Avoidance games are PSPACE-Complete.
Valentin Gledel, Nacim Oijid

To cite this version:
Valentin Gledel, Nacim Oijid. Avoidance games are PSPACE-Complete.. Leibniz International Proceedings in Informatics , 2023, 254 (40), pp.34:1–34:19. 10.4230/LIPIcs.STACS.2023.34 . hal-03787958v2

HAL Id: hal-03787958
https://hal.science/hal-03787958v2
Submitted on 20 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Avoidance games are PSPACE-Complete

Valentin Gledel
Department of Mathematics and Mathematical Statistics, Umeå University, Sweden

Nacim Oijid
Univ. Lyon, Université Lyon 1, LIRIS UMR CNRS 5205, F-69621, Lyon, France

Abstract
Avoidance games are games in which two players claim vertices of a hypergraph and try to avoid some structures. These games have been studied since the introduction of the game of SIM in 1968, but only few complexity results have been found out about them. In 2001, Slany proved some partial results on Avoider-Avoider games complexity, and in 2017 Bonnet et al. proved that short Avoider-Enforcer games are Co-W[1]-hard. More recently, in 2022, Miltzow and Stojaković proved that these games are NP-hard. As these games correspond to the misère version of the well-known Maker-Breaker games, introduced in 1963 and proven PSPACE-complete in 1978, one could expect these games to be PSPACE-complete too, but the question has remained open since then. Here, we prove here that both Avoider-Avoider and Avoider-Enforcer conventions are PSPACE-complete. Using the PSPACE-hardness of Avoider-Enforcer, we provide in appendix proofs that some particular Avoider-Enforcer games also are.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Games, Avoider-Enforcer, Maker-Breaker, Complexity, Avoider-Avoider, PSPACE-complete

Funding Valentin Gledel: supported by the Kempe Foundation Grant No. JCK-2022.1 (Sweden).
Nacim Oijid: supported by the ANR project P-GASE (ANR-21-CE48-0001-01)

Acknowledgements We want to thank Eric Duchêne, Marianne Fortin, Aline Parreau and the anonymous referees for their help in the writing of this article

1 Introduction

1.1 Related works
Avoidance games belong to the class of positional games, that were introduced by Hales and Jewett in 1963 [14] and popularized by Erdős and Selfridge in 1973 [11]. In this class of games, the board is a hypergraph and two players alternately claim a vertex of the hypergraph that has not been claimed before. Winning conditions depend on the convention and are related to the hyperedges. Tic-Tac-Toe and Hex are two famous examples of positional games.

To learn more about positional games, we refer the reader to the recent survey of Hefetz et al. [18].

Among positional games, a natural dichotomy exists: on the one hand, there are games in which players seek to build a structure, and on the other hand, there are games in which players want to avoid a structure. The former set contains both Maker-Maker and Maker-Breaker conventions, in which the hyperedges are winning sets, and the player either wants to fill up a winning set (Maker role), or to play at least once in each of them (Breaker role). The latter contains Avoider-Avoider and Avoider-Enforcer conventions, that can be seen as the misère version of the former. In these games, the hyperedges are losing sets, and the players either want not to fill up one losing set (Avoider role), or to force their opponent to fill up one of them (Enforcer role).

When positional games were introduced, the focus was on Maker-Breaker games, i.e. games in which one player, Maker, aims to fill up a hyperedge, and the second one, Breaker,
Avoidance games are \textsc{PSPACE}-Complete

wants to prevent it by claiming at least one vertex in each hyperedge. This convention is
the most popular one, and several games have been studied according to this convention. In
particular, the survey of Beck [5] presents several results obtained for Maker-Breaker games.
The field of Maker-Breaker games is still well investigated today, and some Maker-Breaker
games were introduced recently [10, 25].

The first Avoider-Avoider game was introduced in 1968 with the game of SIM and is
presented in [28], but the first study of the complexity of Avoidance games was done by
Schaefer in 1978 [27]. Avoider-Enforcer games were introduced later by Lu in 1991 [22, 23]
under the name of Antimaker-Antibreaker games and correspond to the misère version
of Maker-Breaker games. The standard name for this convention, Avoider-Enforcer, was
popularized by Hefetz and different co-authors in 2007 [15, 16, 19, 20]. In this game, Enforcer
wins if at some point during the game, Avoider has claimed all the vertices of a hyperedge,
otherwise Avoider wins.

Even if most of the studies of positional games are focused on Maker-Breaker games,
Avoider-Enforcer games have become more and more relevant: the famous Ramsey game was
As it was done in the Maker-Breaker convention, some games on graphs were introduced in
Avoider-Enforcer or Avoider-Avoider conventions, where the loosing sets correspond to some
structure in the graph, see [2, 3, 13, 17].

In terms of complexity, an overview of the field is proposed by Demaine [9]. In positional
games, as they are perfect information games, one player always has a winning strategy
(or both players can ensure a draw). The natural decision problem related to games is
therefore: does the first player have a winning strategy? This problem was quickly proven to
be \textsc{PSPACE}-complete for Maker-Breaker games by Schaefer in 1978 [27] even restricted to
11-uniform hypergraphs (i.e. hypergraphs in which all hyperedges have size 11). This bound
was recently improved by Rahman and Watson in 2021 [26], proving that the problem is still
\textsc{PSPACE}-complete if the hypergraph is 6-uniform. These two proofs are very technical and
a simpler proof of the \textsc{PSPACE}-completeness was provided by Byskov in 2004 [8], proving
at the same time that Maker-Maker games are also \textsc{PSPACE}-complete. The complexity
of Maker-Breaker games is still studied today, as Galliot et al. [12] have proven that the
winner of a 3-uniform Maker-Breaker game can be computed in polynomial time, but the
gap between the complexity of 6-uniform hypergraphs and 3-uniform hypergraphs remains
to be closed.

Despite the fact that Avoidance games were introduced at the same time as Maker-Breaker
games, only partial results on complexity are known: determining the winner in Avoider-
Avoider games, was proven to be \textsc{PSPACE}-complete by Slany in 2002 [29] for endgames,
i.e. games in which some vertices are already attributed to the players, but there are no
mentioned that the complexity of this problem is still open, when they proved that short
games, i.e. games in which a player only has few moves to make, are co-W[1]-hard, with the
number of moves taken as a parameter. The best known result today is due to Miltzow and
Stojaković in 2022 [24] that states the \textsc{NP}-hardness of this decision problem and conjectures
its \textsc{PSPACE}-completeness.

1.2 Presentation of the results

The Avoider-Enforcer game is played as follows: given a hypergraph $H$, two players, called
Avoider and Enforcer, alternately claim an unclaimed vertex of $H$ with Avoider starting. The
game ends when all the vertices have been claimed. If Avoider has claimed all the vertices of
a hyperedge, Enforcer wins. Otherwise, Avoider wins. The related decision problem is the following one.

**Problem 1. Avoider-Enforcer**

*Input*: A hypergraph $H$.

*Output*: True if and only if Avoider has a winning strategy in the Avoider-Enforcer game on $H$.

This paper will focus on the proof of the following result:

**Theorem 2.** The Avoider-Enforcer problem is PSPACE-complete, even when the entry is restricted to hyperedges of size at most 6.

Our proof of Theorem 2 follows a similar idea to the proof of Rahman and Watson [26] and the proof of Schaefer [27], by constructing some hyperedges forcing the order of the moves. Contrary to Maker-Breaker games, in Avoider-Enforcer convention, there is no vertex in which the players are urged to play, as in general, players do not want to move in avoidance games. The key idea of this reduction is to create some structures in which playing first is a losing move. In the provided construction, at any moment of the game, only few moves are not losing moves. Thus, we can control the vertices played by the two players.

The proof provided for PSPACE-completeness of Avoider-Enforcer games, enables us to state the following corollary for Avoider-Avoider games that will also be proven later:

**Problem 3. Avoider-Avoider**

*Input*: A hypergraph $H$.

*Output*: True if and only if the second player has a winning strategy in the Avoider-Avoider game on $H$.

**Corollary 4.** The Avoider-Avoider problem is PSPACE-complete, even when the entry is restricted to 7-uniform hypergraphs.

This paper is organized as follows. In Section 2, we introduce two lemmas that will be used in the proof of Theorem 2. In particular, we show that pairing strategies that are often used in Maker-Breaker conventions can also be applied to Avoider-Enforcer games. Section 3 describes the reduction used to prove the PSPACE-completeness and define an order on the move that we call the legitimate order. We also show in this section that the proof holds if both players follow the legitimate order. In Section 4, we show that if a player does not follow the legitimate order then it cannot be a disadvantage to the other player, completing the proof of Theorem 2. Finally, in Section 5, we reduce the Avoider-Enforcer problem to 6-uniform hypergraphs and prove Corollary 4. One use of these results is to prove the PSPACE-completeness of particular Avoider-Enforcer games. In appendix we give two examples of such reductions with the cases of the Avoider-Enforcer domination game and the Avoider-Enforcer vertex $H$-game.

## 2 Preliminaries

In Maker-Breaker games, if a vertex is in all the hyperedges, it is always an optimal move for both players to play it. Here, we present a similar result for Avoider-Enforcer games. This result was proved by Miltzow and Stojaković [24], and intuitively states that if a vertex $v$ is in all the hyperedges that contains another vertex $u$, then claiming $v$ before $u$ cannot benefit any player.
Avoidance games are PSPACE-Complete

Lemma 5. Let $H$ be a hypergraph, and $u, v$ two vertices of $H$ such that, for every hyperedge $e$ containing $u$, $e$ also contains $v$. If a player has a winning strategy, then this player has a winning strategy in which he never claims $v$ while $u$ is unclaimed.

The second tool we introduce here is pairing strategies in Avoider-Enforcer games. In Maker-Breaker convention, these strategies are often described by using the fact that a player can claim at least one vertex in each pair of vertices. Here in Avoider-Enforcer convention, the main idea of pairing strategies is that this is always possible to force the opponent to claim at least one vertex in each pair.

In this section, we will refer to the players as Alice and Bob, as the strategy can be applied both by Avoider and by Enforcer.

Lemma 6. Let $H = (V, E)$ be a hypergraph. Suppose that Alice plays last in $H$, i.e. if the game is played until all the vertices have been claimed, Alice will claim the last one. Let $(a_1, b_1), \ldots, (a_n, b_n)$ be pairwise disjoint pairs of vertices, and let $v \notin \bigcup_{i=1}^{n} \{a_i, b_i\}$.

Alice has a strategy which forces Bob to claim at least one vertex in each pair $(a_i, b_i)$.

Bob has a strategy which forces Alice to claim $v$ and at least one vertex in each pair $(a_i, b_i)$.

A strategy satisfying the hypothesis of Lemma 6 will be called a pairing strategy.

Proof. Consider the following strategy for Alice:
- If Bob claims a vertex in a pair $(a_i, b_i)$, she claims the other vertex of the pair.
- Otherwise, she claims any vertex that is not in a pair.

By construction; when it is Bob’s turn, in any pair in which he has played, Alice has also played. Therefore, when it is Alice’s turn, there is at most one pair of vertices in which she has to play. As Alice plays the last move, whenever it is her turn to play, the number of remaining vertices is odd. Therefore, if Bob does not play in a pair, at least one vertex in no pair will be available for Alice. Thus, Alice has a strategy to force Bob to claim at least one vertex in each pair $(a_i, b_i)$.

Now, consider the following strategy for Bob:
- If Alice claims a vertex in a pair $(a_i, b_i)$, he claims the other vertex of the pair.
- If Alice claims $v$, if there exists at least one pair $(a, b)$ in which Alice has not played, he claims $a$ and he considers now that $b$ is the new vertex that Alice will be forced to claim.
- Otherwise, he claims any vertex that is not in a pair nor $v$.

For the same reason, with this strategy, when it is Alice’s turn, in any pair in which she has played, Bob has played too. When it is Bob’s turn, note that the number of remaining moves is even, and there always exists exactly one vertex that is not in a pair, and that Bob wants Alice to claim. Thus, the number of vertices on which Bob cannot play before Alice is odd. Therefore, he always has an available move that fulfills this strategy.

3 Proof of the main theorem

In this section, we begin the proof of Theorem 2 by describing the reduction from 3-QBF, introducing an order on the move of Avoider and Enforcer called the legitimate order and providing a sketch of the general proof.

Theorem 2. The Avoider-Enforcer problem is PSPACE-complete, even when the entry is restricted to hypergraphs with hyperedges of size at most 6.

The first step of the proof is to prove that this game is in PSPACE.
Lemma 7. The Avoider-Enforcer problem is in PSPACE.

Proof. Let \( H = (V, E) \) be a hypergraph. As the players are not allowed to play an already claimed vertex, any game ends after at most \( |V| \) moves. Therefore, according to Lemma 2.2 of Schaefer [27], as the game has a polynomial length and a polynomial number of moves, its winner can be computed with polynomial space.

3.1 Construction of the hypergraph

We reduce the problem 3-QBF to an Avoider-Enforcer game. This problem has been proven PSPACE-complete by Stockmeyer and Meyer [30], and we use the gaming version of this problem as it was formulated by Rahman and Watson [26]. The game is played on a quantified formula \( \varphi \) of the form \( \forall x_1 \exists x_2 \ldots \forall x_{2n-1} \exists x_{2n} \psi \), with \( \psi \) a 3-SAT formula. Alternately, two players, namely Falsifier and Satisfier, chose valuation for the variables, Falsifier for the odd variables (quantified with a \( \exists \)) and Satisfier for the even ones (quantified with a \( \forall \)). When all the variables have a valuation Satisfier wins if \( \psi \) is satisfied, otherwise, Falsifier wins.

Problem 8. 3-QBF

Input: A 3-SAT quantified formula \( \varphi \) of the form \( \forall x_1 \exists x_2 \ldots \forall x_{2n-1} \exists x_{2n} \psi \).

Output: True if and only if Satisfier has a winning strategy in the 3-QBF game on \( \varphi \)

Given a 3-QBF formula of the form defined in Problem 8, we construct a hypergraph with 10n vertices \( x_1, \overline{x}_1, \ldots, x_{2n}, \overline{x}_{2n}, u_1, u_2, \ldots, u_{6n} \).

A round in the 3-QBF formula corresponds to a step \( i \) during which Falsifier gives a valuation to \( x_{2i-1} \) and then Satisfier gives a valuation to \( x_{2i} \). In this reduction, any round corresponds to ten vertices and eight hyperedges. Four of the ten vertices are \( u_{6i-5}, u_{6i-4}, u_{6i-3}, u_{6i-2}, u_{6i-1}, u_{6i} \). The eight hyperedges are constructed as follows:

\[
\begin{align*}
A_{2i} &= (x_{2i}, \overline{x}_{2i}, u_{6i+1}, u_{6i+3}) \\
C_{6i}^+ &= (u_{6i}, u_{6i+1}, u_{6i+3}, x_{2i}) \\
C_{6i}^- &= (u_{6i-2}, u_{6i-1}, u_{6i+1}, x_{2i}) \\
C_{6i-4}^- &= (u_{6i-4}, u_{6i-3}, u_{6i-1}, x_{2i-1})
\end{align*}
\]

If some of these vertices do not exist, we still add the hyperedges, but with fewer vertices in them. For instance, \( A_{2n} = \{x_{2n}, \overline{x}_{2n}\} \). Moreover, for each clause \( F = l_1^+ \lor l_2^+ \lor l_3^+ \in \psi \) where the vertices \( l_1^+, l_2^+, l_3^+ \) and \( l_3^+ \) are literals either positive or negative, we add a hyperedge \( D_j \). For \( k = 1, 2, 3 \), if \( l_k^+ \) is a positive variable \( x_p \), then \( x_p \) is in \( D_j \), if \( l_k^- \) is a negative one \( -x_p \), then \( \overline{x}_p \) is in \( D_j \). Moreover, if \( p \) is odd, then \( u_{6p-1} \) is in \( D_j \), if \( p \) is even, then \( u_{6p+1} \) is in \( D_j \).

Finally, the CNF game \( \varphi \) is reduced to the hypergraph \( H = (V, E) \) with

\[
V = \{ \{x_i\}_{1 \leq i \leq 2n} \cup \{\overline{x}_i\}_{1 \leq i \leq 2n} \cup \{u_i\}_{1 \leq i \leq 6n} \}
\]

\[
E = \{ \{A_{2i}\}_{1 \leq i \leq n} \cup \{C_{6i}^+\}_{1 \leq i \leq 3n} \cup \{C_{6i}^-\}_{1 \leq i \leq 3n} \cup \{B_{2i-1}\}_{1 \leq i \leq n} \cup \{D_j\}_{1 \leq j \leq m} \}
\]

With this construction, we say that Avoider and Enforcer follow a legitimate order if they claim board elements in the following order for increasing \( i \):

**Legitimate order during round \( i \)**

1. Avoider starts and claims \( u_{6i-5} \).
2. Enforcer claims \( u_{6i-4} \).
3. Avoider claims \( u_{6i-3} \).
4. Enforcer claims one of \( x_{2i-1} \) or \( \overline{x}_{2i-1} \).
Avoidance games are PSPACE-Complete

5. Avoider claims the remaining vertex in \((x_{2i−1}, \overline{x_{2i−1}})\)

6. Enforcer claims \(u_{6i−2}\).

7. Avoider claims \(u_{6i−1}\).

8. Enforcer claims \(u_{6i}\).

9. Avoider claims one of \(x_{2i}\) or \(\overline{x_{2i}}\).

10. Enforcer claims the remaining vertex in \((x_{2i}, \overline{x_{2i}})\).

3.2 Sketch of the proof

To prove that, with our construction, Avoider wins the AVOIDER-ENFORCER game, if and only if Satisfier wins the QBF game, we first prove that this statement is true if the order of the moves is legitimate, as the moves will correspond to a valuation obtained in QBF.

To force the players to play in the legitimate order, the main idea of the construction is that players want to claim some vertices as late as possible. Therefore, we prove that it is always optimal to respect the legitimate order of the moves. We introduce the following three lemmas that will be proved in the next section.

- **Lemma 9.** When the game is restricted to the legitimate order, Avoider has a winning strategy in the Avoider-Enforcer game on \(H\) if and only if Satisfier has a winning strategy for the 3-QBF game on \(\varphi\).

- **Lemma 10.** If Enforcer has a winning strategy in \(H\) when the legitimate order is respected by the two players, then he has a winning strategy in \(H\).

- **Lemma 11.** If Avoider has a winning strategy in \(H\) when the legitimate order is respected by the two players, then she has a winning strategy in \(H\).

We first admit these lemmas and we prove Theorem 2.

**Proof.** First, according to Lemma 7, we know that Avoider-Enforcer is in PSPACE. We now prove the PSPACE-hardness of the problem by reduction from 3-QBF.

Let \(\varphi\) be a 3-SAT quantified boolean formula of the form described in Problem 8. Consider the hypergraph \(H\) obtained from \(\varphi\) by following the construction of Section 3.1. This construction has polynomial size. According to Lemma 9, when the order is respected, if Satisfier (Falsifier resp.) has a winning strategy in \(\varphi\), Avoider (Enforcer resp.) has a winning strategy in \(H\). Thus, according to Lemma 11 (Lemma 10 resp.), if Avoider (Enforcer resp.) has a winning strategy on \(H\) when the legitimate order is respected, she (he resp.) has one in general in \(H\). Thus, Satisfier wins on \(\varphi\) if and only if, Avoider wins on \(H\). Therefore, the AVOIDER-ENFORCER PROBLEM is PSPACE-complete.

As all the construction provides a hypergraph \(H\) in which all the hyperedges have of size at most six, the AVOIDER-ENFORCER PROBLEM is PSPACE-complete even restricted to hypergraphs in which all the hyperedges have size at most six.

3.3 Game in legitimate order

In this section, we suppose that both players follow a legitimate order of moves.

If the order of moves is legitimate, the only choices available for Avoider and Enforcer are on the vertices \(x_i\) and \(\overline{x_i}\). For each \(1 \leq i \leq 2n\), Avoider claims one of \(x_i, \overline{x_i}\) and Enforcer the other. Therefore, if both Avoider and Enforcer play one vertex in \((x_i, \overline{x_i})\), we define the underlying valuation given to \(\psi\) as the following one:
\[ X_i = \begin{cases} \text{True} & \text{if Avoider has claimed } \overline{x}_i \text{ and Enforcer has claimed } x_i \\ \text{False} & \text{if Avoider has claimed } x_i \text{ and Enforcer has claimed } \overline{x}_i \end{cases} \]

We now prove Lemma 9.

**Proof.** Consider a game played on \( H \) for which both Avoider and Enforcer respected the legitimate order through the whole game.

▷ **Claim 12.** Avoider won the game on \( H \) if and only if the formula \( \psi \) is satisfied by the underlying valuation of the \( X_i \).

Proof. Since the legitimate order is respected, Enforcer claimed all the vertices \( u_{2k} \) and thus played at least once in all the hyperedges \( C_{2k}^i \) and \( C_{2j}^i \). Moreover, for each pair of variables \( (x_i, \overline{x}_i) \), Enforcer claimed one of the vertices of the pair, and so he has claimed at least one vertex in all the hyperedges \( A_i \) and \( B_i \). Thus, the only hyperedges that could possibly be fully played by Avoider are the hyperedges \( D_j \).

Since, in the legitimate order, Avoider claimed all the vertices \( u_{2k+1} \), a hyperedge \( D_j \) corresponding to a clause \( F_j \) is fully played by Avoider if and only if she played on all the vertices \( x(l_k^j) \) for \( l_k \in F_j \), where \( x(l_k^j) = x_p \) if \( l_k^j = X_p \) and \( x(l_k^j) = \overline{x}_p \) if \( l_k^j = \overline{X}_p \). If this is the case, then this means that the formula \( \psi \) is not satisfied by the underlying valuation because the clause \( F_j \) has all its literals assigned to False. On the contrary, if the formula \( \psi \) is satisfied by the underlying valuation, then, for all clause \( F_j \), at least one of the literals in it is assigned to True and so Enforcer played at least once in each hyperedge \( D_j \).

Therefore, Avoider won the game on \( H \) if and only if \( \psi \) is satisfied. ◁

Suppose Satisfier has a winning strategy \( S \) on \( \varphi \). We define a strategy for Avoider as follows: Whenever Avoider has to play a vertex \( x_{2k} \) or \( \overline{x}_{2k} \), Avoider considers the underlying valuation given to the \( X_i \) with \( i < 2k \). Then, if Satisfier had put \( X_{2k} \) to True, she claims \( \overline{x}_{2k} \). Otherwise, she claims \( x_{2k} \). With this strategy, at the end of the game, the underlying valuation of the variables of \( H \) will the same as the valuation given by the game that Satisfier played on \( \varphi \). Since Satisfier has a winning strategy on \( \varphi \), the underlying valuation satisfies \( \psi \) and so Avoider wins the game.

Similarly, if Falsifier has a winning strategy, Enforcer can follow the strategy in such a way that at the end the valuation of variables in the game played by Falsifier correspond to the underlying valuation in \( H \). Since Falsifier wins on \( \varphi \), Enforcer wins the game on \( H \). ◀

### 4 Proofs of Lemma 10 and Lemma 11

The first part of our constructions showed that, if the legitimate order is respected, Avoider wins if and only if Satisfier wins the 3-QBF game. We now prove that, if a player has a winning strategy when the order is respected, he has one even if his opponent does not respect the order. We introduce here different sets of variables. These sets will be the main tools of the proofs of Lemma 10 and Lemma 11.

For \( i = 1 \) to \( 4n \), we define the set of vertices \( S_i \) as \( S_{4n} = \{u_{6n}, x_{2n}, \overline{x}_{2n}\} \) and for \( i < 4n \):

- if \( i = 4k \), \( S_i = \{u_{6k}, x_{2k}, \overline{x}_{2k}, u_{6k+1}\} \cup S_{i+1} \)
- if \( i = 4k - 1 \), \( S_i = \{u_{6k-2}, u_{6k-1}\} \cup S_{i+1} \)
- if \( i = 4k - 2 \), \( S_i = \{x_{2k-1}, \overline{x}_{2k-1}\} \cup S_{i+1} \)
- if \( i = 4k - 3 \), \( S_i = \{u_{6k-4}, u_{6k-3}\} \cup S_{i+1} \)
4.1 Proof of Lemma 10

We now prove Lemma 10

Proof. Suppose Enforcer has a winning strategy when the legitimate order is respected.
Consider a strategy for Enforcer in which he plays according to the legitimate order until
Avoider does not. If Avoider respects the order until all the vertices are played, by assumption,
Enforcer wins. Otherwise, the proof of the following claim provides a winning strategy for
Enforcer.

▷ Claim 13. If, during the game, Avoider plays in a set \( S_i \) in which Enforcer has not played
yet, then, after this move, Enforcer has a strategy to win the game.

Proof. The proof is by induction on \( i \).

First, notice that each \( S_i \) has an odd number of vertices and, as the total number of
vertices in \( H \) is \( 10n \), there is also an odd number of vertices outside \( S_i \). Therefore, if Avoider
plays first in an \( S_i \), Enforcer answers by playing an arbitrary vertex that is not in \( S_i \) and
considers an arbitrary pairing outside \( S_i \), which exists as there is an even number of vertices
outside \( S_i \) after his move. This way, Avoider has to be the next player to play in \( S_i \).

Base cases:
- Case \( i = 4n \): If Avoider plays first is \( S_{4n} \), by pairing the two other vertices in \( S_{4n} \), by
using Lemma 6, Enforcer can force Avoider to play another vertex in \( S_{4n} \). Hence, as
\((u_{6n}, x_{2n}), (u_{6n}, x_{2n})\) and \((x_{2n}, x_{2n})\) are three hyperedges, Avoider will claim the two
vertices of one of them and thus lose.
- Case \( i = 4n - 1 \): As shown previously, Enforcer has a strategy such that Avoider is the
next player to play in \( S_{4n-1} \). If Avoider has played at least one of her two first moves
in \( S_{4n-1} \), she has lost by the case \( i = 4n \). Otherwise, she has claimed exactly \( u_{6n-2} \) and
\( u_{6n-1} \). In this case, Enforcer claims \( u_{6n} \) and pairs \( x_{2n} \) and \( x_{2n} \) and by Lemma 6 he forces
Avoider to claim all the vertices of \( C_{6n-2}^+ \) or \( C_{6n-2}^- \).

Induction steps: Suppose that the first time Avoider does not respect the order of the
move, she plays in a set \( S_i \) for \( i \leq 4n - 2 \). If the second move of Avoider in \( S_i \) is in \( S_{i+1} \),
Enforcer wins by induction hypothesis. Thus, we can suppose that Avoider has claimed two
vertices in \( S_i \setminus S_{i+1} \). Moreover, as Enforcer has arbitrarily paired the vertices outside \( S_i \),
we describe here the strategy in \( S_i \), and Enforcer plays according to the pairing outside \( S_i \).
This strategy ensures that the moves in \( S_i \) alternate between both players.

- Case \( i = 4k \): Avoider has played twice in \( \{u_{6k}, x_{2k}, x_{2k}, u_{6k+1}\} \). At least one of
\( \{u_{6k}, x_{2k}, x_{2k}\} \) is available. Enforcer claims it. Avoider has to claim the third vert-
ex in this quadruple, otherwise, she plays first in \( S_{i+1} \) and loses by induction, and
necessarily one of the three vertices she has claimed is \( u_{6k+1} \). Enforcer claims \( u_{6k+2} \).
Avoider either plays first in \( S_{i+2} \) and loses by induction hypothesis, or claims \( u_{6k+3} \). At
this moment, Avoider has played on the vertices \( u_{6k+1} \) and \( u_{6k+3} \), and two of the vertices of
\( \{u_{6k}, x_{2k}, x_{2k}\} \). So she has completed one of the hyperedges \( C_{6k}^+ = (u_{6k}, u_{6k+1}, u_{6k+3}, x_{2k}) \),
\( C_{6k}^- = (u_{6k}, u_{6k+1}, u_{6k+3}, x_{2k}) \) or \( A_{2k} = (x_{2k}, x_{2k}, u_{6k+1}, u_{6k+3}) \).
- Case \( i = 4k - 1 \): Avoider has claimed \( u_{6k-2} \) and \( u_{6k-1} \). Enforcer claims \( u_{6k} \). Avoider
has to play on vertex in \( \{x_{2k}, x_{2k}, u_{6k+1}\} \), (otherwise she plays first in \( S_{i+2} \) and loses by
induction). Enforcer claims either \( x_{2k} \) or \( x_{2k} \), as at least one of them is available. If Avoider
plays a vertex in \( S_{i+2} \) she losses by induction. So she has to play the last vertex available
in \( S_i \setminus S_{i+1} \). With this strategy, Avoider has necessarily claimed \( u_{6k+1} \) and one of \( x_{2k} \) and
\( x_{2k} \). Thus, she has played all the vertices of either \( C_{6k-2}^+ = (u_{6k-2}, u_{6k-1}, u_{6k+1}, x_{2k}) \) or
\( C_{6k-2}^- = (u_{6k-2}, u_{6k-1}, u_{6k+1}, x_{2k}) \).
Case $i = 4k - 2$: Avoider has claimed $x_{2k-1}$ and $x'_{2k-1}$. Enforcer claims $u_{6k-2}$. Either Avoider plays first in $S_{i+2}$ and loses by induction, or she claims $u_{6k-1}$, the last available vertex in $S_{i+1}$ and loses by having played all the vertices in $B_{2k-1} = (x_{2k-1}, x'_{2k-1}, u_{6k-1})$.

Case $i = 4k - 3$: Avoider has claimed $u_{6k-4}$ and $u_{6k-3}$. Enforcer claims $x_{2k-1}$. Avoider has to claim $x'_{2k-1}$, otherwise she plays first in $S_{i+2}$ and loses by induction. Then Enforcer claims $u_{6k-2}$. If Avoider plays in $S_{i+3}$ she loses by induction. The last vertex available in $S_i \setminus S_{i+3}$ is $u_{6k-1}$, and if Avoider claims it, she loses by playing all the vertices $C_{6k-4} = (u_{6k-4}, u_{6k-3}, u_{6k-1}, x'_{2k-1})$.

By applying this induction, at any moment of the game, if Avoider plays first in a set $S_i$, she looses.

Finally, if Enforcer has played according to the legitimate order, at any moment of the game, Avoider has to play in a set $S_i$ in which Enforcer has already played. Therefore, she has to respect the order of the moves. The only moment when she can change this order is by claiming $u_{6k+1}$ instead of one of the vertices $x_{2k}$, $x'_{2k}$. But if she does so, Enforcer can claim one of them, for instance $x_{2k}$, and Avoider will be forced to claim $x'_{2k}$. If this happens, everything happens as if Avoider has claimed $x'_{2k}$ first and $u_{6k+1}$ after. Since these moves could have occurred in the legitimate order, the strategy can then continue as if the order has been respected.

To conclude, if Enforcer has a winning strategy when the legitimate order is respected, Avoider has a winning strategy in $H$ even without this restriction.

4.2 Proof of Lemma 11

In this section, we prove that if Avoider has a winning strategy when the legitimate order is respected, she also has one if Enforcer does not respect the order. The main idea of the strategy is to respect the order, and if Enforcer does not respect the order, Avoider has a pairing strategy to force Enforcer to claim a vertex in some pair $(x_i, x_j)$, or the odd $u_j$ that follows them. By construction, any hyperedge containing a vertex $x_i$ or $x_j$ also contains the next odd vertex $u_j$ in the legitimate order, and this will prove that whenever Enforcer does not respect the order, it benefits Avoider. We now will prove Lemma 11.

**Proof.** Suppose Avoider has a winning strategy when the legitimate order is respected. We now describe a winning strategy for Avoider in the general case.

While Enforcer respects the legitimate order, Avoider also respects it. Suppose that at some moment of the game, Enforcer does not respect the legitimate order. Denote by $y_A$ the vertex he would have played according to the legitimate order, and by $y_E$ the vertex he has claimed instead. If $y_A$ is a vertex $x_{2i}$ or $x'_{2i}$, Avoider pairs it with $u_{6i+1}$ and continues as if Enforcer had to play $u_{6i+2}$. If this is the case, consider $y_A = u_{6i+2}$. Note that, according to Lemma 5, we can suppose that $y_E$ is not a vertex $u_j$ with $j$ odd. Indeed, for each vertex $u_{2k+1}$, the hyperedges that contain the previous vertex in the legitimate order (if this vertex is a vertex $x_j$ or $x'_{j}$ this is true for either of them) also contain $u_{2k+1}$. As any hyperedge containing $x_{2i}$ or $x'_{2i}$ also contains $u_{6i+1}$ which is the next vertex Avoider should have played according to the legitimate order, it benefits her if Enforcer finally claims $u_{6i+1}$ instead of $x_{2i}$ or $x'_{2i}$ according to Lemma 5.

Denote by $k$ the smallest integer such that $y_E \notin S_k$, and by $k'$ the largest integer such that $y_A \in S_{k'}$. We consider $k = 4n + 1$ if $y_E \in S_{4n}$, with $S_{4n+1} = \emptyset$. Note that all the vertices outside $S_k$ have already been played or are paired, and that $S_k \setminus S_i$ is then the set of vertices perturbed by the move of Enforcer. As all the sets $S_i$ have an odd number
of vertices, we know that the number of remaining vertices outside \( S_k \) is odd, as an even number of moves have been played in it. By assumption, Avoider was following a winning strategy for the legitimate order. She can then consider an arbitrary sequence of moves for Enforcer following the legitimate order and her answers according to her strategy until all the vertices in \( S_k \setminus \) \( S_k \) are played. According to these moves, we will denote by \( x_j^E \) the vertex among \( (x_j, y_j) \) claimed by Enforcer and by \( x_j^A \) the vertex played by Avoider.

Avoider claims \( y_A \), the vertex that Enforcer should have claimed according to the legitimate order, and will consider one strategy in \( S_k \) and another one outside \( S_k \):

\( \Rightarrow \) In \( H \setminus S_k \), she plays according to a pairing strategy, that is presented in the next paragraph.

\( \Rightarrow \) In \( S_k \), Avoider considers the strategy she would have played if all the vertices outside \( S_k \) were played according to the legitimate order, with the vertices \( x_j^E \) claimed by Enforcer and the vertices \( x_j^A \) claimed by Avoider.

The pairing we define is the following one: \((u_{6i-4}, x_{2i-1}^A), (u_{6i-3}, u_{6i-6}), (x_{2i-1}^E, u_{6i-1})\), \((u_{6i-2}, x_{2i}^A), (u_{6i+1}, x_{2i}^E)\). This pairing concerns all the vertices that have to be played after \( y_A \) in the legitimate order that are not in \( S_k \), and we consider only pairs containing at least one vertex outside \( S_k \). Note that, by construction, exactly one vertex of this pairing is already played, and exactly one paired with a vertex in \( S_k \). Therefore, to make the pairing contain only vertices not played and outside \( S_k \), some modifications are done. These modifications are presented in Figure 1. By applying Lemma 6, Avoider can ensure that Enforcer plays at least one in each of these pairs.

<table>
<thead>
<tr>
<th>( y_A )</th>
<th>changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_{6i-4} )</td>
<td>( u_{6i-3} \leftrightarrow x_{2i-1}^A )</td>
</tr>
<tr>
<td>( x_{2i-1} ) or ( x_{2i-1}^E )</td>
<td>( x_{2i-1}^A \leftrightarrow u_{6i-1} )</td>
</tr>
<tr>
<td>( u_{6i-2} )</td>
<td>( u_{6i-1} \leftrightarrow x_{2i}^A )</td>
</tr>
<tr>
<td>( u_{6i} )</td>
<td>( u_{6i+3} \leftrightarrow x_{2i}^E )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( y_E )</th>
<th>changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_{6i-4} )</td>
<td>no changes</td>
</tr>
<tr>
<td>( x_{2i-1} ) or ( x_{2i-1}^E )</td>
<td>( x_{2i-1}^A \leftrightarrow u_{6i-4} )</td>
</tr>
<tr>
<td>( u_{6i-2} )</td>
<td>no changes</td>
</tr>
<tr>
<td>( u_{6i} )</td>
<td>no changes</td>
</tr>
<tr>
<td>( x_{2i} ) or ( x_{2i}^E )</td>
<td>( x_{2i}^A \leftrightarrow u_{6i+1}, u_{6i-2} \leftrightarrow u_{6i} )</td>
</tr>
</tbody>
</table>

**Figure 1** Changes of the matching. \( x_j^E \) refers to the variable in \( \{x_k, y_k\} \) that has not been played, arrows represent new pairs of vertices in the matching.

**Claim 14.** The pairing strategy ensures that Enforcer plays at least once in each hyperedge \( A_i, B_i \) or \( C_i \) containing all their vertices in \( S_k \) and at least one outside \( S_k \).

**Proof.** First, if the hyperedge contains no vertex whose pairing has been modified because of their belonging to \( y_A \) or \( y_E \), it contains two paired vertices. We show in bold text the paired vertices:

\[
A_{2i} = (x_{2i}^A, x_{2i}^E, u_{6i+1}, u_{6i+3}) \quad C_{6i-2} = (u_{6i-2}, u_{6i-1}, u_{6i+1}, x_{2i}^E) \\
C_{6i} = (u_{6i}, u_{6i+1}, u_{6i+3}, x_{2i}^A) \quad B_{2i-1} = (x_{2i-1}^A, x_{2i-1}^E, u_{6i-1}) \\
C_{6i}^E = (u_{6i}, u_{6i+1}, u_{6i+3}, x_{2i}^E) \quad C_{6i-2} = (u_{6i-2}, u_{6i-1}, u_{6i+1}, x_{2i}^A) \\
C_{6i} = (u, u_{6i-1}, u_{6i+3}, x_{2i}^A) \quad C_{6i-4} = (u_{6i-4}, u_{6i-3}, u_{6i-1}, x_{2i}^A) \\
C_{6i}^E = (u_{6i-4}, u_{6i-3}, u_{6i-1}, x_{2i}^E) \\
C_{6i-4} = (u_{6i-4}, u_{6i-3}, u_{6i-1}, x_{2i}^E)
\]

For the first hyperedges of the matching, there are two paired vertices.

Recall first that if Enforcer was supposed to play in \( \{x_{2i}, x_{2i}^E\} \), Avoider pairs this vertex with \( u_{6i+1} \) and considers \( y_A = u_{6i+2} \). The only one hyperedge among the \( A_i, B_i \) and \( C_i \) that was concerned with this change is \( A_{2i} \), in which two vertices are now paired. In any other case, the following vertices are paired:

\( \Rightarrow \) If \( y_A = u_{6i-4} \), only the hyperedges \( C_{6i-4}^E \) and \( C_{6i-4}^A \) are concerned by the changes. In the former \( x_{2i-1}^E \) is paired with \( u_{6i-1} \), in the latter \( x_{2i-1}^A \) is paired with \( u_{6i-3} \).
If \( y_A \in \{x_{2i-1}, x_{2i-1}\} \), the only hyperedges concerned by the change is \( B_{2i-1} \). In it, the other vertex in \( \{x_{2i-1}, x_{2i-1}\} \) is paired with \( u_{6i-1} \).

If \( y_A = u_{6i-2} \), only the hyperedges \( C_{6i-2}^E \) and \( C_{6i-2}^A \) are concerned by the changes. In the former \( x_{2i}^E \) is paired with \( u_{6i+1} \), in the latter \( x_{2i}^A \) is paired with \( u_{6i-1} \).

If \( y_A = u_{6i} \), only the hyperedges \( C_{6i}^E \) and \( C_{6i}^A \) are concerned by the changes. In the former \( x_{2i}^E \) is paired with \( u_{6i+1} \), in the latter \( x_{2i}^A \) is paired with \( u_{6i+3} \).

For the last hyperedges that contain vertices of the matching, the following happens:

If \( y_E = u_{6i-4} \), the pairing stops at \( u_{6i-3} \). The only two hyperedges that contain at least one vertex in \( S_k \) and one vertex outside \( S_k \) are \( C_{6i-4}^+ \) and \( C_{6i-4}^- \), in which Enforcer has claimed \( y_E \).

If \( y_E = x_{2i-1} \) or \( x_{2i-1}^E \), the pairing stops after the second vertex in \( \{x_{2i-1}, x_{2i-1}^E\} \). The only one hyperedge containing at least one vertex in \( S_k \) and one outside \( S_k \) is \( B_{2i-1} \), in which Enforcer has already claimed \( y_E \).

If \( y_E = u_{6i-2} \), the pairing stops at \( u_{6i-1} \). The only two hyperedges that contain at least one vertex in \( S_k \) and one vertex outside \( S_k \) are \( C_{6i-2}^+ \) and \( C_{6i-2}^- \), in which Enforcer has claimed \( y_E \).

If \( y_E = u_{6i} \), the pairing stops at \( u_{6i+1} \). The three hyperedges that contain both vertices in \( S_k \) and vertices outside \( S_k \) are \( C_{6i}^+, C_{6i}^- \) and \( A_{2i} \). In \( C_{6i}^+, C_{6i}^- \), Enforcer has claimed \( y_E \), and in \( A_{2i} \), Enforcer will play one of \( x_{2i}^E \) or \( u_{6i+1} \) as these two vertices are paired together.

If \( y_E = x_{2i} \) or \( x_{2i}^E \), the pairing stops at \( u_{6i+1} \). The three hyperedges that contain vertices inside \( S_k \) and outside \( S_k \) are \( C_{6i}^+, C_{6i}^- \) and \( A_{2i} \). As the second vertex in \( \{x_{2i}, x_{2i}^E\} \) is paired with \( u_{6i+1} \), either Enforcer has claimed both \( x_{2i} \) and \( x_{2i}^E \), and any of these three hyperedges contains at least one of them; or Enforcer has claimed \( u_{6i+1} \) which is in these three hyperedges.

If the pairing stops because it goes until the end (i.e. \( k = 4n + 1 \)), one vertex is not paired. According to Lemma 6, as Enforcer plays the last move in \( H \), Avoider can force him to play it and still play once in each pair of the pairing.

Finally, in any hyperedge \( A_i, B_i \) or \( C_i \) containing at least one vertex of the matching, Enforcer has played at least one vertex.

Now, we can prove that the strategy we defined for Avoider is a winning strategy. In all the hyperedges \( A_i, B_i \) or \( C_i \), Enforcer played at least once. Indeed, if Enforcer has respected the order until he plays in one of these hyperedges, there is nothing to do. Otherwise, by Claim 14, Avoider can force Enforcer to play in it as this hyperedge is considered in a set of hyperedges in which Enforcer has not respected the order, and thus contains two paired vertices.

In the hyperedges \( D_j \), as the strategy in the legitimate order is a winning strategy, it forces at least one vertex \( x_i \) or \( x_i^E \) to be claimed by Enforcer in \( D_j \) (since all the vertices \( u_k \) of odd indices are played by Avoider in the legitimate order). Then, by construction, if when this vertex has to be claimed the order was respected, Avoider has played it. If the order was not respected, then Avoider has paired this vertex with the next vertex \( u_k \) of odd index, forcing Enforcer to play one of them. In both cases, Enforcer has played in \( D_j \).
Avoidance games are PSPACE-Complete

5 Uniform hypergraphs and applications

The construction provided in Section 3.1 provided a hypergraph in which any hyperedge have size at most six. We prove here that the PSPACE-hardness can be generalized to uniform hypergraphs.

5.1 From 6-hypergraphs to 6-uniform hypergraphs

A hypergraph \( H = (V,E) \) is a \( k \)-hypergraph if any hyperedge \( e \in E \) has size at most \( k \). It is said to be \( k \)-uniform if any hyperedge \( e \in E \) has size exactly \( k \).

Lemma 15. Let \( H = (V,E) \) be a \( k \)-hypergraph. Let \( m = \min_{e \in E} |e| \). If \( m < k \), there exists a \( k \)-hypergraph \( H' = (V',E') \) where \( \min_{e \in E'} |e| = m + 1 \), having \( |E'| \leq 2|E| \) and \( |V'| \leq |V| + 2 \) such that Avoider has a winning strategy in the Avoider-Enforcer game on \( H \) if and only if she has one in \( H' \).

Proof. Let \( H = (V,E) \) be a \( k \)-hypergraph. Let \( m = \min_{e \in E} |e| \). We define \( H' = (V',E') \) as follows. We start from \( V' = V \). We add two vertices \( \{a_1,a_2\} \) in \( V' \). For each hyperedge \( e \in E \), we add hyperedges in \( E' \) as follows:

- If \( |e| > m \), we add a copy of \( e \) in \( E' \).
- If \( |e| = m \), we add two hyperedges \( e_1 = e \cup \{a_1\} \) and \( e_2 = e \cup \{a_2\} \) in \( E' \).

We have \( |V'| = |V| + 2 \), \( |E'| \leq 2|E| \) and \( \min_{e \in E'} |e| = m + 1 \).

Now, if Avoider (Enforcer resp.) had a winning strategy \( S \) in \( E \), we can define a strategy \( S' \) in \( E' \) as follows:

- If the opponent plays a vertex in \( V \), or if it is the first move of the player, play as in \( S \).
- If the opponent plays a vertex in \( \{a_1,a_2\} \), or if there is no vertex in \( V \) available, play an available vertex in \( \{a_1,a_2\} \).

Following this strategy, Avoider (Enforcer resp.) has played exactly the same vertices in \( H' \) as he (she resp.) would have played in \( H \) according to \( S \) with, in addition, exactly one of \( \{a_1,a_2\} \).

Therefore, if Avoider had a winning strategy in \( H \), then for each \( e \in E \), there exists one vertex \( v \in e \), that Enforcer has played. Let \( e' \in E' \) be a hyperedge. If \( e' \) is a copy of some hyperedge \( e \in E \), then Enforcer has played in it, as he would have played in \( e \) according to \( S \).

If \( e' \) is a hyperedge \( e_1 \) or \( e_2 \) created from a hyperedge \( e \in E \), as Enforcer would have played in \( e \) according to \( S \), he has played the same vertex in \( e' \).

If Enforcer had a winning strategy in \( H \), following this strategy, there exists a hyperedge \( e \in E \) in which Avoider has claimed all the vertices. If \( |e| \geq m + 1 \), Avoider has also played all the vertices of \( e \in E' \), so Enforcer has won. If \( |e| = m \), as the strategy \( S' \) forces Avoider to play at least one of \( \{a_1,a_2\} \), suppose without loss of generality that she has claimed \( a_1 \). Then, she has claimed \( a_1 \) and all the vertices of \( e \), so she has filled up the edge \( e_1 \). Therefore, this strategy is a winning strategy for Enforcer.

Finally, \( H' \) has the same outcome as \( H \) and \( \min_{e \in E'} |e| = m + 1 \).

Corollary 16. Avoider-Enforcer is PSPACE-complete even restricted to 6-uniform hypergraphs

Proof. The construction provided in the proof of Theorem 2 builds a 6-hypergraphs in which all hyperedges have size at least two. Therefore, by applying Lemma 15 four times, with \( m = 2,3,4,5 \), we obtain a hypergraph having at most eight more vertices and eight times
more hyperedges. Thus, this construction is still polynomial and the hypergraph obtained is 6-uniform.

5.2 Avoider-Avoider games

We prove Corollary 4, i.e. that Avoider-Avoider games are PSPACE-complete, even restricted to 7-uniform hypergraphs.

Proof. Consider the construction provided in the proof of Corollary 16. Consider $H'$ the hypergraph obtained by adding a vertex $v_0$ in $H$ and adding it in all the hyperedges of $H$. Note that, as any hyperedge of $H$ has size six, any hyperedge of $H'$ have size exactly seven. According to Lemma 5 (note that the result also apply to Avoider-Avoider games), both players have an optimal strategy in which $v_0$ will be played last, and as the graph has an odd number of vertices, the first player will play it. Therefore, the second player cannot fill up a hyperedge and plays as Enforcer would in the Avoider-Enforcer game. By applying the same strategy as in Avoider-Enforcer, if Avoider wins in Avoider-Enforcer, the game ends by a draw, otherwise the second player wins.

References

Avoidance games are PSPACE-Complete

14


Several games have been proven to be PSPACE-complete in the Maker-Breaker convention, thanks to the proofs of Schaefer or of Rahman and Watson. Due to the similarities between the two conventions, some reductions may be adapted to prove that these games are PSPACE-complete in the Avoider-Enforcer convention. In particular, we prove in this section that the domination game and the vertex $H$-Game are PSPACE-complete in Avoider-Enforcer convention.

### A.1 Avoider-Enforcer domination game.

The Maker-Breaker domination game was introduced by Duchêne et al. in 2020 [10] and follows the study of domination games on graphs, which were investigated since 2002 [1, 7]. In Maker-Breaker, two players, namely Dominator and Staller alternately claim an unclaimed vertex of the graph, Dominator wins if he manages to claim all the vertices from a dominating set. Otherwise, Staller wins. They proved that determining whether Dominator or Staller has a winning strategy is PSPACE-complete using a reduction from the general Maker-Breaker game. The Avoider-Enforcer domination game can be similarly defined, with Anti-Staller winning if Anti-Dominator claims a dominating set and Anti-Dominator winning otherwise. We prove here that determining the winner of the Avoider-Enforcer domination game is PSPACE-complete. Note that the proof is very similar to the reduction from Maker-Breaker games to Maker-Breaker domination game.

▶ **Problem 17. Avoider-Enforcer domination game**

Input: A graph $G$ 

Output: True if and only if Anti-Dominator wins the Avoider-Enforcer domination game on $G$.

▶ **Theorem 18.** The Avoider-Enforcer domination game problem is PSPACE-complete

Proof. First, Avoider-Enforcer domination game is in PSPACE, as the number of moves in a game is the number of vertices, and as determining if a set is a dominating set or not can be done in polynomial time, the game is in PSPACE.

Let $H = (V_H, E_H)$ be a hypergraph. Without loss of a generality, we can suppose that each vertex is in at least one hyperedge, otherwise the vertices in no hyperedge will be played first according to Lemma 5, and we can just remove them, up to change the first player to go. We construct the following graph $G = (V, E)$ as follows:

1. For each vertex $u_i$ in $V_H$, we add a vertex $v_i$ in $V$.
2. For each hyperedge $C$ in $E_H$, we add two vertices $v^1_C$ and $v^2_C$ in $V$.
3. If a vertex $u_i$ of $V_H$ belongs to a hyperedge $C$ of $E_H$, we add the edges $v_i v^1_C$ and $v_i v^2_C$ to $E$.

Note that the graph created here is bipartite

Suppose Avoider (Enforcer resp.) has a winning strategy $S$ in $H$. We define a strategy $S'$ for Anti-Staller (Anti-Dominator resp.) in $G$ as follows.

1. If Avoider (Enforcer resp.) plays the first move in $H$, he claims first a vertex $v_i$ such that $u_i$ is the first vertex claimed in $S$.
2. If the opponent claims a vertex $v_j$, he claims a vertex $v_j$ such that $u_j$ is the answer to the vertex $u_i$ in $S$. 

Avoidance games are PSPACE-Complete

If a player plays a vertex $v^k_C$ for $k \in \{1, 2\}$, he claims the vertex $v^{k'}_C$ for $k' \neq k \in \{1, 2\}$.

Now if Avoider had a winning strategy in $H$, by applying the strategy $S'$, for any vertex $v^1_C$, Anti-Staller has not played all the $v_j$s adjacent to it. Therefore, Anti-Dominator has claimed one of them and all the $v^2_C$s are dominated. Moreover, all the vertex $u_j$s are in at least one edge $C$ of $H$, and thus $v_j$ is dominated by Anti-Dominator, as she has played one of $(v^1_C, v^2_C)$. Finally, Anti-Dominator has dominated the graph and Anti-Staller has won.

Reciprocally, if Enforcer had a winning strategy in $H$, by applying the strategy $S'$, Anti-Dominator knows that there exists a pair of vertices $(v^1_C, v^2_C)$, such that Anti-Staller has played all the $v_j$s adjacent to them. As Anti-Staller has played one of them, and all its neighbors, this vertex is not dominated. Therefore, Anti-Dominator has won. ▶

In Figure 2 we provide an example of reduction. Note that the construction provided is exactly the same as the one used to prove the PSPACE-completeness in Maker-Breaker in [10].

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure2.png}
\caption{Reduction from Avoider-Enforcer game to Avoider-Enforcer domination game.}
\end{figure}

Remark 19. Up to add all the edges between the $v_i$s, the game is still PSPACE-complete on split graphs.

A.2 Avoider-Enforcer vertex $H$-Game

The vertex $H$-Game has been introduced by Kronenberg, Mond and Naor in [21] on random graphs. It is presented in several conventions, but we will focus here on the Avoider-Enforcer one. The game is played as follows:

Let $H$ be a graph. Avoider and Enforcer play on the vertex set of another graph $G$. Alternately, Avoider and Enforcer claim an unclaimed vertex of $G$. Avoider wins if the set of vertices she has claimed do not contain $H$ as a subgraph (not necessarily induced). Otherwise, Enforcer wins.

\begin{problem}{Avoider-Enforcer vertex $H$-Game}
Input: A graph $G$
Output: True if and only if Avoider wins the Avoider-Enforcer vertex $H$-Game played on $G$.
\end{problem}

We first need to define some graphs and operations that will be helpful to describe the graphs of this section:

We will denote by $I_k$ the graph being an independent set of size $k$, i.e. containing $k$ vertices and no edge.
If $G$ and $H$ are two graphs, we denote by $G \bowtie H$ their join, i.e. if $G = (V_G, E_G)$ and
$H = (V_H, E_H)$, we have $G \bowtie H = (V, E)$ with $V = V_G \cup V_H$ and $E = E_G \cup E_H \cup

If $G$ and $H$ are two graphs, we denote by $G \boxtimes H$ their strong product, i.e. if $G = (V_G, E_G)$
and $H = (V_H, E_H)$, we have $G \boxtimes H = (V, E)$ with $V = \{x_{u,v} | u \in V_G, v \in V_H\}$ and
$E = \{(x_{u_1,v_1}, x_{u_2,v_2}) | (u_1 = u_2 \text{ or } (u_1, u_2) \in E_G) \text{ and } (v_1 = v_2 \text{ or } (v_1, v_2) \in E_H)\}$.

Remark that for any graph $G, G \boxtimes P_2$ (where $P_2$ design the path of length 2) is obtained
by taking two copies of $G$ and connecting each vertex to its copy and its copy’s neighbors.

We prove here that determining the winner of the Avoider-Enforcer vertex $H$-Game is a
PSPACE-complete problem for several graphs $H$.

\textbf{Theorem 21.} Let $H_0$ be a graph containing at least one edge or at least 6 vertices, and
let $k \geq 6$. Consider $H = I_k \bowtie H_0$. The Avoider-Enforcer vertex $H$-game problem is
PSPACE-complete.

Note that complete bipartite graphs $K_{n,m}$, with $n, m \geq 6$, are of this type. Indeed,
$K_{n,m} = I_k \bowtie H_0$ for $H_0 = I_m$ and $k = n$.

\textbf{Proof.} First, the Avoider-Enforcer vertex $H$-game is in PSPACE. Indeed, as it is a positional
game, if $G = (V, E)$ is a graph, the game ends after at most $|V|$ moves. After that, determining
whether a graph $H$ is a subgraph of a graph $G$ can be done in polynomial space.

We do our reduction from AVOIDER-ENFORCER on 6-uniform hypergraphs (proven to be
PSPACE-complete in Corollary 16). Let $H_0$ be a graph containing at least one edge or at
least 6 vertices, and let $k \geq 6$. Let $H = I_k \bowtie H_0$. To avoid confusion while describing the
strategies in the two games, we will call the players of the Avoider-Enforcer vertex $H$-Game
Alice and Bob, with Alice avoiding creating a subgraph $H$ and Bob forcing her to create one.

Let $H' = (V', E')$ be a 6-uniform hypergraph. Let $H_0^P$ be the strong product $H_0 \boxtimes P_2$.
We build $G = (V, E)$ an instance of the Avoider-Enforcer vertex $H$-Game as follows:

**Step 1:** For any vertex $v'_i \in V'$, we add a vertex $v_i \in V$.

**Step 2:** For any hyperedge $C \in E'$, we add $2(k-6)$ vertices $v_1^C, \ldots, v_{2(k-6)}^C$ (note that
if $k = 6$, no vertex is added during this step).

**Step 3:** For any hyperedge $C \in E'$, we add a copy $H_0^C$ of the graph $H_0$ in $G$, and we
connect any vertex of $H_0^C$ to all the vertices $v_i$ such that $v'_i \in C$ and to all the vertices
$v_j^C$ for $1 \leq j \leq 2k - 6$.

\textbf{Claim 22.} If Avoider has a winning strategy in $H'$, Alice has a winning strategy in $G$.

\textbf{Proof.} Suppose Avoider has a winning strategy $S'$ in $H'$. We define Alice’s strategy $S$ in $G$
as follows:

She starts by claiming the vertex $v_i$, corresponding to the vertex $v'_i$ that Avoider would
have claimed in $H'$ according to $S'$.

If Bob claims a vertex $v_j$, she answers with the vertex $v_j^C$ corresponding to the vertex $v'_j$
that Avoider would have claimed by $S'$ in $H'$ if Enforcer has claimed $v'_j$.

In $H_0^C$, as it is a strong product $H_0 \boxtimes P_2$, Alice considers the pairing between any vertex
of $H_0^C$ and its copy in the strong product. If Bob plays in one pair of this pairing, she
claims the second vertex of this pair.

For any hyperedge $C$ in $H'$, Alice considers the set of vertices $v_j^C$ for $1 \leq j \leq 2k - 6$. If
Bob claims one of them, she also claims one. As there is an even number of them, this is
always possible.
At the end of the game, by the matching strategy, for each hyperedge \( C \) in \( H' \), Alice will have played exactly a copy of \( H_0 \) in each \( H_C^j \), exactly \( k - 6 \) vertices among the \( v_i^j \) for \( 1 \leq j \leq 2(k - 6) \), and the vertices \( v_i \) corresponding to the \( v_i^j \) that Avoider would have played according to \( S' \) in \( H' \).

Now, consider any copy \( H_1 \) of \( H \) in \( G \). Suppose that Alice has played all the vertices of \( H_1 \).

Suppose that \( H_0 \) has at least one edge. We first prove that \( H_1 \) cannot contain two vertices \( v_C \) and \( v_C' \) for \( C \neq C' \) that have been created by the Steps 2 or Step 3 of our construction. Suppose it does. Consider a decomposition of \( H_1 = I_k \bowtie H_0^1 \). By construction, the \( v_C \)'s and the \( v_C' \)'s are not adjacent. Therefore, as these two components are fully connected one to the other, they must either both in \( I_k \) or both in \( H_0^1 \).

Let \( e = (u_1, u_2) \) be an edge of \( H_0^1 \). \( v_C \) and \( v_C' \) cannot be both adjacent to \( u_1 \), otherwise, by construction, we would have \( C = C' \). Therefore, as only vertices created during Step 1 can be adjacent to both \( v_C \) and \( v_C' \), any vertex in \( I_k \) must be a vertex \( v_i \). Which is not possible otherwise, Alice would have play \( k \geq 6 \) vertices \( v_i \) adjacent to a same \( v_C \), which means that, according to \( S' \) she would have played \( k \geq 6 \) vertices in the same hyperedge in \( H' \), which contradicts the fact that \( S' \) was a winning strategy for Avoider in \( H' \).

Now, as all the vertices of \( H_1 \) are either \( v_C \)'s or were created by considering the same hyperedge \( C \) and \( |H_1| = |H_0| + k \), by the pairing, we know that exactly 6 of them are \( v_C \)'s. As there are no edges between the \( v_C \)'s, they must all be on the same side of the join, and therefore, they are all connected to a same vertex. By construction, this is only the case if these six vertices are in a same hyperedge of \( H' \) which contradicts that \( S' \) was a winning strategy for Avoider in \( H' \), as these six vertices would then form a hyperedge.

Suppose now that \( H_0 \) has no edges and has \( k' \geq 6 \) vertices.

This means that \( H \) can be written \( I_k \bowtie I_{k'} \) for \( k, k' \geq 6 \) (note that \( H \) is a complete bipartite graph). Once again, consider two vertices \( v_C \) and \( v_C' \) for \( C \neq C' \) in \( H_1 \). As they cannot be adjacent, they must be both in \( I_k \) or both in \( I_{k'} \). Thus, \( v_C \) and \( v_C' \) have \( \min(k, k') \geq 6 \) common neighbors. This implies that, if \( C \neq C' \), at least one of their common neighbor is not a vertex \( v_i \) created during Step 1, otherwise Alice would have played six vertices in the same hyperedge of \( H' \). This is not possible by construction. So once again, \( H_1 \) cannot contain \( v_C \) and \( v_C' \) created from different hyperedges \( C \) and \( C' \) in \( H' \). Now, if Alice has played all the vertices of \( H_1 \), by construction as \( |H_1| = k + k' \), and as her pairing strategy ensures her to play \( k - 6 \) vertices created during step 2 and \( k' \) created during step 3, necessarily, she has played six vertices \( v_i \) creating during step 1. As there are no edges between these six vertices, they must all be in the same independent set \( I_k \) or \( I_{k'} \). Thus, they have a common neighbor. This common neighbor must then be a vertex \( v_C \) creating during step 3 as only them are connected to the \( v_C \)'s. Finally, these six vertices corresponds to six vertices \( v_i \)'s that are in the same hyperedge \( C \) of \( H' \). Once again, this contradicts the fact that \( S' \) was a winning strategy for Avoider in \( H' \).

\[ \triangleright \]

\textbf{Claim 23.} If Enforcer has a winning strategy \( S' \) in \( H' \), Bob has a winning strategy in \( G \).

\textbf{Proof.} Let \( S' \) be a winning strategy for Enforcer in \( H' \). We consider a strategy \( S \) for Bob in \( G \) as follows:

- If Alice claims a vertex \( v_i \), he answers with the vertex \( v_j \) that corresponds to the vertex \( v_j^i \) that Enforcer would have claimed in response to \( v_i^j \) in \( S' \).
In $H_0^C$, as it is a strong product $H_0 \boxtimes P_2$, Bob considers the pairing between any vertex and its copy in the strong product. If Alice plays in one pair of the pairing, he plays the second vertex of the pair.

For any edge $C \in H'$, Bob pairs the vertices $v_j^C$ for $1 \leq j \leq 2k - 6$. If Alice claims one of them, he claims one of them too.

If at a certain moment of the game, it is Bob’s turn and the remaining vertices are all in some $H_0^C$ or vertices $v_j^C$’s, he applies the pairing strategy, so that Alice plays once in any pair of the matching by Lemma 6.

Consider the graph at the end of the game. As $S$ was a winning strategy for Enforcer in $H'$, there exists a hyperedge $C \in H'$ in which Avoider has claimed the six vertices. Up to a renaming of the vertices, denote by $v_1', \ldots, v_6'$ be these six vertices. Alice has then claimed $v_1, \ldots, v_6$ in $G$. According to the pairing strategy, Bob knows that Alice will play exactly on $k - 6$ vertices from the $v_j^C$, denote them $v_7, \ldots, v_k$, and exactly one copy $H_1$ of $H_0$ from the vertices of $H_0^C$. Now, by construction, the vertices $v_1, \ldots, v_k$ are a stable set and all the edges exist between any $v_i$ ($1 \leq i \leq k$) and any vertex $v$ of $H_1$. Thus, the subgraph formed by these vertices, which were all played by Alice, is isomorphic to $I_k \bowtie H_0 = H$. Thus, Bob has won.

Finally, Alice has a winning strategy in the Avoider-Enforcer vertex $H$-Game played on $G$ if and only if Avoider has one in the Avoider-Enforcer game played on $H'$, and determining the winner of the Avoider-Enforcer vertex $H$-Game is PSPACE-complete.