



HAL
open science

Complexity of Maker-Breaker Games on Edge Sets of Graphs

Eric Duchêne, Valentin Gledel, Fionn Mc Inerney, Nicolas Nisse, Nacim Oijid,
Aline Parreau, Miloš Stojaković

► **To cite this version:**

Eric Duchêne, Valentin Gledel, Fionn Mc Inerney, Nicolas Nisse, Nacim Oijid, et al.. Complexity of Maker-Breaker Games on Edge Sets of Graphs. *Discrete Applied Mathematics*, 2025, 361, pp.502-522. 10.1016/j.dam.2024.11.012 . hal-03993275

HAL Id: hal-03993275

<https://hal.science/hal-03993275v1>

Submitted on 16 Feb 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.



Distributed under a Creative Commons Attribution 4.0 International License

Complexity of Maker-Breaker Games on Edge Sets of Graphs*

Eric Duchêne¹, Valentin Gledel^{†2}, Fionn Mc Inerney³, Nicolas Nisse⁴, Nacim Oijid¹,
Aline Parreau¹, and Miloš Stojaković^{‡5}

¹Univ Lyon, CNRS, INSA Lyon, UCBL, Centrale Lyon, Univ Lyon 2, LIRIS,
UMR5205, F-69622 Villeurbanne, France

²Department of Mathematics and Mathematical Statistics, Umeå University, Sweden

³Laboratoire d'Informatique et Systèmes, Aix-Marseille Université, CNRS, France

⁴Université Côte d'Azur, Inria, CNRS, I3S, France

⁵Department of Mathematics and Informatics, Faculty of Sciences, University of Novi
Sad, Novi Sad, Serbia

Abstract

We initiate the study of the algorithmic complexity of Maker-Breaker games played on edge sets of graphs for general graphs. We mainly consider three of the *big four* such games: the connectivity game, perfect matching game, and H -game. Maker wins if she claims the edges of a spanning tree in the first, a perfect matching in the second, and a copy of a fixed graph H in the third. We prove that deciding who wins the perfect matching game and the H -game is PSPACE-complete, even for the latter in graphs of small diameter if H is a tree. Seeking to find the smallest graph H such that the H -game is PSPACE-complete, we also prove that there exists such an H of order 51 and size 57.

On the positive side, we show that the connectivity game and arboricity- k game are polynomial-time solvable. We then give several positive results for the H -game, first giving a structural characterization for Breaker to win the P_4 -game, which gives a linear-time algorithm for the P_4 -game. We provide a structural characterization for Maker to win the $K_{1,\ell}$ -game in trees, which implies a linear-time algorithm for the $K_{1,\ell}$ -game in trees. Lastly, we prove that the $K_{1,\ell}$ -game in any graph, and the H -game in trees are both FPT parameterized by the length of the game. We leave the complexity of the last of the big four games, the Hamiltonicity game, as an open question.

Keywords: Maker-Breaker Games, H -game, Perfect Matching Game, Connectivity Game, Graphs, Computational Complexity

*This work has been partially supported by the National Research Agency (ANR) projects P-GASE (ANR-21-CE48-0001-01) and DISTANCIA (ANR-17-CE40-0015).

[†]Supported by the Kempe Foundation Grant No. JCK-2022.1 (Sweden)

[‡]Partly supported by Ministry of Science Technological Development and Innovation of the Republic of Serbia (Grant No. 451-03-47/2023-01/200125). Partly supported by Provincial Secretariat for Higher Education and Scientific Research, Province of Vojvodina (Grant No. 142-451-2686/2021).

1 Introduction

Positional games are a class of combinatorial games that received a lot of attention recently, with the books [2] and [21] giving an overview of the field. Given a finite set X and a family \mathcal{F} of subsets of X , two players alternate claiming unclaimed elements of X until all of the elements are claimed. The set X is referred to as the *board*, and the elements of \mathcal{F} as the *winning sets*. The pair (X, \mathcal{F}) is often called the *game hypergraph*.

As for the rules for determining the winner of the game, there are several conventions – Maker-Breaker games, Avoider-Enforcer games, strong making games, strong avoiding games, *etc.* The most studied convention is Maker-Breaker games, first introduced by Erdős and Selfridge [12], where the players are Maker and Breaker. Maker wins the game if she claims all the elements of a set from \mathcal{F} , and Breaker wins otherwise. Unless stated otherwise, it is assumed that Maker plays first and that both players play optimally, *i.e.*, they play to win.

There is a number of results on Maker-Breaker games where the board of the game is the edge set $E(G)$ of a graph G . For the majority of the games that appear in the literature, the winning sets are all representatives of some predetermined graph structure. The four most prominent examples of the well-studied games of this kind are the connectivity, perfect matching, Hamiltonicity, and H -game, where the winning sets are all, respectively, spanning trees, perfect matchings, Hamiltonian cycles, and copies of a fixed graph H .

Maker-Breaker games on graphs were first introduced and studied by Chvátal and Erdős [9], who looked at the games of connectivity, Hamiltonicity, and the H -game played on the edges of the complete graph, and also in the biased setting, where Breaker may claim more than one edge per round. These games, along with the perfect matching game, were further studied in a series of papers, with major breakthroughs in [1], [3], [16], and [23], resulting in the finding of the leading term of the threshold bias for all four games more than three decades later.

In fast Maker-Breaker games, Maker’s goal is not just to win, but also to win as fast as possible. The duration of the game in this setting was first studied in [19] for the games of connectivity, perfect matching, and Hamiltonicity, with further improvements and generalizations in [22, 25]. These results also have direct implications on strong game outcomes for the same set of games, see, *e.g.*, [14].

Another line of research was initiated in [31], studying Maker-Breaker games played on the edge set of the Erdős-Rényi random graph model, where the main question raised and partially answered was to determine the threshold probabilities for Maker’s win for the *big four* – the games of connectivity, perfect matching, Hamiltonicity, and the H -game. This question was further pursued in a number of papers, including [20, 26, 27].

A general overview of results on these and other games on graphs can be found in [21].

Complexity of positional games. Given a positional game, we want to study the complexity of deciding whether Maker wins. Schaefer was the first to show that the problem of determining the winner of a Maker-Breaker game with winning sets of sizes up to 11 is PSPACE-complete [30]. In [29], the required winning set size was reduced to six. Recently, it was shown that the outcome of any Maker-Breaker game with winning sets of size at most three can be found in polynomial time [15]. Hence, when it comes to positional games on edge sets of graphs, we can conclude the outcome in polynomial time of, for example, the H -game for any H such that $|E(H)| \leq 3$ (*e.g.*, the triangle game).

A simpler proof of the PSPACE-hardness of Maker-Breaker games in full generality is presented in [8], along with a reduction from Maker-Breaker games to Maker-Maker games, showing that deciding who wins Maker-Maker games (a.k.a. “the strong games”) is also PSPACE-complete. The problem of determining the winner in a Maker-Breaker game sometimes appears in the literature under a different name, POS CNF, an equivalent formulation dealing with a game in which the goal is to satisfy a positive CNF formula (see [29, 30]). Regarding other conventions, Avoider-Enforcer and Chooser-Picker are PSPACE-complete [17] and NP-hard [10], respectively, with the former result only recently answering a long-standing open problem.

From a parameterized complexity point of view, a natural parameter for positional games is the length k of a game, *i.e.*, the game ends after Maker’s k^{th} move. Bonnet *et al.* [6] proved that, when parameterized by the length of the game, Maker-Breaker games are W[1]-complete (generalized Hex included), Maker-Maker games are AW[*]-complete, and Avoider-Enforcer games are co-W[1]-complete. However, when the winning sets are more structured, some Maker-Breaker games are fixed-parameter tractable (FPT) parameterized by the length of the game, such as Hex played on a hexagonal grid [7], and a generalization of Tic-Tac-Toe called k -Connect [6].

Our results. The algorithmic complexity of Maker-Breaker games played on vertex sets of graphs has been studied in general graphs (see, *e.g.*, [5, 6, 7, 11, 13]), and, as seen above, there is a large literature on those played on edge sets of graphs. Surprisingly, the algorithmic complexity of the latter has only been considered in restricted graph classes like complete graphs and random graphs. We initiate the study of the algorithmic complexity of Maker-Breaker games played on edge sets of graphs for general graphs. We mainly consider three of the *big four* such games: the connectivity game, perfect matching game, and H -game. First, we introduce notation and preliminary results in Section 2. In Section 3, we prove that deciding the outcomes of the perfect matching game and the H -game is PSPACE-complete. In particular, this holds for the H -game in graphs of diameter at most 6 where H is a tree. Towards finding the smallest graph H such that the H -game is PSPACE-complete, we also prove that there exists such a graph H of order 51 and size 57.

On the positive side, in Section 4, we observe that the connectivity game is polynomial-time solvable as Maker wins the connectivity game (playing second) if and only if the graph contains two edge-disjoint spanning trees [24], and this property can be checked in polynomial time [32]. We also prove that the arboricity- k game can be solved in polynomial-time. We then give several positive results for the H -game, first giving a structural characterization for Breaker to win the P_4 -game, which gives a linear-time algorithm for the P_4 -game. This improves on the algorithm of Galliot *et al.* [15] for this game since their algorithm does not give any structural insight about the graphs in which Maker (Breaker, resp.) wins, and it runs in time $O(|V(\mathcal{H})|^5 E(\mathcal{H})^2 + V(\mathcal{H})^6 \Delta(\mathcal{H}))$, where \mathcal{H} is the hypergraph for the game. We provide a structural characterization for Maker to win the $K_{1,\ell}$ -game (H is a star with ℓ leaves) in trees, which implies a linear-time algorithm for the $K_{1,\ell}$ -game in trees. Lastly, in Section 5, we prove that the $K_{1,\ell}$ -game in any graph, and the H -game in trees are both FPT parameterized by the length of the game. The former result is the by-product of proving that the $K_{1,\ell}$ -game needs only to be considered in graphs of bounded maximum degree in k and in the ball (of edges) of bounded diameter in the length of the game centered at the first edge claimed by Maker. This is in stark contrast with the result that the H -game is

PSPACE-complete in graphs of diameter at most 6, even when H is a tree. In Section 6, we give directions for further work, such as determining the complexity of the Hamiltonicity game, the last of the *big four* games.

2 Preliminaries

Unless stated otherwise, the graphs considered in this paper are finite, simple, and undirected. The vertex and edge sets of a graph G will be denoted by $V(G)$ and $E(G)$, respectively. For any graph G and any two vertices $x, y \in V(G)$, $dist_G(x, y)$ is the length of a shortest path between x and y in G . A graph H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Let $V' \subseteq V(G)$. The subgraph *induced by* V' , denoted by $G[V']$, is the graph with vertex set V' and whose edge set consists of all the edges of G between two vertices of V' : $E(G[V']) = \{uv \in E(G) \mid u, v \in V'\}$. If E' is a set of edges, the *edge-induced subgraph* $G[E']$ is the subgraph of G whose edge set is E' and whose vertex set consists of all ends of edges of E' .

In this paper, we consider Maker-Breaker games played on the edge set of a graph G . In other words, the game hypergraph is a pair $(E(G), \mathcal{F})$, where $\mathcal{F} \subseteq 2^{E(G)}$. For all of the games we consider, we say that Maker *wins* in G if she has a *winning strategy* in G , that is, regardless of how Breaker plays, Maker can ensure claiming the edges of a winning set $F \in \mathcal{F}$. Similarly, Breaker wins in G if he has a winning strategy in G , that is, regardless of how Maker plays, Breaker can ensure claiming at least one edge in each winning set $F \in \mathcal{F}$. Note that, by definition, only one of the two players may have a winning strategy for a given pair $(E(G), \mathcal{F})$. We mainly consider three games: the *connectivity game*, *perfect matching game*, and *H-game*. In the connectivity game, \mathcal{F} is the set of all the spanning trees of G , and so, Maker wins if she can claim the edges of a tree that contains all the vertices of G . In the perfect matching game, \mathcal{F} is the set of all the perfect matchings of G . Thus, Maker wins if she manages to claim $|V(G)|/2$ vertex-disjoint edges that cover all the vertices of the graph. Note that, during the game, Maker can claim edges incident to other edges she already claimed. In the H -game, H refers to another (fixed) graph and the winning sets are all the copies of H . More formally, a winning set $F \in \mathcal{F}$ is a set of edges such $G[F]$ is isomorphic to H . Note that $G[F]$ is not necessarily a vertex-induced subgraph of G (there might be more edges in G), but its vertices must be in a one-to-one correspondence with the vertex set of H .

As mentioned before, unless stated otherwise, Maker plays first. Note, however, that most of our results can be extended if Breaker starts. For example, assume one can check in polynomial time if Maker wins when Maker starts in some family of graphs. Then, in the case where Breaker starts the game, we can scan through all the possible first moves of Breaker, and treat the resulting graph with one edge removed as the base graph for a new game in which Maker starts. If the family of graphs is stable by edge removal, this leads to a polynomial algorithm.

When it is clear what the set of winning sets is, we will use only the graph G to describe a game. We will also need to consider a *position* of the game where some edges have already been claimed. We will denote a position by a triplet (G, E_M, E_B) , where E_M and E_B are two disjoint sets of edges that correspond to the edges already claimed by Maker and by Breaker, respectively. Note that, in general, $|E_B| \leq |E_M| \leq |E_B| + 1$.

We now state some basic definitions and facts about positional games that will be used throughout the paper. The next lemma expresses the fact that the addition of edges can only

help Maker, whereas, the deletion of edges is favorable for Breaker.

Lemma 2.1 (Folklore). *Let G' be a subgraph of G , $\mathcal{F} \subseteq 2^{E(G)}$ a set of winning sets, and $\mathcal{F}' \subseteq \mathcal{F} \cap 2^{E(G')}$. If Maker wins in $(E(G'), \mathcal{F}')$, then she wins in $(E(G), \mathcal{F})$.*

The union of two games $(E(G_1), \mathcal{F}_1)$ and $(E(G_2), \mathcal{F}_2)$ is simply the game $(E(G_1) \cup E(G_2), \mathcal{F}_1 \cup \mathcal{F}_2)$. Note that $E(G_1)$ and $E(G_2)$ can intersect. It is not difficult to prove the following.

Lemma 2.2 (Folklore). *Let $(E(G_1), \mathcal{F}_1)$ and $(E(G_2), \mathcal{F}_2)$ be Maker-Breaker games. If Maker wins in one of the two games, then she wins in their union $(E(G_1) \cup E(G_2), \mathcal{F}_1 \cup \mathcal{F}_2)$. If $E(G_1) \cap E(G_2) = \emptyset$, then the converse is also true.*

A pairing strategy is a frequently used type of strategy in positional games. Suppose that a subset of edges is partitioned into disjoint pairs. Then, the pairing strategy dictates that, during the game, whenever their opponent claims one edge of a pair, the player responds by claiming the other one. We will use pairing strategies in several places, sometimes as ingredients in more elaborate strategies.

Erdős and Selfridge [12] gave a criterion for Breaker to win:

Theorem 2.3 (Erdős-Selfridge Criterion [12]). *For any positional game with winning sets \mathcal{F} , if $\sum_{F \in \mathcal{F}} 2^{-|F|} < \frac{1}{2}$, then Breaker wins playing second.*

We end this preliminary section with a lemma that states that if two edges are “similar”, we can assume that both players will claim one each.

Lemma 2.4. *Let G be a (not necessarily simple) graph and \mathcal{F} the set of the winning sets of a positional game played in G . Let (G, E_M, E_B) be a position of the game and e_1, e_2 two unclaimed edges of G such that the two sets $\{F \setminus (E_M \cup \{e_1, e_2\}) \mid e_i \in F \in \mathcal{F} \text{ and } F \cap E_B = \emptyset\}$, for $i \in \{1, 2\}$, (the new winning sets for Maker after claiming e_i) are the same. Then, (G, E_M, E_B) and $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$ have the same outcome, regardless of who plays next.*

Proof. Let G be a (not necessarily simple) graph, \mathcal{F} the winning sets of some positional game played in G , and (G, E_M, E_B) some position. Let e_1, e_2 be two unclaimed edges such that the sets $\{F \setminus (E_M \cup \{e_1, e_2\}) \mid e_i \in F \in \mathcal{F} \text{ and } F \cap E_B = \emptyset\}$ are equal for $i \in \{1, 2\}$. Note that a player plays next in (G, E_M, E_B) if and only if they play next in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$. Suppose first that Maker wins in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$ and let \mathcal{S} be a corresponding winning strategy for her. We define her winning strategy \mathcal{S}' in (G, E_M, E_B) as follows:

- If it is Maker’s turn, then she claims the edge e_0 she would have claimed by \mathcal{S} in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$.
- If Breaker claims an edge in $\{e_1, e_2\}$, then she claims the second edge in $\{e_1, e_2\}$.
- If Breaker claims any other edge e , then she claims the edge e' she would have claimed by \mathcal{S} if Breaker had claimed e in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$.
- If e_1 and e_2 are the only unclaimed edges and it is Maker’s turn, then she claims e_1 .

Since \mathcal{S} is a winning strategy, there exists a set of edges $F \in \mathcal{F}$ for which Maker will claim all the edges following \mathcal{S} in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$. If $F \cap \{e_1, e_2\} = \emptyset$, then the same set of edges F are claimed following \mathcal{S}' in (G, E_M, E_B) , and so, she wins in (G, E_M, E_B) . Otherwise, $F \cap \{e_1, e_2\} = \{e_1\}$, as e_2 is attributed to Breaker. By the construction of \mathcal{S}' , Maker either claimed e_1 or e_2 . If she claimed e_1 , then she claimed all the edges of F , and therefore, she wins in (G, E_M, E_B) . Otherwise, she claimed e_2 , and by the definitions of e_1 and e_2 , there exists a set $F' \in \mathcal{F}$ such that $F \setminus (\{E_M\} \cup \{e_1\}) = F' \setminus (E_M \cup \{e_2\})$. Thus, she claimed all the edges of F' and she wins in (G, E_M, E_B) .

Reciprocally, suppose now that Breaker wins in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$ and let \mathcal{S} be a corresponding winning strategy for him. We define his winning strategy \mathcal{S}' in (G, E_M, E_B) as follows:

- If it is Breaker's turn, then he claims the edge e_0 he would have claimed by \mathcal{S} in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$.
- If Maker claims an edge in $\{e_1, e_2\}$, then he claims the second edge in $\{e_1, e_2\}$.
- If Maker claims any other edge e , then he claims the edge e' he would have claimed by \mathcal{S} if Maker had claimed e in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$.
- If e_1 and e_2 are the only unclaimed edges and it is Breaker's turn, then he claims e_1 .

Since \mathcal{S} is a winning strategy, for all $F \in \mathcal{F}$, there exists an edge $e_B \in F$ that Breaker will claim following \mathcal{S} in $(G, E_M \cup \{e_1\}, E_B \cup \{e_2\})$. Since, by definition, the two sets $\{F \setminus (E_M \cup \{e_1, e_2\}) \mid e_i \in F \in \mathcal{F} \text{ and } F \cap E_B = \emptyset\}$ are equal for $i \in \{1, 2\}$, we can suppose that $e_B \neq e_2$, as otherwise $F \cup \{e_1\} \setminus \{e_2\}$ would be a winning set claimed by Maker. Therefore, as any edge that would have been claimed by Breaker by \mathcal{S} has been claimed by him by \mathcal{S}' , he has claimed at least one edge in each winning set, and Breaker wins in (G, E_M, E_B) . \square

3 Hardness Results

In this section, we prove that it is PSPACE-complete to determine the outcome of the perfect matching game and the H -game. Our reductions are from UNIFORM POS CNF 6 that is known to be PSPACE-complete [29]. Specifically, UNIFORM POS CNF 6 is a game played on a CNF formula \mathcal{F} over a set of variables $X = \{X_1, \dots, X_n\}$, where all the clauses contain exactly six variables, all the variables appear in their positive forms, and every variable appears in at least one clause. Two players, named Satisfier and Falsifier, respectively, alternate claiming a variable that has not been claimed yet, with Satisfier playing first. Variables claimed by Satisfier are set to true, and those claimed by Falsifier are set to false. Satisfier wins if and only if \mathcal{F} is true once every variable has been claimed. The associated decision problem, *i.e.*, deciding whether Satisfier wins or not, is denoted by UNIFORM POS CNF 6. Note that, in [29], it is also proven that UNIFORM POS CNF 6 remains PSPACE-complete, even if Falsifier goes first and/or the number of variables n is odd, which we make use of in our reductions.

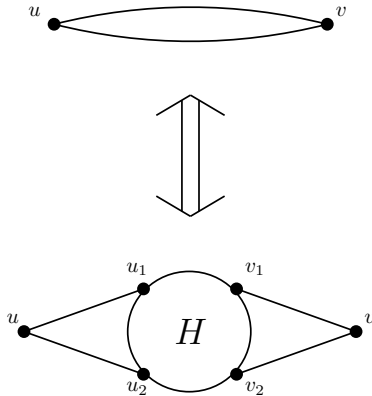


Figure 1: How parallel edges are removed in the perfect matching game in Lemma 3.1.

3.1 Perfect-matching game

We first prove that it is PSPACE-complete to determine the outcome of the perfect matching game. To simplify the reduction, the proof will be done allowing for parallel edges, that the following lemma enables to remove, see Figure 1.

Lemma 3.1. *Let G be any graph containing two parallel edges e_1 and e_2 connecting two of its vertices u and v . Then, there exists a complete bipartite graph H with bipartition (A, B) such that, if e_1 and e_2 are removed from G , and a copy of H is added instead, where u (v , respectively) is adjacent to two vertices of A (B , respectively), then the outcome of the perfect matching game in the resulting graph G' is the same as that in G .*

Proof. With the parallel edges e_1 and e_2 in G , Maker can clearly claim the edge uv “for free”, independent of the play in the rest of the graph. That means that, playing in G , she has two possibilities to claim a perfect matching:

- (i) claim a perfect matching in $G - \{e_1, e_2\}$, or
- (ii) claim a perfect matching in $G \setminus \{u, v\}$.

Let H be a balanced complete bipartite graph with parts A and B , and let $u_1, u_2 \in A$ and $v_1, v_2 \in B$. We will show that when $|A| = |B| = k$, for a suitable constant k to be chosen later, Maker, playing second, has a strategy in H that satisfies the following five conditions:

- Maker has a perfect matching in H , and
- for every $i, j \in \{1, 2\}$, Maker has a perfect matching in $H \setminus \{u_i, v_j\}$.

Once this is shown, forming G' by replacing the parallel edges e_1 and e_2 in G by H , and adding the two pairs of edges connecting u to u_1 and u_2 , and v to v_1 and v_2 , as depicted in Figure 1, finishes the proof. Indeed, assume Maker, playing second, has a strategy in H to satisfy the above-mentioned five conditions, and, on top of that, in G' , she pairs uu_1 with uu_2 , and vv_1 with vv_2 . Then, she can claim a perfect matching in G' if and only if she can satisfy (i) or (ii) in G , which corresponds to her being able to claim a perfect matching in G .

To show that Maker can satisfy the five conditions playing second in H , we define an auxiliary positional game in H . Let

$$\mathcal{F}_H := \{E(X, Y) \mid X \subseteq A, Y \subseteq B, |X| + |Y| = k + 1\},$$

and, for every $i, j \in \{1, 2\}$, let

$$\mathcal{F}_{i,j} := \{E(X, Y) \mid X \subseteq A \setminus \{u_i\}, Y \subseteq B \setminus \{v_j\}, |X| + |Y| = k\}.$$

In the auxiliary game, Maker will assume the role of Breaker (to avoid confusion, we will call this player Auxiliary Breaker), trying to claim one edge in every winning set in $\mathcal{F} := \mathcal{F}_H \cup (\cup_{i,j \in \{1,2\}} \mathcal{F}_{i,j})$. If she achieves that, then Hall's condition [18] for the existence of a perfect matching in a bipartite graph implies that Maker will have a claimed perfect matching in H , as well as in $H \setminus \{u_i, v_j\}$, for every $i, j \in \{1, 2\}$.

To show that Auxiliary Breaker can win the auxiliary game in H , we apply the Erdős-Selfridge Criterion (see Theorem 2.3):

$$\begin{aligned} \sum_{E' \in \mathcal{F}} 2^{-|X|} &\leq \sum_{\ell=1}^k \binom{k}{\ell} \binom{k}{k-\ell+1} 2^{-\ell(k-\ell+1)} + 4 \sum_{\ell=1}^{k-1} \binom{k-1}{\ell} \binom{k-1}{k-\ell} 2^{-\ell(k-\ell)} \\ &\leq \sum_{\ell=1}^k \binom{k}{\ell} \binom{k}{k-\ell+1} 2^{-\ell(k-\ell)} + 4 \sum_{\ell=1}^k \binom{k}{\ell} \binom{k}{k-\ell} 2^{-\ell(k-\ell)} \\ &= \sum_{\ell=1}^k \binom{k}{\ell} \binom{k}{\ell-1} 2^{-\ell(k-\ell)} + 4 \sum_{\ell=1}^k \binom{k}{\ell}^2 2^{-\ell(k-\ell)} \\ &\leq 2 \sum_{\ell=1}^{\lceil k/2 \rceil} \binom{k}{\ell}^2 2^{-\ell(k-\ell)} + 8 \sum_{\ell=1}^{\lceil k/2 \rceil} \binom{k}{\ell}^2 2^{-\ell(k-\ell)} \\ &\leq 10 \sum_{\ell=1}^{\lceil k/2 \rceil} k^{2\ell} 2^{-\ell(k-\ell)} \\ &\leq 10 \sum_{\ell=1}^{\lceil k/2 \rceil} k^{2\ell} 2^{-\ell(k-\lceil k/2 \rceil)} \\ &= 10 \sum_{\ell=1}^{\lceil k/2 \rceil} \left(2^{2 \log_2 k - \lceil k/2 \rceil} \right)^\ell. \end{aligned}$$

This expression tends to zero when k grows, and hence, for k large enough (say, for $k = 100$) it is less than $1/2$, and so, the Erdős-Selfridge criterion implies Auxiliary Breaker's win. \square

Theorem 3.2. *Deciding whether Maker wins the Maker-Breaker perfect matching game in a given graph G is PSPACE-complete.*

Proof. The problem is clearly in PSPACE since both the number of turns and the number of possible moves at each turn are bounded from above by n^2 . To prove it is PSPACE-hard, we give a reduction from UNIFORM POS CNF 6 where there are an odd number of variables, which, as mentioned before, is PSPACE-hard [29].

Let ϕ be an instance of UNIFORM POS CNF 6 where there are an odd number of variables. Denote the variables in ϕ by x_1, \dots, x_{2n+1} , and the clauses in ϕ by C_1, \dots, C_m . From ϕ , we construct the graph G as follows, and recall that, by Lemma 3.1, we can allow pairs of parallel edges.

- For all $1 \leq i \leq 2n + 1$, introduce two vertices $v_{i_0}^i$ and $\bar{v}_{i_0}^i$, and the edge $e_i = v_{i_0}^i \bar{v}_{i_0}^i$.
- Then, add n new vertices a_1, \dots, a_n , and, for all $1 \leq i \leq 2n + 1$ and $1 \leq \ell \leq n$, add two parallel edges between $v_{i_0}^i$ and a_ℓ . See Figure 2 for an illustration.
- For each clause C_j in ϕ , add a vertex C^j in G .
- For each variable x_i in ϕ , let $C_{i_1}, \dots, C_{i_{k_i}}$ be the clauses containing x_i in ϕ . For all $1 \leq i \leq 2n + 1$ and for all $1 \leq j \leq k_i$, add the vertices $u_{i_j}^i, \bar{u}_{i_j}^i, v_{i_j}^i, \bar{v}_{i_j}^i, x_{i_j}^i$, and $y_{i_j}^i$. Also, for all $1 \leq i \leq 2n + 1$, add the vertices $y_{i_{k_i+1}}^i$. Then, connect them as follows for all $1 \leq i \leq 2n + 1$ and $1 \leq j \leq k_i$ (see Figures 3 and 4):

- Add the two edges $\bar{v}_{i_{j-1}}^i u_{i_j}^i$ and $v_{i_{j-1}}^i \bar{u}_{i_j}^i$.
- Add two parallel edges between $u_{i_j}^i$ and $x_{i_j}^i$.
- Add two parallel edges between $x_{i_j}^i$ and C^{i_j} .
- Add two parallel edges between $x_{i_j}^i$ and $y_{i_j}^i$.
- Add two parallel edges between $u_{i_j}^i$ and $\bar{u}_{i_j}^i$.
- Add two parallel edges between $\bar{u}_{i_j}^i$ and $v_{i_j}^i$.
- Add two parallel edges between $v_{i_j}^i$ and $\bar{v}_{i_j}^i$.

Also, for all $1 \leq i \leq 2n + 1$, add two parallel edges between $\bar{v}_{i_{k_i}}^i$ and $y_{i_{k_i+1}}^i$.

- If $|V(G)|$ is currently an odd number, then add the vertex $y_{0_0}^0$ in G .
- For each pair of vertices among the $y_{i_j}^i$'s (including $y_{0_0}^0$ if it exists), add two parallel edges between them (see Figure 5).

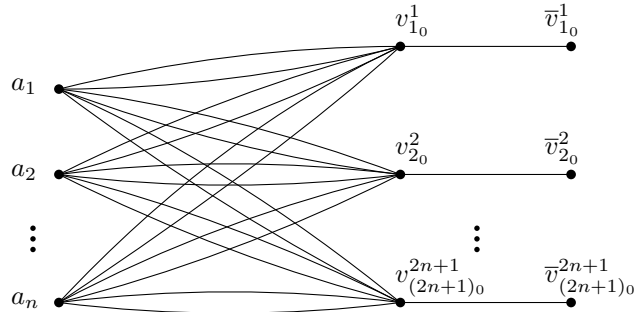


Figure 2: The variable gadget in the graph G constructed in the proof of Theorem 3.2.

Note that G is clearly constructed in polynomial time. We prove that Satisfier wins in ϕ if and only if Maker wins the perfect matching game in G . By Lemma 2.4, the outcome of the

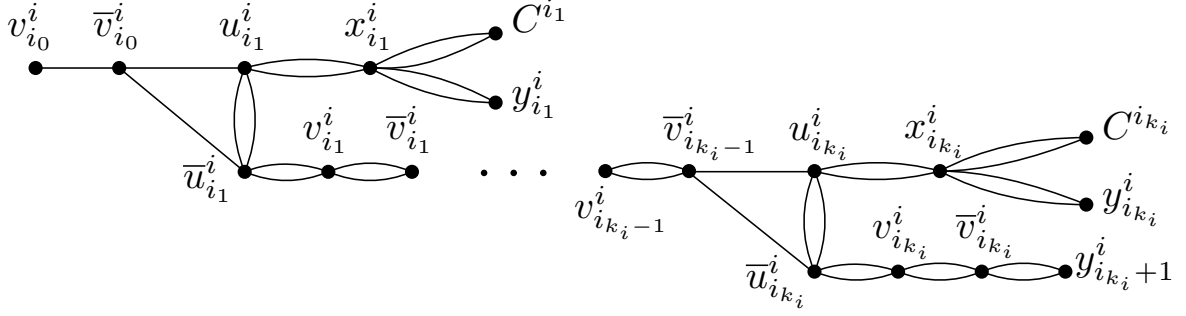


Figure 3: Construction for a variable x_i in clauses $C^{i_1}, \dots, C^{i_{k_i}}$ in ϕ in the graph G constructed in the proof of Theorem 3.2.

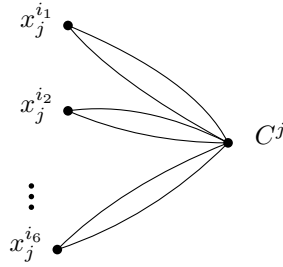


Figure 4: Construction for a clause $C_j = (x_{i_1} \vee \dots \vee x_{i_6})$ in ϕ in the graph G constructed in the proof of Theorem 3.2.

perfect matching game in G is the same as the outcome in G , where, for each pair of parallel edges, both Maker and Breaker have claimed one of the two edges, and so, we can assume they have done so in what follows.

First, we prove the simpler of the two directions, that is, if Falsifier wins in ϕ , then Breaker wins the perfect matching game in G . Assume that Falsifier has a winning strategy \mathcal{S} in ϕ . Consider the following strategy for Breaker in G :

- If Maker claims an edge $e_i = v_{i_0}^i \bar{v}_{i_0}^i$, then Breaker answers by claiming an edge $e_j = v_{j_0}^j \bar{v}_{j_0}^j$, where x_j is the variable that would have been claimed by Falsifier in ϕ according to \mathcal{S} if Satisfier claimed x_i in ϕ .
- If Maker claims any edge other than an e_i before all the e_i 's have been claimed, then Breaker claims an arbitrary e_i . Then, as there is an odd number of e_i 's, by pairing them, Breaker can ensure claiming at least $n + 1$ of them. Thus, by construction, Maker will not be able to have a perfect matching containing all the $v_{i_0}^i$'s. Indeed, after this step, at least $n + 1$ of the e_i 's are claimed by Breaker, and so, their respective $n + 1$ $v_{i_0}^i$'s must be matched with their only remaining neighbors, the a_ℓ 's, of which there are only n , and thus, this is not possible.

Hence, we can assume that all the edges e_i have been claimed during the first $2n + 1$ moves. Consider the valuation obtained in ϕ if all the x_i variables associated to the e_i edges claimed by Maker are the variables set to True, and all the x_i variables associated to the e_i edges claimed by Breaker are the variables set to False. By the hypothesis that \mathcal{S} is a winning

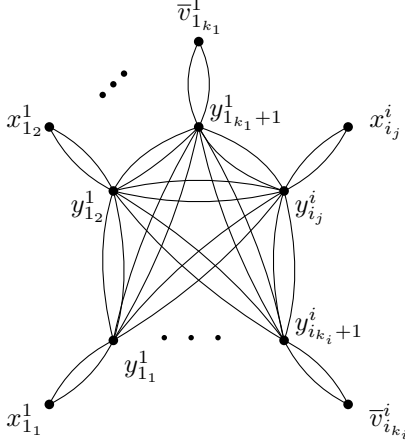


Figure 5: The $y_{i_j}^i$ vertices in the graph G constructed in the proof of Theorem 3.2.

strategy for Falsifier in ϕ , there exists a clause C_z that is not satisfied by this valuation in ϕ . Let x_{f_1}, \dots, x_{f_6} be the variables in C_z in ϕ . Since Breaker claimed all the edges e_i corresponding to the variables x_i in ϕ that Falsifier would have claimed according to \mathcal{S} , the edges e_{f_1}, \dots, e_{f_6} are claimed by Breaker in G . Recall that, as the number of variables is odd, it is Breaker's turn. Breaker plays as follows for all $\ell \in \{f_1, \dots, f_6\}$ and for $j = 1$ to k_ℓ while $\ell_j \leq z$:

- If $\ell_j < z$, then Breaker claims $\bar{v}_{\ell_{j-1}}^\ell \bar{u}_{\ell_j}^\ell$. As the only remaining edge available to match $\bar{v}_{\ell_{j-1}}^\ell$ is $\bar{v}_{\ell_{j-1}}^\ell u_{\ell_j}^\ell$, Maker has to claim it. Now, all the edges adjacent to $\bar{u}_{\ell_j}^\ell$ have been claimed, and Maker has claimed only two of them, of which only one does not interfere with the edges already forced in the matching: $\bar{u}_{\ell_j}^\ell v_{\ell_j}^\ell$. Thus, $\bar{u}_{\ell_j}^\ell v_{\ell_j}^\ell$ has to be in any perfect matching claimed by Maker, which forces the edge $v_{\ell_j}^\ell \bar{v}_{\ell_j}^\ell$ claimed by Maker to not be in the matching.
- If $\ell_j = z$, then Breaker claims $\bar{v}_{\ell_{j-1}}^\ell u_z^\ell$, forcing Maker to claim $\bar{v}_{\ell_{j-1}}^\ell \bar{u}_z^\ell$ to match $\bar{v}_{\ell_{j-1}}^\ell$. Now, the only edge that Maker can use to match u_z^ℓ is the edge $u_z^\ell x_z^\ell$.

By the above strategy for Breaker, any perfect matching contained in the edges claimed by Maker has to contain the edges $u_z^\ell x_z^\ell$ for all $\ell \in \{f_1, \dots, f_6\}$. Therefore, all the vertices adjacent to C^z are already matched, and thus, it cannot be matched, and Breaker wins.

Now, we prove that if Satisfier wins in ϕ , then Maker wins the perfect matching game in G . Assume that Satisfier has a winning strategy \mathcal{S} in ϕ . We construct a strategy for Maker in G as follows:

First, Maker claims the edge $e_i = v_{i_0}^i \bar{v}_{i_0}^i$ corresponding to the variable x_i that Satisfier would have claimed first in ϕ according to \mathcal{S} . Then,

- If Breaker claims an edge $e_j = v_{j_0}^j \bar{v}_{j_0}^j$, then Maker claims the edge $e_i = v_{i_0}^i \bar{v}_{i_0}^i$ corresponding to the variable x_i that Satisfier would have claimed according to \mathcal{S} if Falsifier had claimed x_j in ϕ .
- For any variable x_i in a clause C_j in ϕ , Maker pairs the edges $\bar{v}_{i_{j-1}}^i u_{i_j}^i$ and $\bar{v}_{i_{j-1}}^i \bar{u}_{i_j}^i$ in G , and so, if Breaker claims one of them, she claims the other one.

Note that, by construction, one of these moves is always available, as after the first move of Maker, any set where moves are considered has an even number of unclaimed edges remaining. Suppose that Maker employs this strategy until the end of the game. Then, she will have claimed the e_i edges corresponding to the x_i variables claimed by Satisfier in ϕ according to \mathcal{S} . We extract a perfect matching from the edges she claimed as follows:

- Add in the matching, the $n + 1$ e_i 's claimed by Maker. The n $v_{i_0}^i$'s that are not in these edges are paired with the n a_ℓ 's.
- Let i_1, \dots, i_{n+1} be the indices of the e_i edges claimed by Maker. For $\ell = i_1$ to i_{n+1} , add in the matching all the edges $u_j^\ell \bar{u}_j^\ell$ and $v_j^\ell \bar{v}_j^\ell$ such that x_ℓ is in C_j in ϕ .
- For each clause C_j , as at least one variable x_i in C_j in ϕ corresponding to an edge x_i claimed by Maker is True, consider such an i , and add the edge $x_j^i C^j$ in the matching. For each other variable x_ℓ in C_j in ϕ corresponding to an edge x_ℓ claimed by Maker ($\ell \neq i$), add the edge $x_j^\ell y_j^\ell$ in the matching.
- For each variable x_i with $i \notin \{i_1, \dots, i_{n+1}\}$ being in clauses $C_{i_1}, \dots, C_{i_{k_i}}$ in ϕ :
 - For $j = 1$ to k_i , while Maker has claimed $\bar{v}_{i_{j-1}}^i u_{i_j}^i$, add in the matching the edges, $\bar{v}_{i_{j-1}}^i u_{i_j}^i$, $x_{i_j}^i y_{i_j}^i$, and $\bar{u}_{i_j}^i v_{i_j}^i$. If and once Maker has claimed an edge $\bar{v}_{i_{z-1}}^i \bar{u}_{i_z}^i$, add in the matching $\bar{v}_{i_{z-1}}^i \bar{u}_{i_z}^i$, $u_{i_z}^i x_{i_z}^i$, and $v_{i_z}^i \bar{v}_{i_z}^i$, and exit the for loop. Then, for $j = z + 1$ to k_i , add in the matching the edges $u_{i_j}^i \bar{u}_{i_j}^i$, $x_{i_j}^i y_{i_j}^i$, and $v_{i_j}^i \bar{v}_{i_j}^i$.
 - If Maker has claimed no edge $\bar{v}_{i_{z-1}}^i \bar{u}_{i_z}^i$ at the end, add in the matching $\bar{v}_{i_{k_i}}^i y_{i_{k_i}+1}^i$.
- Note that this strategy matches all the vertices except some of the $y_{i_j}^i$'s (including $y_{0_0}^0$ if it exists). As G contains an even number of vertices, and every pair of remaining vertices is connected by two parallel edges, by considering any matching among the remaining $y_{i_j}^i$'s, Maker has claimed a perfect matching and wins. \square

3.2 H -game

We begin by proving that the H -game is PSPACE-complete, even when H is a tree and in graphs of diameter at most 6. Hence, even when H is a relatively basic graph, determining the outcome of the H -game remains hard. Furthermore, the H -game remaining hard in graphs of constant diameter contrasts with a later result (Corollary 5.2) that shows that when H is a star, the H -game is FPT parameterized by the length of the game since the diameter of the graph (after Maker's first move) can be bounded by this parameter too.

Theorem 3.3. *There exists a tree H , such that deciding whether Maker wins the H -game in a given graph G is PSPACE-complete, even if G has diameter at most 6.*

Proof. We show that the statement holds for the tree H in Figure 6.

The problem is clearly in PSPACE since both the number of turns and the number of possible moves at each turn are bounded from above by n^2 . To prove it is PSPACE-hard, we give a reduction from UNIFORM POS CNF 6 where Falsifier plays first, which as mentioned before, is known to be PSPACE-hard [29].

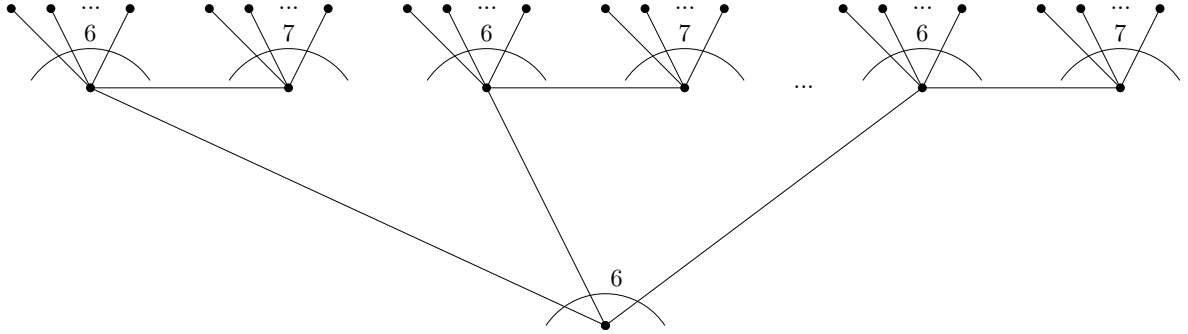


Figure 6: The tree H in the proof of Theorem 3.3.

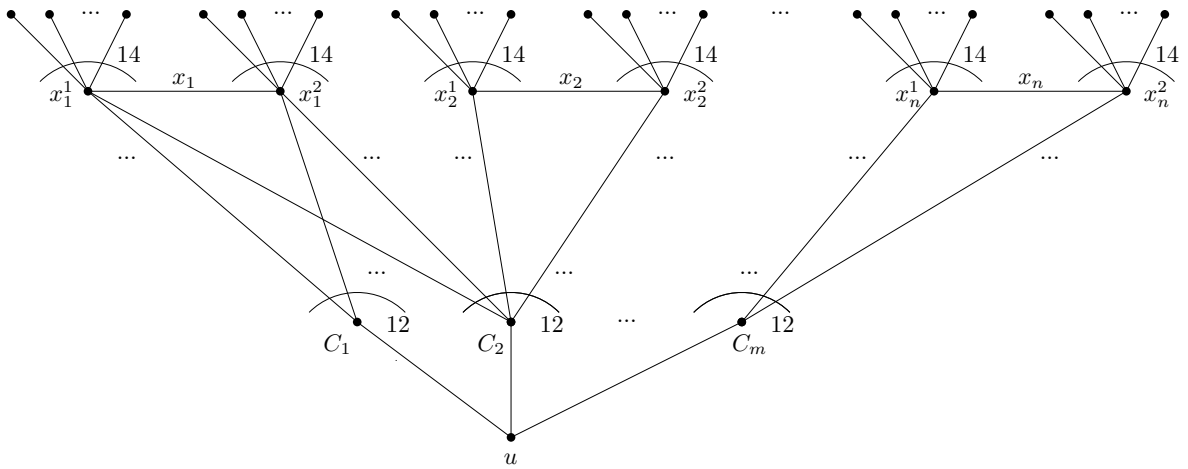


Figure 7: The graph G constructed in the proof of Theorem 3.3. In this example, the variable x_1 appears in the clauses C_1 and C_2 , the variable x_2 appears in the clause C_2 , and the variable x_n appears in the clause C_m .

Let ϕ be an instance of UNIFORM POS CNF 6 in which Falsifier plays first. From ϕ , we construct the graph G as follows. For each clause C_j in ϕ , introduce a new clause vertex C_j in G , and, for each variable x_i in ϕ , introduce a new variable edge $x_i = x_i^1 x_i^2$ in G , all pairwise vertex-disjoint from each other. For all $1 \leq i \leq n$ and $1 \leq j \leq m$, if the variable x_i is contained in the clause C_j in ϕ , then add the edges $x_i^1 C_j$ and $x_i^2 C_j$ in G . For each $1 \leq i \leq n$, add 28 vertices, and make 14 of them adjacent to x_i^1 , and the other 14 adjacent to x_i^2 . Lastly, to ensure G has diameter at most 6, add a vertex u and, for all $1 \leq j \leq m$, make it adjacent to C_j . See Figure 7 for an illustration of G , and note that G is clearly constructed in polynomial time.

We prove that Satisfier wins in ϕ (recall that Falsifier plays first) if and only if Breaker wins the H -game in G . First, we prove the following useful claims.

Claim 3.4. *Suppose that, for all $1 \leq i \leq n$ and $1 \leq j \leq m$ such that the edges $x_i^1 C_j$ and $x_i^2 C_j$ exist, Breaker claims at least one of $x_i^1 C_j$ and $x_i^2 C_j$. In that case, if Maker is to claim a copy of H in G , then the unique vertex of degree 6 in H must be a clause vertex in G .*

Proof of the claim. Every other vertex in G either has degree 1 or cannot be adjacent to 6 vertices that have at least 8 edges incident to each of them that Maker can claim, due to

Breaker's strategy. ◇

Claim 3.5. *Suppose that, for all $1 \leq i \leq n$ and $1 \leq j \leq m$ such that the edges $x_i^1 C_j$ and $x_i^2 C_j$ exist, Breaker claims at least one of $x_i^1 C_j$ and $x_i^2 C_j$. In that case, if Maker is to claim a copy of H in G , then each of the pairs of adjacent degree-8 vertices in H must be the two vertices of a variable edge in G .*

Proof of the claim. By Claim 3.4, the unique vertex of degree 6 in H is a clause vertex. Thus, the 6 vertices of degree 8 adjacent to this vertex of degree 6 in H must each be a vertex of 6 different variable edges in G . Indeed, u cannot be one of these vertices since all of u 's neighbors can have at most 7 incident edges claimed by Maker due to Breaker's strategy. Due to Breaker's strategy, for each of these vertices of degree 8, the only vertex adjacent to them that has at least 8 edges incident to it that Maker can claim, is the other vertex in each of the same variable edges. ◇

First, we prove that if Satisfier wins in ϕ , then Breaker wins the H -game in G . Assume that Satisfier wins in ϕ . Breaker employs the following pairing strategy. If Maker claims a variable edge x_i , then Breaker follows his winning strategy as Satisfier in ϕ by claiming the variable edge in G corresponding to the variable he wants to set to true in ϕ assuming that Maker just set the variable x_i to false in ϕ . If Maker claims an edge $x_i^1 C_j$ ($x_i^2 C_j$, resp.), then Breaker claims $x_j^2 C_j$ ($x_j^1 C_j$, resp.). If Maker claims an edge incident to a degree-1 vertex, then Breaker claims an edge incident to the same vertex of that variable edge and a degree-1 vertex (there are an even number of these edges). Lastly, if Maker claims an edge incident to u , then Breaker claims another edge incident to u . Whenever Breaker cannot employ his strategy, he claims an arbitrary edge, and then goes back to following his strategy. For a contradiction, assume that, at the end of the game, Maker claimed a copy of H . Then, by Claims 3.4 and 3.5, there exists a clause such that all of the variable edges corresponding to the variables it contains in ϕ have been claimed by Maker. This contradicts the fact that Satisfier wins in ϕ since Breaker followed Satisfier's winning strategy in ϕ on the variable edges of G .

Now, we prove that if Falsifier wins in ϕ , then Maker wins the H -game in G . Assume that Falsifier wins in ϕ . Maker first claims a variable edge in G that corresponds to the variable she wants to set to false in ϕ according to her winning strategy as Falsifier in ϕ . Then, Maker employs the following pairing strategy. If Breaker claims a variable edge x_i , then Maker follows her winning strategy as Falsifier in ϕ by claiming the variable edge in G corresponding to the variable she wants to set to false in ϕ assuming that Breaker just set the variable x_i to true in ϕ . If Breaker claims an edge $x_i^1 C_j$ ($x_i^2 C_j$, resp.), then Maker claims $x_j^2 C_j$ ($x_j^1 C_j$, resp.). If Breaker claims an edge incident to a degree-1 vertex, then Maker claims an edge incident to the same vertex of that variable edge and a degree-1 vertex (there are an even number of these edges). Lastly, if Breaker claims an edge incident to u , then Maker claims another edge incident to u . Whenever Maker cannot employ her strategy, she claims an arbitrary edge, and then goes back to following her strategy. Since Maker followed Falsifier's winning strategy in ϕ on the variable edges of G , for at least one clause, she will have claimed all of the variable edges corresponding to the variables contained in that clause in ϕ . We can easily locate a Maker's copy of H containing that clause's vertex as the unique vertex of degree 6 in H . □

As can be seen in Figure 6, the tree H from the proof of Theorem 3.3 has order 91. It would be interesting to know the order or size of the smallest graph H for which the H -game remains

PSPACE-complete. As a step in this direction, in a more involved proof, we now show that the H -game is PSPACE-complete for a graph H of order 51 and size 57. It is worth noting that the orders and sizes of our graphs H in our reductions are dependent on the fact that UNIFORM POS CNF 6 is PSPACE-hard, but it is not known whether the analogously defined UNIFORM POS CNF 5 or UNIFORM POS CNF 4 are PSPACE-hard, which would allow for smaller H 's to be constructed.

Theorem 3.6. *There exists a graph H of order 51 and size 57, such that deciding whether Maker wins the H -game in a given graph G is PSPACE-complete.*

Proof. The problem is in PSPACE by the proof of Theorem 3.3. Let H be the graph consisting of a triangle and six 5-cycles, pairwise vertex-disjoint, where one vertex of each 5-cycle is connected to the same vertex of the triangle by a path of length 4, as depicted in Figure 8.

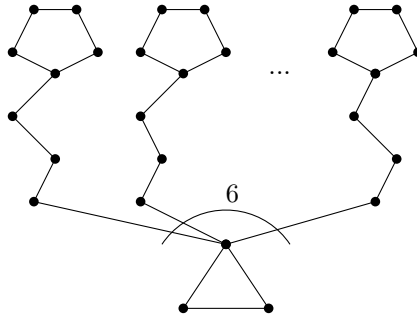


Figure 8: The graph H in the proof of Theorem 3.6.

Before describing the reduction to prove it is PSPACE-hard, we need to introduce some auxiliary graphs, starting with the graph X (see Figure 9), and prove several intermediate results.

Claim 3.7. *Playing on the edges of X as the second player, Maker can claim a path of length 3 connecting t to u_1 or u_2 .*

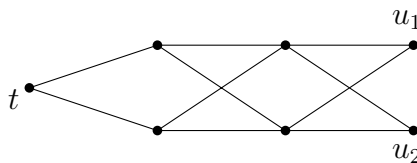


Figure 9: The graph X in the proof of Theorem 3.6.

Proof of the claim. Maker as the second player employs the following pairing strategy that clearly accomplishes her goal. For each vertex $w \in V(X) \setminus \{u_1, u_2\}$, she pairs the two edges between w and its neighbors on the right. \diamond

The next auxiliary graph is Y , whose structure is shown in Figure 10. Note that Y contains four copies of X shown in gray (where the vertex t in Y corresponds to the identification of the vertex t of each copy of X), as well as the following *additional edges* that are not shown in the figure. Each of the vertices a_1 and a_2 (that correspond to the vertices u_1 and u_2 of the corresponding copy of X) is connected by two edges to the vertices x_L and y_L (this is indicated by the letters "LL" above the first copy of X). Similarly, each of the vertices b_1 and

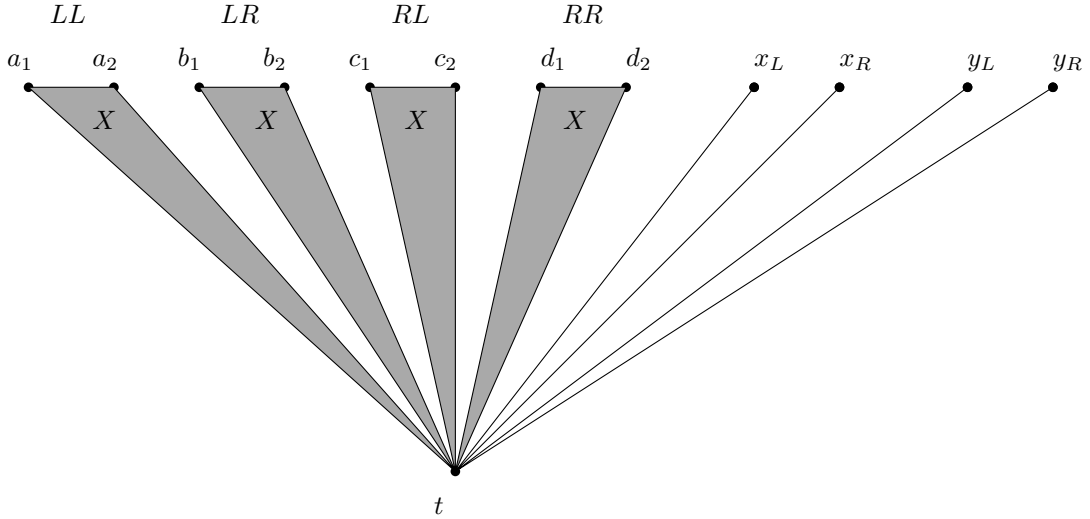


Figure 10: The structure of the graph Y (some edges are omitted) in the proof of Theorem 3.6.

b_2 is connected by two edges to the vertices x_L and y_R (indicated by "LR"). The vertices c_1 and c_2 , and d_1 and d_2 are connected accordingly as in Figure 10.

Claim 3.8. *Playing on the edges of Y as the second player, Maker can claim a C_5 containing the vertex t .*

Proof of the claim. Maker partitions the edge set of Y into several parts, and plays on each of them separately, responding in each of them as the second player.

On each copy of X , Maker plays to claim a path connecting t and one of the two top vertices (the two vertices corresponding to u_1 and u_2 in that copy of X), which is possible by Claim 3.7. For each of the vertices $a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2$, Maker pairs the two edges going towards (two out of four of) the vertices x_L, x_R, y_L, y_R . Finally, she pairs the two edges between t and x_L and x_R , and the two edges between t and y_L and y_R .

Following this strategy, after all the edges are claimed, Maker will claim, w.l.o.g., the edges tx_L and ty_L . Also, in the leftmost copy of X , she will claim, w.l.o.g., a path from t to a_1 . Now, due to pairing, one of the two edges a_1x_L and a_1y_L is claimed by Maker, completing a C_5 through t that is fully claimed by her. \diamond

Claim 3.9. *Playing on the edges of the clique K_6 as the second player, and given any vertex $p \in V(K_6)$, Maker can claim a triangle containing the vertex p .*

Proof of the claim. For her first two moves, Maker claims two edges incident to p , denote them by pv_1 and pv_2 . If, by Maker's next move, Breaker has not claimed v_1v_2 , then Maker does it and completes her goal.

Otherwise, there must be another unclaimed edge incident to p , denote it by pv_3 , and Maker claims it. After the following move of Breaker, the edges v_1v_3 and v_2v_3 must be claimed by him, as otherwise Maker is done. But, then there is another unclaimed edge incident to p , denote it by pv_4 , and Maker claims it. By Maker's next move, Breaker cannot have claimed all the edges $v_iv_j, i, j \in \{1, 2, 3, 4\}$, so Maker claims one of them in the following move completing her triangle at p . \diamond

The last auxiliary graph is the graph Z as depicted in Figure 11.

Claim 3.10. *Playing on the edges of Z as the second player, Maker can claim a path of length 3 from the vertex ℓ to the vertex r .*

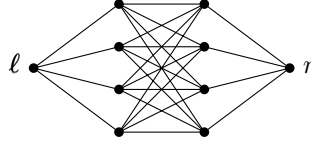


Figure 11: The graph Z in the proof of Theorem 3.6.

Proof of the claim. In her first four moves, Maker claims two edges incident to ℓ , and two edges incident to r . If, at that point, one of the four edges closing a Maker’s path of length 3 between ℓ and r is unclaimed, she claims it and she is done.

Otherwise, all these edges are claimed by Breaker, and so, at most one edge claimed by Breaker is incident to either ℓ or r . In her next two moves, Maker claims one more edge at both ℓ and r , and so, there are nine edges that complete Maker’s path of length 3 from ℓ to r . As Breaker claimed only seven edges before Maker’s next move, at least two of those edges are unclaimed, and Maker completes her path from ℓ to r . \diamond

We are now ready to describe the reduction. To prove it is PSPACE-hard, we give a reduction from UNIFORM POS CNF 6 where Falsifier plays first, as in the proof of Theorem 3.3.

Let ϕ be an instance of UNIFORM POS CNF 6 in which Falsifier plays first. From ϕ , we construct the graph G in the following way. For each clause C_j in ϕ , $j \in \{1, \dots, m\}$, we add a new copy of the clique K_6 and denote one of its vertices as C_j . For each variable x_i in ϕ , $i \in \{1, \dots, n\}$, we introduce a new edge denoted by x_i , all pairwise vertex-disjoint. For each variable edge x_i , we add a new copy of the graph Y and identify one of the ends of x_i , called its “upper” vertex, with the vertex t of Y . The other vertex of the variable edge will be called the “lower” vertex. Finally, for each variable x_i and each clause C_j that contains that variable in ϕ , we introduce a new copy of the graph Z , identifying the vertex ℓ (r , respectively) of this copy of Z with the “lower” vertex of x_i (with C_j , respectively). See Figure 12 for an illustration of G , and note that G is clearly constructed in polynomial time.

We prove that Satisfier wins in ϕ if and only if Breaker wins the H -game in G .

First, we prove that if Satisfier wins in ϕ , then Breaker wins the H -game in G . Assume that Satisfier wins in ϕ . Then, Breaker can apply the winning strategy of Satisfier in ϕ on the variable edges of G , and play arbitrarily in the rest of the graph G . For a contradiction, assume that, at the end of the game, Maker claimed a copy of H .

The graph H contains a triangle, the graph Y is triangle-free, and the middle part of G (what remains of G when all of the copies of Y and K_6 are removed) is bipartite. Hence, the triangle in Maker’s copy of H needs to be within one of the copies of K_6 on one of the clause vertices, denote it by \tilde{C} . But, in H , Maker also claimed six 5-cycles connected by Maker’s paths of length 4 to that Maker’s triangle. As the other K_6 ’s are too far away, the only remaining non-bipartite parts of the base graph that are close enough are the six copies of Y glued to the edges corresponding to the variables in the clause \tilde{C} in ϕ . Since Breaker wins as Satisfier in ϕ on the variable edges in G , at least one of these six variable edges is claimed by Breaker,

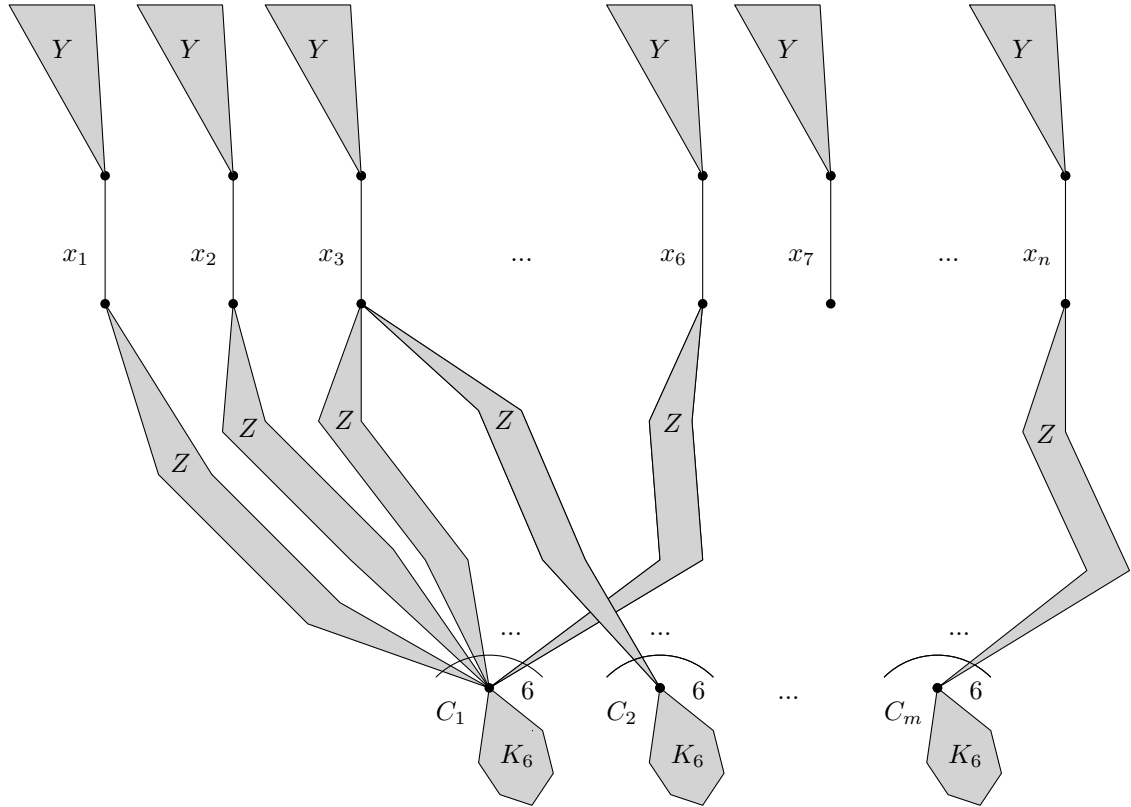


Figure 12: The graph G constructed in the proof of Theorem 3.6.

and so, Maker does not have six internally disjoint paths of length 4 from the clause vertex \tilde{C} to six different 5-cycles, a contradiction.

Now, we prove that if Falsifier wins in ϕ , then Maker wins the H -game in G . Assume that Falsifier wins in ϕ . Maker first claims a variable edge in G that corresponds to the variable she wants to set to false in ϕ according to her winning strategy as Falsifier in ϕ . Maker then partitions the edge set of the base graph into several parts, and plays on each of them separately, responding in each of them as the second player. She plays arbitrarily in G whenever she cannot follow her strategy, and then she resumes her strategy. Specifically, she uses Falsifier's winning strategy in ϕ on the variable edges of G . On each copy of Y , she claims a C_5 through the attachment vertex, as guaranteed by Claim 3.8. In each copy of Z , she claims a path connecting the end vertices, as guaranteed by Claim 3.10. Finally, in each copy of K_6 , she claims a C_3 on the attachment vertex, as guaranteed by Claim 3.2.

Now, we can spot a clause for which all six of the variable edges are claimed by Maker, and locate a Maker's copy of H on that clause's vertex. Indeed, Maker's graph will contain a triangle on that clause's vertex, six paths of length 3 that connect that clause's vertex with the six Maker's variable edges, and on the "upper" vertex of each of those edges, there will be a Maker's 5-cycle. \square

4 Polynomial-time Algorithms

Some games can be solved easily using known results. This is the case for the connectivity game. Indeed, Maker wins the connectivity game (playing second) if and only if G contains two edge-disjoint spanning trees [24]. This leads to a polynomial-time algorithm since this graph property can be checked in polynomial time [32]. This is also the case for any H -game where H has at most three edges. Indeed, such a game can also be solved in polynomial time since the corresponding winning sets have size at most 3 [15].

In this section, we first prove that the arboricity- k game and the cycle game are solvable in polynomial time. Then, we give a concrete linear-time algorithm to solve the P_4 -game in any graph. Finally, we prove that the $K_{1,\ell}$ -game in trees is solvable in linear time for any fixed integer $\ell \geq 1$.

4.1 Arboricity- k game and cycle game

In the arboricity- k game, where $k \geq 2$, Maker wins if her graph at the end of the game has arboricity at least k , *i.e.*, Maker's graph cannot be decomposed into less than k forests. A special case of this game for $k = 2$ is particularly interesting as this turns out to be the cycle game (see, *e.g.*, [4]), the game in which Maker's goal is to claim a cycle.

Proposition 4.1. *For any graph G , Maker wins the arboricity- k game in G if and only if $k \leq \lceil ar(G)/2 \rceil$.*

Proof. Let $\ell := ar(G)$. It was shown in the proof of [27, Proposition 17] that Breaker can ensure that Maker's graph at the end of the game has arboricity at most $\lceil \ell/2 \rceil$. On the other hand, if we denote Maker's and Breaker's graphs at the end of the game by M and B , respectively, then this defines a 2-partition of the edge set of G , and so, $ar(M) + ar(B) \geq \ell$.

We can apply strategy stealing to ensure the existence of a strategy for Maker that ensures her graph has high arboricity. Indeed, suppose that Breaker has a strategy to keep Maker's arboricity strictly below $\ell/2$. Then, by strategy stealing, Maker as the first player can apply the same strategy to keep Breaker's arboricity strictly below $\ell/2$. But, these two strategies put against each other would violate $ar(M) + ar(B) \geq \ell$, a contradiction. Hence, Maker can ensure that $ar(M) \geq \ell/2$ at the end of the game. Since $ar(M)$ is an integer, we are done. \square

Hence, to determine the outcome of the arboricity- k game, it is enough to determine $ar(G)$, which can be done in polynomial time [28], completing the proof of the following statement.

Theorem 4.2. *It can be decided in polynomial time if Maker wins the k -arboricity game in a given graph G , for any $k \geq 2$.*

Remark. In the proof of Proposition 4.1, we use strategy stealing to prove the existence of a strategy for Maker, but this does not provide us with an explicit strategy. However, we can use the Nash-Williams arboricity theorem which gives $ar(G) = \left\lceil \max_{G' \subseteq G} \frac{|E(G')|}{|V(G')|-1} \right\rceil$. Therefore, there exists $G' \subseteq G$ with $\frac{|E(G')|}{|V(G')|-1} > ar(G) - 1$. If Maker restricts her playing to G' for as long as there are unclaimed edges in G' , she will claim at least $|E(G')|/2$ edges in G' . This ensures that $ar(M) \geq \frac{\ell}{2}$, which makes this explicit strategy optimal.

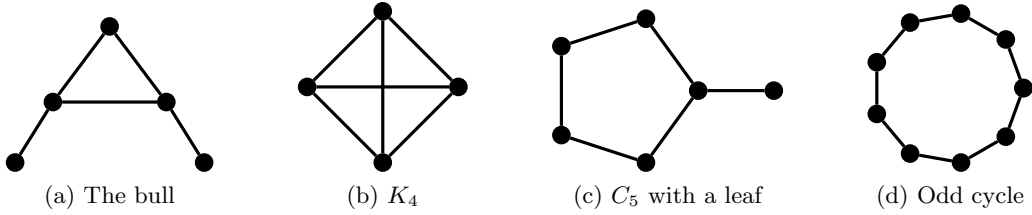


Figure 13: Maximal non-bipartite graphs for which Breaker wins the P_4 -game.

4.2 Linear-time algorithm for the P_4 -game

This subsection is dedicated to the P_4 -game where P_4 is the path with three edges. The algorithm of Galliot *et al.* in the framework of Maker-Breaker 3-uniform positional games [15] does not give any structural idea about the graphs in which Maker (Breaker, resp.) wins, and it runs in time $O(|V(\mathcal{H})|^5 E(\mathcal{H})^2 + V(\mathcal{H})^6 \Delta(\mathcal{H}))$, where \mathcal{H} is the hypergraph for the game. We give a necessary and sufficient structural condition for Breaker to win the P_4 -game in any graph G . This leads to a linear-time algorithm to decide the outcome of the P_4 -game.

Theorem 4.3. *For any connected graph G , Breaker wins the P_4 -game in G if and only if*

1. G is bipartite and all the vertices of degree at least 3 are in the same part; or
2. G is an odd cycle; or
3. G is a subgraph of the bull, K_4 , or a C_5 with a leaf attached to one vertex (see Figure 13).

Proof. We first prove the “if” part. One can check with a small case analysis that Breaker wins in the first three graphs of Figure 13, and thus, in all their subgraphs by Lemma 2.1. For the odd cycle, Breaker wins with the following strategy. Let $(e_1, \dots, e_{2\ell+1})$ be the edges of the cycle in order. W.l.o.g., let e_1 be the first edge claimed by Maker. Then, Breaker claims $e_{2\ell+1}$, and then follows a pairing strategy by pairing edges e_{2i} with e_{2i+1} , for $1 \leq i < \ell$ ($e_{2\ell}$ is not paired).

Thus, we can assume that G is bipartite with all the vertices of degree at least 3 in the same part. Let $V(G) = A \cup B$ be a bipartition of the vertices of G with the part A containing all the vertices of degree at least 3, and so, the vertices of B have degree at most 2. Note that any path on four vertices must contain an inner vertex in B , and thus, two edges incident to the same vertex in B . Thus, Breaker can win by following a pairing strategy where, for each vertex v of B , the (potentially) two edges incident to v are paired together.

We now prove the “only if” part. Let G be a graph that does not satisfy the three conditions of Theorem 4.3. The proof is divided into four cases:

1. G contains a diamond (K_4 minus one edge).
2. G contains a triangle, but not a diamond.
3. G contains an odd path between two vertices of degree at least 3, but not a triangle.
4. G contains an odd cycle with a unique vertex of degree at least 3.

These cases will cover all the possible graphs where Maker wins. Indeed, if G is bipartite, then it would be treated in Case 3. If G is not bipartite and not an odd cycle, then it has a vertex of degree at least 3 in an odd cycle C . If C contains a unique vertex of degree at least 3, then we are in Case 4. Otherwise, C contains two vertices of degree at least 3, and thus, there is an odd path between vertices of degree at least 3, and we are in Case 3 if G is triangle-free. Otherwise, G contains a triangle, but not a diamond (Case 2) or G contains a diamond (Case 1).

Case 1. G contains a diamond. Since G is not restricted to a subgraph of K_4 , G contains a subgraph that is a diamond with a leaf connected to it (either to a vertex of degree 3 or to a vertex of degree 2 in the diamond). One can easily check that Maker wins the P_4 -game in these subgraphs, and thus, by Lemma 2.1, she wins in all the graphs containing them.

Case 2. G contains a triangle, but not a diamond. Let uvw be this triangle. Since G is not a triangle (it would be a subgraph of the bull), at least one vertex, say u , must be connected to another vertex z . Note that z is not connected to any other vertex of the triangle, as otherwise G would contain a diamond. Since G is not restricted to the graph on the vertices $\{u, v, w, z\}$ (because otherwise it would be a subgraph of K_4), there is another vertex x in the graph.

Assume first that x is connected to z . Then, Maker claims zu . Then, she has a pairing strategy with pairs (xz, vw) and (uv, uw) . Assume now that x is connected to u . Then, Maker claims vu , and then follows a pairing strategy with pairs (xu, zu) and (uv, uw) . Finally, assume that x is connected to the triangle, without loss of generality, by v . Since G is not a bull and does not contain a diamond, there must either be an additional edge, and it can only be xz (otherwise G would contain a diamond) or there is another vertex connected to the graph. The first case returns to the case where x was connected to z . For the second case, the only possibility not covered yet is that there is a vertex t connected to w . Then, Maker claims uw . If Breaker then claims vx or vu , then Maker claims uz , and pairs tw with wv . Otherwise, w.l.o.g., Breaker then claims uz . Then, Maker claims uv and pairs tw with vx .

From now on, we can assume that G is triangle-free. Thus, Maker just needs to claim any three consecutively incident edges and will be sure to obtain a P_4 .

Case 3. G contains an odd path between two vertices of degree at least 3, but not a triangle. Let u and v be two vertices of degree at least 3 connected by an odd path P . We choose u and v such that P has minimum length. Let $e_1, \dots, e_{2\ell+1}$ be the edges of P with e_1 incident to u , and $e_{2\ell+1}$ incident to v . Let e_u and e'_u be the two other edges incident to u , and e_v and e'_v the two other edges incident to v .

If $\ell = 0$, then Maker claims e_1 and follows a pairing strategy with pairs (e_u, e'_u) and (e_v, e'_v) (e_u and e'_u are vertex-disjoint from e_v and e'_v since G is triangle-free). Assume now that $\ell > 0$. Maker starts by claiming e_2 . Assume first Breaker does not answer by claiming e_1 . Then, Maker can claim e_1 as her second move. Then, either she pairs e_u with e'_u if Breaker did not claim any of these edges on his first move, or she pairs e_u or e'_u (the one that is unclaimed) with e_3 . Therefore, we can assume that Breaker answers by claiming e_1 .

Assume now, by induction on $1 < i < \ell$, that before their i^{th} moves, Maker has claimed all the even edges e_{2j} , and Breaker has claimed all the odd edges e_{2j-1} for $j = 1, \dots, i-1$ (we

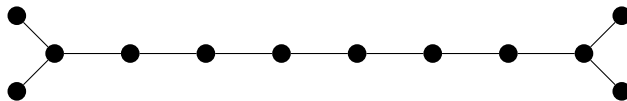


Figure 14: A tree where Maker wins the P_4 -game, but must play disconnected. By adding an even number of degree-2 vertices in the middle path, this gives a family of trees where the number of moves to win for Maker is unbounded.

have shown this is true for $i = 2$ above). Then, on her i^{th} move, Maker claims e_{2i} . Breaker has to answer by claiming e_{2i-1} , since otherwise Maker can claim e_{2i-1} , creating a P_4 with e_{2i} and e_{2i-2} . Then, the inductive hypothesis holds for $i + 1$. Repeating this argument, for her ℓ^{th} move, Maker claims $e_{2\ell}$, and Breaker has to answer by claiming $e_{2\ell-1}$. Then, Maker claims $e_{2\ell+1}$ and pairs e_v with e'_v , which will create a P_4 .

Case 4. G contains an odd cycle with a unique vertex of degree at least 3. Let u be the unique vertex of degree at least 3 in an odd cycle in G . Let $e_1, \dots, e_{2\ell+1}$ be the edges of the cycle, with u incident to e_1 and $e_{2\ell+1}$. Let v be a vertex adjacent to u , but not in the cycle (it exists since u has degree at least 3). Since G is triangle-free, $\ell > 1$. Assume first that $\ell = 2$. Since G is not restricted to a C_5 with a leaf, and since any additional edge will create another vertex of degree at least 3 in the cycle, there must be another vertex w in the graph. If w is adjacent to u , then Maker claims e_2 . Then, as before for the odd path, Breaker should answer by claiming e_1 . Then, Maker claims e_4 , and Breaker should answer by claiming e_3 . Finally, Maker claims e_5 , and then pairs uv with uw . If w is adjacent to v , then Maker claims uv . Breaker should answer by claiming vw , as otherwise Maker can make a P_3 with two free extremities. Then, Maker can force moves by claiming e_2 (Breaker then claims e_1), then e_4 (Breaker then claims e_3), and then win by claiming e_5 .

Assume now that $\ell > 2$, *i.e.*, the cycle has length at least 7. Maker claims e_4 . Breaker should claim either e_3 or e_5 to avoid a P_3 with two free extremities. If, w.l.o.g., Breaker claims e_3 , Maker can again force moves by claiming the even edges $e_6, \dots, e_{2\ell}$. Breaker always has to answer by claiming the preceding odd edges $e_5, \dots, e_{2\ell-1}$. Then, Maker claims $e_{2\ell+1}$ and pairs e_1 with uv , making a P_4 . \square

The conditions given in Theorem 4.3 are checkable in linear time, which implies the following:

Corollary 4.4. *The P_4 -game can be solved in linear time in any graph.*

Solving the P_ℓ -game for $\ell > 4$ seems difficult, even for trees. Indeed, already for $\ell = 4$, there are some trees for which Maker needs an unbounded number of moves to win and needs to play disconnected. An example is given in Figure 14, where there can be as many (but an even number of) vertices of degree 2 as one wants in the middle path. This can be generalized for the P_ℓ -game for any $\ell \geq 4$.

However, if one extremity of the path is fixed, the game becomes easy. Consider the rooted path game in a tree T rooted in r : Maker wants to make the longest possible path that starts from r . Let $\ell_M(T, r)$ ($\ell_B(T, r)$, respectively) be the longest path starting from r that Maker can construct when she plays first (when Breaker plays first, resp.).

Lemma 4.5. *Let T be a tree rooted in r , x_1, \dots, x_s the children of r , and T_1, \dots, T_s the subtrees rooted in x_1, \dots, x_s . W.l.o.g., assume that $\ell_B(T_1) \geq \ell_B(T_2) \geq \dots \geq \ell_B(T_s)$. Then,*

$\ell_M(T, r) = \ell_B(T, r) = 0$ if r is the single vertex in T . Otherwise:

$$\ell_M(T, r) = 1 + \ell_B(T_1, x_1)$$

$$\ell_B(T, r) = 1 + \ell_B(T_2, x_2).$$

In particular, they can be computed in linear time.

Proof. Consider first $\ell_M(T, r)$. Maker can construct a path of length $1 + \ell_B(T_1, x_1)$ by claiming the edge rx_1 , and then following her strategy as the second player in T_1 rooted in x_1 . If Maker claims rx_i as her first edge, then Breaker can assure her path has length at most $1 + \ell_B(T_i, x_i) \leq 1 + \ell_B(T_1, x_1)$ by following his strategy in T_i rooted in x_i .

Consider now Breaker as the first player. By claiming rx_1 , the game is then equivalent to the game in the tree T' where T_1 has been removed and Maker starts. As before, Maker will claim a path of length $1 + \ell_B(T_2, x_2)$. Breaker cannot do better: if he claims another edge that is not in T_1 , then Maker claims rx_1 and can construct a path of length $1 + \ell_B(T_1, x_1)$, which is worse for Breaker. If he claims an edge $e = uv$ in T_1 , the game is equivalent to the game played in T'' , the tree obtained by cutting the tree on the edge e and keeping the part containing r . Since T'' contains T' , Maker will be able to construct a path of length at least $1 + \ell_B(T_2, x_2)$. \square

4.3 $K_{1,\ell}$ -game in trees

In this section, we consider the $K_{1,\ell}$ -game in trees. Otherwise said, Maker needs to claim ℓ edges adjacent to the same vertex. We prove this game is tractable in trees.

Theorem 4.6. *For any integer $\ell \geq 1$, it can be decided in linear time whether Maker wins the $K_{1,\ell}$ -game in a tree.*

To prove this theorem, we need three structural lemmas. The first one is true for the $K_{1,\ell}$ -game in any graph.

Lemma 4.7. *Let G be a graph and $\ell \geq 1$ an integer. If G contains a vertex of degree at least $2\ell - 1$, then Maker wins the $K_{1,\ell}$ -game.*

Proof. Let u be a vertex of degree at least $2\ell - 1$. Maker claims any ℓ edges incident to u in her first ℓ moves. \square

Lemma 4.8. *For any integer $\ell \geq 1$, if T is a tree with maximum degree at most $2\ell - 2$ and at most one vertex of degree $2\ell - 2$, then Breaker wins the $K_{1,\ell}$ -game in T .*

Proof. Let r be a vertex of maximum degree (possibly $2\ell - 2$) and root T in r . We define a pairing strategy for Breaker as follows. For any vertex $u \in V(T)$, let v_1, \dots, v_{t_u} be its children, and pair together the edges $\{uv_{2i-1}, uv_{2i}\}$ for $i = 1$ to $i = \lfloor t_u/2 \rfloor$. Breaker, playing this pairing strategy, will claim at least half of the edges incident to r (so Maker will claim at most $\ell - 1$ edges incident to r) and, for any other vertex $w \in V(T)$, Breaker will claim at least $\lfloor \frac{t_w-1}{2} \rfloor$ edges incident to w . Since $t_w < 2\ell - 2$ as r is the only vertex that could have degree $2\ell - 2$, Maker will claim at most $\ell - 1$ edges incident to u . \square

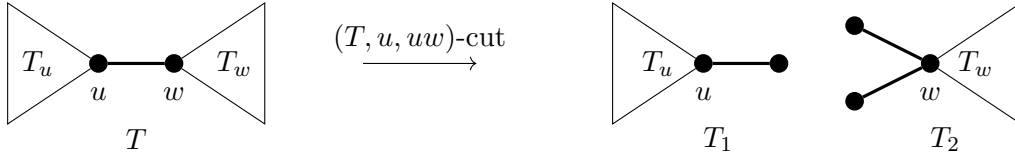


Figure 15: Illustration of a (T, u, uw) -cut. If u has degree $2k - 2$, then the $K_{1,\ell}$ -game in T or in the cut has the same outcome.

The next lemma will enable us to cut a tree into several components when there is a vertex of degree $2\ell - 2$. We first describe the cut operation we are using. Let T be a tree and let $uw \in E(T)$. Let T_u and T_w be the two trees composing the forest $T \setminus \{uw\}$, with $u \in V(T_u)$ and $w \in V(T_w)$. Let T_1 be the tree obtained from T_u by adding one pendent edge incident to u , and let T_2 be the tree obtained from T_w by adding two pendent edges incident to w . We call the forest $T_1 \cup T_2$ a (T, u, uw) -cut. See Figure 15 for an illustration.

Lemma 4.9. *For any integer $\ell \geq 1$, let T be a tree with a vertex u of degree $2\ell - 2$, w any neighbor of u , and $T_1 \cup T_2$ the (T, u, uw) -cut. Maker wins the $K_{1,\ell}$ -game in T if and only if she wins the $K_{1,\ell}$ -game in $T_1 \cup T_2$.*

Proof. Let e be the edge added to u in T_1 , and e_1, e_2 the two edges added to w in T_2 . Assume that Maker wins in T . In $T_1 \cup T_2$, Maker plays as follows. She pairs the edges e_1 and e_2 , and associates all the other edges with their corresponding edges in T (e is associated with uw). Then, she plays as in T . If Breaker claims e_1 or e_2 , then she claims the other edge. At the end of the game in T , there is a vertex x in T that has ℓ incident edges claimed by Maker. If $x \neq w$, it has the same ℓ incident edges at the end of the game in $T_1 \cup T_2$ (with possibly the edge uw replaced by e if $x = u$). If $x = w$, and Maker claimed the edge uw that is not present in T_2 , in the game in T , then Maker claimed the same ℓ edges in the game in $T_1 \cup T_2$ except that uw has been replaced by one of the edges e_1 or e_2 .

For the other direction, assume that Maker has a winning strategy in $T_1 \cup T_2$. By Lemma 2.2, she wins either in T_1 or in T_2 . Assume she wins in T_1 . Then, she can follow the same strategy in T without taking care of Breaker's moves in the rest of T (and considering that $e = uw$). Assume now she wins in T_2 . By Lemma 2.4, since e_1 and e_2 are interchangeable in any winning set that contains one of them, Maker also wins in $(T_2, \{e_1\}, \{e_2\})$, (i.e., T_2 where e_1 is claimed by Maker and e_2 by Breaker).

To win in T , Maker first claims uw . Then, Breaker should answer by claiming an edge incident to u since u has degree $2\ell - 2$ (otherwise, Maker can claim ℓ edges incident to u). Then, Maker follows her winning strategy in $(T_2, \{e_1\}, \{e_2\})$ in T_w . The unclaimed edges are in a one-to-one correspondence, and the vertices have the same number of edges claimed by Maker (that is, one for w , and 0 for the other vertices). Thus, Maker will win in T . \square

The next theorem gives a necessary and sufficient structural condition for Maker to win the $K_{1,\ell}$ -game in trees. This will imply Theorem 4.6 since it is easy to check if a tree has this structure (see the proof of Theorem 4.6 below).

Theorem 4.10. *Let T be a tree and $\ell \geq 1$ an integer. Maker wins the $K_{1,\ell}$ -game in T if and only if there is a subtree T' of T such that every vertex x of T' has degree at least $2\ell - 1 - d_{T'}(x)$ in T , where $d_{T'}(x)$ denotes the degree of x in T' .*

Note that if T' is reduced to a single vertex, it must have degree $2\ell - 1$ in T , which corresponds to Lemma 4.7. Otherwise, all the leaves of T' must be of degree $2\ell - 2$ in T . The theorem says that these leaves must be connected by vertices of sufficiently large degree.

Proof. We prove the equivalence by induction on the number of inner vertices of T (i.e., vertices of degree at least 2). Assume that T has only one inner vertex u . The only possibility for T' is to be the single vertex u , and Maker wins if and only if u has degree $2\ell - 1$. Thus, the equivalence is true.

Assume now, by induction, that the result is true for any tree with at most $i - 1 \geq 1$ inner vertices. Let T be a tree with i inner vertices. If T has at least one vertex u of degree at least $2\ell - 1$, then take T' to be the single vertex u , and Maker wins by Lemma 4.7. If T has maximum degree at most $2\ell - 2$, and T contains at most one vertex of degree $2\ell - 2$, it is not possible to have a subtree T' satisfying the property. Indeed, since T' cannot be a single vertex, it must have at least two leaves of degree $2\ell - 2$ in T . By Lemma 4.8, Breaker wins, and so, the equivalence is true.

Thus, we can assume that T has at least two vertices of degree exactly $2\ell - 2$. Let u and v be two vertices of degree $2\ell - 2$ at maximum distance from each other. Let w be the neighbor of u on the path between u and v . Let $T_1 \cup T_2$ be the (T, u, uw) -cut. Note that, by the maximality of the distance between u and v , the vertex u is the only vertex of T_1 of degree $2\ell - 2$. In particular, Breaker wins in T_1 by Lemma 4.8. By Lemma 4.9, Maker wins in T if and only if she wins in $T_1 \cup T_2$. Then, since Breaker wins in T_1 , by Lemma 2.2, Maker wins in T if and only if she wins in T_2 . The tree T_2 has strictly less than i inner vertices, and thus, by induction, Maker wins in T_2 if and only if there is a tree T'_2 such that every vertex of degree t in T'_2 has degree $2\ell - 1 - t$ in T_2 .

Assume Maker wins in T . This means that such a T'_2 exists. If $w \notin V(T'_2)$, then it is a valid tree T' for T . If $w \in V(T'_2)$, then T'_2 does not contain the two pendent edges incident to w (since no leaf of T_2 can belong to T'_2). Then, let T' be the subtree of T obtained from T'_2 by adding u . Note that T'_2 cannot have leaves of T as vertices, and thus, all its vertices are inner vertices of T . Since u is a leaf of T' and has degree $2\ell - 2$ in T , it satisfies the degree condition. This is also true for all the other vertices of T' except w . Let t_2 be the degree of w in T'_2 . By hypothesis, its degree in T_2 , $d_{T_2}(w)$, is at least $2\ell - 1 - t_2$. The vertex w has degree $t_2 + 1$ in T' and $d_{T_2}(w) - 1$ in T . Thus, $d_T(w) = d_{T_2}(w) - 1 \geq 2\ell - 1 - t_2 - 1 = 2\ell - 1 - d_{T'}(w)$, and T' satisfies the property.

For the other direction, assume that there is a subtree T' valid in T . Let uw' be the pendent edge in T_1 (i.e., $w' \notin V(T)$). If T' is a subtree of $T_1 \setminus \{w'\}$, then, by induction, Maker wins in T_1 , and, by Lemmas 2.2 and 4.9, she wins in T . Otherwise, let $V(T'_2) = V(T') \cap V(T_2)$. As before, one can prove that the degree condition is correct for each vertex in T'_2 in T_2 (indeed, if u was in T' , then w has one less neighbor in T'_2 than it did in T' , but this is compensated by the fact that w has one more neighbor in T_2). Thus, by induction, Maker wins in T_2 , and so, by Lemmas 2.2 and 4.9, Maker wins in T .

Therefore, the equivalence is true for T , and, by induction, is true for all T . \square

This characterization implies a linear-time algorithm for this game.

Proof of Theorem 4.6. To find such a T' , one can do a breadth-first search starting from any vertex of T . Label each vertex with its degree in T . Then, consider each vertex x , starting

from the deepest level. If x has label $2\ell - 2$, then increment the label of the parent of x by 1, and otherwise, do nothing. If at some point a vertex is labeled with $2\ell - 1$, then Maker wins, and otherwise, Breaker wins. Note that when we increment the label of a vertex, it corresponds to the cut operation. Precisely, if a vertex v receives a label $2\ell - 1$, let T_v be the subtree rooted in v . Then, a subtree T' satisfying the requirement is the inclusion-minimal subtree of T_v containing v and whose leaves ($\neq v$) have degree $2\ell - 2$ in T . \square

Whether the $K_{1,\ell}$ -game is polynomial-time solvable for general graphs is still open, but we prove in the next section that it is FPT parameterized by the length of the game.

5 FPT Algorithms

In this section, we consider the H -game parameterized by the length of the game. That is, given a fixed integer $k \in \mathbb{N}$, a fixed graph H , and a graph G as input, the problem consists in deciding whether Maker manages to claim edges of G inducing a copy of (a supergraph of) H in at most k moves (we say that Maker can *create* H in G in at most k moves). In particular, we prove that determining the outcome for the $K_{1,\ell}$ -game in any graph and the H -game in any tree are both FPT parameterized by the length of the game. The next theorem is crucial for both of these results, and is interesting on its own since it could lead to other positive results for the H -game, but first we need to introduce some notation.

For any graph G , $e = uv \in E(G)$, and $r \in \mathbb{N}$, let $B_G(e, r) = \{ww' \in E(G) \mid \max\{\text{dist}_G(u, w), \text{dist}_G(v, w), \text{dist}_G(u, w'), \text{dist}_G(v, w')\} \leq r\}$. Given a graph G and $X, Y \subseteq E(G)$ (with $|X| = |Y| + 1$), Breaker wins the H -game in position (G, X, Y) in $i \geq 1$ moves if it is Breaker's turn and Maker cannot create H in at most $i - 1$ moves after Breaker's next move.

Theorem 5.1. *Let H be a connected graph, G any graph, and k a positive integer. Maker wins the H -game in G if and only if there exists $e \in E(G)$ such that Maker wins the H -game in $G[B_G(e, 3^k)]$ in at most k moves.*

Proof. First, if there exists $e \in E(G)$ such that Maker wins the H -game in $G[B_G(e, 3^k)]$ in at most k moves, then Maker wins the H -game in G in at most k moves by Lemma 2.1. Now, assume that Breaker wins the H -game in $G[B_G(e, 3^k)]$ in at most k moves for each $e \in E(G)$. We describe a winning strategy for Breaker in the H -game in G that takes at most k moves. Let e_1 be the first edge claimed by Maker, and let $G_1 = G[B_G(e_1, 3^k)]$. Note that Breaker wins the H game in $(G_1, \{e_1\}, \emptyset)$ in at most k moves by the initial assumption. In particular, if Maker always claims an edge in G_1 , then Breaker wins.

First, Breaker answers to Maker claiming e_1 by following his winning strategy in G_1 . Let $2 \leq i \leq k$ be the i^{th} round of the game, before the i^{th} move of Maker, and let $M_i = \{e_1, \dots, e_{i-1}\}$ be the edges claimed by Maker, and $B_i = \{f_1, \dots, f_{i-1}\}$ the edges claimed by Breaker. Assume, by induction on i , that there exist edge-disjoint subgraphs G_1, \dots, G_{s_i} such that:

- for every $1 \leq j \leq s_i$, there exists $e_j \in E(G)$ such that $E(G_j) \subseteq B_G(e_j, 3^k)$;
- for every $e \in M_i$, there exists a unique $1 \leq j_e^i \leq s_i$ such that $e \in E(G_{j_e^i})$ and, moreover, $B_G(e, 3^{k-i+1}) \subseteq E(G_{j_e^i})$;

- for every $1 \leq j \leq s_i$, Breaker wins the H -game in $(G_j, E(G_j) \cap M_i, E(G_j) \cap B_i)$ in at most $k - i + 1$ moves.

The inductive hypothesis holds if $i = 2$ by remarks above (in particular, $s_2 = 1$). Assume that the inductive hypothesis holds for $i \geq 2$. Let e_i be the i^{th} edge claimed by Maker.

- If there exists $1 \leq j \leq s_i$ such that $B_G(e_i, 3^{k-i}) \subseteq E(G_j)$, then Breaker answers by following his winning strategy in G_j . Note that, in this case, j is unique since G_1, \dots, G_{s_i} are edge-disjoint subgraphs by the inductive hypothesis for i . Then, the inductive hypothesis holds for $i + 1$ with the same subgraphs G_1, \dots, G_{s_i} (in particular, $s_i = s_{i+1}$).
- If $B_G(e_i, 3^{k-i}) \cap E(G_j) = \emptyset$ for all $1 \leq j \leq s_i$, then let $s_{i+1} = s_i + 1$ and $G_{s_{i+1}} = G[B_G(e_i, 3^{k-i})]$. Then, Breaker answers by following his winning strategy in $G_{s_{i+1}}$, which exists by the assumption that Breaker wins the H -game in $G[B_G(e, 3^k)]$ in at most k moves for each $e \in E(G)$, and since $G_1, \dots, G_{s_{i+1}}$ are edge-disjoint subgraphs by the inductive hypothesis for i and the case we are in. Indeed, by Lemma 2.1, Breaker has a winning strategy in $G[B_G(e_i, 3^{k-i})]$ in $k - i$ moves since he has one in $G[B_G(e_i, 3^k)]$ in k moves (if Maker cannot create H in k moves, then she clearly cannot do it in $k - i$ moves in a subgraph). Then, the inductive hypothesis holds for $i + 1$.
- Lastly, if there exists $\emptyset \neq J \subseteq \{1, \dots, s_i\}$ such that, for all $j \in J$, $B_G(e_i, 3^{k-i}) \cap E(G_j) \neq \emptyset$ and $B_G(e_i, 3^{k-i}) \setminus E(G_j) \neq \emptyset$, then let $s_{i+1} = s_i + 1$ and $G_{s_{i+1}} = G[B_G(e_i, 3^{k-i})]$. Now, for every $j \in J$, let $E_j = \{f \in E(G_j) \mid B(f, 2 \cdot 3^{k-i}) \not\subseteq E(G_j)\}$. Note that, for every $f' \in M_i$ and $j \in J$, $E_j \cap B(f', 3^{k-i}) = \emptyset$ by the second assumption of the inductive hypothesis for i . For all $j \in J$, let $G_j = G[E(G_j) \setminus E_j]$ (intuitively, the edges of G_j that are “too close” to $G_{s_{i+1}}$ are removed from G_j), and note that $G_1, \dots, G_{s_{i+1}}$ are now edge-disjoint subgraphs since G_1, \dots, G_{s_i} were edge-disjoint subgraphs by the inductive hypothesis for i . Now, Breaker plays his next move according to his winning strategy in $G_{s_{i+1}}$, which, as in the previous case, exists by the assumption that Breaker wins the H -game in $G[B_G(e, 3^k)]$ in at most k moves for each $e \in E(G)$, and since $G_1, \dots, G_{s_{i+1}}$ are edge-disjoint subgraphs. Then, the inductive hypothesis holds for $i + 1$.

The inductive hypothesis and the strategy described above guarantee that Breaker wins, *i.e.*, Maker cannot win in G in at most k moves. \square

With Theorem 5.1 in hand, we now have one of the main tools to prove our FPT results, which rely on the fact that we only need to consider the ball (of edges) of bounded diameter in the length of the game centered at the first edge claimed by Maker.

Corollary 5.2. *The $K_{1,\ell}$ -game, for any fixed integer $\ell \geq 1$, is FPT parameterized by the length of the game.*

Proof. Consider the $K_{1,\ell}$ -game, for a positive constant ℓ (recall that H is a fixed graph in the H -game), in any graph G . Let k be the length of the game. If there is a vertex of degree at least $2\ell - 1$, then Maker wins in ℓ moves by Lemma 4.7. Hence, we can assume that the maximum degree is at most $2\ell - 2$. By Theorem 5.1, Maker wins in G in k moves if and only if she wins in k moves in one of the balls $B(e, 3^k)$ for some edge $e \in E(G)$. Since $\Delta(G) \leq 2\ell - 2$,

for any edge $f \in E(G)$, the ball $B(f, 3^k)$ has size at most $(2(\Delta(G) - 1))^{3^k} = (4\ell - 6)^{3^k}$, *i.e.*, a function of k since ℓ is a constant. Therefore, one can check if Maker wins by first checking the maximum degree, and then checking the outcomes of all possible games in the $|E(G)|$ balls (of edges) of diameter 3^k which have size bounded by a function $f(k)$. Indeed, this leads to an FPT algorithm since, in any graph of size bounded by a function $f(k)$, the output of the H -game in at most k moves can be determined by an exhaustive search in time $f'(k)$ for some computable function f' (the length of the game is k , and the number of possible moves at each step is at most the number of edges which is at most $f(k)$). \square

Theorem 5.1 combined with the particular structure of trees leads to the following result.

Theorem 5.3. *Let H be any tree. Then, the H -game is FPT parameterized by the length of the game in trees.*

Proof. We prove that the H -game parameterized by the length of the game k admits a kernel in trees. That is, given a tree T , we build, in polynomial time, a forest F of size at most a function of k (precised below) such that Maker wins in T in at most k moves if and only if there exists a connected component of F in which Maker wins in at most k moves.

First, for every edge $e \in E(T)$, let T_e be the subtree of T induced by the edges at distance at most 3^k from e , *i.e.*, T_e is the subtree induced by $B(e, 3^k)$. Let F be the forest that consists of the disjoint union of the T_e 's, $e \in E(T)$. By Theorem 5.1, Maker wins in T in at most k moves if and only if there exists $e \in E(T)$ such that Maker wins in T_e in at most k moves.

The *depth* of a rooted tree is the maximum distance from its root to a leaf. For every $e \in E(T)$, let us root T_e in such a way that it has depth $d_e \leq 3^k$ (this is possible by the definition of T_e). A vertex $v \in V(T_e)$ has *level* $i \geq 0$ if the subtree of T_e rooted in v has depth i . Let us iteratively, for $i = 1$ to d_e , replace T_e^{i-1} ($T_e = T_e^0$) by a tree T_e^i such that: Maker wins in T_e^{i-1} in at most k moves if and only if Maker wins in at most k moves in T_e^i ; and, for every vertex $v \in V(T_e^i)$ at level i , the subtree of T_e^i rooted in v has size at most $n_i(k)$ (a function of k whose recursive definition is given below).

First, for $i = 1$, for every vertex v at level 1 in T_e , (*i.e.*, all children of v are leaves), if v has more than $2k$ children, then remove all but $2k$ of its children. Let T_e^1 be the obtained tree. By construction, every vertex $v \in V(T_e^1)$ at level 1 is the root of a subtree of size at most $n_1(k) = 2k + 1$. Moreover, since, for every vertex v at level 1 in T_e , at most $2k$ edges between v and leaves can be claimed (as the length of the game is k), then the output of the H -game is the same in T_e and T_e^1 .

Now, by induction on $i \geq 1$, let us assume that we have built a tree T_e^i such that Maker wins in T_e^i in at most k moves if and only if Maker wins in at most k moves in T_e ; and, for every vertex $v \in V(T_e^i)$ at level i , the subtree of T_e^i rooted in v has size at most $n_i(k)$. Let $g_i(k)$ be the number of rooted trees of depth at most i and of size at most $n_i(k)$. For every $v \in V(T_e^i)$ at level $i + 1$, let S_1, \dots, S_r be the subtrees rooted in the children of v (note that each of these subtrees has depth at most i and size at most $n_i(k)$). For every possible rooted subtree S of depth i and size at most $n_i(k)$, if there are more than $2k$ copies of S in the multiset of trees $\{S_1, \dots, S_r\}$, then remove all but $2k$ copies of S . Let T_e^{i+1} be the resulting tree (after having done the above process for every vertex at level $i + 1$ of T_e^i). In T_e^{i+1} , every vertex at level $i + 1$ has at most $2kg_i(k)$ children and those children are the roots of subtrees of size at most $n_i(k)$, and hence, every vertex at level $i + 1$ is the root of a subtree of size at most $n_{i+1}(k) = 2kg_i(k)n_i(k)$. Moreover, for every vertex v at level $i + 1$ in T_e^i , at most $2k$ edges in

the subtree rooted at v can be claimed. Therefore, the output of the H -game is the same in T_e^i and T_e^{i+1} .

After the above process has been done for $i = d_e \leq 3k$ for each subtree T_e , $e \in E(T)$, F consists of the disjoint union of the trees T_e^{3k} , $e \in E(T)$, each of size at most $n_{3k}(k)$, and Maker wins in T in at most k moves if and only if she wins in at most k moves in some connected component of F . While two of these subtrees are isomorphic, let us remove one of the two isomorphic subtrees. This clearly preserves the fact that Maker wins the H -game in T in at most k moves if and only if she wins in at most k moves in some connected component of F . Moreover, eventually, F has size at most $g_{3k}(k)n_{3k}(k)$, *i.e.*, this is the desired kernel.

To conclude, this leads to an FPT algorithm since, in any graph of size bounded by a function $f(k)$, the output of the H -game in at most k moves can be determined by an exhaustive search in time $f'(k)$ for some computable function f' (the length of the game is k , and the number of possible moves at each step is at most the number of edges which is at most $f(k)$). \square

6 Further Work

As a first direction, it would be intriguing to study the algorithmic complexity of other classic Maker-Breaker games on edge sets of general graphs, such as the Hamiltonicity game. Regarding the H -game, it would be interesting to know the order and/or size of the smallest graph H for which the H -game is PSPACE-complete. Due to the positive results in trees, we wonder whether the H -game in trees can be solved in polynomial time when H is a path. As the H -game is PSPACE-complete in graphs of diameter at most 6 (Theorem 3.3), intuitively, the H -game should be W[1]-hard parameterized by the length of the game for many graphs H . Is this true or does bounding the length of the game have consequences, other than bounding the diameter of the graph, that may lead to FPT algorithms? Concerning the perfect matching game, are there interesting graph classes for which the outcome of the game is decidable in polynomial-time? Lastly, what about these games for other conventions like Avoider-Enforcer?

References

- [1] József Beck. Remarks on positional games. I. *Acta Mathematica Academiae Scientiarum Hungaricae*, 40:65–71, 1982.
- [2] József Beck. *Combinatorial games: Tic-Tac-Toe Theory*, volume 114 of *Encyclopedia of Mathematics and Its Applications*. Cambridge: Cambridge University Press, 2008.
- [3] Małgorzata Bednarska and Tomasz Luczak. Biased Positional Games for Which Random Strategies are Nearly Optimal. *Combinatorica*, 20(4):477–488, 2000.
- [4] Małgorzata Bednarska and Oleg Pikhurko. Odd and even cycles in Maker–Breaker games. *European Journal of Combinatorics*, 29(3):742–745, 2008.
- [5] Julien Bensmail, Foivos Fioravantes, Fionn Mc Inerney, Nicolas Nisse, and Nacim Oijid. The Maker-Breaker Largest Connected Subgraph Game. *Theoretical Computer Science*, 943:102–120, 2023.

- [6] Édouard Bonnet, Serge Gaspers, Antonin Lambilliotte, Stefan Rümmele, and Abdallah Saffidine. The Parameterized Complexity of Positional Games. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 90:1–90:14, 2017.
- [7] Édouard Bonnet, Florian Jamain, and Abdallah Saffidine. On the complexity of connection games. *Theoretical Computer Science*, 644:2–28, 2016.
- [8] Jesper Makholm Byskov. Maker-Maker and Maker-Breaker games are PSPACE-complete. *BRICS Report Series*, 11(14), 2004.
- [9] Vašek Chvátal and Paul Erdős. Biased positional games. In *Annals of Discrete Mathematics*, volume 2, pages 221–229. Elsevier, 1978.
- [10] András Csernenszky, Ryan R. Martin, and András Pluhár. On the Complexity of Chooser–Picker Positional Games. *Integers*, 12(3):427–444, 2012.
- [11] Eric Duchêne, Valentin Gledel, Aline Parreau, and Gabriel Renault. Maker-Breaker domination game. *Discrete Mathematics*, 343(9):111955, 2020.
- [12] Paul Erdős and John Lewis Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14(3):298–301, 1973.
- [13] Shimon Even and Robert Endre Tarjan. A Combinatorial Problem Which Is Complete in Polynomial Space. *Journal of the ACM*, 23(4):710–719, 1976.
- [14] Asaf Ferber and Dan Hefetz. Winning Strong Games through Fast Strategies for Weak Games. *The Electronic Journal of Combinatorics*, 18(1):#P144, 2011.
- [15] Florian Galliot, Sylvain Gravier, and Isabelle Sivignon. Maker-Breaker is solved in polynomial time on hypergraphs of rank 3. *CoRR*, abs/2209.12819, 2022. <https://arxiv.org/abs/2209.12819>.
- [16] Heidi Gebauer and Tibor Szabó. Asymptotic random graph intuition for the biased connectivity game. *Random Structures & Algorithms*, 35(4):431–443, 2009.
- [17] Valentin Gledel and Nacim Oijid. Avoidance games are PSPACE-complete. In *Proceedings of the 40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023)*, Leibniz International Proceedings in Informatics (LIPIcs), 2023.
- [18] Philip Hall. On representatives of subsets. *Journal of the London Mathematical Society*, 10(1):26–30, 1935.
- [19] Dan Hefetz, Michael Krivelevich, Miloš Stojaković, and Tibor Szabó. Fast winning strategies in Maker-Breaker games. *Journal of Combinatorial Theory, Series B*, 99(1):39–47, 2009.
- [20] Dan Hefetz, Michael Krivelevich, Miloš Stojaković, and Tibor Szabó. A sharp threshold for the Hamilton cycle Maker–Breaker game. *Random Structures & Algorithms*, 34(1):112–122, 2009.

- [21] Dan Hefetz, Michael Krivelevich, Miloš Stojaković, and Tibor Szabó. *Positional Games*, volume 44 of *Oberwolfach Seminars*. Birkhäuser/Springer, 2014.
- [22] Dan Hefetz and Sebastian Stich. On two problems regarding the Hamiltonian cycle game. *The Electronic Journal of Combinatorics*, 16(1):#R28, 2009.
- [23] Michael Krivelevich. The critical bias for the Hamiltonicity game is $(1+o(1))n/\ln n$. *Journal of the American Mathematical Society*, 24(1):125–131, 2011.
- [24] Alfred Lehman. A Solution of the Shannon Switching Game. *Journal of the Society for Industrial and Applied Mathematics*, 12(4):687–725, 1964.
- [25] Mirjana Mikalački and Miloš Stojaković. Fast strategies in biased Maker–Breaker games. *Discrete Mathematics & Theoretical Computer Science*, 20:2:#5, 2018.
- [26] Tobias Müller and Miloš Stojaković. A threshold for the Maker-Breaker clique game. *Random structures & algorithms*, 45(2):318–341, 2014.
- [27] Rajko Nenadov, Angelika Steger, and Miloš Stojaković. On the threshold for the Maker-Breaker H -game. *Random Structures & Algorithms*, 49(3):558–578, 2016.
- [28] Jean-Claude Picard and Maurice Queyranne. A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory. *Networks*, 12(2):141–159, 1982.
- [29] Md Lutfar Rahman and Thomas Watson. 6-uniform Maker-Breaker game is PSPACE-complete. In *Proceedings of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, volume 187 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:15, 2021.
- [30] Thomas Jerome Schaefer. On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16(2):185–225, 1978.
- [31] Miloš Stojaković and Tibor Szabó. Positional games on random graphs. *Random Structures & Algorithms*, 26(1–2):204–223, 2005.
- [32] Robert Endre Tarjan. Edge-disjoint spanning trees and depth-first search. *Acta Informatica*, 6(2):171–185, 1976.