The Raymond and Beverly Sackler
Faculty of Exact Sciences
Tel Aviv University

הפקולטה למדעים מדויקים
ע״ש ריימונד ובברלי סאקלר
אוניברסיטת תל אביב

The Raymond and Beverly Sackler
Faculty of Exact Sciences
School of Mathematical Sciences

Positional games

in various settings

Thesis submitted in partial fulfillment of the requirements for the degree "Master of Science" in mathematics

by

Adva Mond

Under the supervision of
Prof. Michael Krivelevich

September 2018

Abstract

Positional games are two-player games of perfect information. They are played on a finite set X, called the *board* of the game, equipped with a family of subsets $\mathcal{F} \subset 2^X$, called the *winning sets* of the game. Two players alternately claim unclaimed elements of X until all elements are claimed and then the winner is determined by the rules of the specific game. In this thesis, we mainly focus on a special class of positional games, so called *Maker-Breaker games*. In the $(1 : b)$ Maker-Breaker game $(X, \mathcal{F})$, two players, Maker and Breaker, alternately claim p and q unclaimed elements of $X = \cup_{F \in \mathcal{F}} F$ respectively. Maker wins in the game if by the time all elements have been claimed he has claimed all the elements of some $F \in \mathcal{F}$, and loses otherwise (there are no draws). Other classical positional games such as *Avoider-Enforcer*, *Waiter-Client* and *Client-Waiter* will also be discussed in this thesis and will be defined in the Introduction.

It is natural to play positional games on graphs. In this case, one can take the board of the game to be either the edges of the graph or its vertices. In both options each player's final graph is a subgraph of the original graph, where in the vertex case it is the graph induced by the vertices each player claimed. The winning sets are the subgraphs satisfying some given graph property. In this thesis we study both positional games played on the vertices of a graph, and those played on its edges.

Positional games played on the edge set of a graph are the more studied kind of games so far. One of the games that was studied in this setting is the so-called *odd-cycle game*, where the winning sets are all subgraphs containing a cycle of an odd length, where the board is $E(K_n)$. Investigating this game, we improve a result of Bednarska and Pikhurko: we show that Maker wins the $(1 : b)$ Maker-Breaker odd cycle game if $b \leq ((4 - \sqrt{6})/5 + o(1))n$. We then introduce new rules for various positional games played on the edges of a graph, which we call "connected rules", and study Maker-Breaker and Client-Waiter odd cycle games under these rules.

In the second part of this thesis we address positional games, played on a vertex set of a graph. Given a graph G, two players claim vertices of G, where the outcome of the game is determined by the subgraphs of G induced by the vertices claimed by each player (or by one of them). We study classical positional games such as Maker-Breaker games, Avoider-Enforcer games, Waiter-Client and Client-Waiter games, where the board of the game is the vertex set of the binomial random graph $G \sim G(n, p)$. Under these settings, we consider those games where the winning sets are all graphs containing a copy of a fixed graph H, that is, H-games, and focus on those cases where H is a clique or a cycle. We show that, similarly to the edge version of the game, there is a strong connection between the threshold probability for these games and the one for the vertex-Ramsey property (that is, the property that every 2-vertex-coloring of $G(n, p)$ spans a monochromatic copy of H). We also show that the cases where H is a triangle or a forest sometimes have a different behavior.

Acknowledgments

First and foremost, I wish to thank my advisor Professor Michael Krivelevich, for kindly guiding me in my very first steps in academia, for being always encouraging and supportive, for being a great teacher introducing me the world of positional games, as well as the world of academic research and how it should be done, for being a role model as a mathematician, and for always believing in me.

Second, I wish to thank Gal Kronenberg for being like a second advisor for me, for working with me patiently, for teaching me and guiding me through the path of writing a thesis, and for being a friend during the whole time.

I would like to thank the Combinatorics group: Noga Alon, Michael Krivelevich, Wojciech Samotij, and Asaf Shapira for teaching me, inspiring me, and for introducing me to the world of Combinatorics.

I want to thank my co-authors in the papers on which this thesis is based on: Jan Corsten, Gal Kronenberg, Alon Naor, Alexey Pokrovskiy, Christoph Spiegel, and Tibor Szabó, for a pleasant joint work experience and for all the qualities I learned from each and every one of them about being a researcher.

I would like to thank my friends and co-workers: Clara Shikhelman, Peleg Michaeli, Yinon Spinka, Michal Amir, Limor Friedman and Or Baruch for many valuable discussions, for creating a great workspace, and for moral support during struggles.

Last, I wish to thank my parents, my family and my friends for always being there, for supporting me and believing in me without doubt throughout all these years.

Contents

Chapter 1

Introduction

Positional games are finite, perfect information games with no chance moves, played by two players $\mathcal{A}$ and $\mathcal{B}$ (which usually have more informative names, in correspondence to the particular game in discussion). In its most general form, a positional game is a 4-tuple $(a, b, X, \mathcal{F})$, where a and b are two positive integers (called the *bias* of the players), X is a finite set (called the *board*) and $\mathcal{F} \subseteq 2^X$ is a family of subsets of X. The pair $(X, \mathcal{F})$ is referred to as the *hypergraph of the game*, and every member in $\mathcal{F}$ is called a *target set* (according to the game in discussion we sometimes refer to the target sets as either *winning sets* or *losing sets*). The definition of the game is complete by identifying the first player to move (when this is relevant) and by specifying the winning criteria in the game.

The course of the game goes as follows: the two players alternately claim previously unclaimed elements of the board (each such element is called *free*), until there are none left. In each round, $\mathcal{A}$ claims a elements, and $\mathcal{B}$ claims b elements. The last player to play may claim fewer elements than his bias, if not enough free elements remain. The most basic case is $a = b = 1$, the so-called *unbiased* game, while for all other choices of a and b, the game is called *biased*. Positional games have drawn much attention in the past decade, and numerous papers investigating them have been published. We refer the reader to the extremely thorough book on the subject by Beck [1], and to the more recent book by Hefetz, Krivelevich, Stojaković and Szabó [28].

The positional games discussed in this thesis cannot end in a draw. Hence, and given the nature of positional games in general, each such game must satisfy exactly one of the following: either $\mathcal{A}$ has a strategy to ensure his win (which works against **any** strategy of $\mathcal{B}$), or $\mathcal{B}$ has such strategy. Thus, we may (and systematically do) refer to every given positional game as either $\mathcal{A}$'s win or $\mathcal{B}$'s win.

It is natural to play positional games on the edge set of a graph G. In this case, $X = E(G)$ and $\mathcal{F}$ consists of all edge sets of subgraphs of G satisfying some monotone increasing graph property. Such a property could be, for example, "being connected and spanning" (the *connectivity* game), "containing a perfect matching" (the *perfect matching* game), "containing a Hamilton cycle" (the *Hamiltonicity* game), "containing a copy of a predetermined fixed graph H", and so on. The latter family of games is called H-games, and they are the subject of research of Chapter 3 in this thesis.

1.1 Maker-Breaker

Let us now define the main type of positional games investigated in this thesis, called *Maker-Breaker games*. In the $(a : b)$ Maker-Breaker $(X, \mathcal{F})$ game, the two players — who are now called

Maker and Breaker — take turns in claiming the elements of X. In every round Maker claims a board elements and Breaker claims b elements. Maker wins the game if he occupies all elements of some target set (a winning set in this case) by the end of the game; if he fails to do so, Breaker wins the game (so indeed, a draw is not possible).

Maker-Breaker games in general, and especially those who are played on graphs, are probably the most studied family of positional games. The most natural choice for the graph whose edges the players claim is K_n, the complete graph on n vertices. As it turns out, many natural games played on it, such as the connectivity, perfect matching and Hamiltonicity games, are drastically in favor of Maker: he wins these games in their unbiased version in (almost) the minimal number of moves required to fully claim a winning set (for more details and for similar results, see [26, 32, 37]). Therefore, in order to even out the odds and make these games more interesting, two main approaches are considered, many times simultaneously.

The first is to give Breaker a larger bias than that of Maker, and typically the $(1 : b)$ version is considered. An important (and quite easy for observation) property of Maker-Breaker games is that they are *bias monotone*: if Maker wins some game $\mathcal{F}$ with bias $(a : b)$, he also wins this game with bias $(a' : b')$, for every $a' \geq a$ and $b' \leq b$. In other words, no player can be harmed by claiming more elements per move. This bias monotonicity enables the definition of the *threshold bias* (also known as the *critical* bias): for a given hypergraph $\mathcal{F}$, its threshold bias b^* is the unique integer for which Maker wins the $(1 : b)$ game $\mathcal{F}$ if and only if $b < b^*$. For example, it was shown [12, 22, 35] that for the connectivity, perfect matching and Hamiltonicity games played on K_n, the threshold bias is $(1 + o(1))n/\ln n$. Bednarska and Łuczak analyzed H-games in [6] and showed that for any graph H (under the technical assumption that H contains at least two edges), the threshold bias for the H-game played on K_n satisfies $b^* = \Theta(n^{1/m_2(H)})$, where $m_2(H) := \max\{\frac{e(H)-1}{v(H)-2} \mid H' \subseteq H,\ v(H') \geq 3\}$.

The other main approach towards balancing Maker-Breaker games is to play on sparse graphs. Here the typical case study is that of random graphs, and specifically the binomial graph $G \sim G(n, p)$, a graph on n vertices where each of the $\binom{n}{2}$ potential edges is included with probability p, independently of all other edges. It is well known by the seminal (and more general) result of Bollobás and Thomason [11] that every monotone increasing graph property $\mathcal{P}$ has a *threshold probability*. Now, given a monotone increasing graph property and the bias of the players, it is easy to see that "being Maker's win" is a monotone increasing graph property as well (Maker's winning strategy for a graph G is applicable to any graph containing G on the same vertex set). This allows us to consider the *threshold probability of the game*, i.e., look for the turning point of the game, where $G \sim G(n, p)$ goes through a phase transition, from being w.h.p. (with high probability, that is, with probability tending to 1 as n tends to infinity) Breaker's win to being w.h.p. Maker's win.

In Chapter 3 we investigate the second approach when playing the Maker-Breaker H-game for a fixed graph H, and in Chapter 2 we investigate the first approach when playing the Maker-Breaker odd-cycle game in two different settings.

1.2 Avoider-Enforcer

We move on to a different type of positional games, called Avoider-Enforcer games, which are the misère version of Maker-Breaker games. An $(a, b, X, \mathcal{F})$ Avoider-Enforcer game is played in the same manner as the corresponding Maker-Breaker game: the two players, called Avoider and Enforcer, alternately claim a and b free elements of X per move, respectively. The difference is that the target sets are now losing sets, and so at the end of the game Avoider loses if he has fully claimed some

$F \in \mathcal{F}$, and wins otherwise.

Despite being closely related to Maker-Breaker games, Avoider-Enforcer games are unfortunately (and perhaps surprisingly) not bias monotone in general (see e.g. [25],[29]). Even though one may assume intuitively that no player can be harmed by claiming fewer elements per move, this is not always the case. This behavior motivated Hefetz, Krivelevich, Stojaković and Szabó to propose in [25] a bias monotone version for Avoider-Enforcer games: in the new version Avoider and Enforcer claim **at least** a and b elements per move, respectively. It is easy to see that this new version is indeed bias monotone, and no player can be harmed from lowering his bias. As it turns out, this change of rules may change dramatically the outcome of the game. We refer to the traditional and new sets of rules as the *strict* and *monotone* rules, respectively, and accordingly refer to either strict games or monotone games. For every monotone game there exists a threshold bias, defined in a similar way to that of Maker-Breaker games. However, for strict games it is only possible to define lower and upper threshold biases. We do not elaborate on that. For more information about the differences between the two sets of rules and about Avoider-Enforcer games in general, see for example [1, 25, 28, 29].

1.3 Waiter-Client and Client-Waiter

Waiter-Client and Client-Waiter games resemble Maker-Breaker and Avoider-Enforcer games and were introduced by Beck [1, 2] under the names Picker-Chooser and Chooser-Picker, respectively. Since the original names of the players were confusing, it is now conventional to use the new names Waiter and Client as suggested in [5]. As in Maker-Breaker and Avoider-Enforcer games, the parameters of these games are a set X, a family $\mathcal{F} \subseteq 2^X$, and two positive integers a and b which denote the bias of Client and Waiter, respectively (note that a denotes Client's bias even in the Waiter-Client game, which might be confusing).

The course of every round, however, is different. In an $(a : b)$ Waiter-Client game $(X, \mathcal{F})$, in every round Waiter selects $a+b$ previously unclaimed elements of X. Client then chooses a of those elements to claim and the remaining elements are claimed by Waiter. For the last round of the game, let $t \leq a+b$ be the number of free elements remaining. If $t \leq b$ then Waiter claims all these elements, and otherwise Client claims $t-b$ elements and the rest go to Waiter. Waiter wins if by the end of the game Client has claimed all elements of some $F \in \mathcal{F}$, and otherwise Client wins. We can think of Waiter as the *builder* of the game and of Client as the *spoiler*.

The definition of $(a : b)$ Client-Waiter games is similar, but with a few differences. In every round Waiter selects t free elements of X, where $a \leq t \leq a+b$, from which Client chooses a to claim, and the rest are claimed by Waiter. In the last round Client chooses a of the remaining elements to claim (and all others go to Waiter), or he claims all of them if less than a free elements remain. Client wins if by the end of the game he has claimed all elements of some $F \in \mathcal{F}$, and otherwise Waiter wins. In this game we can think of Client as the builder of the game and of Waiter as the spoiler.

The reason that Waiter may offer less than $a+b$ elements per move in the Client-Waiter game is that otherwise the game would not be monotone in Waiter's bias, as first observed by Bednarska-Bzdęga [3] (the game is monotone in Client's bias even without this relaxation). Waiter-Client games as defined here are monotone in Waiter's bias only. Since in the study of Waiter-Client (and Client-Waiter) games the typical case is that the bias of Client is 1, no similar adjustment of the rules is usually considered (although there exists one), including in this thesis.

1.4 Organization of the thesis

This thesis is based on two papers: *On the odd cycle game* by Corsten, Mond, Pokrovskiy, Spiegel and Szabó (see [15]), and *Positional games on the vertex set of random graphs* by Kronenberg, Mond and Naor (see [36]). In Chapter 2 we discuss the odd cycle games (played on the edges of K_n), and in Chapter 3 we discuss the positional H-games that played on the vertices of random graphs. In the beginning of each chapter we give the relevant background and tools.

1.5 Notation

Our graph-theoretic notation is standard and follows that of [46]. In particular we use the following. For a graph G, let $V = V(G)$ and $E = E(G)$ denote its set of vertices and edges, respectively. In addition denote $v(G) = |V|$ and $e(G) = |E|$. For a set of vertices $U \subseteq V$, let $G[U]$ denote the corresponding vertex-induced subgraph of G, and let $N_G(U) = \{v \in V \setminus U \mid \exists u \in U \text{ such that } uv \in E\}$ denote the external neighborhood of U in G. For a vertex $v \in V$ we abbreviate $N_G(\{v\})$ to $N_G(v)$ and let $d_G(v) = |N_G(v)|$ denote the degree of v in G. Often, when there is no risk of ambiguity, we omit the subscript G in the above notation.

Given a natural number n we denote by $[n]$ the set $\{1, \ldots, n\}$. For a vertex $v \in V(G)$ and a set of vertices $A \subset V(G)$ let $\deg(v, A)$ be the degree of v to A, that is, the number of neighbors of v in A. Moreover, given sets $A, B \subset V(G)$ we use $e(A, B)$ to denote the number of edges in G connecting a vertex of A with a vertex of B, and we use $e(A)$ to denote the number of edges in G between vertices of A.

Our results are asymptotic in nature and we assume that n is large enough where needed. We omit floor and ceiling signs whenever these are not crucial.

Chapter 2

Playing on the edges of the complete graph

2.1 Introduction

In this chapter we study the *odd cycle game* $\mathcal{OC}_n$, where the winning sets are all subgraphs of K_n containing a cycle of odd length, played on the edges of the complete graph K_n.

As mentioned in Chapter 1, it is very common to study Maker-Breaker games on graphs, and in particular when the board is the edge-set of some graph, usually the complete graph, and the winning sets are all subgraphs satisfying some given graph property. For example, in the *cycle game* $\mathcal{C}_n$, where Maker's goal is containing a copy of a cycle in his graph, the board of the game is $X = E(K_n)$ and the winning sets are all subgraphs of K_n containing a cycle (given by their edge-set). We consider biased Maker-Breaker games, where in each turn Maker claims one element of the board whereas Breaker claims b, for some bias $b \geq 1$, and we assume that Breaker claims his b elements in each turn an element after an element, instead of altogether. In this chapter we are interested in determining the threshold bias for several games and settings, and hence we use various notations for b^*, in order to be more accurate with respect to the specific game and setting we consider.

For the cycle game, Bednarska and Pikhurko proved in [8] that the threshold bias satisfies $b_{mb}(\mathcal{C}_n) = \lceil n/2 \rceil - 1$. Furthermore, Krivelevich [34] proved that for any $0 < \varepsilon < 1$ and large enouch n Maker can always build a linearly-long cycle when $b \leq (1/2 - \varepsilon)n$. In [7] Bednarska and Pikhurko discussed even-cycle and odd-cycle games. Since building a cycle of odd length is certainly more difficult for Maker than building just any cycle, we have $b_{mb}(\mathcal{OC}_n) \leq b_{mb}(\mathcal{C}_n) = \lceil n/2 \rceil - 1$. However, no upper bound separating the parameters $b_{mb}(\mathcal{OC}_n)$ and $b_{mb}(\mathcal{C}_n)$ is known yet. In [7], the authors proved that $b_{mb}(\mathcal{OC}_n) \geq (1 - 1/\sqrt{2} - o(1))n$ (approximately $(0.2929 - o(1))n$) and asked the following question.

Question 1. Do we have $b_{mb}(\mathcal{OC}_n) = (1/2 + o(1))n$?

We give the following small improvement to the lower bound.

Theorem 2.1.1. *Maker wins the* $(1 : b)$ *Maker-Breaker game for every* $\varepsilon > 0$*, every large enough* n *and every* $b \leq \left((4 - \sqrt{6})/5 - \varepsilon\right) n$*. In other words,* $b_{mb}(\mathcal{OC}_n) \geq \left((4 - \sqrt{6})/5 + o(1)\right) n$ *(approximately* $(0.3101 - o(1))n$*).*

Suppose we are playing the Maker-Breaker game $(E(K_n), \mathcal{F})$. Then, for every $t \geq 0$, we define

Maker's graph $G_M(t)$ after t rounds to be the graph on $V(K_n)$ consisting of all edges which Maker claimed in the first t rounds of the game. Maker's final graph, we simply denote by G_M. We further define $G_M^{ind}(t)$ and G_M^{ind} to be the induced subgraphs of $G_M(t)$ and G_M, i.e. the graphs formed by deleting all isolated vertices. Similarly we define Breakers graphs $G_B(t), G_B, G_B^{ind}(t)$ and G_B^{ind}.

2.1.1 Connected games

Walker-Breaker games are a well-studied variant of Maker-Breaker games on graphs, which were introduced by Espig, Frieze, Krivelevich and Pegden [17]. Walker-Breaker games are similar to Maker-Breaker games with the constraint that Walker has to choose his edges in each turn according to a walk. In particular, Walker's graph is always connected. We will now introduce another related natural variant in which Maker is allowed to claim edges incident to any vertex he has visited so far.

The $(1 : b)$ *connected Maker-Breaker game* $(E(K_n), \mathcal{F})$ is exactly as the $(1 : b)$ Maker-Breaker game $(E(K_n), \mathcal{F})$ except that, in every round $t \geq 2$, Maker is only allowed to claim edges which are incident to some vertex $v \in V\left(G_M^{ind}(t-1)\right)$. Equivalently, Maker has to ensure that $G_M^{ind}(t)$ is connected for every $t \geq 1$. If, at some point of the game, this is not possible any more, Maker loses the game. It is not hard to see that connected Maker-Breaker games are also bias-monotone. Thus, for every connected Maker-Breaker game $(E(K_n), \mathcal{F})$, there also exists a threshold bias $b^c_{mb}(\mathcal{F})$, so that Breaker wins the $(1 : b)$ game $(X, \mathcal{F})$ if and only if $b \geq b^c_{mb}(\mathcal{F})$.

Both Maker's strategy in the proof of Theorem 2.1.1 and in Bednarska's and Pikhurko's proof in [7] follow the connected rules. It is therefore natural to try to answer Question 1 under the additional assumption that Maker follows these rules. As connected games are a restriction for Maker, we have $b^c_{mb}(\mathcal{OC}_n) \leq b_{mb}(\mathcal{OC}_n) \leq \lceil n/2 \rceil - 1$ for every $n \in \mathbb{N}$. We give the following improvement, which shows that the answer to Question 1 is no if there is an optimal strategy of Maker which follows the connected rules. In other words, to prove $b_{mb}(\mathcal{OC}_n) = (1/2 - o(1))n$, it is necessary to find a strategy for Maker in which his graph might be disconnected during the game.

Theorem 2.1.2. *Breaker wins the $(1 : b)$ Maker-Breaker game under connected rules for every large enough n and every $b \geq 0.498n$, i.e. $b^c_{mb}(\mathcal{OC}_n) \leq 0.498n$ for every large enough n.*

2.1.2 Client-Waiter games

We also study the *Client-Waiter* odd-cycle game. Similar as for Maker-Breaker games, it is common to study games played on the edge-set of the complete graph. In this case, we analogously define Client's and Waiter's graphs $G_C(t), G_C, G_C^{ind}(t), G_C^{ind}, G_W(t), G_W, G_W^{ind}(t)$ and G_W^{ind}.

Considering the cycle-game $\mathcal{C}_n$, Hefetz, Krivelevich and Tan [31] proved that $b_{cw}(\mathcal{C}_n) = \lceil n/2 \rceil - 1$. Moreover, Krivelevich [34] proved that for any $0 < \varepsilon < 1$ and large enough n Client can always build a linearly long cycle if $b \geq (1/2 - \varepsilon)n$ and n is large enough.

For the odd-cycle game, we trivially have $b_{cw}(\mathcal{OC}_n) \leq b_{cw}(\mathcal{C}_n) = \lceil n/2 \rceil - 1$. Using a random strategy for Client, Hefetz, Krivelevich and Tan [31] proved that $b_{cw}(\mathcal{OC}_n) \geq \left(\frac{1}{4 \log 2} - o(1)\right) n$ (approximately $(0.3607 - o(1))n$) and conjectured that the upper bound is asymptotically tight.

Conjecture 2.1.3. *We have $b_{cw}(\mathcal{OC}_n) = (1/2 - o(1))\, n$.*

We will answer the analogous question for connected Client-Waiter games. The connected Client-Waiter game $(E(K_n), \mathcal{F})$ is exactly as the Client-Waiter game $(E(K_n), \mathcal{F})$ except that, in every round $t \geq 1$, Waiter is only allowed to offer edges which are adjacent to some vertex $v \in V\left(G_C^{ind}(t-1)\right)$

(for technical reasons, we assume that $V\left(G_C^{ind}(0)\right)$ is a single vertex). If there are no such edges left, Client wins the game. It is trivial from the definition that connected Client-Waiter games are bias-monotone. Therefore there again exists a threshold bias $b_{cw}^c(\mathcal{F})$, so that Waiter wins the connected $(1:b)$ Client-Waiter game $(X,\mathcal{F})$ if and only if $b \geq b_{cw}^c(\mathcal{F})$.

As playing connected is a restriction for Waiter, we have $b_{cw}^c(\mathcal{F}) \geq b_{cw}(\mathcal{F})$ for every game $(X,\mathcal{F})$. Furthermore, Waiter's strategy presented in [31] follows the connected rules and therefore shows $b_{cw}^c(\mathcal{C}_n) = b_{cw}(\mathcal{C}_n) = \lceil n/2 \rceil - 1$.

For odd cycles, we trivially have $b_{cw}^c(\mathcal{OC}_n) \leq b_{cw}^c(\mathcal{C}_n) = \lceil n/2 \rceil - 1$ and $b_{cw}^c(\mathcal{OC}_n) \geq b_{cw}(\mathcal{OC}_n) \geq \left(\frac{1}{4\log 2} - o(1)\right) n$. We show that the upper bound is tight, i.e. $b_{cw}^c(\mathcal{OC}_n) = \lceil n/2 \rceil - 1$.

Theorem 2.1.4. *If $b = \lceil n/2 \rceil - 2$, then Client has a winning strategy in the connected $(1:b)$ Client-Waiter odd-cycle game. In particular $b_{cw}^c(\mathcal{OC}_n) \geq \lceil n/2 \rceil - 1$.*

2.2 Maker-Breaker

2.2.1 Lower bound

In this section, we prove Theorem 2.1.1. We begin by presenting a strategy for Maker in the $(1:b)$ Maker-Breaker odd-cycle game. The basic idea is to build a tree, in which one side of its bipartition is very large. Maker achieves this by building a star around an arbitrary vertex until Breaker stops her. Then she connects an arbitrary new vertex to her tree and builds a star around it (using only vertices which are not in her current tree) until Breaker stops her again. That way, she keeps building a star around every new vertex she claims whenever Breaker stops her from building the previous star. At any point of the game let V be the set of vertices of positive Maker-degree and $R = [n] \setminus V$ be the set of vertices which are untouched by Maker.

Strategy 2.2.1. Suppose n and the bias b are fixed and choose an arbitrary vertex $w_0 \in [n]$. The strategy works in phases, starting with phase 0. In every round of phase k, $k \geq 0$, do the following.

(i) If there is an unclaimed edge closing an odd cycle, claim it.

(ii) Otherwise, if there is an unclaimed edge between w_k and R, claim it.

(iii) Otherwise, if there is a vertex $u \in R$ which is adjacent to V via an unclaimed edge and its degree in Breaker's current graph satisfies $\deg(u,R) \leq |R| - b - 2$, claim this edge. Name the vertex w_{k+1} and proceed to the next phase.

(iv) Otherwise, forfeit.

Let B_k be the set of neighbors of w_k in Maker's graph claimed in phase k, and note that these edges are all claimed in case (ii). We now define a class of graphs $\mathbb{G}_{n,b}$ with certain properties and later show that if Breaker wins the game then his final graph belongs to $\mathbb{G}_{n,b}$.

Definition 2.2.1. Let $\mathbb{G}_{n,b}$ be the set of tuples $(G, v_0, A_0, \ldots, v_s, A_s)$ with the following properties.

(I) $s \in \mathbb{N} \cup \{0\}$, $v_0, \ldots, v_s \in [n]$ are distinct, and $A_0, \ldots, A_s \subset [n] \setminus \{v_0, \ldots v_s\}$ are pairwise disjoint non-empty sets.

(II) G is a graph on the vertex set $[n]$ that contains all edges which are

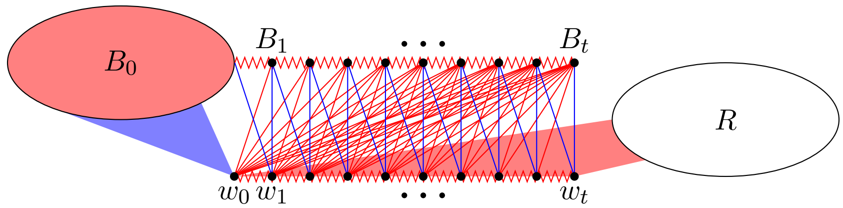

Figure 2.1: A possible final position, when Maker plays according to Strategy 2.2.1. Breaker's edges are colored red and Maker's edges blue.

- inside $\{v_0, \ldots, v_s\}$ and inside $\bigcup_{i=0}^{s} A_i$,
- between $\{v_0, \ldots, v_s\}$ and $R := [n] \setminus (\{v_0, \ldots v_s\} \cup \bigcup_{i=0}^{s} A_i)$ and
- between v_i and A_j for all $0 \le i < j \le s$.

(III) For every $v \in R$, we have

- v is fully connected to $T := [n] \setminus R$, or
- $\deg(v, R) \ge |R| - b - 1$.

When the elements of the tuple are clear from the context we will sometimes abbreviate $(G, v_0, A_0, \ldots, v_s, A_s)$ to G for simplicity.

Proof of Theorem 2.1.1. Suppose, for contradiction, we are playing the odd cycle game on n vertices with bias $(1 : b)$ and that Maker plays according to Strategy 2.2.1, but Breaker wins the game in phase t for some $t \ge 0$. Let G_M be Maker's final graph, let G_B be Breaker's final graph and let $w_0, \ldots, w_t$ and $B_0, \ldots, B_t$ be as produced by Strategy 2.2.1.

Observation 2.2.1. *G_M is a tree with biparts $\{w_0, \ldots, w_t\}$ and $\bigcup_{i=0}^{t} B_i$ and we have $(G_B, w_0, B_0, \ldots, w_t, B_t) \in \mathbb{G}_{n,b}$.*

Proof. G_M is a tree with biparts $\{w_0, \ldots, w_t\}$ and $\bigcup_{i=0}^{t} B_i$ because Breaker wins the game and Maker never closes an even cycle when following Strategy 2.2.1. We shall now check the items of Definition 2.2.1 one by one. (I) follows immediately from Strategy 2.2.1, since for every $1 \le k \le t$ the vertex w_k, when chosen, has at least one non-neighbor in R (in both Maker's and Breaker's graphs) so the set B_k is not empty. Moreover, Breaker must have claimed all edges inside $\{w_0, \ldots, w_t\}$ and $\bigcup_{i=1}^{s} B_i$ since otherwise, by Strategy 2.2.1, Maker would have won the game. Furthermore, for each $k = 0, \ldots, t$, Breaker must have claimed all edges between w_k and $R \cup \bigcup_{i=k+1}^{t} B_i$ because otherwise Maker would have kept playing in phase k for at least one more round. This verifies (II). Finally, for verifying (III), note first that, assuming Breaker wins the game, Maker must forfeit when the game reaches the point where $|R| \le b - 2$, unless she gave up earlier. In either case, the game ends with Maker giving up. If (III) was not true, Maker would move to phase $t + 1$ instead of forfeiting, which contradicts the assumption about G_M being her final graph. ∎

Define $f : \mathbb{G}_{n,b} \to \mathbb{R}$ by

$$f(G, v_0, A_0, \ldots, v_s, A_s) = \frac{e(G)}{\sum_{i=0}^{s} |A_i| + s}$$

and let $m = \min_{\mathbb{G}_{n,b}} f$ (note that the minimum is attained since $\mathbb{G}_{n,b}$ is a finite set). By Observation 2.2.1 G_M is a tree with $N := \sum_{i=0}^{t} |B_i| + t$ edges. Hence, Breaker claimed at most Nb edges during the game, and thus $b \geq f(G_B, w_0, B_0, \ldots, w_t, B_t) \geq m$. We thus turned the problem into a minimization problem.

We will start by reducing the domain over which we have to minimise. Let $R = [n] \setminus (\{v_0, \ldots, v_s\} \cup \bigcup_{i=0}^{s} A_i)$ and let $\mathcal{G} \subset \mathbb{G}_{n,b}$ be the set of all $G = (G, v_0, A_0, \ldots, v_s, A_s) \in \mathbb{G}_{n,b}$ for which $f(G) = m$.

Claim 2.2.2. *For every $G = (G, v_0, A_0, \ldots, v_s, A_s) \in \mathcal{G}$, we have*

(i) If $s \geq 1$, $|A_1| = |A_2| = \ldots = |A_s| = 1$,

(ii) $\deg(v, R) \geq |R| - b - 1$ for all $v \in R$ and

(iii) If $s \geq 1$, then $|R| + s \leq |A_0| \leq |R| + s + 3$.

Proof. Let $G = (G, v_0, A_0, \ldots, v_s, A_s) \in \mathcal{G}$. Assume that $s \geq 1$ and $|A_j| > 1$ for some $j \in [s]$. Then, by moving one vertex $u \in A_j$ to A_0 and deleting all edges between u and $\{v_0, \ldots, v_s\}$ we get a new graph $\tilde{G}$ and new sets $\tilde{A}_0, \ldots, \tilde{A}_s$ satisfying $(\tilde{G}, v_0, \tilde{A}_0, \ldots, v_s, \tilde{A}_s) \in \mathbb{G}_{n,b}$. This contradicts the choice of G since $f(\tilde{G}) < f(G)$ and hence (i) holds as well.

Let now $R_1 \subset R$ be the set of vertices $v \in R$ for which $\deg(v, R) < |R| - b - 1$. Note that all $v \in R_1$ are completely connected to $[n] \setminus R$ by Definition 2.2.1. We again form a new instance $(\tilde{G}, v_0, \tilde{A}_0, \ldots, v_s, \tilde{A}_s) \in \mathbb{G}_{n,b}$ by completely moving R_1 to A_0. If R_1 is non-empty, it is easy to see that $f(\tilde{G}) < f(G)$ again and hence (ii) holds.

Assume now that $s \geq 1$ and that $|A_0| < |R| + s$. Move v_s and the vertex $u \in A_s$ to A_0, remove the $2s$ edges between $\{u, v_s\}$ and $\{v_0, \ldots, v_{s-1}\}$ and the $|R|$ edges between v_s and R, and add the $|A_0| + s$ edges between v_s and $\bigcup_{i=0}^{s} A_i$. Call the new graph $\tilde{G}$, denote the new sets by $\tilde{A}_0, \ldots, \tilde{A}_{s-1}$, and note that $(\tilde{G}, v_0, \tilde{A}_0, \ldots, v_{s-1}, \tilde{A}_{s-1}) \in \mathbb{G}_{n,b}$. Since we removed $|R| + 2s$ edges and added $|A_0| + s$ edges, we have $e(G) > e(\tilde{G})$, and $\sum_{i=0}^{s-1} |\tilde{A}_i| + s - 1 = \sum_{i=0}^{s} |A_i| + s$. Thus $f(G) > f(\tilde{G})$, a contradiction. If $|A_0| > |R| + s + 3$, repeat this argument the other way around (i.e. move two vertices v, u from A_0 to be $v = v_{s+1}$ and $A_{s+1} = \{u\}$, remove the $|A_0| + s - 1$ edges between v_{s+1} and $\bigcup_{i=0}^{s} A_i \setminus \{v_{s+1}\}$, add the $2s + 2$ edges between $\{v_{s+1}, u\}$ and $\{v_0, \ldots, v_s\}$, and add the $|R|$ edges between v_{s+1} and $|R|$). This leads to a similar contradiction and hence (iii) also holds. ∎

Now, for every $G = (G, v_0, A_0, \ldots, v_s, A_s) \in \mathcal{G}$, we have

$$\begin{aligned} e(G) &\geq \binom{|A_0| + s}{2} + \binom{s+1}{2} + (s+1)|R| + \sum_{i=1}^{s} i + e(R) \\ &= \frac{1}{2}\left(|A_0|^2 + 3s^2 + 2\,|A_0|\,s + 2\,|R|\,s\right) + e(R) + o(n^2). \end{aligned} \tag{2.2.1}$$

Define β, α, ρ and σ by $b = \beta n$, $|A_0| = \alpha n$, $|R| = \rho n$ and $s = \sigma n$, and note that G and all these parameters depend on n. We split into two cases.

Case 1 ($|R| \leq b + 1$). In this case, we will neglect $e(R)$. Hence, plugging the above parameters into Equation (2.2.1) and using $b \geq f(G)$ and $\alpha > 0$, gives

$$\beta \geq \frac{\alpha^2 + 3\sigma^2 + 2\alpha\sigma + 2\rho\sigma}{2(\alpha + 2\sigma)} + o(1). \tag{2.2.2}$$

It follows from $n = |A_0| + 2s + 1 + |R|$ that $\alpha + 2\sigma + \rho = 1 + o(1)$. Assume first that $s = 0$, then in Equation (2.2.2) we have

$$\beta \geq \frac{\alpha}{2} + o(1) \geq \frac{1-\rho}{2} + o(1) \geq \frac{1-\beta}{2} + o(1)$$
$$\Rightarrow \beta \geq \frac{1}{3} + o(1) \geq \frac{4-\sqrt{6}}{5}$$

as required. Assume now that $s \geq 1$. By Claim 2.2.2 *(iii)* we have $\alpha = \rho + \sigma + o(1)$ and hence in Equation (2.2.2) we get

$$\beta \geq \frac{2-\rho^2-2\rho}{6(1-\rho)} + o(1) =: h(\rho) + o(1).$$

By assumption we have $\beta \geq \rho + o(1)$, and hence $\beta \geq \min\{h(\rho) : \rho \leq h(\rho)\} + o(1)$. This minimization problem is easy to solve and the solution is given by $(4-\sqrt{6})/5$.

Case 2 ($|R| > b+1$). In this case, we have $e(R) \geq |R|(|R|-b-1)/2$, and hence plugging the above parameters into Equation (2.2.1) and using $b \geq f(G)$ gives

$$\beta \geq \frac{\alpha^2 + 3\sigma^2 + 2\alpha\sigma + 2\sigma\rho + \rho(\rho-\beta)}{2(\alpha+2\sigma)} + o(1). \tag{2.2.3}$$

Again, assuming first that $s = 0$, we get $\alpha + \rho = 1 + o(1)$ and hence in Equation (2.2.3) we have

$$\beta \geq \frac{\alpha^2 + \rho(\rho-\beta)}{2\alpha} + o(1)$$
$$\Rightarrow \beta \geq \frac{\alpha^2+\rho^2}{2\alpha+\rho} + o(1) = \frac{(1-\rho)^2+\rho^2}{2-\rho} + o(1) \geq 0.32 \geq \frac{4-\sqrt{6}}{5}$$

as required. Assume that $s \geq 1$. Using $\alpha + 2\sigma + \rho = 1 + o(1)$ and $\alpha = \rho + \sigma + o(1)$ again, implies

$$\beta \geq \frac{2-2\rho+2\rho^2}{6-3\rho} + o(1) =: h(\rho) + o(1).$$

By assumption we have $\beta \leq \rho + o(1)$, and hence $\beta \geq \min\{h(\rho) : \rho \geq h(\rho)\} + o(1)$. This minimization problem is also easy to solve and the solution is again given by $(4-\sqrt{6})/5$.

In both cases the minimum is attained when $\rho = \beta = (4-\sqrt{6})/5$. $\square$

2.2.2 An upper bound under connected rules

In this section we consider the connected $(1:b)$ Maker-Breaker odd-cycle game and prove Theorem 2.1.2. Let $G_M^{ind}(s) = (V_s, E_s)$ denote Maker's graph after s of her turns and let $R_s := [n] \setminus V_s$ be the set of vertices not touched by Maker. As long as Maker does not win the game, $G_M^{ind}(s)$ is bipartite and, since $G_M^{ind}(s)$ is connected, there is a unique (up to labelling) bipartition $V_s = V_s^1 \cup V_s^2$, which we may choose in a way that $V_s^i \subset V_{s+1}^i$ holds for all $s \geq 0$ and $i = 1, 2$.

Unless stated otherwise, we will always mean Breaker's graph when we mention edges and degrees without specifying the graph. A *state* is a tuple (s,k) with $s \geq 1$ and $0 \leq k \leq b$ and describes the situation of the game after s moves of Maker and after Breaker claimed k edges in his s-th turn. For example, state $(1,0)$ describes the situation right after Maker claimed her first edge. We denote by $\deg_k(v)$ the degree of a vertex $v \in [n]$ in Breaker's graph at state (s,k), where s is clear from the context. Similarly, we define other quantities like $e_k(R_s, V_s), e_k(R_s, V_s^i)$ where $i = 1, 2$. When we refer to these quantities at the end of round s (i.e. when $k = b$) we omit the extra parameter and simply denote $\deg(v, V_s)$ and $e(R_S, V_s^i)$.

To prove Theorem 2.1.2, we provide Breaker with the following strategy. In this strategy Breaker's goal is to distribute his edges between the set of vertices not touched by Maker and the parts of her bipartite graph as evenly as possible. This way Breaker minimises the number of edges ending up being between the parts of Maker bipartite graph, saving more edges for blocking her from building an odd cycle.

Strategy 2.2.2. After s moves of Maker, do the following with decreasing priority.

(i) Kill all threats: Claim every unclaimed edge between vertices inside V_s^1 or inside V_s^2. If this is not possible, forfeit.

(ii) Claim all remaining edges of this round by repeating the following part. Assume that so far $0 \leq k \leq b-1$ edges were already claimed in round s. Further assume that $e_k(R_s, V_s^{i_1}) \leq e_k(R_s, V_s^{i_2})$ for $1 \leq i_1 \neq i_2 \leq 2$.

   (a) If $k = 0$, $|R_s| \leq b$ and for some $i = 1, 2$ we have that all vertices of V_s^i, but one, are completely connected to R_s, then claim all edges connecting that vertex to R_s.

   (b) Otherwise, if there are unclaimed edges between R_s and $V_s^{i_1}$, do as follows. Find a vertex $v \in R_s$ for which $\deg_k(v, V_s^{i_1})$ is minimal among all vertices in R_s and claim an arbitrary edge connecting it to $V_s^{i_1}$.

   (c) Otherwise, if there are unclaimed edges between R_s and $V_s^{i_2}$, do as follows. Find a vertex $v \in R_s$ for which $\deg_k(v, V_s^{i_2})$ is minimal among all vertices in R_s and claim an arbitrary edge connecting it to $V_s^{i_2}$.

   (d) Otherwise, claim an arbitrary edge.

Proof of Theorem 2.1.2

From now on, we assume that the bias is $b = \frac{1-\varepsilon}{2}n$ for some fixed $\varepsilon > 0$ which will be specified later, but Maker has a winning strategy. We further assume that she follows this strategy and Breaker plays according to Strategy 2.2.2, and hence Maker wins the game after $t+2$ moves for some $t \geq 0$. We begin with some easy facts.

Observation 2.2.3. *Throughout the game the following properties may be assumed to hold.*

(i) Maker's graph is a tree for all $1 \leq s \leq t$, in particular $|V_s| = s + 1$.

(ii) Maker must create at least $b + 1$ threats in round $t + 1$.

(iii) We have $t \leq n - 2$.

Proof. Suppose that at some point Maker claims an edge e which closes an even cycle C_1 in her graph. Let C_2 be the odd cycle she fully claimed to win the game. If $e \in C_2$, then $(C_2 \setminus C_1) \cup (C_1 \setminus C_2)$ is an odd cycle not containing e. Hence, we may assume that she took any other edge instead of e. If there are only edges left which close an even cycle, Maker lost the game, because she cannot create any new threats. This proves (i).

For (ii) note that the only way Breaker loses is when he cannot defend all threats, i.e. following part (i) of Strategy 2.2.2, and therefore gives up the game.

After $n-1$ rounds Maker would have a spanning tree, so in every further round she cannot create any new threats. Hence $t+1 < n$, which implies (iii). ∎

Observation 2.2.4. *Assuming Maker wins the game, Breaker never gets to implement parts $(ii)(a)$ and $(ii)(d)$ when following Strategy 2.2.2. In particular, Breaker never claims an edge inside R_s for every $s \leq t+1$.*

Proof. We start by noting that if at some point Breaker implements part $(ii)(d)$ of Strategy 2.2.2 then it means that for some $s \leq t+2$ Maker's graph is bipartite with parts which are cliques in Breaker's graph, and is disconnected from R_s by Breaker's edges. Hence Maker can neither enlarge her graph with vertices, nor closing an odd cycle, with contradiction to the assumption that she wins the game.

If at some round $s \leq t+1$ Breaker implements part $(ii)(a)$ of Strategy 2.2.2, then at the end of this round one of the parts, say V_s^1, is completely connected to R_s. By Observation 2.2.3 (i), Maker, in her next move must claim an edge incident to R_s and V_s^2. By claiming an edge incident to R_s and V_s^2 Maker creates no new threats. This allows Breaker to keep and follow part $(ii)(a)$ of the strategy until Maker's graph cannot get larger in vertices, remains bipartite with parts which are cliques in Breaker's graph, and thus Maker loses the game, with contradiction to the assumption that she wins. ∎

Lemma 2.2.5. *After any $s \leq t$ rounds we have $|\deg_k(u, V_s^i) - \deg_k(v, V_s^i)| \leq 1$ and $|\deg_k(u, V_s) - \deg_k(v, V_s)| \leq 2$ for all $u, v \in R_s$, $i = 1, 2$ and for every $0 \leq k \leq b$.*

Proof. According to Strategy 2.2.2, whenever there is an unclaimed edge between R_s and V_s^i, for some $i \in \{1, 2\}$, then there is an unclaimed edge between any vertex $v \in R_s$ with minimal $\deg_k(v, V_s^i)$ and V_s^i, which immediately implies the claim. ∎

For $i = 1, 2$ let

$$d_s^i := \frac{e(V_s^i, R_s)}{|R_s|} \quad \text{and} \quad d_s := \frac{e(V_s, R_s)}{|R_s|}$$

be the average number of neighbors inside V_s^1, V_s^2 and V_s of vertices $v \in R_s$.

Lemma 2.2.6. *After any $s \leq t$ rounds, we have $|e(V_s^1, R_s) - e(V_s^2, R_s)| < |R_s|$ and thus $|d_s^1 - d_s^2| < 1$. In particular, we have $(d_s - 1)/2 \leq d_s^i \leq (d_s + 1)/2$ for $i = 1, 2$.*

Proof. Let us call a state (s, k) *safe*, if at this state we have $\left|e_k(R_s, V_s^1) - e_k(R_s, V_s^2)\right| \leq 1$, and $\deg_k(v, V_s^{i_1}) \leq \deg_k(v, V_s^{i_2})$ for all $v \in R_s$ whenever $e_k(R_s, V_s^{i_1}) \leq e_k(R_s, V_s^{i_2})$. Note that the initial state $(1, 0)$ is safe. Note also that a state can only become unsafe after an action of Breaker. Indeed, if (s_0, b) is safe for some round s_0, then so is $(s_0 + 1, 0)$, since a move of Maker does not change any of the relevant quantities. Suppose that (s, k) is not safe for some $s \leq t$ and some $1 \leq k \leq b$, but the previous state $(s, k-1)$ is safe. We show that Breaker will restore a safe state without violating the conditions of the lemma.

By definition of Strategy 2.2.2, by Observation 2.2.4, and by the fact that (s, k) is not safe, all edges between R_s and one part, say V_s^1, are claimed at state $(s, k-1)$. From this point Breaker claims at most $b-1$ additional edges in round s, all incident to R_s and V_s^2. Hence, at the end of this round we have

$$|e(R_s, V_s^1) - e(R_s, V_s^2)| \leq b. \tag{2.2.4}$$

Furthermore, by Observation 2.2.3 (1), in her $(s+1)$-st turn Maker must claim an edge incident to R_s and V_s^2. However, by doing that Maker does not create any new threats. Denote by vu the edge that Maker claims in round $s+1$, where $u \in V_s^2$ and $v \in R_s$. Note that at the beginning of Breaker's turn in round $s+1$ all vertices of V_{s+1}^1 are fully connected to R_{s+1} except for v. Hence we get that $|R_s| \geq b$, or otherwise this contradicts Observation 2.2.4. In particular, by Equation (2.2.4) we get $|e(R_s, V_s^1) - e(R_s, V_s^2)| \leq |R_s|$. In particular in round $s+1$ there are $|R_{s+1}| \geq b-1$ unclaimed edges between R_{s+1} and V_{s+1}^1, so Breaker can even out the difference between $e(R_{s+1}, V_{s+1}^1)$ and $e(R_{s+1}, V_{s+1}^2)$, and return to a safe state within round $s+1$. This finishes the proof. ∎

Now, as we know that Breaker's edges between V_s and R_s are nicely distributed, we shall investigate how many of the edges claimed by Breaker are of this type. The next proposition shows that, assuming Maker wins, in every round s of the game, Maker forces Breaker to claim almost all edges between V_s^1 and V_s^2. We will use this result repeatedly.

Definition 2.2.2. For any $s \leq t$, we call all unclaimed edges between V_s^1 and V_s^2 edges that Breaker *saved.*

Proposition 2.2.7. *After any $s \leq t$ rounds of the game, there are at most $(\varepsilon n^2 - n)/2$ edges that Breaker saved.*

Proof. All edges which are saved at some point of the game, remain saved throughout the game, because neither player ever claims them, by Strategy 2.2.2 and by Observation 2.2.3.

Assume for contradiction that Breaker saved more than $(\varepsilon n^2 - n)/2$ edges after t moves. By Observation 2.2.3, after t moves, there must be some vertex $v \in R_t$ for which there are at least $b+2$ unclaimed edges between it and V_t. Therefore, $\deg(v, V_t) \leq t - b - 1$. By Lemma 2.2.5 we have $\deg(u, V_t) \leq t - b + 1$ for all $u \in R_t$. Moreover, Breaker claimed less than $\binom{t+1}{2} - t - \varepsilon n^2/2 + n/2$ edges inside V_t (all except Maker's edges and his saved edges). Hence we have

$$bt = \text{ number of Breaker's edges} < \binom{t+1}{2} - t - \varepsilon n^2/2 + n/2 + (n-t-1)(t-b+1).$$

It follows directly that

$$(n-1)b < nt - t^2/2 - 5t/2 - \varepsilon n^2/2 + 3n/2 - 1.$$

Simple calculus shows that the right hand side is maximized over $t \in \mathbb{R}$ when $t = n - 2.5$. It follows that $b < \frac{1-\varepsilon}{2} n$, a contradiction. ∎

Using Proposition 2.2.7, it is easy to deduce good bounds on the number of edges between V_s and R_s and hence also on their density.

Lemma 2.2.8. *For large enough n, after any $s \leq (3-\varepsilon)n/4 - 1$ rounds, we have*

$$\frac{s - 3\varepsilon n}{2} < d_s < \frac{s + \varepsilon n}{2} + 1.$$

Proof. By Proposition 2.2.7, after s rounds, we have $\binom{s+1}{2} - s - (\varepsilon n^2 - n)/2 \leq e(V_s) \leq \binom{s+1}{2} - s$, which means

$$(s^2 - s - \varepsilon n^2 + n)/2 \leq e(V_s) \leq (s^2 - s)/2$$

Hence

$$\begin{aligned} e(V_s, R_s) \leq bs - (s^2 - s - \varepsilon n^2 + n)/2 &= \frac{(1-\varepsilon)ns - s^2 + \varepsilon n^2 - n + s}{2} \\ &= \frac{(s + \varepsilon n)(n - s - 1) + 2s - (1-\varepsilon)n}{2} \end{aligned}$$

On the other hand, since $s \leq (3-\varepsilon)b/4 - 1$, we have

$$e(V_s, R_s) \geq bs - s^2/2 = \frac{(1-\varepsilon)ns - s^2}{2} > \frac{(s - 3\varepsilon n)(n - s - 1)}{2}.$$

Dividing both equations by $|R_s| = (n - s - 1)$ we get

$$\frac{s - \varepsilon n}{2} < d_s \leq \frac{s + \varepsilon n}{2} + \frac{2s - (1-\varepsilon)n}{2(n-s-1)} < \frac{s + \varepsilon n}{2} + 1$$

where the last inequality holds by our assumption on the size of s. ∎

Combining Lemma 2.2.5, Lemma 2.2.6 and Lemma 2.2.8, we get the following proposition.

Proposition 2.2.9. *Let n be large enough, let $s \leq t$, let $u, v \in R_s$ and let $i \in 1, 2$. Then,*

(i) $|\deg(u, V_s^i) - \deg(v, V_s^i)| \leq 1$ *and* $|\deg(u, V_s) - \deg(v, V_s)| \leq 2$,

(ii) $|d_s^1 - d_s^2| \leq 1$ *and* $(d_s - 1)/2 \leq d_s^i \leq (d_s + 1)/2$,

(iii) $\frac{s-3\varepsilon n}{2} < d_s < \frac{s+\varepsilon n}{2} + 1$ *and* $\frac{s-3\varepsilon n}{4} - 1 < d_s^i < \frac{s+\varepsilon n}{4} + 1$ *if* $s \leq \frac{(3-\varepsilon)}{4}n - 1$.

We need two more technical lemmas to finish the proof of Theorem 2.1.2.

Lemma 2.2.10. *After any $s \leq (3/4 - \sqrt{\varepsilon})n$ rounds, we have $|V_s^i| < d_s^i + \sqrt{\varepsilon}n$ for some $i \in \{1, 2\}$.*

Proof. Suppose for contradiction that $|V_s^i| \geq d_s^i + \sqrt{\varepsilon}n$ for both $i = 1, 2$ and some fixed $s \leq (3/4 - \sqrt{\varepsilon})n$. Whenever $s' \leq 3n/4$ and n is large enough, using Proposition 2.2.9, we have

$$\begin{aligned} d_{s'+1} = \frac{e(V_{s'+1}, R_{s'+1})}{|R_{s'+1}|} &\leq \frac{e(V_{s'}, R_{s'}) + b - (d_{s'} - 2)}{|R_{s'+1}|} \\ &= \frac{e(V_{s'}, R_{s'}) - d_{s'}}{|R_{s'}| - 1} + \frac{b + 2}{|R_{s'}| - 1} \\ &= d_{s'} + \frac{(1-\varepsilon)n/2 + 2}{n - s' - 2} \\ &\leq d_{s'} + \frac{n/2}{n - s'} \leq d_{s'} + 2. \end{aligned}$$

By iterating this and using Proposition 2.2.9, we get

$$\deg(u, V_{s+j}^i) \leq d_{s+j}^i + 1 \leq \frac{d_{s+j}}{2} + 2 \leq \frac{d_s + 2j}{2} + 2 \leq d_s^i + j + 3$$

for every $u \in R_{s+1}$, $i \in \{1, 2\}$ and $j = 0, \ldots, \sqrt{\varepsilon}n - 4$. Therefore Breaker saves at least $|V_s^i| - d_s^i - j - 3 \geq d_s^i + \sqrt{\varepsilon}n - d_s^i - j - 3 = \sqrt{\varepsilon}n - j - 3$ edges in round $s + j$ for every $j = 0, \ldots, \sqrt{\varepsilon}n - 4$ and some $i \in \{1, 2\}$. Hence, in total, Breaker saves at least

$$\sum_{j=0}^{\sqrt{\varepsilon}n-4} \sqrt{\varepsilon}n - j - 3 = \sum_{j=1}^{\sqrt{\varepsilon}n-3} j > \frac{\varepsilon n^2 - n}{2}$$

edges, which contradicts Proposition 2.2.7. ■

Lemma 2.2.11. *Maker does not win early in the game, more precisely $t > (3 - \varepsilon)n/4 - 1$.*

Proof. Assume for contradiction that $t \leq (3 - \varepsilon)n/4 - 1$. Without loss of generality Maker wins in her $(t + 2)$-nd turn by claiming an edge inside V_{t+1}^2. Hence, in her $(t + 1)$-st turn she must have claimed an edge between V_t^1 and some vertex $v \in R_t$ with at least $b + 1$ unclaimed edges into V_t^2. This implies that there is some $v \in R_t$ satisfying $|V_t^2| - \deg(v, V_t^2) \geq b + 1$. By Proposition 2.2.9, we have $|V_t^2| - d_t^2 \geq b$ and moreover, we have $|V_t^2| = t + 1 - |V_t^1| \leq t + 1 - d_t^1$. Putting both inequalities together and using again Proposition 2.2.9 in the last inequality (note here that $t \leq (3 - \varepsilon)n/4 - 1)$ implies

$$\frac{(1 - \varepsilon)n}{2} = b \leq |V_t^2| - d_t^2 \leq t + 1 - d_t \leq \frac{t + 2 + 3\varepsilon n}{2},$$

which leads to a contradiction. ■

We are finally able to complete the proof of Theorem 2.1.2.

Proof. Let $\varepsilon = 0.002$, $x = 1/3 - \varepsilon$ and $s = xn - 3$. By Lemma 2.2.10 and Proposition 2.2.9 we have without loss of generality

$$|V_s^1| \leq d_s^1 + \sqrt{\varepsilon}n \leq \left(\frac{x + \varepsilon}{4} + \sqrt{\varepsilon}\right) n + 1. \tag{2.2.5}$$

Using also $|V_s^1| + |V_s^2| = s + 1$, we have

$$|V_s^2| \geq xn - \left(\frac{x + \varepsilon}{4} + \sqrt{\varepsilon}\right) n - 3 \geq \left(\frac{3x - \varepsilon}{4} - \sqrt{\varepsilon}\right) n - 3. \tag{2.2.6}$$

Note that $2s \leq \frac{(3-\varepsilon)}{4}n - 1$. Hence, the game goes on for another s moves (by Lemma 2.2.11), and we have (using Proposition 2.2.9)

$$|V_{2s}^1| \geq d_{2s}^1 \geq \left(\frac{2x - 3\varepsilon}{4}\right) n - 3. \tag{2.2.7}$$

Hence Maker connects a vertex to V_s^1 at least $\left(\frac{x-4\varepsilon}{4} - \sqrt{\varepsilon}\right) n - 4$ times between round $s + 1$ and $2s$. Note that $d_{2s}^1 \leq \frac{2s+\varepsilon n}{4} + 1$, so by (2.2.6) and Proposition 2.2.9, Breaker saves in each round s', where $s + 1 \leq s' \leq 2s$, at least

$$\begin{aligned} |V_{s'}^2| - d_{s'}^1 - 1 &\geq |V_s^2| - \frac{s' + \varepsilon n}{4} - 2 \\ &\geq \left(\frac{3x - \varepsilon}{4} - \sqrt{\varepsilon}\right) n - \frac{2s + \varepsilon n}{4} - 5 \end{aligned}$$

$$\begin{aligned}&\geq \left(\frac{x-2\varepsilon}{4}-\sqrt{\varepsilon}\right)n-4\\&> \left(\frac{x-4\varepsilon}{4}-\sqrt{\varepsilon}\right)n-4\end{aligned}$$

edges. Therefore (for n large enough), Breaker saves at least

$$\begin{aligned}\left(\left(\frac{x-4\varepsilon}{4}-\sqrt{\varepsilon}\right)n-4\right)^2 &= \left(\left(\frac{1/3-5\varepsilon}{4}-\sqrt{\varepsilon}\right)n-4\right)^2\\&= \left(\frac{1/3-5\varepsilon}{4}-\sqrt{\varepsilon}\right)^2 n^2 - 8\left(\frac{1/3-5\varepsilon}{4}-\sqrt{\varepsilon}\right)n+16\\&> 0.001n^2 \geq (\varepsilon n^2-n)/2\end{aligned}$$

edges, a contradiction. □

2.3 Client-Waiter

In this section we consider the $(1:b)$ Client-Waiter odd-cycle game under connected rules and prove Theorem 2.1.4.

Proof of Theorem 2.1.4. Suppose from now on the bias is $b = \lceil n/2 \rceil - 2$ but Waiter has a winning strategy, and assume he plays according to this strategy. Let $G_C^{ind}(s) = (V_s, E_s)$ be Client's graph and $R_s = [n] \setminus V_s$ be the vertices not touched by Client after s rounds. By assumption $G_C^{ind}(s)$ is bipartite for any $s \geq 0$ and since $G_C^{ind}(s)$ is connected, there is a unique (up to labeling) bipartition $V_s = V_s^1 \cup V_s^2$, which we may choose in a way that $V_s^i \subset V_{s+1}^i$ holds for all $s \geq 0$ and $i = 1, 2$. Unless stated otherwise, we always mean Waiter's graph when talking about degrees and edges without specifying the graph.

Note that if there is an unclaimed edge inside V_s^1 or V_s^2, Client will win sooner or later, because this edge closes an odd cycle in her graph and at some point Waiter must offer this edge. We will see below that this forces Waiter to offer quite many edges incident to every new vertex he offers if it is not completely connected to one part of the bipartition. This motivates the following definition.

Definition 2.3.1. Let $i \in \{1, 2\}$. A vertex $v \in R_s$ is called *critical* with respect to V_s^i at time s, if it is completely connected to V_s^i. A part V_s^i is called *critical* at time s if there exists a vertex $v \in R_s$ which is critical with respect to V_s^i at time s.

The following strategy for Client tries to avoid having many critical vertices.

Strategy 2.3.1. Let $s \geq 0$ and let W be the set of edges offered by Waiter in round $s+1$. Do the following with decreasing priority.

(i) If there is some $e \in W$ closing an odd cycle in Client's graph, claim it.

(ii) If there is some $e \in W$ so that $E(V_{s+1}^1) \cup E(V_{s+1}^2)$ contains a unclaimed edge if Client claims e, claim it.

(iii) If there is some $e \in W$ which is incident to a non-critical part and to R_s, claim it.

(iv) If there is some $e \in W$ which is adjacent to R_s, claim it.

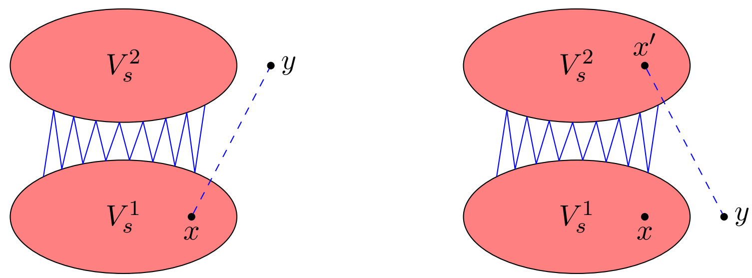

Figure 2.2: The situation in Lemma 2.3.3. Waiter's edges are red and Client's edges are blue.

(v) Give up.

Remark 2.3.1. The first two items basically say that Client should win the game whenever she gets a chance to do so. Hence every reasonable strategy for Client should follow this part of the strategy.

Remark 2.3.2. Note that Client rather gives up than claiming an edge closing an even cycle in her graph. In particular Client's graph will always be a tree, since we are assuming Waiter wins the game. Moreover, Client connects a new vertex $v \in R_{s-1}$ to her graph in every round $s \geq 1$. By the connected rules, v is independent from R_s and hence there is always at most one critical part.

We can now state precisely why critical vertices play a crucial role.

Lemma 2.3.3. *Suppose $y \in R_s$ is not critical at time $s \geq 0$ to any of the parts V_s^1, V_s^2 and Waiter offers an edge incident to y in round $s+1$. Then he offers every unclaimed edge incident to y and V_s in round $s+1$.*

Proof. Figure 2.2 might help to follow this proof. Suppose, w.l.o.g., that Waiter offers the edge xy for some $x \in V_s^1$. If Waiter does not offer all unclaimed edges incident to y and V_s^2, then Client, by following Strategy 2.3.1, will claim an edge such that there will be an unclaimed edge inside V_{s+1}^2, contradicting the fact that Waiter wins the game. Since y is not critical with respect to V_s^2, there is some $x' \in V_s^2$ such that $x'y$ is unclaimed yet and hence Waiter offers this edge. Repeating this argument with x' and y, we conclude that Waiter also offers all unclaimed edges incident to y and V_s^1. ■

The next lemma deals with critical vertices.

Lemma 2.3.4. *Suppose $v \in R_s$ is critical at some time $s \geq 0$ with respect to V_s^i for some $i \in \{1, 2\}$. Then there is exactly one unclaimed edge incident to v and V_s.*

Proof. Assuming otherwise, let s be the first time such that there is a critical vertex $v \in R_s$ and at least two unclaimed edges incident to v and V_s. Without loss of generality v is critical with respect to V_s^1 and thus there are at least two unclaimed edges incident to v and V_s^2. It follows that v was not critical with respect to V_{s-1}^2 after $s-1$ rounds. If v was not critical with respect to V_{s-1}^1 either, then Waiter offered at least one edge incident to v and hence all edges incident to v and V_{s-1} in round s (by Lemma 2.3.3), a contradiction.

Hence v was critical with respect to V_{s-1}^1 and by the choice of s there was exactly one unclaimed edge incident to v and V_{s-1}^2 (see Figure 2.3). Thus Client claimed an edge xy with $x \in V_{s-1}^1$ and $y \in R_{s-1}$ in round s to enlarge V_{s-1}^2. But y was not critical with respect to V_{s-1}^2 by Remark 2.3.2

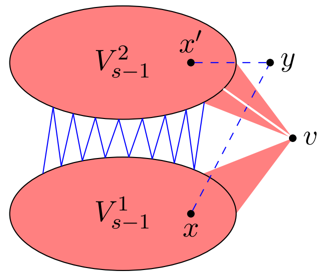

Figure 2.3: The situation in Lemma 2.3.4. Waiter's edges are red and Client's edges are blue. The white edge denotes a unclaimed edge and the dashed edges are two of the edges which are offered to Client.

and hence Waiter offered at least one edge incident to y and V^2_{s-1} by Lemma 2.3.3. But Client would have chosen this edge instead following Strategy 2.3.1, a contradiction. ■

Equipped with these two lemmas, it is easy now to finish the proof of Theorem 2.1.4. By Lemmas 2.3.3 and 2.3.4 Waiter always offers all edges incident to v and V_s whenever he offers one edge incident to some $v \in R_s$. Hence there will never be an unclaimed edge inside V_s and thus Client does not give up the game. Hence the game ends after t rounds, for some integer t, with all edges being claimed. We have

$$|E_t| = t \geq \frac{\binom{n}{2}}{q+1} = \frac{n(n-1)}{2\lceil n/2 \rceil - 2} \geq \frac{n(n-1)}{n-1} = n,$$

so $G^{ind}_C(t)$ cannot be a tree, a contradiction. □

2.4 Concluding remarks and open questions

Other positional game types that can be analyzed in the context of odd-cycle games are Avoider-Enforcer games and Waiter-Client games. In the $(1 : b)$ Avoider-Enforcer odd-cycle game, Enforcer wants to force Avoider claiming an odd cycle. In the *strict* version of the game, Avoider claims one unclaimed element per round, whereas Enforcer claims b. It turns out that this game is not bias monotone, which motivates the definition of the *monotone* version of this game, where in each round Avoider and Enforcer claim at least one and b unclaimed elements, respectively. For more details about Avoider-Enforcer games see, e.g., [1, 25, 28, 29]. In [26] the authors considered the unbiased Avoider-Enforcer odd-cycle game and proved that Enforcer wins rather fast. In [14] the authors proved that for bias $b \geq 200n \ln n$, Avoider can ensure that in the end of both strict and monotone $(1 : b)$ games his graph is a forest for every but maybe the last round of the game. In [27] various Avoider-Enforcer games were considered, among them the biased odd-cycle, and it was shown that in the strict game we have the following bounds on the *upper* and the *lower* threshold biases, $cn \leq b^-_{ae}(\mathcal{OC}_n) \leq b^+_{ae}(\mathcal{OC}_n) \leq n^{3/2}$, for some constant $c > 0$. It is only natural to raise the question about the values of the threshold biases for both strict and monotone Avoider-Enforcer odd-cycle game.

Question 2. What is the threshold biases for the monotone/strict Avoider-Enforcer odd-cycle games?

Waiter-Client games are Similar to Client-Waiter games except that Client's goal is to avoid a full winning set instead of aiming to claim one. In [4] Bednarska-Bzdęga, Hefetz, Krivelevich, and Łuczak considered Waiter-Client games. A special case of their result, regarding the Waiter-Client odd-cycle game, states that $n - 4\lceil n^{3/4} \rceil + 1 \leq b_{wc}(\mathcal{OC}_n) \leq 1.00502n$. They further conjectured that the lower bound is closer to the truth.

Conjecture 2.4.1 ([4]). *The threshold bias for the Waiter-Client odd-cycle game is* $b_{wc}(\mathcal{OC}_n) = (1 + o(1))\, n$.

One of our main contributions in this chapter is that in case the answer to Question 1 is yes, we separate the unrestricted threshold from the connected one. In this context it is an interesting problem to determine the connected thresholds of various games and to see whether the they differ from the unrestricted ones. One of the classical games is the connectivity game, where Gebauer and Szabó [22] showed that the threshold bias for the Maker-Breaker game is asymptotically equal $n/\ln n$. Another classical game is the Hamiltonicity game, where Krivelevich showed in [35] that the threshold bias for the Maker-Breaker game is $(1 + o(1))n/\ln n$. However, it is easy to see that with bias $b = 2$ Breaker can isolate a vertex when playing against connected Maker. This means that for the Hamiltonicity and the connectivity games the unrestricted biases differ from the connected ones, and this is true for other games where Maker's goal is to occupy a spanning subgraph of K_n.

Taking to another direction our main remaining question, whether the connected Maker-Breaker odd-cycle game and the unrestricted one have the same threshold or not, one can think of the odd-cycle game as the non-2-colorability game. General non-k-colorability games for a general integer $k \geq 2$ are well known and studied games. It was proved in [27] that the threshold bias for the Maker-Breaker non-k-colorability game, denoted by $b_{mb}(\mathcal{NC}_n^k)$, satisfies $s'_k n \leq b_{mb}(\mathcal{NC}_n^k) \leq s_k n$ where s'_k, s_k are constants depending only on k. It will be interesting to determine whether in the general non-k-colorability game the threshold biases of the connected game and the unrestricted one are equal or not.

Question 3. Considering both the unrestricted and the connected Maker-Breaker non-k-colorability games, do we have $b_{mb}(\mathcal{NC}_n^k) = b^c_{mb}(\mathcal{NC}_n^k)$?

Chapter 3

Playing on the vertices of random graphs

3.1 Introduction

In this chapter we introduce a new type of positional games, played on the vertex set of a graph. We focus on the case where the board is the vertex set of a random graph, and we are interested to determine the probability threshold of the game. Sometimes we are interested in the level of *sharpness* of threshold functions. Since this characterization is far from being our main focus in this paper, we do not use the standard definitions, but the following simplified ones, which are sufficient for our purposes. We say that a monotone increasing graph property $\mathcal{P}$ has a *sharp* threshold at p^* if in (**??**) we can write $(1+\varepsilon)p^*$ and $(1-\varepsilon)p^*$ instead of $\omega(p^*)$ and $o(p^*)$, respectively, for any constant $\varepsilon > 0$; if we can replace these expressions with Cp^* and cp^*, for some constants c, C independent of n, we say it is a *semi-sharp* threshold (which can be in fact sharp in case the two constants may be arbitrarily close to each other); and if p^* is a threshold function for $\mathcal{P}$, but for every constant c there exists a constant $\delta = \delta(c) \in (0,1)$ such that $\lim_{n\to\infty} \Pr\left[G \sim G(n, cp^*) \in \mathcal{P}\right] = \delta$, we say that p^* is a *coarse* threshold for $\mathcal{P}$.

The first to study positional games on random graphs were Stojaković and Szabó [45], who analyzed Maker-Breaker games played on $E(G)$ where $G \sim G(n,p)$. In that paper they investigated the threshold probability of the unbiased connectivity, perfect matching, Hamiltonicity, and K_k games. They also considered the biased $(1 : b)$ versions of these games, and provided bounds for the threshold bias b^* as a function of p. Since then, much progress has been made in understanding positional games played on the edge set of $G \sim G(n,p)$ (see e.g. [9, 13, 16, 18, 39, 40]). In particular, the study of Maker-Breaker H-games in this setting was continued by Müller and Stojaković [39], who found the threshold probability for the unbiased K_k-game where $k \geq 4$, by giving a lower bound on the threshold probability matching the upper bound given in [45]. For the K_3-game they provided a *hitting time* result, thus achieving a better understanding of this game, whose threshold probability was already determined in [45]. We discuss the K_3-game and the meaning of hitting time results more thoroughly in Subsection 3.1.2. In [40], Nenadov, Steger, and Stojaković solved the unbiased Maker-Breaker H-game for almost all H. Their main result is the following. For a graph H, let $d_2(H) := \frac{e(H)-1}{v(H)-2}$ to be the *2-density* of H and recall that $m_2(H) := \max\{d_2(H') \mid H' \subseteq H,\ v(H') \geq 3\}$ is the *maximum* 2-*density* of H. A graph H is called *strictly* 2-*balanced* if $d_2(H) > d_2(H')$ for every $H' \subsetneq H$.

Theorem 3.1.1 (Theorem 2 in [40]). *Let H be a graph for which there exists $H' \subseteq H$ such that $d_2(H') = m_2(H)$, H' is strictly 2-balanced and it is not a tree or a triangle. Then there exist constants $c, C > 0$ such that*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Maker's win in the } (1:1)\ H\text{-game}\right] = \begin{cases} 1 & p \geq Cn^{-1/m_2(H)}, \\ 0 & p \leq cn^{-1/m_2(H)}. \end{cases}$$

This result is very strongly correlated with a result concerning the following *edge Ramsey property*: for graphs G, H and an integer $r \geq 2$, let $G \to (H)_r^e$ be the property that every r-edge-coloring of G yields a monochromatic H-copy. For $G \sim G(n,p)$ we have the following.

Theorem 3.1.2 (Rödl and Ruciński [41, 42, 43]). *Let $r \geq 2$ be an integer and let H be a graph which is not a forest of stars (and not a path of length 3 if $r = 2$). Let $G \sim G(n,p)$. Then there exist constants $c, C \geq 0$ such that*

$$\lim_{n\to\infty} \Pr\left[G \to (H)_r^e\right] = \begin{cases} 1 & p \geq Cn^{-1/m_2(H)}, \\ 0 & p \leq cn^{-1/m_2(H)}. \end{cases}$$

The resemblance between the two theorems is not coincidental. In fact, if Maker moves first, the 1-statement of Theorem 3.1.1 can be proved almost directly from the 1-statement of Theorem 3.1.2 (and even with the same constant C) by applying a simple *strategy stealing* argument. Without going into much details, such an argument states that if the two players have the same bias, the first player can mimic (steal) any strategy of the second player with a slight modification, and thus can always do at least as well as the second player. In this case, since in the end of the unbiased game at least one of the players has an H-copy in his graph by the Ramsey property, Maker as a first player can ensure his graph does, and win (the authors of [40], however, deduced the 1-statement of Theorem 3.1.1 from some stronger claim they have in the paper). In Subsection 3.1.2 we further discuss the connection between the game and the Ramsey property, the relation to the parameter $m_2(H)$, and those graphs excluded from Theorem 3.1.1.

3.1.1 Our setting and first results

Naturally, an equivalent of the above edge Ramsey property is the following *vertex Ramsey property*: for graphs G and H, and an integer r, let $G \to (H)_r^v$ be the property that every r-vertex-coloring of G yields a monochromatic H-copy. For $G \sim G(n,p)$ we have the analogue of Theorem 3.1.2. Denote by $d_1(H) := \frac{e(H)}{v(H)-1}$ the *1-density* of H, and by $m_1(H) := \max\{d_1(H') \mid H' \subseteq H,\ v(H') \geq 2\}$ the *maximum* 1-*density* of H.

Theorem 3.1.3 (Theorem 1' in [38]). *Let $r \geq 2$ be an integer and let H be a graph with at least one edge (containing a path of length 3 if $r = 2$). Let $G \sim G(n,p)$. Then there exist constants $c, C \geq 0$ such that*

$$\lim_{n\to\infty} \Pr\left[G \to (H)_r^v\right] = \begin{cases} 1 & p \geq Cn^{-1/m_1(H)}, \\ 0 & p \leq cn^{-1/m_1(H)}. \end{cases}$$

It is thus interesting to ask whether a **vertex**-version of Maker-Breaker games on random graphs presents a similar behavior to the vertex Ramsey property, in the same way the **edge**-version of the game resembles the edge Ramsey property.

First, one has to define what a vertex version of the game would be. In this chapter, we suggest the following setting. In a Maker-Breaker game on $V(G)$, the players alternately claim vertices of a graph G according to their bias. For a graph property $\mathcal{P}$, Maker wins the game if the graph induced by his vertices satisfies $\mathcal{P}$, otherwise Breaker wins. Note that in this setting, playing the game on the vertex set of a sparse graph is a very natural choice, since the case $G = K_n$ — which is usually the most basic choice when the players claim edges — is completely trivial: Maker's graph in the end of the game is always a clique on $n/(b+1)$ vertices, no matter how the players play.

It is important to note that for a fixed graph H, the *vertex H-game* (that is, the H-game played on the vertices of a graph) is bias monotone, as claiming more vertices cannot harm Maker. This is not necessarily the case for other vertex games, see Section 3.11 for more details. Furthermore, given a graph H and an integer $b \geq 1$, "being Maker's win in the $(1:b)$ vertex H-game" is also a monotone increasing graph property. Thus we can study the threshold function for this game, namely the function $p^* = p^*(n, b, H)$ that satisfies

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Maker's win in the } (1:b) \text{ vertex } H\text{-game}\right] = \begin{cases} 1 & p = \omega(p^*), \\ 0 & p = o(p^*). \end{cases}$$

For the remainder of this chapter, $p^*_{b,H}$ stands for the threshold probability of being Maker's win in the $(1:b)$ vertex H-game played on $G \sim G(n,p)$; we abbreviate to p^* when there is no risk of confusion. Our main result in this chapter is that the $(1:b)$ vertex H-game is indeed correlated with the aforementioned vertex Ramsey property whenever a subgraph of H of maximal 1-density is either a clique or a cycle, where the only exception is that this subgraph is a triangle and the game is unbiased.

Theorem 3.1.4. *Let k, b be positive integers such that either $k \geq 4$, or $k = 3$ and $b \geq 2$. Let H be a graph for which there exists $H' \subseteq H$ such that $d_1(H') = m_1(H)$, and either $H' = K_k$ or $H' = C_k$. Then there exist constants $c, C > 0$ such that*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Maker's win in the } (1:b) \text{ vertex } H\text{-game}\right] = \begin{cases} 1 & p \geq Cn^{-1/m_1(H)}, \\ 0 & p \leq cn^{-1/m_1(H)}. \end{cases}$$

Note that the case $H' = K_3$ and $b = 1$ was excluded from Theorem 3.1.4. One can easily see that when $H = H' = K_3$, Maker wins the $(1:1)$-game playing only on the graph presented in Figure 3.2, which appears w.h.p. for smaller edge densities than the one given in the 1 -statement of Theorem 3.1.4. It turns out, that in this case Maker wins at the game in the exact same moment that the special graph appears. We elaborate more about this case in the next subsection.

Similarly to the edge version of the game, if the game is unbiased and Maker moves first, the 1-statement of the theorem follows from the 1-statement of Theorem 3.1.3 by applying strategy stealing. However, in order to prove the 1-statement of the theorem in its full generality we need something stronger.

Theorem 3.1.5. *Let $r \geq 2$ and let H be a graph with at least one edge (containing a path of length 3 if $r = 2$). Then there exists a constant $C > 0$ such that for $G \sim G(n,p)$, if $p \geq Cn^{-1/m_1(H)}$, then w.h.p. every subset of $V(G)$ of size $\lfloor \frac{n}{r} \rfloor$ spans an H-copy.*

The proof of this theorem follows easily from the proof of Theorem 3.1.3 in [38]; we omit the straightforward details. In the end of Section 3.4 we go into some more details as we better estimate the constant C from the theorem in case H is a clique with at least four vertices. In any case, the proof of Maker's side in Theorem 3.1.4 is now immediate.

Proof of the 1-statement in Theorem 3.1.4. Consider a $(1:b)$ Maker-Breaker vertex H-game played on $G \sim G(n,p)$. In the end of the game Maker's graph is spanned by a $\frac{1}{b+1}$-fraction of the vertices, no matter how he plays. By Theorem 3.1.5, if $p \geq Cn^{-1/m_1(H)}$, where C is the constant from the theorem corresponding to H and $r = b+1$, then w.h.p. Maker's graph contains an H-copy. $\square$

Note that in fact we got that the 1-statement of Theorem 3.1.4 holds for **any** fixed graph H and not only those specified in the theorem (if H does not meet the requirements of Theorem 3.1.5 then Maker's win is trivial). The proof of the 0-statement of Theorem 3.1.4 will be presented in Sections 3.4 (cliques), 3.5 (cycles), and 3.6 (triangle, biased game).

In contrast to the graphs specified in Theorem 3.1.4, the correlation between the Ramsey property and the game is not maintained in case of forests. Indeed, if H is a forest, then $m_1(H) = 1$. Theorem 3.1.3 therefore implies that the threshold function for the corresponding vertex Ramsey property is $p = 1/n$. However, the following theorem shows that the order of magnitude of the threshold function for the vertex H-game is significantly smaller.

Theorem 3.1.1. *Let H be a forest consisting of trees $T_1, \dots, T_k$, and let b be a positive integer.*

1. *If H is a tree, i.e. $k = 1$, then there exists a tree T such that Maker wins the $(1:b)$ H-game played on $V(T)$.*

2. *For any integer $k \geq 1$, and for every $1 \leq i \leq k$, let $T^{(i)}_{\min}$ be a tree of minimal size such that Maker, as a first player, wins when playing the $(1:b)$ T_i-game on its vertex set. Let $T_{\max}$ be a tree of maximal size among all trees $T^{(i)}_{\min}$, and let $v = v(T_{\max})$, $e = e(T_{\max})$. Then $p^*_{b,H} = n^{-v/e}$.*

In the next subsection, one can find our precise result regarding the unbiased K_3-game, in particular, we show that $p^*_{1,K_3} = n^{-7/10} = o(n^{-1/m_1(K_3)})$, as $m_1(K_3) = 3/2$. We conjecture that as in the edge version of the game, forests and triangles are the only exceptions for the very strong connection between the game and the Ramsey property. A graph H is called *strictly* 1-*balanced* if $d_1(H) > d_1(H')$ for every $H' \subsetneq H$.

Conjecture 3.1.2. *Let $b \geq 1$ be an integer, and let H be a graph for which there exists $H' \subseteq H$ such that $d_1(H') = m_1(H)$, H' is strictly 1-balanced and is not a single edge, and in case $b = 1$ also not a triangle. Then there exist constants $c, C > 0$ such that*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \textit{ is Maker's win in the } (1:b) \textit{ vertex } H\textit{-game}\right] = \begin{cases} 1 & p \geq Cn^{-1/m_1(H)}, \\ 0 & p \leq cn^{-1/m_1(H)}. \end{cases}$$

Remark 3.1.6. The 1-statement in Conjecture 3.1.2 follows immediately from Theorem 3.1.5, as shown in the proof of the 1-statement of Theorem 3.1.4. Thus the interesting part of the conjecture is Breaker's side.

Note that there are two types of graphs not covered by Conjecture 3.1.2. In the terminology of the conjecture, observe that H' is an edge if and only if H is a forest, so this case is covered by Theorem 3.1.1. In the next subsection we consider the unbiased triangle game. It remains to deal with unbiased H-games where $H \neq K_3$ is a graph that satisfies $m_1(H) = 3/2$, and every strictly 1-balanced subgraph $H' \subset H$ with $d_1(H') = 3/2$ is a triangle. Let $\mathcal{H}$ be the family of all such graphs and let $H \in \mathcal{H}$. By Corollary 3.1.5 (in the next subsection), if $p = o\left(n^{-7/10}\right)$, then w.h.p. Breaker can prevent Maker from claiming a triangle, thus winning the H-game. This implies that $p^*_{1,H} = \Omega\left(n^{-7/10}\right)$. On the other hand, the fact that $p^*_{1,H} = O\left(n^{-2/3}\right)$ is an immediate corollary of Theorem 3.1.5. We show that if Conjecture 3.1.2 is confirmed, then there exist infinitely many rational values $\alpha \in \left(\frac{10}{7}, \frac{3}{2}\right)$ for which there exists a graph $H \in \mathcal{H}$ such that $p^*_{1,H} = n^{-1/\alpha}$. We actually prove something more general.

Theorem 3.1.3. *Let H' be a graph for which there exists $H'' \subseteq H'$ such that $d_1(H'') = m_1(H') = \alpha$, H'' is strictly 1-balanced and is not a triangle. Let $v \in V(H')$ and let H be the graph obtained by connecting v to a triangle via a path of length four (see Figure 3.1). Assume that Conjecture 3.1.2 holds.*

- *In case $\alpha > 10/7$ there exist positive constants c, C such that*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Maker's win in the } (1:1) \text{ vertex } H\text{-game}\right] = \begin{cases} 1 & p \geq Cn^{-1/\alpha}, \\ 0 & p \leq cn^{-1/\alpha}. \end{cases},$$

  where the 0-statement holds if the 0-statement of Conjecture 3.1.2 holds (and for the same c).

- *In case $\alpha \leq 10/7$ we have*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Maker's win in the } (1:1) \text{ vertex } H\text{-game}\right] = \begin{cases} 1 & p = \omega(n^{-7/10}), \\ 0 & p = o(n^{-7/10}). \end{cases}.$$

Note that the most interesting case is where $\alpha \in \left(\frac{10}{7}, \frac{3}{2}\right)$, as it shows that there exists an infinite family of graphs for which the threshold probability of the game is not determined by a subgraph on which the maximum 1-density is obtained (which in this case is a triangle). This is the equivalent of the phenomenon demonstrated in Theorem 4 of [40].

3.1.2 Global vs. local and the random graph process

In this section we consider a somewhat different model for the random graph on which the game is played. Let $[n] = \{1, \ldots, n\}$ be the set of vertices, and let $\pi : \binom{[n]}{2} \to \left[\binom{n}{2}\right]$ be an arbitrary permutation. For $G_i = ([n], \pi^{-1}([i]))$, the sequence of graphs $\tilde{G} = \{G_i\}_{i \in \left[\binom{n}{2}\right]}$ is called a graph process. The *random graph process* is the graph process obtained by choosing π uniformly at random from all possible permutations. This random setting generates a random graph model that is closely related to the standard random graph model $G(n,p)$ we have considered so far (see e.g. [33], and more discussion in Section 3.2).

For a given graph process $\tilde{G}$, the *hitting time* of a monotone increasing graph property $\mathcal{P}$ is defined to be $\tau(\tilde{G}, \mathcal{P}) = \min\{i \mid G_i \in \mathcal{P}\}$. We would like to examine the hitting time of the property "being Maker's win in the $(1:1)$ vertex K_3-game". It turns out that w.h.p. the graph becomes Maker's win at the same moment a certain fixed graph appears in G for the first time. Before stating the result formally we need to describe this graph and to introduce new notation.

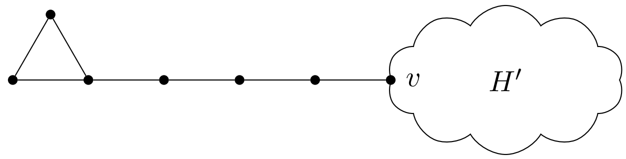

Figure 3.1: The construction of H from H'

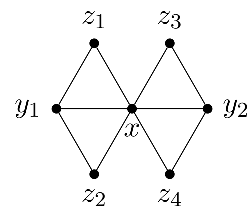

Figure 3.2: The graph DD

Definition 3.1.7. A *diamond* is a copy of K_4 with one edge missing. A *double diamond*, denoted by DD, consists of two diamonds with vertex sets $\{x, y_1, z_1, z_2\}$ and $\{x, y_2, z_3, z_4\}$ where z_1z_2 and z_3z_4 are the missing edges (see Figure 3.2). The intersection of the two diamonds, namely the vertex x, is called *the center* of DD.

Throughout this chapter we use the following notation: for an integer k and a fixed graph H, let $\mathcal{G}_{kH}$ denote the graph property of containing k (possibly intersecting) copies of H. We abbreviate $\mathcal{G}_{1H}$ to $\mathcal{G}_H$.

Theorem 3.1.4. *Let $\mathcal{M}^1_{K_3}$ and $\mathcal{M}^2_{K_3}$ be the graph properties of being Maker's win in the $(1:1)$ vertex K_3-game, where Maker moves first or second, respectively. For a random graph process $\tilde{G}$, w.h.p. $\tau(\tilde{G}, \mathcal{M}^1_{K_3}) = \tau(\tilde{G}, \mathcal{G}_{DD})$ and $\tau(\tilde{G}, \mathcal{M}^2_{K_3}) = \tau(\tilde{G}, \mathcal{G}_{2DD})$.*

The following corollary of Theorem 3.1.4 is due to the asymptotic connection between $G(n,p)$ and the random graph process, combined with Theorem 3.2.13.

Corollary 3.1.5. *Let $p = p(n)$ and let $x = np^{10/7}$. Then assuming Maker moves first we have*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Maker's win in the } (1:1) \text{ vertex } K_3\text{-game}\right] = \begin{cases} 0 & if\ x \to 0, \\ 1 - e^{-c^7/8} & if\ x \to c \in \mathbb{R}^+, \\ 1 & if\ x \to \infty. \end{cases}$$

In particular, the game has a (coarse) threshold at $p = n^{-7/10}$. If Maker moves second, the same holds, with the term $1 - e^{-c^7/8}$ being replaced with $1 - e^{-c^7/8} - c^7/8 \cdot e^{-c^7/8}$.

In particular, this game has a coarse threshold (at $p = n^{-7/10}$), and note that this is not the case in Theorem 3.1.4 (and Conjecture 3.1.2). Recall that the proof of Maker's side in Theorem 3.1.4 and Conjecture 3.1.2 is trivial: the graph is such that Maker wins no matter how he plays. We say that in these cases Maker wins due to a *global* reason, that is, the structure of the entire graph. This stands in sharp contrast to Maker's side in the vertex H-game when H is a forest or a triangle, as shown in Theorems 3.1.1 and 3.1.4, where Maker applies a straightforward winning strategy on some small, fixed graph $\hat{H}$. We say that in these cases Maker wins due to a *local* reason, that is, the appearance of $\hat{H}$ in the random graph. The question is what is more likely to appear first in the random graph process — a local reason or a global reason. In this chapter we show that when H is either a triangle or a forest, w.h.p. the local reason appears first, and conjecture that these are the only cases (we actually show that a triangle is an exception only in the unbiased game).

The exact same phenomenon — of Maker winning globally unless H is either a forest or a triangle — was proven for the unbiased edge version of H-games. Theorem 3.1.1 shows that the threshold for most graphs is the one matching the global reason, that is, $n^{-1/m_2(H)}$ (this was already shown for cliques in [45]). The case that H is a forest was considered by Stojaković in [44, Lemma 36], where he showed (and by that inspired our Theorem 3.1.1) that in this case Maker wins due to a local reason, and so there exists some constant $\alpha(H) > 1$ such that the threshold function of the game is $n^{-\alpha(H)}$, while $m_2(H) = 1$ (in fact, he only considered trees, but his result may be easily generalized to forests, see Remark 3.8.1 in Section 3.8). The case $H = K_3$ was first considered in [45], where the threshold function was shown to be $n^{-5/9}$, since Maker wins locally on a copy of K_5^-, the clique on five vertices with one edge missing. Later, in [39], this result was improved to a hitting time result, that is, assuming Maker moves first, in the random graph process w.h.p. the graph becomes his win at the same moment the first copy of K_5^- appears. This is of course the equivalent of Theorem 3.1.4.

3.1.3 Avoider-Enforcer games

To the best of our knowledge, Avoider-Enforcer games played on random graphs were only considered in [18], where the authors analyzed the k-connectivity, Hamiltonicity, and perfect matching monotone biased games played on the edge set of $G \sim G(n,p)$, determining the threshold bias as a function of p.

It is worth mentioning that when playing on the vertex set of a graph, "being Enforcer's win" is trivially a monotone increasing graph property for both strict and monotone settings (a winning strategy for Enforcer remains such if we add edges to the graph). Thus we can define the threshold probability for these games, just as in Maker-Breaker games. Note that this is not necessarily the case when playing the edge version of the game: in general, Avoider-Enforcer games – in contrast to Maker-Breaker games – do not have hypergraph monotonicity in the following sense. It is possible for Enforcer to win an $(a : b)$ game $(X, \mathcal{F})$, but lose an $(a : b)$ game $(X', \mathcal{F}')$, even if $X \subseteq X'$ and $\mathcal{F} \subseteq \mathcal{F}'$. We now state our results, starting with the monotone game.

Theorem 3.1.6. *Let H be a fixed graph and let a, b be two positive integers. Then the threshold probability for Enforcer's win in the monotone $(a : b)$ vertex H-game played on $G \sim G(n,p)$ is $p = n^{-1/m(H)}$. Furthermore, if H is strictly balanced, then there exists a constant $N = N(a, b, v(H))$, such that in the random graph process, $\tau(\tilde{G}, \mathcal{E}) = \tau(\tilde{G}, \mathcal{G}_{NH})$ holds w.h.p., where $\mathcal{E}$ denotes the property "being Enforcer's win in the monotone $(a : b)$ vertex H-game".*

In Section 3.10 we have a short discussion about strict H-games in general. However, we state an explicit result only for the $(1 : 1)$ triangle-game, for two reasons. First, this is an interesting game: the threshold probability for it when played on a random graph is unique comparing to other H-games or to biased triangle games, whether we consider the Maker-Breaker games for both edge and vertex versions, or Client-Waiter games (which will be introduced shortly), again for both edge and vertex versions. Second, it turns out that this game presents an analogous behavior to that of the Maker-Breaker game in the following way.

Recall that when the random graph process is considered, Maker wins the $(1 : 1)$ triangle-game as soon as the first or the second DD-copy appears in the graph, depending on the identity of the **first** player. The reason for this difference is that both players **wish to claim** the center of a DD-copy. Analogously, Enforcer wins in the $(1 : 1)$ triangle Avoider-Enforcer game as soon as the first or the second DD-copy appears in the graph, depending on the identity of the **last** player to play (that is, the player who claims the last free vertex in the game; not to be confused with the player who plays second). Here both players **wish to avoid claiming** the center of a DD-copy. It is important to note that unlike the edge version of positional game played on random graphs, the identity of the last player is determined by the identity of the first player and the number of vertices in the graph, which are both part of the definition of the game. It does not depend on the random graph process itself or on the number of edges in the graph in any way, which makes the following theorem, and this whole discussion, well defined.

Theorem 3.1.7. *Let $\mathcal{E}^A_{K_3}$ and $\mathcal{E}^E_{K_3}$ be the graph properties of being Enforcer's win in the strict $(1:1)$ vertex K_3-game, where Avoider or Enforcer, respectively, makes the last move in the game. For a random graph process $\tilde{G}$, w.h.p. $\tau(\tilde{G}, \mathcal{E}^A_{K_3}) = \tau(\tilde{G}, \mathcal{G}_{DD})$ and $\tau(\tilde{G}, \mathcal{E}^E_{K_3}) = \tau(\tilde{G}, \mathcal{G}_{2DD})$.*

3.1.4 Waiter-Client and Client-Waiter games

Waiter-Client and Client-Waiter games have drawn much interest in the last several years, resulting in quite a few papers. We do not intend to provide a full background on this subject, as we limit our focus to H-games and games on random boards. We refer the reader to the papers [3, 5, 16, 30], and to the many other works cited in them.

It is trivial to see that whether Waiter offers exactly $a+b$ elements per move or not, in either of the games, the following holds. For any monotone increasing graph property $\mathcal{P}$, and for both games, when playing an $(a:b)$ game $\mathcal{P}$ on the vertex set of a graph, "being the builder's win" is a monotone increasing graph property (note that this is not true in the edge version of the Client-Waiter game if we do not allow Waiter to offer fewer edges per round, which is another important motivation for this adjustment of rules). Indeed, if $G \subseteq G'$ and $V(G) = V(G')$, then in both games a winning strategy for the builder in the game played on $V(G)$ remains such without any changes for the game played on $V(G')$. Once again, this important — and non-trivial — property allows us to define the threshold probability for these games.

We now present some results, which demonstrate that Waiter-Client and Client-Waiter games not only resemble Avoider-Enforcer and Maker-Breaker games, respectively, in the roles of the players, but also in the outcome of the corresponding H-games. Theorem 3.1.6 basically shows that for any a,b and H, Enforcer wins the $(a:b)$ H-game as soon as G contains sufficiently many H-copies. The following theorem shows that the same holds for Waiter in the Waiter-Client game.

Theorem 3.1.8. *Let H be a fixed graph and let a, b be two positive integers. Then the threshold probability for Waiter's win in the $(a:b)$ Waiter-Client vertex H-game played on $G \sim G(n,p)$ is $p^* = n^{-1/m(H)}$. Furthermore, if H is strictly balanced, then there exists a constant $N = N(a, b, v(H))$, such that in the random graph process, $\tau(\tilde{G}, \mathcal{W}) = \tau(\tilde{G}, \mathcal{G}_{NH})$ holds w.h.p., where $\mathcal{W}$ denotes the property "being Waiter's win in the $(a:b)$ Waiter-Client vertex H-game".*

Moving to Client-Waiter games, we observe that they feature an almost identical behavior to that of the corresponding Maker-Breaker games, and we have the following perfect analogues of Theorems 3.1.4 and 3.1.1, and Conjecture 3.1.2.

Theorem 3.1.9. *Let k, b be positive integers such that either $k \geq 4$, or $k = 3$ and $b \geq 2$. Let H be a graph for which there exists $H' \subseteq H$ such that $d_1(H') = m_1(H)$, and either $H' = K_k$ or $H' = C_k$. Then there exist constants $c, C > 0$ such that*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Client's win in the } (1:b) \text{ vertex } H\text{-game}\right] = \begin{cases} 1 & p \geq Cn^{-1/m_1(H)}, \\ 0 & p \leq cn^{-1/m_1(H)}. \end{cases}$$

Theorem 3.1.10. *Let H be a forest consisting of trees $T_1, \dots, T_k$, and let b be a positive integer. The following hold for the $(1:b)$ Client-Waiter vertex H-game.*

1. *If H is a tree, i.e. $k = 1$, then there exists a tree T such that Client wins the game played on T.*
2. *For any integer $k \geq 1$, and for every $i \in [k]$, let $T_{\min}^{(i)}$ be a tree of minimal size such that Client wins the $(1:b)$ T_i-game on its vertex set. Let $T_{\max}$ be a tree of maximal size among all trees $T_{\min}^{(i)}$, and let $v = v(T_{\max})$, $e = e(T_{\max})$. Then the threshold for Client's win in the H-game is $p^* = n^{-v/e}$.*

Conjecture 3.1.11. *Let $b \geq 1$ be an integer, and let H be a graph for which there exists $H' \subseteq H$ such that $d_1(H') = m_1(H)$, H' is strictly 1-balanced and is not a single edge, and in case $b = 1$ also not a triangle. Then there exist constants $c, C > 0$ such that*

$$\lim_{n\to\infty} \Pr\left[G \sim G(n,p) \text{ is Client's win in the } (1:b) \text{ vertex } H\text{-game}\right] = \begin{cases} 1 & p \geq Cn^{-1/m_1(H)}, \\ 0 & p \leq cn^{-1/m_1(H)}. \end{cases}$$

More discussion about the similarities between Client-Waiter games and Maker-Breaker games can be found in Sections 3.3, 3.8 and 3.10.

We finish with the $(1:1)$ Client-Waiter triangle-game, where once again we observe the phenomenon of "the $(1:1)$ triangle-game behaves differently". This game in fact presents the most interesting behavior of all games considered in this chapter, and, if Conjectures 3.1.2 and 3.1.11 turn out to be true, then of all vertex H-games.

Recall that in the unbiased edge version, Maker wins the triangle game locally, but wins due to a global reason for most other H-games. Dean and Krivelevich showed in [16] that for any given graph H, the threshold probability for the Client-Waiter H-game is exactly the same as in the Maker-Breaker H-game even when Waiter is allowed an arbitrary fixed bias. Furthermore, they showed that not only Client wins locally the unbiased triangle game, but also that the fixed graph for which Client awaits is the same one Maker waits for, namely K_5^-.

However, in the vertex version of these games, Client and Maker need different fixed graphs to apply their winning strategies on. While Maker wins as soon as a double diamond appears, Client has to wait further for the appearance of a *triple diamond*, which we describe in Section 3.10. In particular, unlike any other game mentioned in this chapter (for both edge and vertex versions), in the unbiased triangle-game on the vertex set, Waiter is significantly stronger then Breaker.

But there is more to it. As it turns out, the threshold probability for Client's "local win" is of the same order of magnitude as the threshold probability for Client's "global win". It follows that the threshold probability for the $(1:1)$ Client-Waiter triangle-game is coarse from below but (semi-)sharp from above. Consequently, there is a range of values of p for which all of the following occur with probability bounded away from zero in $G \sim G(n,p)$: Client wins due to a local reason, Client wins due to a global reason, Waiter wins. This is all stated explicitly in the following theorem.

Theorem 3.1.12. *There exist positive constants c, C, α such that in the $(1:1)$ Client-Waiter vertex K_3-game played on $G \sim G(n,p)$ the following hold.*

(1) *Waiter wins w.h.p. for any $p = o\left(n^{-2/3}\right)$;*

(2) *For any constant* $0 < d < c$ *and for* $p = dn^{-2/3}$, *we have*

$$\lim_{n\to\infty} \Pr[\textit{Waiter wins the game}] \geq \alpha;$$

(3) *For any constant* $d > 0$ *there exists a constant* $\beta > 0$, *such that for* $p = dn^{-2/3}$ *we have*

$$\lim_{n\to\infty} \Pr[\textit{Client wins the game}] \geq \beta;$$

(4) *Client wins w.h.p. for any* $p \geq Cn^{-2/3}$.

3.1.5 Organization of the chapter

The organization of the chapter is as follows. In Section 3.2 we provide notation and technical preliminaries, and present some random graph results. In Section 3.3 we describe a strategy for the spoiler (either Breaker or Waiter) which can be used in any H-game, and additionally focus on the case $H = K_3$. The content of this section is the basis for the remainder of the chapter. We then prove Theorem 3.1.4, dividing the proof into the following three sections: in Section 3.4 we handle the case $H = K_k$ and $k \geq 4$, in Section 3.5 the case $H = C_k$ and $k \geq 4$, and in Section 3.6 we address the case $H = K_3$ and $b \geq 2$. The proof of Theorem 3.1.4 (the unbiased Maker-Breaker triangle-game) is given in Section 3.7. In Section 3.8 we deal with forests, and in particular prove Theorems 3.1.1 and 3.1.10. In Section 3.9 we prove Theorem 3.1.3. Avoider-Enforcer, Waiter-Client and Client-Waiter games are all discussed in Section 3.10, and the proofs for all corresponding theorems are given (except for Theorem 3.1.10). Finally, in Section 3.11 we provide some concluding remarks and open problems.

3.2 Preliminaries

Our graph-theoretic is standard and follows that of [46]. In particular we use the following. For a graph $G = (V, E)$, let $v(G) = |V(G)|$ and $e(G) = |E(G)|$. For a set of vertices $U \subseteq V$, let $G[U]$ denote the corresponding vertex-induced subgraph of G, and let $N_G(U) = \{v \in V \setminus U \mid \exists u \in U \text{ such that } uv \in E\}$ denote the external neighborhood of U in G. For a vertex $v \in V$ we abbreviate $N_G(\{v\})$ to $N_G(v)$ and let $d_G(v) = |N_G(v)|$ denote the degree of v in G. Often, when there is no risk of ambiguity, we omit the subscript G in the above notation.

Considering a fixed graph H, we say that a graph G is *H-free* if it does not contain a copy of H as a subgraph. More generally, for a family $\mathcal{F}$ of fixed graphs, we say that G is *$\mathcal{F}$-free* if it H-free for any $H \in \mathcal{F}$. We say that two H-copies in a graph G *intersect* if they are not vertex disjoint.

Our results are asymptotic in nature and we assume that n is large enough where needed. We omit floor and ceiling signs whenever these are not crucial.

3.2.1 Intersecting H-copies

Let H be a connected graph. We now present notation for some specific structures involving intersecting H-copies, which will be very useful in our proofs.

Definition 3.2.1. A graph Γ is an *H-chain* of length t if it consists of $t \geq 1$ copies $H_1, \ldots, H_t$ of H, such that

$$\forall\ 1 \leq i < j \leq t: \ |V(H_i) \cap V(H_j)| = \begin{cases} 1 & j - i = 1 \\ 0 & o.w. \end{cases}$$

Note that if H is a clique then for every integer t there exists exactly one H-chain of length t (up to isomorphism), and therefore we can refer to **the** H-chain of length t in this case. This is not the case for any other graph, as there are different chains of each length, according to which vertices are in the intersections between consecutive H-copies.

Definition 3.2.2. A graph Γ is an *H-cycle* of length t if it consists of $t \geq 3$ copies $H_1, \ldots, H_t$ of H, such that

$$\forall\ 1 \leq i, j \leq t:\ |V(H_i) \cap V(H_j)| = \begin{cases} 1 & |j-i| \equiv 1 \pmod{t} \\ 0 & o.w. \end{cases}$$

Definition 3.2.3. Given a graph G, an edge $e \in E(G)$ belonging to two distinct H-copies in G is called *dangerous* with respect to H. Since H is always clear from the context, we simply refer to such edges as dangerous.

3.2.2 Pairings

Consider a $(1:b)$ Maker-Breaker game $(X, \mathcal{F})$, and let $\mathcal{A} = \{A_1, A_2, \ldots\}$ be a family of pairwise disjoint subsets of X, each of size at most $b+1$. Suppose that for every i, whenever Maker claims an element of A_i, Breaker responds by claiming all free elements of A_i (and, if necessary, completes his move according to some strategy). Then at the end of the game Maker occupies at most one element from each member of $\mathcal{A}$. This seemingly trivial strategy, called the *pairing strategy*, turns out to be one of the most useful and basic strategies in positional games. It can be Breaker's entire strategy, or a part of a more involved one. Waiter can use the same strategy in a Client-Waiter game, by offering an entire set A_i in each move (and offer the remaining elements of X, if there are any, according to some other strategy).

Pairing strategies are fundamental for our proofs in this chapter: our main method is to provide the spoiler (either Breaker or Waiter) with a winning strategy for a game played on a "simple" graph, which can be extended via a pairing strategy to a winning strategy for the original game. This is explained in more details in Section 3.3. Furthermore, the winning strategy for this simple graph usually involves another pairing strategy, applied to each connected component separately.

Two of the basic connected components we have to deal with when analyzing K_3-games are the following.

- Let Tr be the graph with vertex set $V(Tr) = \{v_1, v_2, v_3, v_4, v_5\}$ and edge set $E(Tr) = \{v_1v_2, v_1v_3, v_2v_3, v_2v_4, v_3v_4, v_3v_5, v_4v_5\}$, as shown in Figure 3.3.
- For an integer $t \geq 2$ let DD_t be the graph obtained by taking a K_3-chain of length t consisting of the triangles $\{a_ib_ic_i\}_{i=1}^{t}$ where $c_i = a_{i+1}$ for every $0 < i < t$, and adding to it two triangles, a_1c_1y and $c_{t-1}c_tx$, where x, y are new vertices (see Figure 3.4). Note that we got two diamonds connected by a K_3-chain of length $t-2$, and in particular $DD_2 = DD$ (recall Definition 3.1.7).

We now provide pairing strategies for these two graphs, and for two others, which are used in several unbiased K_3-games. For each of these graphs we provide a list of pairs Λ, to which we refer as its *natural pairs*. A pairing strategy with respect to these pairs is called *the natural pairing strategy* for the graph. Note that for the first two graphs specified below, if the spoiler uses the natural pairing strategy, he prevents the builder from creating a triangle. This is also true for the latter two, with the additional assumption that the spoiler claims the vertex x, with respect to the labeling of Figures 3.4 and 3.2.

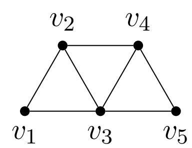

Figure 3.3: The graph Tr

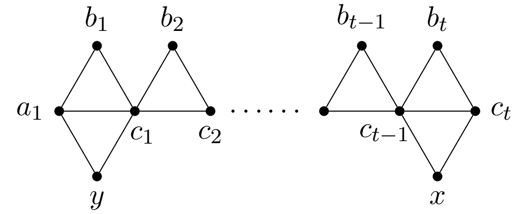

Figure 3.4: The graph DD_t

Definition 3.2.4. Given a K_3-cycle of length $t \geq 4$, denote the vertices of its triangles by $\{a_i, b_i, c_i\}_{i=1}^t$, where $c_t = a_1$ and $c_i = a_{i+1}$ for every $0 < i < t$. The natural pairs for this graph are $\Lambda = \{\{a_i, b_i\}\}_{i=1}^t$.

Definition 3.2.5. The set of natural pairs of the graph Tr, using the labeling of Figure 3.3, is $\Lambda = \{\{v_2, v_3\}, \{v_4, v_5\}\}$.

Definition 3.2.6. For any integer $t \geq 3$, the set of natural pairs of the graph DD_t, using the labeling of Figure 3.4, is $\Lambda = \{a_1, c_1\} \cup \{\{b_i, c_i\}\}_{i=2}^t$.

Definition 3.2.7. The set of natural pairs of the graph DD, using the labeling of Figure 3.2, is $\Lambda = \{\{y_1, y_2\}, \{z_1, z_2\}, \{z_3, z_4\}\}$.

The natural pairs of the graph DD are also used in a different manner in the proof of Theorem 3.1.7, where we use the following simple observation.

Observation 3.2.8. *If $U \subseteq V(DD)$ contains the center of DD (the vertex x in the labeling of Figure 3.2) and at least one vertex from each of the natural pairs of DD, then $DD[U]$ contains a triangle.*

3.2.3 Strictly balanced and strictly 1-balanced graphs

Let $d(H) := e(H)/v(H)$ be the *density* of H, and $m(H) := \max\{d(H') \mid H' \subseteq H\}$ is the *maximum density* of H. A graph H is called *balanced* if $d(H) = m(H)$, and *strictly balanced* if $d(H) > d(H')$ for every $H' \subsetneq H$.

Claim 3.2.9. *For every strictly balanced graph H, if Γ is a graph consisting of two H-copies with a non-empty intersection H', then $m(\Gamma) > m(H) + \frac{1}{2v(H)^2}$.*

Proof. Let $v(H) = h$, $e(H) = m$, $v(H') = h'$ and $e(H') = m'$, and note that h, m and h' are all necessarily positive. Since $H' \subseteq H$, and since H is strictly balanced, we get $\frac{m'}{h'} < \frac{m}{h}$, which implies that $mh' - m'h$ is a positive integer. It follows that

$$m(\Gamma) - m(H) \geq d(\Gamma) - m(H) = \frac{2m - m'}{2h - h'} - \frac{m}{h} = \frac{mh' - m'h}{h(2h - h')} > \frac{1}{2h^2}. \qquad \square$$

Claim 3.2.10. *Every strictly 1-balanced graph with at least three vertices is 2-vertex-connected.*

Proof. Let H be a strictly 1-balanced graph with at least three vertices, and assume for contradiction that H is not 2-vertex-connected. Then there exist two subgraphs $H_1, H_2 \subseteq H$, each containing at least two vertices, such that $H = H_1 \cup H_2$ and $|V(H_1) \cap V(H_2)| = 1$. For $i = 1, 2$, let $v(H_i) = v_i$ and $e(H_i) = e_i$. Since H is strictly 1-balanced we have $d_1(H) > d_1(H_1)$, that is

$$\frac{e_1 + e_2}{v_1 + v_2 - 2} > \frac{e_1}{v_1 - 1}.$$

Rearranging, we get $e_2 v_1 - e_1 v_2 > e_2 - e_1$. Similarly, from $d_1(H) > d_1(H_2)$ we get $e_1 v_2 - e_2 v_1 > e_1 - e_2$. Putting the two inequalities together, we get $e_2 - e_1 < e_2 v_1 - e_1 v_2 < e_2 - e_1$, a contradiction. $\square$

3.2.4 Random graph results

For the proofs of Theorem 3.1.4 and Corollary 3.1.5 (see Section 3.7) we rely on the well-known asymptotic connection between the random graph model $G(n, p)$ and the random graph process, given in the following proposition. Roughly speaking, if a typical graph from one of these models satisfies some monotone increasing graph property $\mathcal{P}$ with some given probability, then a typical graph from the other model (with the corresponding parameters) also satisfies $\mathcal{P}$ with the same probability (see [10, 21, 33] for more details).

Proposition 3.2.11. *Let $\mathcal{P}$ be a monotone graph property, let $p = p(n)$, $0 < a < 1$, let $\tilde{G} = (G_i)$ be a random graph process and let $G \sim G(n, p)$. Then there are constants $c_1, c_2 > 0$ such that:*

- $\Pr[G \in \mathcal{P}] = a \;\Rightarrow\; \Pr[G_i \in \mathcal{P}] = a$ *for* $i = c_1 n^2 p$.
- $\Pr[G_i \in \mathcal{P}] = a \;\Rightarrow\; \Pr[G \in \mathcal{P}] = a$ *for* $i = c_2 n^2 p$.

The next two theorems deal with the appearance of H-copies in $G \sim G(n, p)$, where the order of magnitude of p is compared to $n^{-1/m(H)}$.

Theorem 3.2.12 (Theorem 5.3 in [21])**.** *For any fixed graph H and for any fixed positive integer N, w.h.p. $G \sim G(n, p)$ contains N vertex disjoint copies of H for every $p = \omega(n^{-1/m(H)})$, and is H-free for every $p = o(n^{-1/m(H)})$.*

Theorem 3.2.13 (Theorem 3.19 in [33])**.** *Let H be a strictly balanced graph and let $p = cn^{-1/m(H)}$ for some constant $c > 0$. The number of H-copies in $G \sim G(n, p)$ converges in distribution to Poisson(λ), the Poisson distribution with parameter $\lambda := c^{e(H)}/|Aut(H)|$, where $Aut(H)$ is the automorphism group of H. In particular,*

$$\lim_{n\to\infty} \Pr(G \sim G(n, p) \text{ is } H\text{-free}) = e^{-\lambda}.$$

Using Theorem 3.2.12 we obtain the following characterization of the typical structure of sparse random graphs.

Claim 3.2.14. *Let k be a positive integer and let $G \sim G(n, p)$ for $p = o\left(n^{-\frac{k+1}{k}}\right)$. Then w.h.p. every connected component of G is a tree with at most k vertices.*

Proof. Let $\mathcal{F}$ be the family of all trees with $k + 1$ vertices and all cycles of length at most k. Every $H \in \mathcal{F}$ satisfies either $m(H) = \frac{k}{k+1}$ (if H is a tree) or $m(H) = 1$ (if H is a cycle). Since $\mathcal{F}$ is a finite family, a simple union bound on the members of $\mathcal{F}$ implies that G is w.h.p. $\mathcal{F}$-free by Theorem 3.2.12. Hence every connected component of G has at most k vertices, because every larger component contains a tree with $k + 1$ vertices as a subgraph. The absence of all short cycles from G completes the proof. $\square$

We conclude this section with two results concerning the appearance of any number of H-copies in a random graph process.

Claim 3.2.15. *For any fixed graph H and every finite family of fixed graphs $\mathcal{F} = \{H_1, \dots, H_k\}$, each satisfying $m(H_i) > m(H)$, and for every positive integer N, the appearance of N vertex disjoint copies of H in a random graph process occurs w.h.p. before the appearance of any member of $\mathcal{F}$.*

Proof. Let $\mathcal{G}'_{NH}$ be the graph property of containing N vertex disjoint H-copies, and let $\mathcal{G}_{\mathcal{F}}$ be the graph property of not being $\mathcal{F}$-free. By Theorem 3.2.12, for $p = n^{-1/m(H)} \ln n$, w.h.p. $G \sim G(n,p)$ contains N vertex disjoint H-copies, and by the same theorem, for $p = n^{-1/m(H)} \ln^2 n$, w.h.p. $G \sim G(n,p)$ is $\mathcal{F}$-free (since $\mathcal{F}$ is finite and this value of p satisfies $p = o(n^{-1/m(H_i)})$ for every $i \in [k]$). Proposition 3.2.11 implies that w.h.p. $\tau(\tilde{G}, \mathcal{G}'_{NH}) = O(n^{2-1/m(H)} \ln n)$ and $\tau(\tilde{G}, G_{\mathcal{F}}) = \Omega(n^{2-1/m(H)} \ln^2 n)$, which completes the proof. Note that we had to consider each property separately since "containing N vertex disjoint H-copies and being $\mathcal{F}$-free" is not a monotone graph property. $\square$

Corollary 3.2.16. *For every strictly balanced graph H and for any integer N, w.h.p. the first N copies of H which appear in a random graph process are all vertex disjoint.*

Proof. Let $\mathcal{F}$ be the family of all graphs consisting of two H-copies with a non-empty intersection. Then $\mathcal{F}$ is finite, and by Claim 3.2.9 every member in it has maximal density larger than $m(H)$. The result then immediately follows from Claim 3.2.15. $\square$

3.3 General results and tools for H-Games

In this section we provide general tools, which will be fundamental for the proofs of the 0-statements of Theorems 3.1.4 and 3.1.9. We discuss Maker-Breaker and Client-Waiter H-games, both played on $V(G)$ where $G \sim G(n,p)$ and H is a fixed, strictly 1-balanced graph on at least three vertices. In particular, H is connected by Claim 3.2.10. Since the proofs for both theorems are almost identical, in this section we sometimes refer to Breaker and Waiter collectively as "the spoilers".

First we present a process of deleting vertices and edges from an arbitrary graph G, until we get a subgraph of $G^* \subseteq G$ with the following property: any winning strategy for the spoiler for the H-game played on $V(G^*)$ may be extended to a winning strategy for the original game, played on $V(G)$. Second, we characterize the possible connected components of G^*. Last, in the light of this characterization, we focus on clique games and provide more specific results.

3.3.1 The (H, b) graph deletion algorithm

Given a connected graph H and a positive integer b, we define the (H, b) deletion algorithm, applicable to any graph G. We first need the following definitions.

Definition 3.3.1. Let G, H be graphs, where H is connected, and let b be a positive integer. All the following are defined with respect to G, H and b. A *bad vertex* is a vertex $v \in V(G)$ which is not contained in any H-copy in G. A *bad edge* is an edge $e \in E(G)$ which is not contained in any H-copy in G. A *bad set* is a set $U \subseteq V(G)$ of size $2 \leq |U| \leq b+1$ such that there is no H-copy $\hat{H} \subseteq G$ satisfying $|V(\hat{H}) \cap U| = 1$. A bad set of size two is referred to as a *bad pair*. A *small component* is a connected component of G with at most $(b+1)(v(H)-1)$ vertices.

The algorithm Begin with $G_0 = G$. For as long as possible, obtain G_{i+1} from G_i by deleting arbitrarily either a bad vertex, a bad edge, a bad set or a small component, where these are all defined with respect to the current graph G_i and the fixed H and b. The *output* of the algorithm is a sequence $U = \{U_1, U_2, \dots\}$ of all bad sets that were deleted during the process (if there were any) in the order of deletion (that is, U_i was the ith bad set to be deleted), a sequence $W = \{W_1, W_2, \dots\}$ of the vertex sets of the small components that were deleted (again, if there were any and in the order of deletion), and the resulting graph when no deletion can be made anymore. Note that all of the sets in U and W are clearly pairwise disjoint.

Given the (H, b) graph deletion algorithm, and before further discussing it, we introduce some more terminology.

Definition 3.3.2. Given a connected graph H and a positive integer b, an (H, b)-stable graph is a graph for which no deletion step of the (H, b) deletion algorithm can be made. For any graph G, the (H, b)-core of G is the union of all (H, b)-stable subgraphs of G. Assuming H and b are clear from the context, we usually denote the (H, b)-core of G by G^*. For similar reasons, we also omit b, or both H and b, when talking about stability.

Remark 3.3.3. It is immediate to see that a graph is (H, b)-stable if it is $(H, b+1)$-stable, and in particular every (H, b)-stable graph is also $(H, 1)$-stable.

Claim 3.3.4. *Let $G \subseteq G_1$ be graphs such that G is (H, b)-stable, and let G_2 be a graph obtained from G_1 by applying one deletion step of the (H, b) deletion algorithm, if possible. Then $G \subseteq G_2$.*

Proof. By the stability of G, for any $v \in V(G)$ there exists an H-copy $\hat{H} \subseteq G \subseteq G_1$ containing v, therefore v is not bad with respect to G_1. For the same reason, no edge $e \in E(G)$ is bad with respect to G_1. Next, for every nonempty set $U \subseteq V(G)$ of size at most $b+1$ there exists an H-copy $\hat{H} \subseteq G \subseteq G_1$ such that $|U \cap V(\hat{H})| = 1$. It follows that every bad set $U \subseteq V(G_1)$ must be vertex disjoint from $V(G)$. Otherwise, let $U' = U \cap V(G) \neq \emptyset$. By the stability of G, U' is not bad in it. Hence there exists an H-copy $\tilde{H} \subseteq G \subseteq G_1$ such that $|V(\tilde{H}) \cap U'| = 1$. In particular it means that $|V(\tilde{H}) \cap U| = 1$ since $\tilde{H} \subseteq G$, which contradicts the assumption that U is bad in G_1. Finally, every connected component $\Gamma \subseteq G \subseteq G_1$ satisfies $|V(\Gamma)| > (b+1)(v(H)-1)$. We conclude that any deletion step applied to G_1 contains no vertices and no edges from G, and thus $G \subseteq G_2$. □

Corollary 3.3.5. *Let G, H be graphs where H is connected, and let b be a positive integer. The (H, b) deletion algorithm applied to G terminates with the (H, b)-core of G, regardless of the arbitrary choices made during the process. In particular, G^* is (H, b)-stable.*

Proof. Let $\hat{G}$ be a graph received by applying the deletion algorithm on G. Since $\hat{G}$ is stable, $\hat{G} \subseteq G^*$ trivially holds. On the other hand, by Claim 3.3.4 we get that every stable subgraph of G is contained in $\hat{G}$, and thus $G^* \subseteq \hat{G}$ as well. □

Claim 3.3.6. *Let G, H be graphs where H is connected, let b be an integer, and let U, W, G^* be the output of an arbitrary application of the (H, b) deletion algorithm on G. Let $S \subseteq V(G)$ be such that S contains at most one vertex from any set $U_i \in U$ and at most $v(H)-1$ vertices from any set $W_j \in W$. Then every H-copy contained in $G[S]$ is also contained in G^*.*

Proof. If G is H-free there is nothing to prove. Assume then that it is not, and let $\hat{H} \subseteq G$ be an H-copy not contained in G^*. We have to show that $\hat{H}$ is not contained in $G[S]$ either. By the assumption on $\hat{H}$, when considering the deletion algorithm that was applied on G, there exists an integer $k \geq 0$ such that $\hat{H} \subseteq G_k$ but $\hat{H} \not\subseteq G_{k+1}$. Furthermore, this is due to a removal of either a bad set U_i or a small component Γ_j, as all vertices and edges contained in $\hat{H}$ are obviously not bad in G_k.

In the first case, it follows that $|V(\hat{H}) \cap U_i| \geq 2$, since $V(\hat{H})$ must intersect U_i, and no H-copy in G_k contains exactly one vertex from U_i by definition of a bad set. However, S contains at most one vertex from every bad set, and so $V(\hat{H}) \not\subseteq S$. In the latter case, that is, a small component Γ_j with vertex set W_j was removed, note that $\hat{H} \subseteq \Gamma_j$ by the connectivity of H. Since S contains at most $v(H) - 1$ vertices of W_j, once again we get $V(\hat{H}) \not\subseteq S$. □

Lemma 3.3.7. *Let G, H be graphs where H is connected and b a positive integer, and let G^* be the (H, b)-core of G. When playing the $(1 : b)$ Maker-Breaker or Client-Waiter H-game, if the spoiler has a winning strategy when playing the game on $V(G^*)$, then he has a winning strategy when playing the game on $V(G)$.*

Proof. We start with the Maker-Breaker game, assume a winning strategy $\mathcal{S}_B^*$ for Breaker for the game played on $V(G^*)$, and provide him with the following winning strategy $\mathcal{S}_B$ for the game played on $V(G)$. First, Breaker runs the deletion algorithm on G and obtains an output U, W, G^*. During every round of the game, denote the last vertex claimed by Maker by v. Breaker responds according to the following cases.

(1) If $v \in V(G^*)$, Breaker plays according to $\mathcal{S}_B^*$. If there are less than b free vertices in $V(G^*)$ before his move, or if he is supposed to claim some vertices he already occupies, he completes his move arbitrarily.

(2) Otherwise, if $v \in U_i$ for some $U_i \in U$, Breaker claims all the free vertices in U_i, and, if necessary, completes his move arbitrarily.

(3) Otherwise, if $v \in W_j$ for some $W_j \in W$, Breaker claims b arbitrary free vertices of W_j, and, if there were less than b free vertices in W_j, completes his move arbitrarily.

(4) Otherwise, Breaker makes an arbitrary move.

The strategy is well defined (recall that all the sets in U and W are pairwise disjoint, and clearly they are disjoint from $V(G^*)$ as well), and Breaker can follow it. Case (2) of $\mathcal{S}_B$ ensures that Maker claims at most one vertex from every bad set, and Case (3) of $\mathcal{S}_B$ ensures that Maker claims at most $\left\lceil |W_j|/(b+1) \right\rceil \leq v(H) - 1$ vertices of any $W_j \in W$. Therefore, by Claim 3.3.6, Breaker wins the game if he prevents Maker from claiming any H-copy $\hat{H} \subseteq G^*$, and this is guaranteed by $\mathcal{S}_B^*$.

Now assume a winning strategy $\mathcal{S}_W^*$ for Waiter in the game played on $V(G^*)$. Waiter uses an analogous strategy to $\mathcal{S}_B$ in the following way. He first runs the deletion algorithm and obtains the list of all bad sets and small components which were deleted during the process. He then offers all vertices of G^* according to $\mathcal{S}_W^*$ (recall that Waiter may offer less than $b+1$ elements per move). Next, For every bad set that was removed he offers all of its vertices in one move, ensuring that Client claims only one of them. For every small component he offers arbitrary $b+1$ of its vertices repeatedly until all vertices of the component have been offered (he may offer less than $b+1$ vertices when he offers the last free vertices of the component). By this he ensures that Client claims at most $v(H) - 1$ vertices of any small component. He offers all the remaining vertices arbitrarily.

It is easy to see that Waiter can follow this strategy and win the game played on G, by Claim 3.3.6 and by the fact that the strategy $\mathcal{S}_W^*$ ensures that by the end of the game Client cannot fully claim any H-copy $\hat{H} \subseteq G^*$. $\square$

3.3.2 The structure of G^*

By Lemma 3.3.7, in order to provide the spoilers with winning strategies for their $(1:b)$ H-games played on the vertex set of a graph G, it suffices to provide them with winning strategies for the same games played on $V(G^*)$. Since H is connected by assumption, they only need to prevent their opponents from claiming an H-copy in every connected component of G^*. In fact, it suffices for the spoilers to play separately on each component, as stated in the following observation.

Observation 3.3.8. *Let H be a connected graph. In any $(1:b)$ Maker-Breaker or Client-Waiter H-game played on the vertex set of a graph G, the spoiler has a winning strategy in the game if for every connected component $\Gamma \subseteq G$, he has a winning strategy when playing the game on $V(\Gamma)$. It is immediate for Waiter, as he can avoid offering vertices from different components in the same round, and apply the winning strategy for each component separately. Similarly, since Maker claims only one vertex per move, Breaker can respond according to his winning strategy for this component, and, if necessary, complete his move arbitrarily, which can obviously cannot harm him.*

For better understanding of the possible structures of the connected components of G^*, we describe an exploration process that can be applied to every stable component. This is done step by step, by starting with one vertex and then slowly expand our view of the component by adding to it one piece of structure at a time. We describe here the general method, and then in Sections 3.4, 3.5, 3.6 and 3.10 we go into more details according to the graph H and the bias b in discussion.

We first need some terminology. Let H and Γ be connected graphs and let Γ' be a connected subgraph of Γ. The following definitions are made with respect to H, Γ and Γ'. An *internal edge* is an edge $uv \in E(\Gamma) \setminus E(\Gamma')$ such that $u, v \in V(\Gamma')$. For an H-copy $\hat{H} \subseteq \Gamma$ such that $V(\hat{H}) \not\subseteq V(\Gamma')$ let $U := V(\hat{H}) \cap V(\Gamma')$. We say that $\hat{H}$ is an *external* copy if $|U| = 1$; we say it is an *internal* copy if $|U| > 1$ and in addition $E(\hat{H}[U]) \subseteq E(\Gamma')$, that is, $\hat{H}$ contains no internal edges. We do not consider any other kind of H-copies. Whenever we say we add an internal edge e or an H-copy $\hat{H}$ (either external or internal) to Γ' we mean that we now expand our view from Γ' to Γ'', where $\Gamma'' = \Gamma' \cup \{e\}$ or $\Gamma'' = \Gamma' \cup \hat{H}$, respectively. In either case we get a new connected subgraph of Γ. With respect to the transition from Γ' to Γ'' as above, a vertex $v \in V(\Gamma'')$ is called an *existing* vertex if $v \in V(\Gamma')$ and a *new* vertex otherwise. Existing and new edges are defined analogously.

We are now ready to describe the exploration process, applicable to any connected H-stable graph Γ (by Remark 3.3.3 it suffices to consider $(H,1)$-stable connected graphs). We start by setting $\Gamma_0 = v$ for an arbitrary vertex $v \in V(\Gamma)$. Then, while $\Gamma_i \neq \Gamma$, we expand Γ_i to Γ_{i+1} by adding either an internal edge, an external H-copy or an internal H-copy, with one restriction: if Γ_i is obtained from Γ_{i-1} by adding an external H-copy, and if Γ_{i+1} is obtained from Γ_i by adding an H-copy H' (either internal or external), then $V(H') \cap (V(\Gamma_i) \setminus V(\Gamma_{i-1})) \neq \emptyset$ must hold. In other words, after the addition of an external H-copy, which added $v(H) - 1$ new vertices to the subgraph, we are looking for either an H-copy containing at least one of them, or an arbitrary internal edge. Note that this means that adding t consecutive external copies is in fact adding an H-chain of length t to the subgraph; we often use this terminology.

Since each vertex and each edge in an H-stable graph must be a part of some H-copy, it is evident we can explore every H-stable connected component by sequentially adding internal edges and H-copies. It only remains to see why any component can be explored under the aforementioned restriction. We use the following claim.

Claim 3.3.9. *Let Γ be an H-stable connected graph. Suppose that during an arbitrary exploration of Γ, the ith step was the addition of an external H-copy H_1, consisting of an existing vertex v and a set U of new vertices. Then Γ contains an H-copy $H_2 \not\subseteq \Gamma_i$ such that $V(H_2) \cap U \neq \emptyset$.*

Proof. Note that $|U| \geq 2$, and that any two vertices in U not contained in any other H-copy in Γ but H_1 form a bad pair, which contradicts the stability of Γ. It therefore suffices to show that H_1 is the only H-copy in Γ_i containing any vertices from U.

Indeed, recall that H is 2-vertex-connected by Claim 3.2.10, and let $H' \subseteq \Gamma_i$ be an H-copy satisfying $U_1 := V(H') \cap U \neq \emptyset$. Now, if $H' \neq H_1$ then $U_2 := V(H') \setminus V(H_1) \neq \emptyset$. But since there are no edges between U_1 and U_2 in Γ_i, we have that either H' is not connected or v is a cut vertex of it. In either case we get a contradiction. ☐

It is now immediate to see that the exploration process is well defined: in the terminology of Claim 3.3.9, after the addition of H_1 we can either add H_2 if possible, or add an internal edge required for H_2 otherwise. Claim 3.3.9 also leads to the following useful corollary.

Corollary 3.3.10. *In every exploration of an H-stable connected graph Γ, the last step cannot be an addition of an external H-copy. In particular, if after step i in the exploration of Γ we argue that we can only continue by adding external H-copies, then $\Gamma = \Gamma_i$.*

Our general approach in our analysis is to show that when exploring a connected component of G^*, we may consider only those in which the number of times we can add internal edges or internal H-copies is very limited, as we do not expect G to contain any dense components. We then investigate the possible stable components that can be constructed under these restrictions. We therefore use the following classification of connected components.

Definition 3.3.11. For every connected graph H and three non-negative integers q, t, s, let $X^H_{q,t,s}$ be the family of all connected graphs (not necessarily H-stable) which can be explored when adding exactly q internal edges, t external H-copies, and s internal H-copies during the exploration process. When H is clear from the context we abbreviate to $X_{q,t,s}$. Throughout this chapter we assume that q, t, s are non-negative, even if we do not state that explicitly. Note that $t > 0$ as Γ_1 is always obtained from Γ_0 by the addition of an external H-copy, regardless of H, q, or s.

Note that in general, a connected component Γ could be explored in many different ways (all of them terminating with Γ, of course). The number of times each addition type is used during the exploration may vary between different explorations, so the families $X_{q,t,s}$ are not pairwise disjoint. However, when referring to a graph $\Gamma \in X_{q,t,s}$, we always consider an exploration of Γ in such a way that each addition type is applied exactly q, t, or s times, respectively.

3.3.3 Results for clique games

In this subsection we consider K_k-games where $k \geq 3$ is some fixed integer (this is implicit for the remainder of the section). In particular, since $m_1(K_k) = k/2$, we consider the random graph $G \sim G(n, p)$ for $p = O(n^{-2/k})$. We provide general results which are later used in the analysis of several games in Sections 3.4, 3.6 and 3.10. We start by showing that any component whose exploration contains too many additions of internal edges and K_k-copies is unlikely to appear in G.

Claim 3.3.12. *Let $0 < a < 1$ (not necessarily a constant), let $G \sim G(n, an^{-2/k})$, let q, t, s be three integers satisfying $2q + (k-2)s > k$, and let $\Gamma \in X_{q,t,s}$. Then $\Pr[\Gamma \subseteq G] \le n^{-1/k}$.*

Proof. Consider an arbitrary exploration of Γ, and for every $1 \le i \le s$ let r_i denote the number of existing vertices contained in the ith internal clique added during the exploration. We have:

$$v(\Gamma) = 1 + t(k-1) + \sum_{i=1}^{s}(k - r_i)$$

and

$$e(\Gamma) = q + t\binom{k}{2} + \sum_{i=1}^{s}\left(\binom{k}{2} - \binom{r_i}{2}\right),$$

and therefore

$$\begin{aligned}
\Pr[\Gamma \subset G] &\le n^{v(\Gamma)} \cdot p^{e(\Gamma)} \\
&= a^{e(\Gamma)} n^{1+t(k-1)+\sum_{i=1}^{s}(k-r_i) - \frac{2}{k}\left[q + t\binom{k}{2} + \sum_{i=1}^{s}\left(\binom{k}{2} - \binom{r_i}{2}\right)\right]} \\
&= a^{e(\Gamma)} n^{1-\frac{1}{k}\left[2q + \sum_{i=1}^{s}\left((kr_i - k^2) + (k^2 - k) - r_i(r_i - 1)\right)\right]} \\
&= a^{e(\Gamma)} n^{1-\frac{1}{k}\left[2q + \sum_{i=1}^{s}(k-r_i)(r_i-1)\right]}.
\end{aligned}$$

By definition of an internal copy we have $2 \le r_i \le k-1$ for every i, and since $(k-x)(x-1) \ge k-2$ for every $2 \le x \le k-1$ we obtain

$$\Pr[\Gamma \subset G] \le a^{e(\Gamma)} n^{1-\frac{1}{k}[2q+(k-2)s]}, \tag{3.3.1}$$

which completes the proof by the assumptions on a, q and s. $\square$

We next provide an upper bound on the maximal length of K_k-chains we expect G to contain.

Claim 3.3.13. *Let $0 < c < 1$ be a real number, let $G \sim G(n, cn^{-2/k})$, and let Γ be the K_k-chain of length $t = -\frac{1}{\ln c}\ln n$ (note that $\ln c$ is negative). Then w.h.p. G is Γ-free.*

Proof. By observing that $v(\Gamma) = 1 + t(k-1)$ and $e(\Gamma) = t\binom{k}{2}$ we get

$$\Pr[\Gamma \subseteq G] \le n^{v(\Gamma)} p^{e(\Gamma)} = n^{1+t(k-1)-\frac{2}{k}t\binom{k}{2}} c^{t\binom{k}{2}} \le nc^{-\frac{2}{\ln c}\ln n} = \tfrac{1}{n} = o(1). \qquad \square$$

Claim 3.3.13 shows that we can limit our focus to components containing no long K_k-chains. For every three integers q, t, s, let $Y_{q,t,s}$ denote the family of all members of $X_{q,t,s}$ containing no K_k-chains of length more than $-\frac{1}{\ln c}\ln n$ as subgraphs, and let $Y_{q,s} = \bigcup_{t \ge 1} Y_{q,t,s}$.

Claim 3.3.14. $|Y_{q,s}| = O\left((\ln n)^{3q+ks}\right)$ *holds for any fixed integers q, s.*

Proof. Fix q and s and let $\Gamma \in Y_{q,s}$. In the exploration of Γ we start with a single vertex and then add elements $A_1, B_1, A_2, B_2, \ldots, A_{q+s}, B_{q+s}$ where every A_i is a K_k-chain (possibly empty), and every B_i is either an internal edge or an internal K_k-copy. Since $v(A_i) = O(\ln n)$ and $v(B_i) = O(1)$ for every i, and q, s are fixed, we get $v(\Gamma) = O(\ln n)$.

Now let us bound from above the number of different graphs we can obtain in this way. For every A_i there are $O(\ln n)$ options to choose the length of the chain, and recall that there is only one type of K_k-chain of each length (up to isomorphism). For each of the q internal edges there are at most $\binom{v(\Gamma)}{2} = O(\ln^2 n)$ options to choose its endpoints. For each of the s internal K_k-copies there are at most $\sum_{r=2}^{k-1} \binom{v(\Gamma)}{r} = O(\ln^{k-1} n)$ options to choose the existing vertices it contains. Multiplying all these factors we get the desired result. $\square$

Using the previous three claims we obtain our main result for clique games, which we use extensively in the following sections.

Claim 3.3.15. *Let $0 < c < 1$ be a real number and let $G \sim G(n, cn^{-2/k})$. The following holds w.h.p.: all families $X_{q,t,s}$ for which at least one of their members appears in G satisfy $2q + (k-2)s \leq k$.*

Proof. By Claim 3.3.13 we may assume that G contains no K_k-chains of length more than $-\frac{1}{\ln c} \ln n$, thus we only consider the families $Y_{q,t,s}$ and $Y_{q,s}$. Let q, t, s be three integers such that $2q+(k-2)s > k$, let $\Gamma \in Y_{q,t,s}$, and consider an arbitrary exploration of Γ. Denote by q_j and s_j the number of internal edges and internal copies added during the first j steps of the process, respectively. Let i be the maximal integer such that after i steps in the process $2q_i + (k-2)s_i \leq k$ holds. It follows immediately that $q_i \leq k/2$ and $s_i \leq k/(k-2) \leq 3$. Let

$$Y_k = \bigcup_{\substack{q \leq k, s \leq 4 \text{ s.t.} \\ 2q+(k-2)s > k}} Y_{q,s}.$$

It follows that $\Gamma_{i+1} \in Y_k$, and so in order to prove the claim it suffices to show that w.h.p. G is Y_k-free. By Claim 3.3.12 we have that $\Pr[\Gamma \subseteq G] \leq n^{-\frac{1}{k}}$ for every $\Gamma \in Y_k$. For any $q \leq k$ and $s \leq 4$ we get by Claim 3.3.14 that $|Y_{q,s}| = O\left((\ln n)^{7k}\right)$, and thus $|Y_k| = O\left((\ln n)^{7k}\right)$ for any fixed k. A simple union bound now yields

$$\Pr[G \text{ is not } Y_k\text{-free}] \leq \sum_{\Gamma \in Y_k} \Pr[\Gamma \subseteq G] \leq |Y_k| n^{-\frac{1}{k}} = o(1). \qquad \square$$

In the light of Claim 3.3.15, we show the strong correlation between dangerous edges and stable graphs, when considering only the probable subgraphs of the random graph.

Lemma 3.3.16. *Let q, t, s be integers satisfying $2q + (k-2)s \leq k$, and let $\Gamma \in X_{q,t,s}$ be a $(K_k, 1)$-stable connected graph. Then either Γ contains at least one dangerous edge, or $k = 3$ and Γ is a K_3-cycle of length at least four.*

Proof. Assume that Γ contains no dangerous edges and consider the last addition in an arbitrary exploration of Γ. By Corollary 3.3.10 it is not the addition of an external K_k-copy. However, it cannot be an addition of an internal edge e either. Indeed, by stability e must be a part of some K_k-copy $\hat{H}$. By the assumption that Γ contains no dangerous edges, $\hat{H}$ cannot use any edges from other K_k-copies, and therefore all the edges of $\hat{H}$ must have been added to Γ as internal edges. But that would imply that $q \geq \binom{k}{2} > k/2$, a contradiction. Hence, the last step of the exploration must be an addition of an internal K_k-copy $\hat{H}$. Since $\hat{H}$ contains at least two existing vertices by definition, and since Γ contains no dangerous edges, it follows that at least one internal edge uv was added to Γ prior to the addition of $\hat{H}$. We therefore get $q \geq 1$ and $s \geq 1$, which by the restriction $2q + (k-2)s \leq k$ leads to $q = s = 1$. That is, every addition during the exploration is of an external copy, except for uv and $\hat{H}$.

Note that no external K_k-copies can be added between the additions of uv and $\hat{H}$. Otherwise, $\hat{H}$ must contain a vertex w from the last external copy, and also use the edge uv. However, as $\hat{H}$ is a clique, the edges uw and vw also need to be added as internal edges, which is impossible since $q = 1$. We can now describe the entire exploration process: it starts with a K_k-chain, then an internal edge uv is added, then an internal K_k-copy $\hat{H}$ is added, containing u, v and $k-2$ new vertices, and then the exploration terminates.

If $k \geq 4$, the internal K_k-copy $\hat{H}$ contains $k-2 \geq 2$ vertices which only belong to $\hat{H}$. Since any two of these new vertices form a bad pair, Γ is not stable, which leads to the conclusion that there are no stable graphs with no dangerous edges in this case.

For $k = 3$, note that just before the addition of uv both the first and last triangles in the triangle chain contain a bad pair. By stability, the set $\{u, v\}$ contains exactly one vertex from each of these pairs. The internal triangle wuv that is added in the last step (where w is a new vertex) completes the creation of a K_3-cycle. It is immediate to see that a K_3-cycle contains no dangerous edges if and only if it is of length at least four. □

Results for triangle games

We now focus on the case $k = 3$. Here the term $2q + (k-2)s$ translates to $2q + s$. We first observe that this term depends on Γ alone and not on the way we explore it.

Observation 3.3.17. *Let Γ be a connected graph and let $q_1, t_1, s_1, q_2, t_2, s_2$ be integers such that Γ belongs to both $X^{K_3}_{q_1,t_1,s_1}$ and $X^{K_3}_{q_2,t_2,s_2}$. Then $2q_1 + s_1 = 2q_2 + s_2$.*

Proof. Since every internal triangle contains exactly one new vertex and two new edges, we get that if $\Gamma \in X^{K_3}_{q_i,t_i,s_i}$ then $v(\Gamma) = 1 + 2t_i + s_i$ and $e(\Gamma) = q_i + 3t_i + 2s_i$. The first equation yields $t_1 - t_2 = (s_2 - s_1)/2$ and the second one yields $t_1 - t_2 = (2s_2 - 2s_1 + q_2 - q_1)/3$. Comparing the right hand side of the last two equations, and rearranging, we get the desired result. □

We often perform case analysis on the possible connected graphs under some restriction on the value of $2q + s$. Observation 3.3.17 leads to the following, extremely useful, corollary.

Corollary 3.3.18. *When exploring a connected component Γ under no other restrictions but the value of $2q + s$, we may choose to explore Γ however we like without the risk of overlooking some possible components. In particular, if we assume that Γ contains at least one dangerous edge we can start the exploration of Γ with two triangles sharing an edge, xyz_1 and xyz_2 (that is, we start with x, add xyz_1 as an external triangle, and then add xyz_2 as an internal triangle).*

We next use Corollary 3.3.18 to characterize the possible stable graphs under the restriction $2q + s \leq 2$.

Claim 3.3.19. *Let q, t, s be integers satisfying $2q + s \leq 2$ and let $\Gamma \in X_{q,t,s}$ be a $(K_3, 1)$-stable graph. Then Γ is either a Tr or a DD_t.*

Proof. All the arguments in this proof are done with respect to graph isomorphism. First observe that if Γ does not contain dangerous edges then it is a K_3-cycle by Lemma 3.3.16. However, the only way to explore a K_3-cycle is to start with a K_3-chain, then add an internal edge e between the first and last triangles in the chain, and terminate with an internal triangle containing e. That would imply $q = s = 1$, in contradiction to the assumption.

It follows that Γ must contain a dangerous edge, thus by Corollary 3.3.18 we can start the exploration of Γ by looking at two triangles sharing an edge, xyz_1 and xyz_2. We have $s \geq 1$ by the fact that the triangle xyz_2 is internal and hence by the assumption we get $q = 0$ and $s \leq 2$. By Corollary 3.3.10, the remainder of the exploration must be a series of length $t-1$ of additions of external triangles, followed by an addition of an internal triangle T, and nothing more. We distinguish between two cases, keeping in mind that $\{x, y\}$ is a bad pair in Γ_2.

If $t = 1$, then T must contain an existing edge other than xy, thus $T = wxz_1$ (where w is a new vertex), and Γ is a Tr.

If $t > 1$, denote the vertices of the last external triangle added by a, b, c, where a is the existing vertex and b, c are the new ones. Since $\{b, c\}$ is a bad pair, T must contain an existing edge incident to exactly one of them by stability, and so $T = abw$, where w is a new vertex. Furthermore, the triangle chain must intersect $V(\Gamma_2)$ at x, otherwise $\{x, y\}$ remains a bad pair in Γ. Thus Γ is a DD_t. □

We conclude this section with the following claim, which will be crucial for the analysis of unbiased triangle games involving random graph processes.

Claim 3.3.20. *Let $\mathcal{F}$ be the family of all graphs on less than 25 vertices with density larger than $10/7$, and let G be an $\mathcal{F}$-free K_3-stable graph. Then every connected component of G is either a Tr or a DD.*

Proof. Let Γ be a component in G and let $\Gamma \in X_{q,t,s}$ for some integers q, t, s. Observe that $v(\Gamma) = 1 + 2t + s$ and $e(\Gamma) = q + 3t + 2s$, and that $d(\Gamma) = e(\Gamma)/v(\Gamma) \leq 10/7$ if and only if $7q + t + 4s \leq 10$. It follows that when exploring Γ there can be at most ten addition steps as after eleven steps we get a component $\Gamma_{11} \subseteq \Gamma$ on at most 23 vertices and of density larger than $10/7$ in contradiction to the assumption. It means that **every** connected component of G has at most 21 vertices and therefore density at most $10/7$. Thus, if $\Gamma \in X_{q,t,s}$ is a component of G then $7q + t + 4s \leq 10$ must hold, and in particular $2q + s \leq 2$ must hold as well (since either $q = 1$ and $s = 0$ or $q = 0$ and $s \leq 2$).

We can therefore apply Claim 3.3.19 and deduce that every component in G is either a Tr or a DD_t. In the latter case we have $s = 2$ and $t \geq 2$, and by the restriction $7q + t + 4s \leq 10$ the DD_t must be of length 2, i.e., a DD. □

3.4 Maker-Breaker clique games

In this section we prove the 0-statement in Theorem 3.1.4 whenever $H' = K_k$ and $k \geq 4$ (recall that H' is a subgraph of H of maximal 1-density). If Breaker prevents Maker from claiming a K_k-copy then Maker cannot occupy any H-copy, hence it is enough to prove Breaker's side in the K_k-game. By bias monotonicity, it is enough to prove Breaker's side for $b = 1$. Now let $c < 1$ and let $G \sim G(n, cn^{-2/k})$. By Lemma 3.3.7 it suffices to show that w.h.p. Breaker can win the game played on the vertex set of G^*, the $(K_k, 1)$-core of G. Clearly, is suffices to show that w.h.p. G^* is empty. Recall that Claim 3.3.15 states that w.h.p. G (and therefore G^*) contains only members of families $X_{q,t,s}$ for which $2q + (k-2)s \leq k$. We are therefore done by the following lemma.

Lemma 3.4.1. *Let $k \geq 4$ and let Γ be a non-empty $(K_k, 1)$-stable graph. Then there exist integers q, t, s such that $\Gamma \in X_{q,t,s}$ and $2q + (k-2)s > k$.*

Proof. Assume for contradiction that in every exploration of Γ we have $2q+(k-2)s \leq k$. Lemma 3.3.16 implies that Γ contains a dangerous edge. Let H_1, H_2 be two K_k-copies in Γ sharing at least one edge and let $v \in V(H_1)$. We start the exploration of Γ with $\Gamma_0 = \{v\}$, $\Gamma_1 = H_1$, and $\Gamma_2 = H_1 \cup H_2$.

Since we add H_2 as an internal copy we have $s \geq 1$. Note that $v(\Gamma_2) \leq 2k-2$, i.e. Γ_2 is a small component, and so $\Gamma_2 \neq \Gamma$. By Corollary 3.3.10 the exploration of Γ cannot end with the addition of an external copy. Hence, by the assumption $2q+(k-2)s \leq k$, we only have to consider the following two cases.

1. $q = s = 1$.

2. $q = 0$, $s = 2$ and $k = 4$.

That is, after the addition of H_2 we continue with the addition of a K_k-chain of length $t-1$ (recall that H_1 was added as an external copy), followed by the addition of either an internal edge e or an internal copy H_3 as the last step of the exploration, where the latter case is only possible if $k = 4$. We analyze each case separately, showing that none of them is possible, thus completing the proof.

Case 1: $q = s = 1$.
In this case we have $t > 1$ as otherwise $V(\Gamma) = V(\Gamma_2)$ and Γ is a small component. Let $\hat{H}$ be the last external K_k-copy added in the exploration. Since it is a clique, the edge e (which completes the exploration of Γ) contains at most one of its vertices. It follows that there is a set of $k-2 \geq 2$ vertices of $\hat{H}$ not contained in any other K_k-copy in Γ, as they each have degree $k-1$. Each pair in this set forms a bad pair, in contradiction to the stability of Γ.

Case 2: $q = 0$, $s = 2$, $k = 4$.
Here we have $H = K_4$. Observe first that whenever an H-copy H' is added during the exploration, no new H-copies appear in the explored component other than H'. Indeed, every new vertex of H' has degree $k-1$ and so it belongs only to H', and all new edges are incident to new vertices, so none of them can be a part of any other H-copy as well. It follows that H_3 contains exactly one new vertex, as two new vertices would form a bad pair. Furthermore, the existing three vertices must form a triangle prior to the addition of H_3.

Now, assume that $t > 1$ and let $\hat{H}$ be the last external H-copy added in the exploration. Since H_3 must contain a new vertex of $\hat{H}$, and two of its neighbors, H_3 in fact contains (at least) two of the new vertices of $\hat{H}$. These two vertices form a bad pair in Γ as they both belong to the same set of H-copies (namely, $\hat{H}$ and H_3), in contradiction the stability of Γ.

We thus have $t = 1$, and $\Gamma = \Gamma_3 = H_1 \cup H_2 \cup H_3$. Recall that $v(\Gamma_2) \leq 6$. Since H_3 adds one new vertex to Γ, and since $v(\Gamma) > 6$ or otherwise it would be a small component, we conclude that $v(\Gamma_2) = 6$, and we can write $V(H_1) = \{v_1, v_2, v_3, v_4\}$ and $V(H_2) = \{v_3, v_4, v_5, v_6\}$. Note that in Γ_2 there is no edge between the pairs $\{v_1, v_2\}$ and $\{v_5, v_6\}$, so H_3 contains vertices from only one of these pairs. However, both pairs are bad pairs in Γ_2, so at least one of them remains a bad pair in Γ, which once again contradicts its stability. $\square$

We conclude this section with a short discussion about the two constants c and C appearing in Theorem 3.1.4 for the case considered above (namely, $H' = K_k$ and $k \geq 4$). We later refer to this discussion in Section 3.11. The proof of the 0-statement works for any $c < 1$. It turns out that for any $C > 1$ the 1-statement holds for any $k \geq k_0$, where k_0 is a constant determined by b and C.

More specifically, recall that the proof of the 1-statement of Theorem 3.1.4 is based on the argument that for some constant $C = C(b,k)$, if $p \geq Cn^{-2/k}$, then w.h.p. in $G \sim G(n,p)$ every $\frac{1}{b+1}$-fraction of $V(G)$ induces a K_k-copy in G and thus w.h.p. Maker wins the $(1:b)$ K_k-game no matter how he plays in this case. Let us examine this constant C a bit more carefully. Although not very complicated, we omit most of the technical details below. In the proof of Theorem 3.1.3 the authors of [38] used the following theorem, which was originally stated in a more general form.

Theorem 3.4.2 ([33])**.** *For every strictly balanced graph H there exists a constant $c(H)$ such that for $G \sim G(n,p)$ the probability that G is H-free is at most $e^{-c(H)\mu(H)}$, where $\mu(H)$ is the expected number of H-copies in G.*

One can verify that if H is strictly 1-balanced and $p = \Theta(n^{-1/m_1(H)})$, then the constant $c(H)$ from Theorem 3.4.2 can be set arbitrarily close to 1 from below (this follows from Janson's inequality). Using this fact and conducting the calculations in the proof of Theorem 3.1.3 more carefully, one obtains that for $p \geq Cn^{-2/k}$ (where C is undetermined yet), the probability that there exists a K_k-free subgraph of G induced by a $\frac{1}{b+1}$-fraction of its vertices is at most $\exp\left(n\left(1-(bk)^{-k}C^{\binom{k}{2}}\right)\right)$, i.e. this probability is exponentially small whenever $C > (bk)^{\frac{2}{k-1}}$. For any fixed b, the right hand side of the last inequality is a monotone decreasing function of k, tending to 1 as k tends to infinity. It follows that indeed, the 1-statement holds for any fixed $C > 1$ and $b \geq 1$, if k is large enough.

3.5 Maker-Breaker cycle games

At the beginning of this section we follow the ideas of Section 3.3.3 while paying attention to the computational and structural differences between the clique game and the cycle game, showing eventually that the stable components which are likely to appear in the random graph are quite limited in their structure. The analysis is much more delicate in this case though. From now on, fix an integer $k \geq 4$, let $c < k^{-1/k}$ be a constant, and consider the random graph $G \sim G(n,p)$ for $p = cn^{-1/m_1(C_k)} = cn^{-(k-1)/k}$.

We start with an equivalent of Claim 3.3.13, showing that w.h.p. G does not contain any C_k-chain of length $d \ln n$ for $d = -2/\ln\left(kc^k\right)$ (note that d is a positive constant as $kc^k < 1$). First observe that for any given t there are less than k^t different C_k-chains of length t (up to isomorphism). Indeed, given t, if we denote by d_i the distance on the ith cycle between the two vertices it shares with the previous and next cycles in the chain, then the structure of the chain is determined by $d_2, \ldots, d_{t-1}$. Since there are $\lfloor k/2 \rfloor$ options for each d_i the upper bound on the number of different chains is established. Now, if Γ is a C_k-chain of length t than we have $v(\Gamma) = 1 + t(k-1)$ and $e(\Gamma) = kt$. Therefore, if $\mathcal{T}$ denotes the family of all C_k-chains of length $t = d \ln n$ we get

$$\begin{aligned}\Pr[G \text{ is not } \mathcal{T}\text{-free}] &\leq \sum_{\Gamma \in \mathcal{T}} \Pr[\Gamma \subseteq G] \leq \sum_{\Gamma \in \mathcal{T}} n^{v(\Gamma)} p^{e(\Gamma)} \\ &\leq k^t n^{1+t(k-1)} \left(cn^{-(k-1)/k}\right)^{kt} = n\left(kc^k\right)^{d \ln n} = 1/n = o(1).\end{aligned}$$

Next, for fixed q and s we wish to bound the size of the families $X_{q,t,s}$ containing no C_k-chains of length $d \ln n$, which are once again denoted by $Y_{q,t,s}$. This is done similarly to Claim 3.3.14, but with a few differences. First, as already argued, determining the length of each of the C_k-chains in a component $\Gamma \in Y_{q,t,s}$ is not enough and we have to account for the number of all different choices for their structures. For a given t (that is, the sum of lengths of all C_k chains in the exploration), by using basically the same argument as above, we can upper bound this number by k^t. Consequently, we bound separately the size of each family $Y_{q,t,s}$, and not the size of their union over all values of t. However, while q and s are fixed, we allow t to grow to infinity with n. Lastly, when choosing existing vertices for an internal cycle we also have to choose their location on the cycle. This results in a factor of at most $(k!)^s$. For all other considerations, almost identical calculations to those in Claim 3.3.14 yield the factor $O\left((\ln n)^{3q+ks}\right)$. All in all we get that

$$|Y_{q,t,s}| = O\left(k^t (\ln n)^{3q+ks}\right) \tag{3.5.1}$$

holds for any two fixed integers q, s and any integer $t := t(n)$.

When considering the cycle game, knowing that a component belongs to $X_{q,t,s}$ is not enough for us and we need more information. For every non-negative integer s and for every s-tuple $\overrightarrow{s} = ((v_1, e_1), \dots, (v_s, e_s))$ (including the empty tuple if $s = 0$), let $X_{q,t,\overrightarrow{s}}$ be the subset of all graphs in $X_{q,t,s}$ with the additional condition that the ith internal C_k-copy added during the exploration process contains exactly v_i existing vertices and e_i existing edges. Define $Y_{q,t,\overrightarrow{s}}$ in a similar fashion. Mind the abuse of notation here, as the integer s denotes the length of $\overrightarrow{s}$. This is done in order to reduce the number of different variables used. Whenever discussing such an s-tuple, we assume it is "legal". In particular, for every $i \in [s]$ we have $1 < v_i < k$, since it represents an internal copy, and $v_i > e_i$, since the existing vertices and edges of each internal C_k-copy always form vertex-disjoint paths.

Now we wish to prove an analogue of Claim 3.3.15 that characterizes the families $X_{q,t,\overrightarrow{s}}$ which are likely to appear in G. Once again, we do this by showing that there exist a family of relatively small size, such that G w.h.p. does not contain any of its members, but each graph obtained via an exploration process which does not meet some condition on q and $\overrightarrow{s}$ must contain one of its members as a subgraph. In order to formulate this condition we define

$$f(\overrightarrow{s}) := \sum_{i=1}^{s} (k(v_i - e_i - 1) + e_i).$$

Note that we cannot have $v_i = 1,\ e_i = 0$ for the same $i \in [s]$, as these values represent the addition of an *external* copy, and thus for any given $\overrightarrow{s}$, every summand in $f(\overrightarrow{s})$ is a positive integer.

Claim 3.5.1. *The following holds w.h.p.: all families $X_{q,t,\overrightarrow{s}}$ for which at least one of their members appears in G satisfy $(k-1)q + f(\overrightarrow{s}) \le k$.*

Proof. First, we consider only the families $Y_{q,t,\overrightarrow{s}}$, since w.h.p. there exist no long C_k-chains in G. Now, for every non-negative integer s and a corresponding s-tuple $\overrightarrow{s}$, and for every $\Gamma \in Y_{q,t,\overrightarrow{s}}$, we have $v(\Gamma) = 1 + t(k-1) + \sum_{i=1}^{s}(k - v_i)$ and $e(\Gamma) = tk + q + \sum_{i=1}^{s}(k - e_i)$. Hence,

$$\begin{aligned}
\Pr[\Gamma \subseteq G] &\le n^{v(\Gamma)} p^{e(\Gamma)} \\
&= c^{e(\Gamma)} n^{1+t(k-1)+\sum_{i=1}^{s}(k-v_i) - \frac{k-1}{k}\left(tk+q+\sum_{i=1}^{s} k-e_i\right)} \\
&\le c^{kt} n^{1-\frac{1}{k}\left[(k-1)q+\sum_{i=1}^{s}((k-1)(k-e_i)-k(k-v_i))\right]} \\
&= c^{kt} n^{1-\frac{1}{k}\left[(k-1)q+f(\overrightarrow{s})\right]}.
\end{aligned} \tag{3.5.2}$$

Let q, t and $\overrightarrow{s} = ((v_1, e_1), \dots, (v_s, e_s))$ such that $(k-1)q + f(\overrightarrow{s}) > k$, and let $\Gamma \in Y_{q,t,\overrightarrow{s}}$. For every $i \in [s]$ let $\overrightarrow{s_i} = ((v_1, e_1), \dots, (v_i, e_i))$ be the ith prefix of $\overrightarrow{s}$. Consider an arbitrary exploration of Γ, and let q_j and γ_j be the number of internal edges and internal copies added during the first j steps of the exploration, respectively. Let i be the maximal integer such that $(k-1)q_i + f(\overrightarrow{s_{\gamma_i}}) \le k$. Such an i exists since the first step in any exploration is the addition of an external copy. Clearly we have $q_i \le 1$

and $f(\overrightarrow{s_{\gamma_i}}) \leq k$. Recalling that every summand in $f(\overrightarrow{s})$ is a positive integer, the latter inequality implies $\gamma_i \leq k$. By these observations and the definition of i, we have $(k-1)q_{i+1} + f(\overrightarrow{s_{\gamma_{i+1}}}) > k$, while $q_{i+1} \leq 2$ and $\gamma_{i+1} \leq k+1$. Let

$$Y_{k,t} = \bigcup_{\substack{q,\overrightarrow{s} \text{ s.t. } q \leq 2, s \leq k+1 \\ \text{and } (k-1)q + f(\overrightarrow{s}) > k}} Y_{q,t,\overrightarrow{s}}$$

and

$$Y_k = \bigcup_{t \geq 1} Y_{k,t}.$$

Then $\Gamma_{i+1} \in Y_k$, and so it suffices to show that w.h.p. G is Y_k-free. Clearly, for every t we have

$$Y_{k,t} \subseteq \bigcup_{q \leq 2,\ s \leq k+1} Y_{q,t,s},$$

and so $|Y_{k,t}| \leq k^t (\ln n)^{k^3}$ by (3.5.1). Note also that $Y_{k,t} = \emptyset$ for every $t > (q+s)d \ln n$ by the restriction on the maximal length of a C_k-chain. By (3.5.2) we have that $\Pr[\Gamma \subseteq G] \leq c^{kt} n^{-1/k}$ for every $\Gamma \in Y_{k,t}$. Applying the union bound and using the fact that $kc^k < 1$ we obtain

$$\Pr[G \text{ is not } Y_k\text{-free}] \leq \sum_{t \geq 1} \sum_{\Gamma \in Y_{k,t}} \Pr[\Gamma \subseteq G] \leq \sum_{t=1}^{(q+s)d \ln n} (\ln n)^{k^3} k^t c^{kt} n^{-\frac{1}{k}} \leq (\ln n)^{k^4} n^{-\frac{1}{k}} = o(1),$$

which completes the proof. $\square$

In the light of Claim 3.5.1 we define the following.

Definition 3.5.2. A nonempty connected component Γ is called *feasible* if it is stable, and any $q, t, \overrightarrow{s}$ for which $\Gamma \in X_{q,t,\overrightarrow{s}}$ satisfy $(k-1)q + f(\overrightarrow{s}) \leq k$.

Now, in a similar fashion to that of the proof given in Section 3.4, we focus on the unbiased C_k-game for $k \geq 4$ played on a feasible component, and from this we eventually deduce the more general result. Our main and most basic tool is the fact that stable components cannot contain any bad pairs. In our analysis we encounter two main types of bad pairs, the first is defined as follows.

Definition 3.5.3. Let G be a graph. Two vertices $u, v \in V(G)$ are called *evil twins* if $uv \in E(G)$ and $d_G(u) = d_G(v) = 2$.

It is easy to see that if u, v are evil twins then they indeed form a bad pair: any C_k-copy containing one of these vertices must use the two edges incident to it, and specifically the edge uv, implying that no C_k-copy can contain exactly one of u, v. Note that evil twins remain such as long as no edge incident to one of them is added to the graph, which is not the case for bad pairs in general. Suppose that $\Gamma' \subseteq \Gamma''$ are two graphs obtained during an exploration of a feasible component Γ. If Γ' contains a pair of evil twins, and the two vertices remain evil twins in Γ'', we say that the pair *survives* in Γ''. The other type of bad pairs we wish to introduce is the one described in the following claim.

Claim 3.5.4. *Let G be a graph and let $x, y \in V(G)$ be two vertices of degree 3, such that G contains two distinct (non-empty) paths P_1, P_2 between x and y, where every internal vertex in either path is of degree 2. Then $\{x, y\}$ is a bad pair.*

Proof. Every C_k copy containing either x or y contains two of the three edges incident to that vertex, and so must contain at least one edge belonging to either P_1 or P_2. Since all internal vertices of that path are of degree 2, the C_k-copy must contain the entire path, and in particular both its endpoints x and y. It means that no C_k-copy contains exactly one of $\{x, y\}$, which is what we had to show. □

We now begin with a series of claims and corollaries, gradually revealing more and more information about the possible structures of feasible components and the restrictions applied to them. First we wish to make the restriction $(k-1)q + f(\overrightarrow{s}) \leq k$ a bit more explicit.

Claim 3.5.5. *Let Γ be a feasible component. In any exploration of Γ in which $s \geq 1$, exactly one of the following holds.*

(i) $q = 0,\ s = 1,\ v_1 = 2,\ e_1 = 0.$

(ii) $q = 0,\ \sum_{i=1}^{s} e_i \leq k$, *and* $e_i = v_i - 1 \geq 1$ *for every* $i \in [s]$.

(iii) $q = 1,\ s = 1,\ v_1 = 2,\ e_1 = 1.$

Proof. Recall that $f(\overrightarrow{s}) = \sum_{i=1}^{s}(k(v_i - e_i - 1) + e_i)$. If $v_{i_0} > e_{i_0} + 1$ for some $i_0 \in [s]$ then $f(\overrightarrow{s}) \geq k(v_{i_0} - e_{i_0} - 1) \geq k$. Since Γ is feasible, it follows that in this case $q = 0$ and $f(\overrightarrow{s}) = k$, and thus by the positivity of the summands of $f(\overrightarrow{s})$ we conclude that $s = 1,\ v_1 = 2,\ e_1 = 0$, which is exactly Option (i).

Assume now that $v_i = e_i + 1$ for every $i \in [s]$, and therefore $f(\overrightarrow{s}) = \sum_{i=1}^{s} e_i$, and recall that $e_i \geq 1$ for every $i \in [s]$ in this case. Since $(k-1)q + \sum_{i=1}^{s} e_i \leq k$ holds there are only two possibilities. If $q = 0$ then the restriction $\sum_{i=1}^{s} e_i \leq k$ remains as is, and we got Option (ii). If $q = 1$ then $\sum_{i=1}^{s} e_i \leq 1$, which can only happen if $s = 1,\ v_1 = 2,\ e_1 = 1$, that is, Option (iii). □

Our next step is to show that not surprisingly, dangerous edges are crucial for stability.

Claim 3.5.6. *Any feasible component contains at least one dangerous edge.*

Proof. Assume for contradiction that Γ is a feasible component containing no dangerous edges. Recall that by Corollary 3.3.10 the last step of the exploration cannot be an addition of an external C_k-copy. However, it cannot be an addition of an internal edge e either. Indeed, by stability e must be a part of some C_k-copy, which contains an existing edge e'. Since $q \leq 1$ by the feasibility of Γ, the edge e' must have been added to Γ as a new edge with some previous C_k-copy. This makes e' a dangerous edge, a contradiction.

The exploration therefore terminates with the addition of an internal C_k-copy H', and in particular we get $s \geq 1$. Moreover, when adding an internal C_k-copy, every existing edge that this copy contains must have been added previously as an internal edge, or otherwise we get a dangerous edge. Hence, if $e_i \geq 1$ for some $i \in [s]$, then $q \geq 1$ must hold as well. Claim 3.5.5 therefore implies that $s = 1$ and $v_1 = 2$, since Option (ii) of the claim leads to a contradiction.

It follows that H' which is added as the last step of the exploration contains $k - 2$ new vertices. Let u, v be two of these vertices, and note that they both have degree 2 in Γ. The two edges incident to each of them belong to H' and to no other H-copy in Γ by assumption, implying that the same holds for u and v as well, which makes them a bad pair in contradiction. □

As any feasible component Γ contains a dangerous edge, there exist two C_k-copies $H_1, H_2 \subseteq \Gamma$ sharing at least one edge. It is extremely beneficial for the analysis to start the exploration of Γ with these two copies. Formally, let Γ be a feasible component. Recall that in any exploration of Γ, the first addition to the initial vertex is an external C_k-copy H_1. A *greedy* exploration of Γ is any exploration in which the second step is the addition of an internal copy H_2 which contains at least one existing edge.

Claim 3.5.7. *There exists a greedy exploration for any feasible component.*

Proof. Let Γ be a feasible component. Since Γ contains a dangerous edge by Claim 3.5.6, we can start the exploration with two C_k-copies containing this edge. It only remains to verify that no internal edges need to be added prior to H_2. Assume otherwise, and note that in this case we must have $q = s = 1$ by Claim 3.5.5. That is, one internal edge is added to H_1, and immediately afterwards H_2 is added. But at this point only external copies may be added, implying that $\Gamma_3 = \Gamma$, which is impossible since Γ_3 is a small component. □

From now on we only consider greedy explorations. Since the addition of H_2 ensures that $s \geq 1$ and $e_1 \geq 1$, Option (i) of Claim 3.5.5 is not possible. It turns out that the same holds for Option (iii).

Claim 3.5.8. *Let Γ be a feasible component. Then $q = 0$ holds for any greedy exploration of Γ.*

Proof. Assume otherwise for contradiction. Then $q = 1, s = 1, v_1 = 2, e_1 = 1$ must hold by Claim 3.5.5, and $t > 1$ must hold as otherwise Γ would be a small component. By definition of greedy explorations, and since no exploration can end with the addition of an external copy, we conclude that any greedy exploration of Γ with $q > 0$ must go as follows. First the external copy H_1 is added, then the internal copy H_2 is added, sharing exactly one edge and its two endpoints with H_1. Then a non-empty C_k-chain $\mathcal{C}$ is added, ending with an external copy H_3, and finally an internal edge e is added. Recall that $\Gamma_2 = H_1 \cup H_2$, and let Γ' be the explored graph just before the addition of e.

For $i = 1, 2$, let $V_i \subseteq V(H_i)$ be the set of $k - 2$ vertices of degree 2 in Γ_2, and let z and V_3 be the existing vertex and the set of new vertices added by H_3, respectively. Note that both V_1 and V_2 contain evil twins in Γ_2, and since the intersection of $\mathcal{C}$ and Γ_2 is a single vertex w (satisfying $w = z$ in case $t = 2$), at least one of these sets — assume WLOG it is V_1 — still contains evil twins in Γ'. Since V_3 also contains evil twins in Γ', and since Γ cannot contain any evil twins, it follows that $w \in V_2$, and that $e = uv$ for some $u \in V_1$ and $v \in V_3$. Moreover, we may assume that $k = 4$ and that v is the non-neighbor of z in H_3, as otherwise V_3 still contains evil twins in Γ.

But now we have a contradiction: by stability uv must be part of a C_4-copy, meaning that in Γ' there exists a path of length three between u and v. However, any such path must contain z, which is of distance at least two from both u and v (since there are no edges between V_1 and V_2, and therefore no edges between u and any vertex of $\mathcal{C}$). □

It remains to analyze greedy explorations in which $q = 0$, $\sum_{i=1}^{s} e_i \leq k$, and $e_i = v_i - 1$ for every $i \in [s]$. The fact that $q = 0$ means that each step of the exploration is the addition of a C_k-copy, either external or internal. The last restriction allows us to describe the addition of internal copies very precisely. Suppose that the ith internal C_k-copy H' is added during the exploration to $\Gamma' \subseteq \Gamma$. This must be done by adding a path of $k - e_i \geq 2$ new edges (recall that $e_i < v_i < k$), where all internal vertices of the path are new vertices, and its two endpoints x, y are existing vertices such that Γ' contains a path of length e_i between them. We say that H' is *attached* to x and y. As an immediate corollary of Claim 3.5.4, we get the following restriction on the vertices the last added copy is attached to.

Corollary 3.5.9. *Let Γ be a feasible component, and let H' and H'' be the penultimate and last C_k-copies added in a greedy exploration of it, respectively. Then H'' is attached to at most one vertex of H'.*

Proof. Assume otherwise, and let x, y be the two vertices H'' is attached to. Then $d_\Gamma(x) = d_\Gamma(y) = 3$, and there exists two paths $P_1, P_2 \subseteq \Gamma$ between x and y, the one added by H' and the one added by H'', and every internal vertex in either path is of degree 2. Thus, the conditions for Claim 3.5.4 hold, and $\{x, y\}$ is a bad pair, a contradiction. □

By the general restrictions of the exploration process, we have that in any greedy exploration the first step is the addition of an external copy, while the second step and the last step (which must be two different steps because otherwise $\Gamma = \Gamma_2$ is a small component) are the additions of internal copies, implying in particular $s > 1$. Given the above characterization of internal copy additions, we immediately deduce that $e_s = k - 2$, that is, only one new vertex is added in the last step of the exploration, as otherwise this step will add evil twins to the graph (in any step of the exploration, each pair of adjacent new vertices is a pair of evil twins). It follows that $\sum_{i=1}^{s-1} e_i \leq 2$, leaving no much options for $\overrightarrow{s}$.

Corollary 3.5.10. *In any greedy exploration of a feasible component exactly one of the following holds.*

(a) $s = 2,\ e_1 = 2,\ e_2 = k - 2,\ t > 1.$

(b) $s = 2,\ e_1 = 1,\ e_2 = k - 2.$

(c) $s = 3,\ e_1 = 1,\ e_2 = 1,\ e_3 = k - 2.$

Proof. We already know that only Option (ii) of Claim 3.5.5 is possible, and that $e_s = k - 2$. Thus it only remains to explain why $t > 1$ in Case (a). If not, then the exploration lasts only three steps, where the second step adds $k - 3$ new vertices and the third step adds just one, resulting in a small component in contradiction. □

The significant next claim narrows down our analysis to the case $k = 4$.

Claim 3.5.11. *There exist no feasible components for $k \geq 5$.*

Proof. Let $k \geq 5$, assume for contradiction that Γ is a feasible component and consider a greedy exploration of Γ. Let H' and H'' be the penultimate and last C_k-copies added, respectively. First observe that Case (a) of Corollary 3.5.10 can be excluded. Indeed, in this case H' is an external copy, meaning it adds a path of $k - 1 \geq 4$ new vertices. By Corollary 3.5.9 the last copy H'' is attached to at most one of them, implying there remain evil twins in Γ.

Since only Cases (b) and (c) are possible, and in both $e_1 = 1$, it follows that in *any* greedy exploration of Γ, the first C_k-copies H_1 and H_2 share exactly one edge. Hence, H' and H'' also share at most one edge, as otherwise we could start the exploration with these two copies (this is guaranteed by Claim 3.5.8). Thus, H' adds a path of at least $k-2 \geq 3$ new vertices (since $e_{s-1} = 1$), and either H'' is attached to none of them, or it is attached to a new vertex adjacent to one of the existing vertices of H'. In either case we get that at least one pair of evil twins added by H' survives in Γ, a contradiction. $\square$

Before performing a detailed case analysis for $k = 4$, and in order to simplify it, we first consider the endings of greedy explorations in the following two simple claims.

Claim 3.5.12. *Let Γ be a feasible component. Suppose that in a greedy exploration of Γ the penultimate step is the addition of an external C_4-copy H'. Let x be the existing vertex of H', and let y be the new vertex of H' who is not adjacent to x. Then the internal C_4-copy which is added in the last step is attached to x and y.*

Proof. By Corollary 3.5.9, the last copy added H'' is attached to at most one of the new vertices of H', and so unless H'' is attached to y, evil twins will remain in Γ (the vertex y and one of its neighbors). Additionally, before H'' is added there must be a path of length two between y and the other vertex H'' is attached to, and x is the only vertex meeting this requirement. $\square$

Claim 3.5.13. *Let Γ be a feasible component. Suppose that $\Gamma' \subseteq \Gamma$ is a graph obtained during a greedy exploration of Γ, such that the remainder of the exploration is the addition of a nonempty C_4-chain and then the addition of an internal C_4-copy. Let u be the existing vertex of the first external copy in the chain. Then $d_\Gamma(v) = d_{\Gamma'}(v)$ for every $v \in V(\Gamma') \setminus \{u\}$.*

Proof. Clearly u is the only vertex in $V(\Gamma')$ whose degree is increased by the addition of the C_4-chain. By Claim 3.5.12, the internal C_4-copy added last is attached to two vertices belonging to the vertex set of the last external copy in the chain. So except perhaps u, no vertex of $V(\Gamma')$ is affected by the last step as well. $\square$

We have come to the last claim of this section, showing that although feasible components do exist for $k = 4$, this is no obstacle for Breaker.

Claim 3.5.14. *Breaker wins the unbiased C_4-game played on any feasible component.*

Proof. Let $k = 4$ and let Γ be a feasible component. We consider all possible greedy explorations of Γ and separate the proof according to the three cases of Corollary 3.5.10. We show that there exists a unique feasible component for each case, which is an easy win for Breaker via a pairing strategy. Recall that H_1 is the external copy initiating the exploration, and that H_2 is the internal copy added in the second step. Let H_3 be the second internal copy added during the exploration, and in case $s = 3$ (that is, Case (c)), let H_4 be the third internal copy added (and finishes the exploration). Keep in mind that $e_s = k - 2 = 2$.

Case (a): $s = 2,\ e_1 = e_2 = 2,\ t > 1$.
Since $t > 1$ and $s = 2$, and the exploration terminates with the addition of the internal copy H_3, an external C_4-copy H' is added in the third step of the exploration. First observe that u, the existing vertex of H', must be of degree 3 in Γ_2. Indeed, Γ_2 consists of three paths of length two between the same pair of endpoints $\{x, y\}$. By Claim 3.5.13, every vertex in $V(\Gamma_2) \setminus \{u\}$ remains with the same degree in Γ. If $u \notin \{x, y\}$, then in Γ all conditions of Claim 3.5.4 hold, and $\{x, y\}$ is a bad pair, a contradiction.

Next, assume for contradiction that $t > 2$ and thus an external copy H'' is added in the fourth step. By the restrictions of the exploration process, the existing vertex of H'' is one of the new vertices of H'. Let $V(H') = \{w, x, y, z\}$, where w and z denote the existing vertices of H' and H'', respectively. By Claim 3.5.13 we have $d_\Gamma(x) = d_\Gamma(y) = 2$, making x and y evil twins if they are adjacent. If they are not, then every C_4-copy containing one of them must contain their two neighbors w and z as well. On the other hand, as $wz \notin E(\Gamma)$ in this case, every C_4-copy containing both w and z must contain their only two common neighbors x and y. It follows that both x and y belong to no other C_4-copy but H', which makes them a bad pair.

In conclusion, if an external copy is added after H' then two of the new vertices of H' will be a bad pair in Γ, a contradiction. Thus $t = 2$ and H_3 is added in the fourth and last step, and by Claim 3.5.12 it is attached to the existing vertex w of H' and to the non-neighbor of w in H'. The resulting graph is $\Gamma^{(a)}$ shown in Figure 3.5. Breaker can win the game by following the pairing strategy with respect to the pairs $\Lambda = \{\{x_1, x_2\}, \{y_1, y_2\}, \{z_1, z_2\}\}$.

Case (b): $s = 2,\ e_1 = 1,\ e_2 = 2.$
Since $e_1 = 1$, in both H_1 and H_2 the two vertices not in the intersection are evil twins. By Claim 3.5.13, if the third exploration step is the addition of an external C_4-copy, then at least one of these pairs survives in Γ. Thus, in the third and last step H_3 is added, and in such a way it is attached to one vertex from each pair of evil twins. Clearly the only possible option is $\Gamma^{(b)}$ shown in Figure 3.5. Breaker can win the game by following the pairing strategy with respect to the pairs $\Lambda = \{\{x_1, x_2\}, \{y_1, y_2\}\}$.

Case (c): $s = 3,\ e_1 = e_2 = 1,\ e_3 = 2.$
As in Case (b), in Γ_2 there exist two pairs of evil twins, T_1 and T_2, which are the two vertices of degree 2 in H_1 and H_2, respectively. At least one of these pairs survives in Γ', the graph obtained after the addition of H_3. Indeed, $e_2 = 1$ and so H_3 is attached to two adjacent vertices. Now, if H_3 is added in the third step it cannot be attached to one vertex from each T_i. Otherwise, H_3 is added after an external copy H', and by the restriction of the exploration must be attached to a new vertex of H' and one of its neighbors, which obviously must also belong to $V(H')$. Thus, in this case only one vertex in $V(\Gamma_2)$ has larger degree in Γ' than in Γ_2.

Assume then WLOG that the pair T_1 survives in Γ', and note that the two new vertices added by H_3 are evil twins as well. Denote this pair by T_3. By Claim 3.5.13, if the next addition to Γ' is of an external C_4-copy, then at least one of the pairs T_1 and T_3 survives in Γ. It follows that H_4 is added immediately after H_3, and must be attached to one vertex from each of the pairs T_1 and T_3. Hence, there must be a path of length two in Γ' between these two vertices, that is, they must have a common neighbor. This common neighbor must belong to $V(H_1) \cap V(H_3)$, since for $i = 1, 3$, the vertices of T_i do not have any neighbors outside H_i.

The only way to meet these requirements, as well as the requirement that T_2 does not survive in Γ, is to add H_3 in the third step, and to attach it to a vertex of T_2 and to its neighbor in $V(H_1) \cap V(H_2)$. This yields the graph $\Gamma^{(c)}$ as shown in Figure 3.5. Breaker can win the game by following the pairing strategy with respect to the pairs $\Lambda = \{\{x_1, x_2\}, \{y_1, y_2\}, \{z_1, z_2\}\}$. $\square$

Claim 3.5.14 provides the last missing piece of the puzzle and we can finally prove formally the main result of this section.

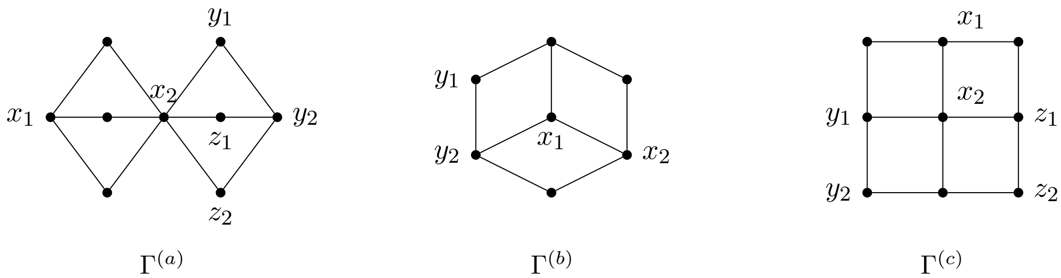

Figure 3.5: The three feasible components for $k = 4$

Proof of the 0*-statement in Theorem 3.1.4 for* $H' = C_k,\ k \geq 4$. Recall that we consider $(1 : b)$ H-games, and that H' is a subgraph of H of maximal 1-density. It suffices to show that breaker can win the C_k-game, as this ensures his win in the H-game as well. By bias monotonicity, it is also sufficient to prove Breaker's win for the case $b = 1$. Fix a constant $c < k^{-1/k}$ and let $G \sim G(n, cn^{-(k-1)/k})$. By Lemma 3.3.7 it suffices to show that w.h.p. Breaker can win the game played on the vertex set of G^*, the $(C_k, 1)$-core of G. By Claim 3.5.1 and Definition 3.5.2, w.h.p. G^* is either empty or contains only feasible components. Claim 3.5.11 implies that for $k \geq 5$ w.h.p. G^* is empty, and thus Breaker wins trivially. Claim 3.5.14 shows that for $k = 4$ w.h.p. Breaker has a winning strategy for the game played on any component of G^*, which implies Breaker's win in the game played on G^* itself by Observation 3.3.8. □

3.6 Biased Maker-Breaker triangle games

In this section we prove the 0-statement in Theorem 3.1.4 for $H' = K_3$ and $b \geq 2$. Similarly to the proofs in Sections 3.4 and 3.5, it suffices to prove that w.h.p. Breaker wins in the $(1 : 2)$ triangle game played on the vertex set of $G \sim G(n, cn^{-2/3})$ where $c < 1$ is an arbitrary constant. Once again, by Lemma 3.3.7 it suffices to consider the game played on the vertex set of G^*, the $(K_3, 2)$-core of G. By Observation 3.3.8 it suffices to consider each connected component of G^* separately. Let $\Gamma \in X_{q,t,s}$ be a connected component of G^*. By Claim 3.3.15 we may assume that $2q + s \leq 3$. Now, if Γ contains no dangerous edges we know by Lemma 3.3.16 that it is a K_3-cycle of length at least four, and Breaker can apply the pairing strategy given in Definition 3.2.4. Breaker's win is thus established by the following lemma, showing that Γ cannot contain dangerous edges, .

Lemma 3.6.1. *Let* $\Gamma \in X_{q,t,s}$ *be a* $(K_3, 2)$*-stable graph containing at least one dangerous edge. Then* $2q + s > 3$.

Proof. First note that $v(\Gamma) = 1 + 2t + s > 6$ or otherwise Γ would be a small component, in contradiction to its stability. Now, assume for contradiction that $2q + s \leq 3$. Recall that the sum $2q + s$ depends only on Γ and not on the way we explore it, so we may start the exploration of Γ with two triangles H_1 and H_2 sharing an edge, as described in Corollary 3.3.18. In particular we have $s \geq 1$, so we only have to consider the following two cases.

1. $q = s = 1$.

2. $q = 0$, $s \leq 3$.

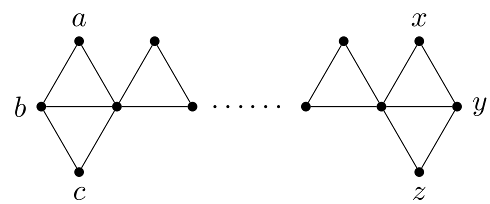

Figure 3.6: A possible structure for Γ' in Case 2

We analyze each case separately, showing that none of them is possible, thus completing the proof.

Case 1: $q = s = 1$.
The restriction $1 + 2t + s > 6$ implies $t > 2$ in this case. Thus, the remainder of the exploration after Γ_2 is the addition of a K_3-chain of length at least two, followed by the addition of an internal edge e as the last step of the exploration. Let $\{a, b, c\}$ be the three vertices in $V(\Gamma_2)$ not contained in the K_3-chain, let H' be the last external copy in the chain, and let $\{x, y\}$ be its new vertices. Right before the addition of e, both $\{a, b, c\}$ and $\{x, y\}$ are bad sets. By stability, e must belong to a triangle in Γ, and so its two endpoints have a common neighbor in Γ. It follows that e cannot contain vertices from both $\{a, b, c\}$ and $\{x, y\}$ since the length of the K_3-chain is at least two. Thus we get a bad set in Γ, which contradicts its stability.

Case 2: $q = 0,\ s \leq 3$.
In this case every step of the exploration process is the addition of a K_3-copy. We first observe that the last two additions are of internal copies. This is trivial for the last step by Corollary 3.3.10. Assume for contradiction that the penultimate addition is of an external K_3-copy consisting of an existing vertex w and new vertices x, y. Since these new vertices form a bad pair, the stability of Γ implies that the internal triangle added in the last step contains an exiting edge incident to exactly one of them, i.e. this triangle is WLOG wyz, where z is a new vertex. At this point Γ is fully explored, however $\{x, y, z\}$ form a bad set, which contradicts the stability of Γ.

Since the restrictions $1 + 2t + s > 6$ and $s \leq 3$ imply $t > 1$ we now conclude that the remainder of the exploration after Γ_2 is the addition of a nonempty K_3-chain, followed by the addition of two internal copies H_3 and H_4 in the last two steps. Let H' be the last triangle in the K_3-chain, let x, y be its new vertices, let z be the new vertex added by H_3, and let Γ' be the explored component after the addition of H_3. By the constraint of the exploration, H_3 must contain an existing edge incident to at least one of $\{x, y\}$, that is, the exiting edge belongs to H'. It follows that $\{x, y, z\}$ form a bad set in Γ'. The same holds for $\{a, b, c\}$, the three vertices of $V(\Gamma_2)$ not contained in the K_3-chain (see an example in Figure 3.6). Note that there is no edge between the triplets $\{a, b, c\}$ and $\{x, y, z\}$, regardless of the length of the K_3-chain. Since H_4 must contain an existing edge when added, it cannot contain vertices from both triplets, so at least one of them remains a bad set in Γ, a contradiction. □

3.7 The unbiased Maker-Breaker triangle game

Proof of Theorem 3.1.4. We begin our proof with two simple sufficient conditions for Maker's win in the $(1:1)$ triangle game played on the vertex set of an arbitrary graph G – one for the case he is the first player and one for the case he is the second player. We refer to the labeling of the vertices as they appear in Figure 3.2. If G contains a DD-copy and Maker plays first, he claims x, y_i and z_j in his first three move (in this order), where $i \in [2]$ and $j \in [4]$ are (possibly) determined according to Breaker's moves. Similarly, if G contains two vertex disjoint DD-copies then Maker, as a second player, does the same, where the DD-copy in which he claims his vertices is a copy whose all its vertices are still free after Breaker's first move. It is immediate to see that Maker indeed wins in these cases, and furthermore achieves his goal already in his third move.

For the main part of the proof, let $\tilde{G} = \{G_i\}$ be a random graph process, let $i_1 = \tau\left(\tilde{G}, \mathcal{G}_{DD}\right) - 1$ and $i_2 = \tau\left(\tilde{G}, \mathcal{G}_{2DD}\right) - 1$, and let G_1^* and G_2^* be the $(K_3, 1)$-cores of G_{i_1} and G_{i_2}, respectively. Maker, as a first player, has a winning strategy for the game played on $V(G_{i_1+1})$. Note that since DD is strictly balanced, the first two DD-copies appearing during the random graph process are w.h.p. vertex disjoint by Corollary 3.2.16, and therefore w.h.p. Maker, as a second player, has a winning strategy for the game played on $V(G_{i_2+1})$. Since "being Breaker's win" is a monotone decreasing graph property, in order to complete the proof of the theorem it is enough to show that w.h.p. Breaker has a winning strategy as a second player in the game played on $V(G_{i_1})$, and as a first player in the game played on $V(G_{i_2})$. By Lemma 3.3.7 it is enough to show that w.h.p. Breaker has a winning strategy as a second player for the game played on $V(G_1^*)$, and as a first player for the game played on $V(G_2^*)$.

The key ingredient for the proof is Claim 3.3.20. Recall the family $\mathcal{F}$ defined in the lemma of all graphs on less than 25 vertices with density larger than 10/7. We now show that if G_{i_2} is $\mathcal{F}$-free, then Breaker has winning strategies for the two games in discussion.

Assume that G_{i_2} is $\mathcal{F}$-free, and note that by containment G_1^* and G_2^* are $\mathcal{F}$-free as well. Since G_1^* is in addition DD-free, and K_3-stable by definition, Claim 3.3.20 implies that every connected component in G_1^* is w.h.p. a Tr-copy, for which there exists a natural pairing strategy. Similarly, G_{i_2} has exactly one DD-copy $\mathcal{D}$, and so every connected component in G_2^* other than $\mathcal{D}$ is a Tr-copy. Breaker, as a first player, can claim the center of $\mathcal{D}$ in his first move (thus ensuring that Maker cannot claim any triangle contained in $\mathcal{D}$), and then apply the natural pairing strategy on all remaining components.

It only remains to observe that since the family $\mathcal{F}$ is finite, and as $m(DD) = \frac{10}{7}$, the graph G_{i_2} is w.h.p. $\mathcal{F}$-free by Claim 3.2.15 and by definition of i_2. $\square$

3.8 Trees and Forests

We begin with the proofs of Theorems 3.1.1 and 3.1.10.

Proof of Theorem 3.1.1. We consider each item separately.

1. We do not attempt to optimize the size of T. Let $\Delta = \Delta(H)$, let $h = v(H)$, let $d = 2b\Delta^h$, and let T be the d-ary tree with h levels rooted at $r \in V(T)$. Maker's strategy when playing on the vertices of T is as follows. He begins by claiming r, then in his next Δ moves he claims Δ arbitrary children of r, then Δ arbitrary children for each of them and so on until he reaches

the leaves level. When he finishes he owns a Δ-ary tree with h levels rooted at r, which clearly contains an H-copy. It only remains to observe that by the time Maker finishes to claim his Δ-ary tree, Breaker claims $b\sum_{i=0}^{h-1}\Delta^i \le b\Delta^h$ vertices, and so Maker has enough children to claim for each vertex in his tree at any point of the game, and he can follow this strategy.

2. Let i such that $T_{\max} = T_{\min}^{(i)}$. If $p = o\left(n^{-\frac{v}{e}}\right)$ then by Claim 3.2.14 every connected component of $G \sim G(n,p)$ is w.h.p. a tree with less than v vertices. By assumption, Breaker can prevent Maker from claiming a T_i-copy on any such component and thus Maker cannot claim an H-copy and loses the game.

   Let $\mathcal{F} = \{T_{\min}^{(i)}\}_{i=1}^{k}$ and let F be the graph consisting of $2bkv$ copies of each $T \in \mathcal{F}$, where all copies of all trees are vertex disjoint. Since $\mathcal{F}$ is finite and every $T \in \mathcal{F}$ satisfies $m(T) \le \frac{e}{v}$, it follows by Theorem 3.2.12 that if $p = \omega\left(n^{-\frac{v}{e}}\right)$ then w.h.p. $G \sim G(n,p)$ contains a copy of the (fixed) graph F. Maker can claim all trees T_i, one at the time, by playing k separate games. For each $i \in [k]$ he plays the $(1:b)$ T_i-game on the vertex set of some $T_{\min}^{(i)}$-copy whose all its vertices are still free at the moment Maker starts. Since Maker can win each such game, and in no more then v moves, by the time he finishes claiming a copy of H, Breaker claims at most bkv vertices, and so for every i Maker can always find a $T_{\min}^{(i)}$-copy with all of its vertices still free and he can follow this strategy. □

Remark 3.8.1. In [44, Lemma 36], Stojaković analyzed the unbiased Maker-Breaker H-game played on the edge set of $G \sim G(n,p)$, when H is a tree. It is immediate to see that his result can be generalized to biased games on forests by using essentially the same proof of Theorem 3.1.1, with slight modifications.

Proof of Theorem 3.1.10. We consider each item separately.

1. Dean and Krivelevich showed in the proof of Proposition 1.9 in [16] that for any two integers b and k, Client has a strategy to build a copy of the k-ary tree of height k when playing the Client-Waiter $(1:b)$ game on the edge set of the m-ary tree of height k, where $m = (k(b+1))^2$. Client can use the same strategy in the vertex version by replacing each parent-child edge with the vertex of the child and build a k-ary tree of height $k-1$ (in fact k copies of it). Clearly building such a tree for $k = v(H)$ is a winning strategy for Client in the H-game.

2. As in the proof of Theorem 3.1.1, let i such that $T_{\max} = T_{\min}^{(i)}$, and note that if $p = o\left(n^{-\frac{v}{e}}\right)$ then by Claim 3.2.14 every connected component of $G \sim G(n,p)$ is w.h.p. a tree with less than v vertices. By offering in every move vertices from the same component, Waiter can prevent Client from claiming a T_i-copy, and thus Client cannot claim an H-copy and loses the game.

   For every $i \in [k]$ let $n_i = \left((bv)^2+1\right)^{i-1}$, let F_i be the forest consisting of n_i vertex disjoint copies of $T_{\min}^{(i)}$, denoted by $\{T_{\min}^{(i,j)}\}_{j=1}^{n_i}$, let $F = \bigcup_{i\in[k]} F_i$ and finally let $F_{<i} = \bigcup_{\ell<i} F_\ell$. We now show that every graph containing an F-copy is Client's win, which completes the proof by Theorem 3.2.12 and the fact that F is a fixed graph satisfying $m(F) = \frac{e}{v}$.

   Given a graph G containing F, Client plays as follows. He ignores all vertices of $V(G) \setminus V(F)$ (if he is offered only vertices of this sort he chooses one arbitrarily). Whenever he is offered a set of vertices U intersecting F, he considers only the vertices belonging to $T_{\min}^{(i,j)}$ for the minimal pair (i,j) which appears in U, where he uses the natural lexicographical ordering of pairs: $(i, j) < (i', j')$ if either $i < i'$ or $i = i'$ and $j < j'$. He chooses a vertex from that tree according to his winning strategy in the T_i-game when playing on $V\left(T_{\min}^{(i)}\right)$, and deletes from F all other trees which intersect U.

Note that this strategy ensures that for every i and j, whenever Waiter offers a vertex of $T_{\min}^{(i,j)}$, either Client claims a vertex in that tree according to his winning strategy on it, or he deletes it from F. It follows that at the end of the game, if a tree $T_{\min}^{(i,j)}$ is still in F, then Client has claimed a T_i-copy in it. It remains to show that for every i at least one $T_{\min}^{(i,j)}$ survives in F until the end of the game.

Now let $i \in [k]$, and note that a tree $T_{\min}^{(i,j)}$ is deleted from F only if one of its vertices is offered to Client in a set containing a vertex from $T_{\min}^{(i',j')}$ for some pair $(i', j') < (i, j)$. Since all trees in F have at most v vertices, every tree in $F_{<i}$ causes the deletion of at most bv trees in F_i. Furthermore, each deleted tree in F_i can cause the deletion of at most $b(v-1)$ additional trees from F_i (one of its vertices is the one causing its own deletion). Therefore, the number of trees deleted from F_i throughout the game is at most

$$bv(1 + b(v-1)) \sum_{\ell=1}^{i-1} n_\ell \leq (bv)^2 \sum_{\ell=1}^{i-1} \left((bv)^2 + 1\right)^{\ell-1} = (bv)^2 \frac{\left((bv)^2+1\right)^{i-1} - 1}{((bv)^2+1) - 1} = n_i - 1,$$

meaning that at least one tree survives in each F_i until the end of the game, and thus Client wins. □

We now show some more equivalencies between the edge and vertex versions of H-games where H is a tree. Let S_d be the star with d leaves. Proposition 38 in [44] (for the Maker-Breaker game) and Claim 4.13 in [16] (for the Client-Waiter game) show that the minimal size of a tree for which the builder of the game (either Maker or Client) has a winning strategy in the unbiased S_d-game played on its edges is $2d - 1$. In both cases a minimal example is a star.

The vertex version of these games presents a similar behavior even when considering the biased game. That is, following the terminology of Theorems 3.1.1 and 3.1.10, for any two positive integers b and d, and with respect to the $(1 : b)$ S_d-game, $v(T_{\min}) = d(b+1)+1$ in both the Maker-Breaker and Client-Waiter games. Indeed, Maker can win the game when playing on $V\left(S_{d(b+1)}\right)$ by first claiming the center of the star and then claiming arbitrary d leaves in his remaining moves. Client's strategy would be to claim the star center whenever it is offered to him, and claim an arbitrary leaf of the star in any other move. This is a winning strategy as the game lasts at least $\lceil (d(b+1)+1)/(b+1) \rceil = d+1$ rounds. As for Breaker's and Waiter's sides, Maker claims at most d vertices when playing the game on the vertex set of any tree with at most $d(b+1)$ vertices, and so does Client assuming Waiter offers $b+1$ vertices in every move (except maybe the last). Hence, both Maker and Client trivially lose their games in this case. In particular, by Theorems 3.1.1 and 3.1.10, both Maker-Breaker and Client-Waiter $(1 : b)$ S_d-games have a threshold at $p = n^{-\frac{d(b+1)+1}{d(b+1)}}$. Note that the same biased games generalization yields that $S_{(d-1)(b+1)+1}$ is a minimal sized tree required for the builder in the $(1 : b)$ edge versions of the games.

By using an almost identical proof to that of Proposition 37 in [44], one can show that $v(T_{\min}) = \Theta\left(2^{\ell/2}\right)$ when considering the unbiased Maker-Breaker P_ℓ-game. The argument in the proof of Claim 4.14 in [16] shows that $v(T_{\min}) = \Omega\left(2^{\ell/2}\right)$ when considering the unbiased Client-Waiter P_ℓ-game. We omit the slight modifications required in both cases.

3.9 Graphs containing a triangle

Proof of Theorem 3.1.3. Throughout this section G stands for the random graph $G(n, p)$, where the values of p vary according to the different parts of the theorem.

We begin the proof with the 0-statements. First, if $\alpha \leq 7/10$ and $p = o(n^{-7/10})$, then by Corollary 3.1.5 Breaker w.h.p. can prevent Maker from claiming a triangle, and thus wins the H-game in this case. Next, if $\alpha > 7/10$ and $p \leq cn^{-7/10}$, then the assertion of Conjecture 3.1.2 implies that Breaker can prevent Maker from claiming an H'-copy, and once again win the H-game.

Moving to the 1-statements, if $\alpha \geq 3/2$ then Maker's win follows immediately from Remark 3.1.6. Assume then that $\alpha < 3/2$, and note that whether $\alpha > 7/10$ or not, we have both $p = \omega(n^{-7/10})$ – and thus G w.h.p. contains a DD-copy by Corollary 3.1.5 – and $p \geq Cn^{-7/10}$, where C is the constant guaranteed in Theorem 3.1.5, with respect to H' and $r = 20$; that is, w.h.p. every induced subgraph of G with $n/20$ vertices contains an H'-copy. From now on we assume G satisfies these two properties. Since "being Maker's win" is a monotone increasing property, we add the technical assumption $p = o(1)$.

The remainder of the proof goes along the same lines as the proof of [40, Theorem 4]. For completeness we repeat it here with the necessary modifications. We first need the following claim about the expansion of G.

Claim 3.9.1 ([40], proof of Theorem 4)**.** *Let $p = \omega(n^{-7/10})$ and $G \sim G(n, p)$. Then w.h.p, for every subset $X \subset V(G)$ of size $|X| \leq 1/(2p)$, we have $|N_G(X)| \geq |X|np/4$.*

We now assume G possesses this expansion property, and describe Maker's winning strategy, which we divide into five phases. For $i \in [5]$, denote the set of vertices claimed by Maker during the ith phase by M_i, and the set of vertices claimed by both players during the phase by R_i.

Phase 1: In his first three moves, Maker claims a triangle $v_1v_2v_3$ by playing on the vertex set of some DD-copy in G (recall Theorem 3.1.4 and its proof).

Phase 2: In his next $100/(n^2p^3)$ moves, Maker claims arbitrary vertices from $N_G(v_1)\backslash R_1$. This is possible since $|N_G(v_1)| \geq np/4 \gg 1/n^2p^3$ by assumption, and $|R_1| = 6$.

Phase 3: In his next $10/np^2$ moves, Maker claims arbitrary vertices from $N_G(M_2) \setminus (R_1 \cup R_2)$. By assumption we have $|N_G(M_2)| \geq |M_2|np/4 = 25/(np^2)$, and since $|R_1 \cup R_2| \ll 1/np^2$ (as $np \gg 1$), we have that $|N_G(M_2) \setminus (R_1 \cup R_2)| \geq 20/np^2$, and so Maker can claim half of these vertices.

Phase 4: In his next $1/(2p)$ moves, Maker claims arbitrary vertices from $N_G(M_3) \setminus (R_1 \cup R_2 \cup R_3)$. By assumption we have $|N_G(M_3)| \geq |M_3|np/4 > 2/p$, and since $|R_1 \cup R_2 \cup R_3| \ll 1/p$ (as $np \gg 1$), we have that $|N_G(M_2) \setminus (R_1 \cup R_2 \cup R_3)| \geq 1/p$, and so Maker can claim half of these vertices.

Phase 5: In his next $n/20$ moves, Maker claims arbitrary vertices from $N_G(M_4) \setminus (R_1 \cup R_2 \cup R_3 \cup R_4)$. By assumption we have $|N_G(M_4)| \geq |M_4|np/4 = n/8$, and since $|R_1 \cup R_2 \cup R_3 \cup R_4| \ll n$ (as $np \gg 1$), we have that $|N_G(M_2) \setminus (R_1 \cup R_2 \cup R_3 \cup R_4)| \geq n/10$, and so Maker can claim half of these vertices.

Now, $|M_5| = n/20$ and so by assumption it contains an H'-copy. Since $v_1v_2v_3$ is a triangle, and since every vertex in M_5 is connected to v_1 via a path of length four disjoint from $\{v_2, v_3\}$ by construction, Maker's graph contains an H-copy at the end of Phase 5 (note that the total number of rounds by the end of Phase 5 is less than $n/10$, so Maker can follow the proposed strategy). $\square$

3.10 Other game types

We begin this section with the description and analysis of a positional game in which all target sets are pairwise disjoint. This game is later used as an auxiliary game in many of the proofs in this section.

3.10.1 Box games

In their seminal paper [12], Chvátal and Erdős introduced the *box game*. This is essentially a Maker-Breaker game where all board elements are partitioned into element disjoint winning sets. Each winning set is referred to as a *box*, and the two players are denoted by BoxMaker and BoxBreaker. Chvátal and Erdős used this game in their analysis of the connectivity game as part of Breaker's strategy, who pretends to be BoxMaker in an auxiliary game, and by this isolates a vertex in Maker's graph. This is of course a winning strategy for Breaker in other games for which positive minimum degree in his graph is a necessary condition for Maker, such as the Hamiltonicity game and the perfect matching game.

Chvátal and Erdős were interested in the $(a:1)$ box game where the sizes of the smallest and largest winning sets differ by at most one. Later, in [24], Hamidoune and Las Vergnas analyzed the box game in full generality (and also corrected a mistake in one of the proofs in [12]). In this chapter we use different variations of box games for some of the proofs in Section 3.10. We are only interested in uniform box games, that is, all boxes have the same size. We denote this setting by $n \times k$, where n indicates the number of boxes and k denotes their size. We first state and prove a trivial result for the $(a:b)$ Waiter-Client version of the uniform game, abbreviated to $WCBox(n \times k, (a:b))$. This will be useful in the proof of Theorem 3.1.8.

Claim 3.10.1. *Let a, b, k be three positive integers. There exists an integer $N = N(a, b, k)$ such that for every $n \geq N$ BoxWaiter has a winning strategy in the game $WCBox(n \times k, (a:b))$.*

Proof. We do not wish to optimize the bound on N and rather show that given a and b, BoxWaiter has a winning strategy in the game $WCBox(n_k \times k, (a:b))$, where $n_k = (a+b)^k$. Clearly, this strategy is also applicable to a game containing $n > n_k$ boxes since BoxWaiter can first offer elements only from the first n_k boxes, achieve his goal in the game, and then offer all remaining elements arbitrarily. We prove the claim by induction on k. For $k = 1$ BoxWaiter simply offers all $(a+b)$ elements in the game, one from each box. BoxClient then claims at least one of them and loses. Now Assume that BoxWaiter has a winning strategy in the game $WCBox(n_k \times k, (a:b))$ and consider the game $WCBox(n_{k+1} \times (k+1), (a:b))$. In the first n_k rounds of the game, BoxWaiter offers arbitrary $a+b$ elements in every move, each from a different box for which none of its elements has been offered yet. After these rounds there are $bn_k \geq n_k$ boxes in which Client has claimed an element, and no other element in them has been offered yet. BoxWaiter can then apply his strategy for the game $WCBox(n_k \times k, (a:b))$ on these boxes and win. $\square$

In [20], Ferber, Krivelevich and Naor analyzed the misère version of the box game. That is, BoxEnforcer is trying to force BoxAvoider to fully claim a box, while BoxAvoider is trying to prevent that. Following the ideas of [12], they also used the monotone version of the game as an auxiliary game in order to provide Avoider with a strategy to isolate a vertex in his graph and win various spanning games. This was indeed Avoider's strategy in the k-connectivity, Hamiltonicity, and perfect matching games considered in [18]. We abbreviate the $(a : b)$ Avoider-Enforcer uniform box game to $AEBox(n \times k, (a : b))$. When considering the monotone game, the situation is very simple. Similarly to BoxWaiter, BoxEnforcer wins if there are sufficiently many boxes, where the number of required boxes depends on the bias of the players and the box size, all of which are fixed.

Theorem 3.10.2 (Theorem 1.7 in [20])**.** *Let a, b, k be three positive integers. There exists an integer $N = N(a, b, k)$ such that for every $n \geq N$ BoxEnforcer has a winning strategy in the monotone game $AEBox(n \times k, (a : b))$ as a first or a second player.*

However, when considering the strict game, things are more complicated and the outcome of the game depends on some divisibility conditions. The following theorem appears in a slightly weaker form as Corollary 1.4 in [20], and can be deduced from Theorems 1.2 and 1.3 of that paper.

Theorem 3.10.3 ([20])**.** *Let a, b, k be three positive integers. If $\gcd(a + b, \ell) \leq a$ holds for every $2 \leq \ell \leq k$, then there exists an integer $N = N(a, b, k)$ such that for every $n \geq N$ BoxEnforcer has a winning strategy in the strict game $AEBox(n \times k, (a : b))$ as a first or a second player. Otherwise, BoxAvoider wins this game for every n.*

We conclude this subsection with the following related claim which will be very useful in the proof of Theorem 3.1.7.

Claim 3.10.4. *Let m and n be two non-negative integers and let $\mathcal{H} = (X, \mathcal{F})$ be a hypergraph where $X = \{a_1, \ldots, a_m, b_1, \ldots, b_m, c_1, \ldots, c_n\}$ and $\mathcal{F} = \{\{a_1, b_1\}, \ldots, \{a_m, b_m\}\}$.*

(*a*) *Avoider wins the strict $(1 : 1)$ Avoider-Enforcer game played on $\mathcal{H}$ as a first or a second player.*

(*b*) *If, in addition, Avoider makes the last move in the game (that is, either n is odd and Avoider plays first, or n is even and Avoider plays second), he also wins the strict $(1 : 1)$ game played on $\mathcal{H}' = (X', \mathcal{F}')$ where $X' = X \cup \{d\}$ and $\mathcal{F}' = \mathcal{F} \cup \{\{d\}\}$ for a new element $d \notin X$ (note that when playing on $\mathcal{H}'$ Enforcer makes the last move in the game).*

Proof. Avoider uses the same strategy for both variants of the game. As long as there exists a free vertex c_i, or a free vertex of one of the pairs $\{a_i, b_i\}$ such that the other vertex in that pair is already claimed by Enforcer, then Avoider claims one of these vertices arbitrarily. If before one of his moves no such vertex exists, then at this point only pairs $\{a_i, b_i\}$ in which both vertices are free remain. Indeed, it is trivial when playing on $\mathcal{H}$, and for $\mathcal{H}'$ it follows from the fact that there is an even number of free vertices before each of Avoider's moves by assumption. From this moment until the end of the game Avoider plays as BoxAvoider in the strict game $AEBox(n' \times 2, (1 : 1))$, where $n' \leq n$ is the number of unclaimed pairs. By Theorem 3.10.3 he can Avoid claiming both elements of a pair (as $\gcd(1 + 1, 2) = 2 > 1$), and by doing so win the game. □

3.10.2 Proofs of Theorems 3.1.6, 3.1.7, 3.1.8, 3.1.9 and 3.1.12

We begin this subsection with the proofs of Theorems 3.1.6 and 3.1.8. Since the statements of the theorems are almost identical, it is not surprising that their proofs are also very similar.

Proof of Theorems 3.1.6 and 3.1.8. Let $G' \subseteq G$ be a collection of vertex disjoint H-copies of maximal size. Enforcer and Waiter win their respective H-games if G' is sufficiently large, in which case they can treat each H-copy in it as a box (where each vertex in the copy is one element in the box) and simulate some box game.

By the proof of Claim 3.10.1, Waiter has a winning strategy if G' contains $(a+b)^{v(H)}$ copies, because he simply plays first the $WCBox$ game on G' and after achieving his goal offers all other vertices in the game arbitrarily. Enforcer's strategy is straight forward as well. We assume for simplicity that he plays first, and the proof is very similar when he is the second player. In his first move he claims all vertices in $V(G) \setminus V(G')$ and from that moment he plays as BoxEnforcer in the $(a : b)$ misère box game. By Theorem 3.10.2, BoxEnforcer wins as long as there are enough boxes, where enough means some constant depending only on a, b and $v(H)$. Winning in this game means of course that Avoider is forced to claim all vertices of some H-copy.

Since Client and Avoider win trivially in their respective games if G is H-free, the result for a general graph H in both theorems follows by Theorem 3.2.12.

For a strictly balanced graph H the result follows immediately by Corollary 3.2.16, where the integer N from the statements of the theorems represent the minimal number of boxes required for the wins of BoxWaiter and BoxEnforcer, respectively, in the games described in this proof. □

It is immediate to see that by replacing the vertices of the H-copies with their edges in the proof above, we can obtain the same results for Avoider-Enforcer and Waiter-Client $(a : b)$ H-games played on the edge set of random graphs. The Avoider-Enforcer result can also be obtained from Corollary 1.8 in [20].

Corollary 3.10.5. *Theorems 3.1.6 and 3.1.8 are still valid when stated for the edge versions of the games, with the only minor difference that the parameter N in their statements depends (in addition to the bias of the players) on $e(H)$ rather than on $v(H)$.*

We continue with the strict Avoider-Enforcer game played on the vertex set of $G \sim G(n,p)$. Before proving Theorem 3.1.7, we would like to discuss the H-game for a given arbitrary graph H. Theorem 3.10.3 imply that there are infinitely many pairs of integers (a,b), for which a strict game equivalent of the statement of Theorem 3.1.6 holds. These are all pairs satisfying $\gcd(a+b,\ell) \leq a$ for every $2 \leq \ell \leq v(H)$. For example, two obvious families of such pairs are all pairs (a,b) such that $a \geq v(H)$, and all pairs (a,b) such that $a+b$ is a prime number larger than $v(H)$. There is one small difference in the statement of the hitting time result, though. The number N of required H-copies in the graph does not depend only on a, b and $v(H)$, but also on the identity of the first player and the residue of $n \mod (a+b)$, where n is the number of vertices in the graph process. It does not depend on the value of n itself, however, and therefore it is a constant. One simple example is that if $n \equiv 0 \pmod{(a+b)}$, Enforcer plays first and $a \geq v(H)$, then $N = 1$ (Enforcer plays arbitrarily and only avoiding the vertices of the single H-copy in G, which will be fully claimed by Avoider eventually).

At the same time, there exist infinitely many integers b such that w.h.p. Avoider wins the $(1 : b)$ H-game for every $p = O\left(n^{-1/\mu(H)}\right)$, where $\mu(H) = m(H) + \frac{1}{2v(H)^2}$. Indeed, let $\hat{H} \subseteq H$ be a strictly balanced graph with $m(\hat{H}) = m(H)$. By Claim 3.2.9, Theorem 3.2.12, the assumption on p, and the fact that two $\hat{H}$-copies can intersect in finitely many ways, all $\hat{H}$-copies in G are w.h.p. vertex disjoint. Thus, if $b+1$ is divisible by an integer $2 \leq \ell \leq v(\hat{H})$, then Avoider can win the H-game. His strategy would be to first play arbitrarily as long as he does not claim the first unclaimed vertex of any $\hat{H}$-copy, and if it at some point he can no longer do so, play until the end of the game as BoxAvoider where each box is an $\hat{H}$-copy. He wins this game by Theorem 3.10.3. This strategy ensures that he avoids fully claiming any $\hat{H}$-copy and thus wins the H-game. For example, Avoider can apply this strategy in any $(1 : b)$ H-game whenever b is odd (and thus $b+1$ is divisible by $2 \leq v(\hat{H})$). In particular, the unbiased H-game is never an "easy" win for Enforcer, in the sense that the threshold probability $p^*_{1,H}$ in this case satisfies $p^*_{1,H} = \omega\left(n^{-1/\mu(H)}\right)$ for every H, including, of course, the case $H = K_3$, which we now discuss.

Proof of Theorem 3.1.7. Not surprisingly, the proof goes along the lines of the proof of Theorem 3.1.4. We omit or abbreviate some of the repeated arguments. We begin with sufficient conditions for Enforcer's win when playing the game on $V(G)$ for an arbitrary graph G. Assume first that Avoider makes the last move of the game and that G contains a DD-copy $\mathcal{D}$. By Part (b) of Claim 3.10.4, Enforcer can play an auxiliary game (as *AuxAvoider*) and avoid claiming the center of $\mathcal{D}$ (the vertex x) and any full pair of the natural pairs of $\mathcal{D}$. It means that Avoider (as *AuxEnforcer*), must claim x and at least one vertex form each pair, which leads to Enforcer's win in the original game by Observation 3.2.8. Similarly, assuming that Enforcer makes the last move, by Part (a) of the same claim, if G contains two vertex disjoint DD-copies, then Enforcer can win by ensuring that Avoider claims at least one vertex from each of their natural pairs, and at least one of their centers (by pairing up the centers as well).

Now let $\tilde{G} = \{G_i\}$ be a random graph process, and recall that $\mathcal{G}_{DD}$, $\mathcal{G}_{2DD}$, $\mathcal{E}^A_{K_3}$ and $\mathcal{E}^E_{K_3}$ are the graph properties of containing one or two vertex disjoint DD-copies, respectively, and being Enforcer's win in the strict $(1 : 1)$ triangle game, where Avoider or Enforcer, respectively, moves last. Then $\tau(\tilde{G}, \mathcal{E}^A_{K_3}) \leq \tau(\tilde{G}, \mathcal{G}_{DD})$ trivially holds, and, since the first two DD-copies appearing during the random graph process are w.h.p. vertex disjoint, $\tau(\tilde{G}, \mathcal{E}^E_{K_3}) \leq \tau(\tilde{G}, \mathcal{G}_{2DD})$ holds w.h.p. as well. Let $i_1 = \tau(\tilde{G}, \mathcal{G}_{DD}) - 1$ and $i_2 = \tau(\tilde{G}, \mathcal{G}_{2DD}) - 1$, and let G^*_1 and G^*_2 be the $(K_3, 1)$-cores of G_{i_1} and G_{i_2}, respectively. It remains to show that w.h.p. Avoider has a winning strategy in the game played on $V(G_{i_1})$, and, provided he is not the last player to play, in the game played on $V(G_{i_2})$. Following the argument presented in the proof of Theorem 3.1.4, from now on we assume that G_{i_2} is $\mathcal{F}$-free (and therefore G^*_1 and G^*_2 are $\mathcal{F}$-free as well), where $\mathcal{F}$ is the family defined in Claim 3.3.20. Since this is true w.h.p., and since all remaining arguments are deterministic, the proof holds.

When playing the game on $V(G_{i_1})$, Avoider first runs the deletion algorithm to obtain an output U, W, G^*_1. By our assumption and by Claim 3.3.20, every connected component of G^*_1 is a Tr-copy. Avoider pairs up the two vertices of any bad pair $U_i \in U$, and for any $W_j \in W$ he chooses arbitrary $\lfloor W_j/2 \rfloor$ disjoint pairs contained in W_j. By Part (a) of Claim 3.10.4 he can avoid claiming all these pairs and all the natural pairs of all components of G^*_1. It follows that by the end of the game he claims at most one vertex from each bad pair U_i and at most two vertices from each small component W_j, and thus by Claim 3.3.6 and the fact that he avoids all triangles in G^*_1, he wins the game.

When playing on $V(G_{i_2})$, every connected component of G^*_2 is a Tr-copy, except for one DD-copy $\mathcal{D}$. If Enforcer makes the last move of the game, then by Part (b) of Claim 3.10.4 Avoider can avoid claiming all the pairs as in the previous case (that is, all the pairs generated by the deletion algorithm and all natural pairs of the Tr-copies in G^*_2), as well as the center of $\mathcal{D}$, and win. □

It remains to prove the theorems regarding Client-Waiter games. The 1-statement of Theorem 3.1.9 follows from Theorem 3.1.5. The 0-statement of the theorem is an immediate corollary of Lemma 3.3.7, the proof of Theorem 3.1.4, and the following trivial observation.

Observation 3.10.6. *For any two graphs G, H and integer b, if Breaker can win the $(1:b)$ Maker-Breaker H-game played on $V(G)$ by using a pairing strategy (with respect to Λ), then Waiter can win the $(1:b)$ Client-Waiter H-game played on $V(G)$ by using the same pairing strategy. That is, Waiter offers each set of at most $b+1$ vertices from Λ in one move, and offers all other vertices arbitrarily.*

We finish this section with the proof of Theorem 3.1.12.

Proof of Theorem 3.1.12. We set $p = an^{-2/3}$ where $a < 1$ is undetermined at this point, and consider the $(1:1)$ Client-Waiter triangle-game played on $G \sim G(n,p)$. We begin with Waiter's side, and so by Lemma 3.3.7 we may focus on G^*, the $(K_3, 1)$-core of G. By Claim 3.3.15 we may assume that all families $X_{q,t,s}$ for which G^* contains at least one of their members satisfy $2q + s \leq 3$. For simpler analysis, it will be convenient to define the following. For any three integers q, t, s, let $Z_{q,t,s}$ be the family of all K_3-stable graphs in $X_{q,t,s}$, which can be explored when beginning with two triangles sharing an edge (i.e., immediately after the addition of the first external triangle we add an internal one). Corollary 3.3.18 and Lemma 3.3.16 imply that every connected components of G^* not contained in any $Z_{q,t,s}$ is a K_3-cycle of length at least four, for which there exists a natural pairing strategy for Waiter (see Definition 3.2.4). From all of the above, and by Observation 3.3.8, it follows that if Client has a winning strategy in the game, then he has a winning strategy when playing on the vertex set of some $\Gamma \in Z_{q,t,s}$, where q, t, s are three integers satisfying $2q+s \leq 3$. The next claim shows that the condition in the last statement may be replaced with $2q + s = 3$.

Claim 3.10.7. *Let q, t, s be integers satisfying $2q + s \leq 2$ and let $\Gamma \in Z_{q,t,s}$. Waiter wins the $(1:1)$ triangle game played on $V(\Gamma)$.*

Proof. By Claim 3.3.19, Γ is either a Tr or a DD_t. If it is a Tr, then Waiter can apply the natural pairing strategy (see Definition 3.2.5). Assume now that Γ is a DD_t. Waiter's winning strategy goes as follows, with respect to the labeling of Figure 3.4. In his first move, he offers the pair $\{b_1, x\}$. By symmetry we may assume that Client chooses b_1. Waiter then applies the natural pairing strategy as described in Definition 3.2.6. □

We now wish to bound from above the probability that G^* contains a member of $Z_{q,t,s}$ for any q, t, s such that $2q + s = 3$. There are only two cases to consider, $q = s = 1$ and $q = 0, s = 3$. As it turns out, it suffices to consider only the latter case.

Claim 3.10.8. $Z_{1,t+1,1} \subseteq Z_{0,t,3}$ *for every integer t.*

Proof. Let $\Gamma \in Z_{1,t+1,1}$. By definition, we start its exploration with two triangles sharing an edge, xyz_1 and xyz_2. By the fact that $q = s = 1$ and by Corollary 3.3.10, the remainder of the exploration must be an addition of a K_3-chain of length t, then the addition of an internal edge, and nothing more. Clearly $t > 0$ as otherwise $\Gamma = K_4$, which is a small component and thus not stable. Denote the series of external triangles by $\{a_i, b_i, c_i\}_{i=1}^t$, where $a_1 \in \{x, y, z_1, z_2\}$, and $c_i = a_{i+1}$ for every $0 < i < t$ in case $t > 1$. Since $\{b_t, c_t\}$ is a bad pair, the addition of the internal edge must create a new triangle containing exactly one of them. Up to isomorphism, this triangle must be $a_t b_t w$, where $w \neq c_t$ is some neighbor of a_t. Therefore, we can explore Γ in a slightly different way: we simply change the last two steps of the exploration. Instead of adding the external triangle $a_t b_t c_t$ and then the edge $b_t w$, we first add the internal triangle $a_t b_t w$ (in this triangle b_t is the new vertex) and then the internal triangle $a_t b_t c_t$ (here c_t is the new vertex). Note that these two triangles are indeed internal with respect to this process and that the beginning of the exploration remains the same, and thus $\Gamma \in Z_{0,t,3}$. □

Since Claims 3.10.7 and 3.10.8 show that all components which could be problematic for Waiter lay in the family $Z_{0,t,3}$, the next natural step is to bound the size of this family.

Claim 3.10.9. $|Z_{0,t,3}| \le 12\binom{t+3}{2}$ *holds for every integer* t.

Proof. Consider a graph $\Gamma \in Z_{0,t,3}$. Once again, we start the exploration of Γ with two triangles sharing an edge, xyz_1 and xyz_2. By Corollary 3.3.10 the remainder of the exploration goes as follows: t_1 additions of external triangles for some $0 \le t_1 \le t-1$, an addition of an internal triangle T_1 containing an existing edge e_1, the addition of $t_2 = t-1-t_1$ external triangles, and finally the addition of an internal triangle T_2 containing an existing edge e_2. Furthermore, for $i = 1, 2$, if $t_i > 0$ then e_i must be part of the last external triangle that was added before it by the restriction of the exploration process.

Given t_1, t_2, let us count the number of graphs that can be created in this manner (some of them may be not stable), where all the arguments are done with respect to isomorphism, and some of them provide upper bounds and not exact values. If $t_1 > 0$ then the first triangle in the chain contains either x or z_1; there is only one option for the external chain; there are two options for e_1 whether $t_1 = 0$ or not; if $t_2 > 0$ there are $5+2t_1$ options for the existing vertex contained in the external chain, and then one option for the chain itself and by stability one for e_2; if $t_2 = 0$ there are $7+3t_1$ options for e_2. All in all we can bound from above the number of such graphs by $4(7+3t_1) \le 12(t_1+3)$. Hence we get

$$|Z_{0,t,3}| \le \sum_{t_1=0}^{t-1} 12(t_1+3) \le 12 \sum_{t_1=0}^{t+2} t_1 = 12\binom{t+3}{2}.$$

□

We are now ready to prove Waiter's side. The discussion and claims above show that Waiter wins the game if G^* contains no components of $Z_{0,t,3}$. For every $\Gamma \in Z_{0,t,3}$ we get by Equation (3.3.1) in the proof of Claim 3.3.12 that $\Pr[\Gamma \subseteq G] \le a^{e(\Gamma)} = a^{3t+6}$. Using the fact that $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$ for any $|x| < 1$ we obtain that

$$\sum_{n=0}^{\infty} \binom{n}{2} x^n = \frac{1}{2}x^2 \sum_{n=0}^{\infty} n(n-1)x^{n-2} = \frac{1}{2}x^2 \frac{d^2}{dx^2}\left(\sum_{n=0}^{\infty} x^n\right) = \frac{1}{2}x^2 \frac{d^2}{dx^2}\left(\frac{1}{1-x}\right) = \frac{x^2}{(1-x)^3}$$

for every such x.

For any $a < 1$ the probability that G^* contains a component $\Gamma \in Z_{0,t,3}$ for some t can therefore be bounded from above by

$$\sum_{t=1}^{\infty} \sum_{\Gamma \in Z_{0,t,3}} \Pr[\Gamma \subseteq G] \le \sum_{t=1}^{\infty} 12\binom{t+3}{2} a^{3t+6} \le 12 \sum_{t=4}^{\infty} \binom{t}{2} a^{3t-3} \le 12 \sum_{t=0}^{\infty} \binom{t}{2} a^{t} = \frac{12a^2}{(1-a)^3}.$$

For $a = o(1)$ it follows that Waiter wins the game with probability at least $1 - \frac{12a^2}{(1-a)^3} = 1 - o(1)$, and thus Claim (1) of the theorem holds.

For every constant a such that $\frac{12a^2}{(1-a)^3} < 1$ (it is easy to see that the inequality holds for every $a \le 0.2$, for example), we obtain that the probability Waiter wins is bounded away from 0, thus Claim (2) of the theorem holds.

Since Client claims half the vertices of G by the end of the game no matter how he plays, Claim (4) of the theorem trivially holds by Theorem 3.1.5.

It remains to prove Claim (3) of the theorem. For this we describe a strictly balanced fixed graph with density 3/2, denoted by DDD (a *triple diamond*), and show that if G contains a copy of this graph Client wins the game. The desired result then follows by Theorem 3.2.13.

Before getting to the triple diamond, we first make some observations and introduce new terminology. For any triangle xyz in G, if at any point during the game Client claims the vertex x while the vertices y and z have not been offered yet, we say that the pair $\{y, z\}$ is *forced* on Waiter, because if this pair will not be offered later in the game, then at the moment the first vertex from the pair is offered Client can abandon any other strategy, claim it, and then whenever the other vertex in the pair is offered claim it as well, and win.

Let D be a diamond on the vertex set $\{x, y, z, w\}$, i.e., a copy of K_4 on this vertex set minus the edge zw. We say that D is a *winning diamond* if the vertices $\{y, z, w\}$ have not been offered yet in the game, and in addition either x was already claimed by Client, or that a pair $\{x, u\}$ is forced on Waiter for some vertex $u \notin \{y, z, w\}$. It is easy to see that the existence of such a diamond guarantees Client's win in the game: assuming the pair $\{x, u\}$ will be offered, Client can claim the vertices x, y and either z or w, as y cannot be offered in a pair with both of them. We now define a triple diamond and show that if G contains a copy of it then Client has a winning strategy in the game.

Let $H \subseteq G$ be a triple diamond composed of the three diamonds D_L, D_M and D_R, with vertex sets $V_L = \{x_1, y_1, z_1, z_2\}$, $V_M = \{x_1, x_2, w_1, w_2\}$, $V_R = \{x_2, y_2, z_3, z_4\}$, respectively (see Figure 3.7). Client's strategy goes as follows. He plays arbitrarily whenever the pair offered to him is disjoint from $V(H)$. Let $\{v, u\}$ denote the first pair offered to him which intersects $V(H)$. Client then plays according to these four cases (all other options are isomorphic to those described here).

1. If $v = x_1$ Client claims v and then either D_L or D_M is a winning diamond (since u belongs to at most one of them).

2. Otherwise, if $v = y_1$ Client claims v. If $u \notin V_L$ then D_L is a winning diamond. If $u \in V_L$, then $u = z_1$, and thus the pair $\{x_1, z_2\}$ is forced on Waiter, and so D_M is a winning diamond.

3. Otherwise, if $v = w_1$ Client claims v, and the pair $\{x_1, x_2\}$ is then forced on Waiter. This means that either D_L or D_R is a winning diamond (since u belongs to at most one of them).

4. Otherwise, $v = z_1$ and $u \notin V_M \cup \{y_1\}$. Client claims v, forces the pair $\{x_1, y_1\}$ on Waiter and D_M becomes a winning diamond. □

3.11 Concluding Remarks and Open Problems

The main open problem raised in this chapter is proving Breaker's side in Conjecture 3.1.2. In the previous sections we provided two families of graphs for which Breaker's side of the conjecture holds. Our method was to first apply the general deletion algorithm that breaks down G into small components with limitations on their structure, and then preform some case analysis on these components. We believe that the same can be done for any given graph H, although the case analysis might be very exhausting. We also believe that the deletion algorithm, as described here, could be helpful in proving the conjecture in its general form. However, the specific case analysis should

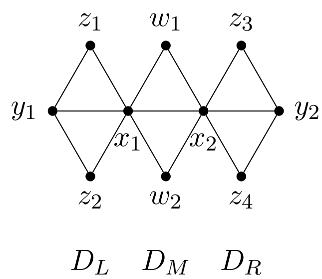

Figure 3.7: Triple Diamond

be replaced by some general argument about the possible structures of the surviving connected components, if there are any.

Another question concerns the sharpness of the threshold probabilities of H-games. We know that whenever Maker "wins locally", i.e. wins as soon as some fixed graph appears in G, the threshold is coarse. This is case for the unbiased triangle game and for forest games, which are not covered by Conjecture 3.1.2. From the discussion at the end of Section 3.4, we have that for any integer b and every $\varepsilon > 0$ there exists a constant $k_0 := k_0(b, \varepsilon)$ such that for any $k \geq k_0$, when playing the $(1 : b)$ K_k-game on the vertex set of $G \sim G(n, p)$, Maker wins w.h.p. for $p \geq (1+\varepsilon)n^{-2/k}$, and Breaker wins w.h.p. for $p \leq (1 - \varepsilon)n^{-2/k}$. Of course, ideally we could switch the order of the quantifiers in the above statement to establish a sharp threshold. More formally, we would like to know the following.

Question 4. Is there a constant k_0 such that for every $k \geq k_0$ the $(1 : b)$ Maker-Breaker K_k-game has a sharp threshold at $p = n^{-2/k}$? If so, does k_0 equal 4? Does the same hold for $(1 : b)$ C_k-games as well? Which other H-games has a sharp threshold at $p = n^{-1/m_1(H)}$?

We conclude this section with a short discussion about the new type of positional games introduced in this chapter. That is, given a graph G, the two players claim vertices of G, where the outcome of the game is determined by the subgraph of G induced by the vertices claimed by one of them. We considered H-games, and it would be interesting to investigate other classical games such as the k-connectivity, the perfect-matching and the Hamiltonicity games. Note that if the graph properties we consider are spanning, like in these examples, then the whole nature of the game changes dramatically in comparison to the edge version of these games. To begin with, the very meaning of the word "spanning" is different. For instance, in the Maker-Breaker Hamiltonicity game, Maker's goal is to claim a set $U \subset V(G)$ such that $G[U]$ is Hamiltonian, and not the impossible goal of claiming a Hamilton cycle of G. For the same reason, these games are not bias monotone – claiming more vertices can harm both Maker and Breaker. Another difference of this sort is the fact that for any given monotone increasing graph property $\mathcal{P}$, when playing on the edge set of a graph G, the family of target sets $\mathcal{T}$ (either winning or losing) is closed upwards, that is, if $E(G_1) \in \mathcal{T}$ and $G_1 \subseteq G_2 \subseteq G$, then $E(G_2) \in \mathcal{T}$. Clearly, this is not the case when playing on the vertex set of G and $\mathcal{P}$ is a spanning graph property: considering the Hamiltonicity example again, if $U \subseteq W \subseteq V(G)$, there is no relation whatsoever between the Hamiltonicity of $G[U]$ and that of $G[W]$. Considering all this, it is not clear whether the behavior of these games is analogous in some sense to the behavior of their respective edge version games, like we have shown in this chapter for H-games.

Acknowledgements

We wish to thank Michael Krivelevich for suggesting the new setting of positional games introduced in this chapter and for helpful discussions.

Bibliography

[1] J. Beck, **Combinatorial Games: Tic-Tac-Toe Theory (Encyclopedia of Mathematics and its Applications)**, Cambridge University Press, (2011).

[2] J. Beck, “Positional games and the second moment method”, Combinatorica 22.2 (2002), 169–216.

[3] M. Bednarska-Bzdęga, “On weight function methods in Chooser-Picker games”, Theoretical Computer Science 475 (2013), 21–33.

[4] M. Bednarska-Bzdęga, D. Hefetz, M. Krivelevich, and T. Łuczak, “Manipulative waiters with probabilistic intuition”, Combinatorics, Probability and Computing 25.6 (2016), 823–849.

[5] M. Bednarska-Bzdęga, D. Hefetz, and T. Łuczak, “Picker-Chooser fixed graph games”, Journal of Combinatorial Theory Series B 119 (2016), 122–154.

[6] M. Bednarska and T. Łuczak, “Biased positional games for which random strategies are nearly optimal”, Combinatorica 20 (2000), 477–488.

[7] M. Bednarska and O. Pikhurko, “Odd and even cycles in Maker–Breaker games”, European Journal of Combinatorics 29.3 (2008), 742–745.

[8] M. Bednarska, and O. Pikhurko, “Biased positional games on matroids”, European Journal of Combinatorics 26.2 (2005), 271–285.

[9] S. Ben-Shimon, A. Ferber, D. Hefetz, and M. Krivelevich, “Hitting time results for Maker-Breaker games”, Random Structures and Algorithms 41 (2012), 23–46.

[10] B. Bollobás, **Random graphs**, 2^{nd} ed. Cambridge University Press, (2001).

[11] B. Bollobás and A. Thomason, “Threshold functions", Combinatorica 7 (1987), 35-–38.

[12] V. Chvátal and P. Erdős, “Biased positional games”, Annals of Discrete Mathematics 2 (1978), 221–228.

[13] D. Clemens, A. Ferber, M. Krivelevich, and A. Liebenau, “Fast Strategies In Maker-Breaker Games Played on Random Boards”, Combinatorics, Probability and Computing 21 (2012), 897–915.

[14] D. Clemens, J. Ehrenmüller, Y. Person, and T. Tran, “Keeping Avoider’s graph almost acyclic”, available in arXiv, arXiv:1403.1482.

[15] J. Corsten, A. Mond, A. Pokrovskiy, C. Spiegel, and T. Szabó, “On the odd cycle game”, Preprint.

[16] O. Dean and M. Krivelevich, “Client-Waiter games on complete and random graphs”, Electronic Journal of Combinatorics 23 (2016), P4.38.

[17] L. Espig, A. Frieze, M. Krivelevich, and W. Pegden, “Walker-Breaker games”, SIAM Journal on Discrete Mathematics 29.3 (2015), 1476–1485.

[18] A. Ferber, R. Glebov, M. Krivelevich, and A. Naor, “Biased games on random graphs”, Random Structures and Algorithms 46 (2015), 651–676.

[19] A. Ferber, D. Hefetz, and M. Krivelevich. “Fast embedding of spanning trees in biased Maker–Breaker games”, European Journal of Combinatorics 33.6 (2012), 1086–1099.

[20] A. Ferber, M. Krivelevich, and A. Naor, “Avoider-Enforcer games played on edge disjoint hypergraphs”, Discrete Mathematics 313 (2013), 2932–2941.

[21] A. Frieze and M. Karoński, **Introduction to random graphs**. Cambridge University Press, (2015).

[22] H. Gebauer and T. Szabó, “Asymptotic random graph intuition for the biased connectivity game", Random Structures and Algorithms 35 (2009), 431—443.

[23] A. W. Hales, and R. I. Jewett, “Regularity and positional games”, Classic Papers in Combinatorics. Birkhäuser Boston (2009), 320–327.

[24] Y. O. Hamidoune and M. Las Vergnas, “A solution to the box game”, Discrete Mathematics 65 (1987), 157–171.

[25] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, “Avoider-Enforcer: The rules of the game”, Journal of Combinatorial Theory Series A 117 (2010), 152–163.

[26] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, “Fast winning strategies in Maker-Breaker games”, Journal of Combinatorial Theory Series B 99 (2009), 39–47.

[27] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, “Planarity, colorability, and minor games”, SIAM Journal on Discrete Mathematics 22.1 (2008), 194–212.

[28] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, **Positional Games (Oberwolfach Seminars)**, Birkhäuser, (2014).

[29] D. Hefetz, M. Krivelevich, and T. Szabó, “Avoider-Enforcer games”, Journal of Combinatorial Theory Series A 114 (2007), 840–853.

[30] D. Hefetz, M. Krivelevich, and W. E. Tan, “Waiter-Client and Client-Waiter Hamiltonicity games on random graphs”, European Journal of Combinatorics 63 (2017), 26–43.

[31] D. Hefetz, M. Krivelevich, and W. E. Tan, “Waiter–Client and Client–Waiter planarity, colorability and minor games”, Discrete Mathematics 339.5 (2016), 1525–1536.

[32] D. Hefetz and S. Stich, "On two problems regarding the Hamilton cycle game", The Electronic Journal of Combinatorics, Vol 16 (1) (2009), R28.

[33] S. Janson, T. Łuczak, and A. Ruciński, **Random graphs**, John Wiley & Sons (2000).

[34] M. Krivelevich, "Long cycles in locally expanding graphs, with applications", Combinatorica (2017), 1–17.

[35] M. Krivelevich, "The critical bias for the Hamiltonicity game is $(1 + o(1))n/\ln n$", Journal of the American Mathematical Society 24.1 (2011), 125–131.

[36] G. Kronenberg, A. Mond, and A. Naor, "Positional games on the vertex set of random graphs", Preprint.

[37] A. Lehman, "A solution of the Shannon switching game", Journal of the Society for Industrial and Applied Mathematics 12.4 (1964), 687–725.

[38] T. Łuczak, A. Ruciński, and B. Voigt, "Ramsey properties of random graphs", Journal of Combinatorial Theory, Series B 56 (1992), 55–68.

[39] T. Müller and M. Stojaković, "A threshold for the Maker-Breaker clique game" Random structures and algorithms 45 (2014), 318–341.

[40] R. Nenadov, A. Steger, and M. Stojaković, "On the threshold for the Maker-Breaker H-game", Random Structures and Algorithms 49 (2016), 558–578.

[41] V. Rödl and A. Ruciński, "Lower bounds on probability thresholds for Ramsey properties", In Combinatorics, Paul Erdős is eighty, Vol. 1, Bolyai Soc. Math. Stud. (1993), pp. 317–346.

[42] V. Rödl and A. Ruciński, "Random graphs with monochromatic triangles in every edge coloring", Random Structures Algorithms 5 (1994), 253–270.

[43] V. Rödl and A. Ruciński, "Threshold functions for Ramsey properties", J. Amer. Math. Soc. 8 (1995), 917–942.

[44] M. Stojaković, "Games on Graphs", PhD thesis, ETH Zurich, Switzerland, (2005).

[45] M. Stojaković and T. Szabó, "Positional games on random graphs", Random Structures and Algorithms 26 (2005), 204–223.

[46] D. B. West, **Introduction to Graph Theory**, 2^{nd} ed. Prentice Hall, (2001).