uniform hypergraph which is neither an isolated vertex, a cycle nor a prism. Let y be a vertex of minimum degree in H . If no connected component of H^{-y} is a linear tree, then we are done. Therefore, we assume from now that there exists a connected component H' of H^{-y} which is a linear tree.

First suppose that y has degree 1, and let $e = \{y, z_1, z_2\}$ be the unique edge containing y . It is impossible that H' is an isolated vertex t , otherwise $t \in \{z_1, z_2\}$ and e would be a leaf-edge of H , contradicting the fact that H is reduced. Moreover, H' cannot consist of a single edge e' , otherwise exactly one vertex of e' would intersect e and e' would be a leaf-edge of H , once again contradicting the fact that H is reduced. Hence, by Proposition 19, there are at least two leaf-edges in H' . Since H is reduced, it contains no leaf-edge, so the leaf-edges of H' are not leaf-edges in H . As such, e must contain at least one vertex of degree 1 of each leaf-edge of H' . Since the vertices of degree 1 of the leaf-edges of H' are pairwise distinct, this means H' has exactly two leaf-edges, which contain z_1 and z_2 respectively as a vertex of degree 1. Therefore, H' is a z_1z_2 -chain by Proposition 19. In particular, since z_1 and z_2 are in the same connected component of H^{-y} , we know that H^{-y} is connected, so $H^{-y} = H'$. All in all, since H^{-y} is a z_1z_2 -chain, H is a cycle, which is a contradiction.

Now, suppose that y has degree at least 2. This means that there is no vertex of degree 1 in H . In particular, H' cannot be an isolated vertex, since that vertex would be of degree 1 in H . Furthermore, all the vertices of degree 1 in H' are of degree 2 in H and must be neighbours of y . Note that, by Proposition 19, H' does have vertices

of degree 1, so the minimality of y ensures that y is of degree exactly 2. Write the two edges containing y as $\{y, v_1, v_2\}$ and $\{y, w_1, w_2\}$. Since H is linear, two vertices of degree 1 in the same leaf-edge of H' cannot share an edge containing y . This means that H' cannot consist of a single edge, as its three vertices would be of degree 1. Therefore, by Proposition 19, H' has at least two leaf-edges, which amounts to at least four vertices of degree 1. Since those vertices are neighbours of y , we conclude that H' has exactly two leaf-edges (so H' is a chain by Proposition 19) and exactly four vertices of degree 1 which are v_1, v_2, w_1 and w_2 (so the chain H' is of length 2). Let z be the fifth vertex of H' : up to exchanging the roles of w_1 and w_2 , we can write the two edges of H' as $\{z, v_1, w_1\}$ and $\{z, v_2, w_2\}$. Since all neighbours of y are in the same connected component of H^{-y} , we know that H^{-y} is connected, so $H^{-y} = H'$. All in all, we have $V(H) = \{y, z, v_1, v_2, w_1, w_2\}$ and $E(H) = \{\{y, v_1, v_2\}, \{y, w_1, w_2\}, \{z, v_1, w_1\}, \{z, v_2, w_2\}\}$, so H is a prism, which is a contradiction. \square

Corollary 35. *Let H be an even connected reduced linear 3-uniform hypergraph. Avoider wins on H as last player if and only if H is either a cycle or a prism.*

Proof. The “if” direction is given by Proposition 24. For the “only if” direction, suppose H is neither a cycle nor a prism. By Lemma 34, there exists $x \in V(H)$ such that no connected component of H^{-x} is a linear tree: Enforcer picks a such y . We must show that Enforcer wins as second-to-last player on H^{-y} . Since H^{-y} is odd, it has an odd connected component H' , and we know H' is not a linear tree. Corollary 32 thus ensures that Enforcer wins on H' as second-to-last player, so Enforcer wins on H^{-y} as second-to-last player by Proposition 9 (Monotonicity Principle). \square

Proof of Theorem 25 and Corollary 26. All cases for the connected components are covered by Proposition 27, Corollary 32, Proposition 33 and Corollary 35, which proves Theorem 25. Since computing the reduced hypergraph and testing whether each connected component is a cycle, a prism, an isolated 2-edge or an isolated vertex can clearly be done in polynomial time, Corollary 26 ensues. \square

Conclusion and perspectives

We have completely solved the Avoider-Enforcer game on hypergraphs of rank 2 in all cases, and on linear hypergraphs of rank 3 when Avoider is the last player.

A first line of research would be to solve the case where Avoider is the last player on general hypergraphs of rank 3, without the linearity constraint. The difference with the linear case is that a non-linear path whose endpoints are picked by Avoider is not always winning for Enforcer. In particular, if it has an even total number of vertices, Enforcer will be forced to play in it first, unlike nunchakus. Furthermore, in contrast to the Maker-Breaker convention where all is decided after a constant number of moves [11], Avoider and Enforcer may stall by picking “useless” vertices before a losing structure is created by Avoider. This makes the analysis much harder. Another perspective, still with

Avoider playing last, is to look at some linear hypergraphs of higher rank. In particular, the case of linear trees seems accessible using the leaf-edges reduction.

When Enforcer is the last player, it seems difficult to find general results, even for linear hypergraphs of rank 3. Indeed, as discussed in Subsection 2.2, one cannot deduce directly from Theorem 25 a complete characterisation when Enforcer plays last. Since there are only few hypergraphs where Avoider wins as last player, Proposition 3 can only be used in very specific circumstances, leaving most cases open. However, when the minimum degree is high enough, the situation often becomes very favorable to Enforcer. Indeed, a consequence of Theorem 25 is that Avoider only wins playing last if there are vertices of degree 1 or 2. We think that a similar result is true when Enforcer is the last player.

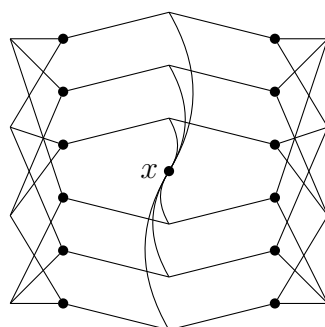


Figure 9: This hypergraph has minimum degree 3 and a cut-vertex x , but Avoider wins as second-to-last player (note that, when removing x , both the left and the right subhypergraphs are prisms).

Conjecture 36. Let H be a linear hypergraph of rank 3. There exists a constant d such that, if H has minimum degree d , then Enforcer wins on H as last player.

This conjecture is true for $d = 4$ when H has a cut-vertex x . Indeed, under these assumptions, H^{-x} has at least two connected components of minimum degree 3. Since Avoider loses as last player on each of these two components, Enforcer wins on their union as last player, so Enforcer wins on H as last player by monotonicity. Note that the conjecture is not true for $d = 3$. Indeed, consider the hypergraph H from Figure 9, obtained by taking two disjoint prisms and adding six 3-edges that pairwise connect the vertices of both prisms with a common additional vertex x . It has minimum degree 3 and a cut vertex x . Since H^{-x} is the disjoint union of two prisms, each of which has outcome \mathcal{A} by Theorem 25, Avoider wins on H^{-x} as last player. By Proposition 3, Avoider thus wins on H as second-to-last player.

Finally, we note that, for linear hypergraphs of rank 3 that are sufficiently connected, two disjoint nunchakus should appear after a small number of moves, thus guaranteeing that Enforcer wins as last player.

Acknowledgements

This research was partly supported by the ANR project P-GASE (ANR-21-CE48-0001-01).

References

- [1] J. Barát and M. Stojaković. On winning fast in Avoider-Enforcer games. *Electronic Journal of Combinatorics*, 17:R56, 2009. doi:[10.37236/328](https://doi.org/10.37236/328).
- [2] J. Beck. *Combinatorial Games: Tic-Tac-Toe Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2008. doi:[10.1017/CB09780511735202](https://doi.org/10.1017/CB09780511735202).
- [3] M. Bednarska-Bzdega, O. Ben-Eliezer, L. Gishboliner, and T. Tran. On the separation conjecture in Avoider-Enforcer games. *Journal of Combinatorial Theory, Series B*, 138:41–77, 2019. doi:[10.1016/j.jctb.2019.01.003](https://doi.org/10.1016/j.jctb.2019.01.003).
- [4] C. Berge. *Graphs and Hypergraphs*, volume 6 of *North-Holland Mathematical Library*. Elsevier, 1973. doi:[10.1016/s0924-6509\(09\)x7013-3](https://doi.org/10.1016/s0924-6509(09)x7013-3).
- [5] A. Bretto. *Hypergraph Theory: an Introduction*. Springer Publishing Company, Incorporated, 2013.
- [6] V. Chvátal and P. Erdős. Biased positional games. In *Algorithmic Aspects of Combinatorics*, volume 2 of *Annals of Discrete Mathematics*, 221–229. 1978. doi:[10.1016/S0167-5060\(08\)70335-2](https://doi.org/10.1016/S0167-5060(08)70335-2).
- [7] D. Clemens, J. Ehrenmüller, Y. Person, and T. Tran. Keeping Avoider’s graph almost acyclic. *The Electronic Journal of Combinatorics*, 22(1):#P1.60, 2015. doi:[10.37236/4859](https://doi.org/10.37236/4859).
- [8] P. Erdős and J. L. Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14, 1973. doi:[10.1016/0097-3165\(73\)90005-8](https://doi.org/10.1016/0097-3165(73)90005-8).
- [9] S. A. Fenner, D. Grier, J. Messner, L. Schaeffer, and T. Thierauf. Game values and computational complexity: an analysis via black-white combinatorial games. In *Algorithms and Computation*, 689–699, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. doi:[10.1007/978-3-662-48971-0_58](https://doi.org/10.1007/978-3-662-48971-0_58).
- [10] A. Ferber, M. Krivelevich, and A. Naor. Avoider-Enforcer games played on edge disjoint hypergraphs. *Discrete Mathematics*, 313(24):2932–2941, 2013. doi:[10.1016/j.disc.2013.09.008](https://doi.org/10.1016/j.disc.2013.09.008).
- [11] F. Galliot, S. Gravier, and I. Sivignon. Maker-Breaker is solved in polynomial time on hypergraphs of rank 3. Preprint, 2025. [arXiv:2209.12819v3](https://arxiv.org/abs/2209.12819v3).
- [12] V. Gledel and N. Oijid. Avoidance games are PSPACE-complete. In *40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023)*, volume 254 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 34:1–34:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:[10.4230/LIPIcs.STACS.2023.34](https://doi.org/10.4230/LIPIcs.STACS.2023.34).

- [13] A. Grzesik, M. Mikalački, Z. L. Nagy, A. Naor, B. Patkós, and F. Skerman. Avoider–Enforcer star games. *Discrete Mathematics & Theoretical Computer Science*, 17(1):145–160, 2015. doi:[10.46298/dmtcs.2124](https://doi.org/10.46298/dmtcs.2124).
- [14] A. Gyárfás, E. Györi, and M. Simonovits. On 3-uniform hypergraphs without linear cycles. *Journal of Combinatorics*, 7, 2014. doi:[10.4310/JOC.2016.v7.n1.a8](https://doi.org/10.4310/JOC.2016.v7.n1.a8).
- [15] A. W. Hales and R. I. Jewett. Regularity and positional games. *Transactions of the American Mathematical Society*, 106(2):222–229, 1963.
- [16] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó. Fast winning strategies in positional games. *Electronic Notes in Discrete Mathematics*, 29:213–217, 2007. doi:[10.1016/j.endm.2007.07.046](https://doi.org/10.1016/j.endm.2007.07.046).
- [17] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó. *Positional Games*, volume 44 of *Oberwolfach seminars*. Springer, 2014. doi:[10.1007/978-3-0348-0825-5](https://doi.org/10.1007/978-3-0348-0825-5).
- [18] D. Hefetz, M. Krivelevich, and T. Szabó. Avoider–Enforcer games. *Journal of Combinatorial Theory, Series A*, 114(5):840–853, 2007. doi:[10.1016/j.jcta.2006.10.001](https://doi.org/10.1016/j.jcta.2006.10.001).
- [19] F. O. Koepke. Solving Maker-Breaker games on 5-uniform hypergraphs is PSPACE-complete. *The Electronic Journal of Combinatorics*, 32(4), #P4.41, 2025. doi:[10.37236/13920](https://doi.org/10.37236/13920).
- [20] M. Kutz. Weak positional games on hypergraphs of rank three. *Discrete Mathematics & Theoretical Computer Science*, DMTCS Proceedings vol. AE, European Conference on Combinatorics, Graph Theory and Applications (EuroComb '05), 2005. doi:[10.46298/dmtcs.3422](https://doi.org/10.46298/dmtcs.3422).
- [21] X. Lu. Hamiltonian games. *Journal of Combinatorial Theory, Series B*, 55(1):18–32, 1992. doi:[10.1016/0095-8956\(92\)90030-2](https://doi.org/10.1016/0095-8956(92)90030-2).
- [22] T. Miltzow and M. Stojaković. Avoider–Enforcer game is NP-hard. Preprint, 2022. [arXiv:2208.06687](https://arxiv.org/abs/2208.06687).
- [23] N. Oijid. *Complexity of positional games on graphs*. PhD thesis, Université Lyon 1, 2024. URL: <http://www.theses.fr/2024LY010113>.
- [24] M. L. Rahman and T. Watson. 6-Uniform Maker-Breaker game is PSPACE-complete. *Combinatorica*, 43(3):595–612, 2023. doi:[10.1007/s00493-023-00026-7](https://doi.org/10.1007/s00493-023-00026-7).
- [25] T. J. Schaefer. On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16:185–225, 1978. doi:[10.1016/0022-0000\(78\)90045-4](https://doi.org/10.1016/0022-0000(78)90045-4).
- [26] M. Stojaković and J. Stratijev. On strong avoiding games. *Discrete Mathematics*, 346(3):113270, 2023. doi:[10.1016/j.disc.2022.113270](https://doi.org/10.1016/j.disc.2022.113270).