Doctoral Thesis

Aspects of games on random graphs

Author(s):
Thomas, Henning

Publication Date:
2013

Permanent Link:
https://doi.org/10.3929/ethz-a-009921594

Rights / License:
In Copyright - Non-Commercial Use Permitted
Aspects of Games on Random Graphs

A dissertation submitted to
ETH ZURICH

for the degree of
DOCTOR OF SCIENCES

presented by
HENNING THOMAS
Master of Science ETH in Computer Science
born 02.06.1985
citizen of Germany

accepted on the recommendation of
Prof. Dr. Angelika Steger, examiner
Prof. Dr. Benjamin Doerr, co-examiner

2013
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Zusammenfassung</td>
<td>vii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>xi</td>
</tr>
<tr>
<td><strong>1 Introduction and Results</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Our Results</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Organization of This Thesis</td>
<td>6</td>
</tr>
<tr>
<td><strong>2 Notation</strong></td>
<td>7</td>
</tr>
<tr>
<td>2.1 Graphs and Hypergraphs</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Asymptotic Notation</td>
<td>9</td>
</tr>
<tr>
<td><strong>3 Background and Related Work</strong></td>
<td>11</td>
</tr>
<tr>
<td>3.1 Random Graphs</td>
<td>11</td>
</tr>
<tr>
<td>3.2 Games in Discrete Mathematics</td>
<td>15</td>
</tr>
<tr>
<td>3.3 Extremal Combinatorics</td>
<td>18</td>
</tr>
<tr>
<td>3.4 Ramsey Games on Random Graphs and Achlioptas Processes</td>
<td>20</td>
</tr>
<tr>
<td><strong>4 Offline Ramsey Games with Giants</strong></td>
<td>31</td>
</tr>
</tbody>
</table>
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>31</td>
</tr>
<tr>
<td>4.2 Proof of Main Result</td>
<td>34</td>
</tr>
<tr>
<td><strong>5 Explosive Percolation in Random Graph Processes</strong></td>
<td>39</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>40</td>
</tr>
<tr>
<td>5.2 Our Results</td>
<td>42</td>
</tr>
<tr>
<td>5.3 Proof for the Half-Restricted Process</td>
<td>44</td>
</tr>
<tr>
<td>5.4 Proof for the Component Process</td>
<td>49</td>
</tr>
<tr>
<td>5.5 Proof for the Mixed Process</td>
<td>52</td>
</tr>
<tr>
<td><strong>6 Ramsey Properties of Random Hypergraphs</strong></td>
<td>55</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>55</td>
</tr>
<tr>
<td>6.2 Proof of Main Result</td>
<td>59</td>
</tr>
<tr>
<td>6.3 The Grow Sequences Approach</td>
<td>61</td>
</tr>
<tr>
<td>6.4 A Large Class of Ramsey-Density-Obeying Hypergraphs</td>
<td>76</td>
</tr>
<tr>
<td>6.5 Exception structures</td>
<td>86</td>
</tr>
<tr>
<td>6.6 Open Problems</td>
<td>91</td>
</tr>
<tr>
<td><strong>7 Playing Mastermind with Many Colors</strong></td>
<td>95</td>
</tr>
<tr>
<td>7.1 Introduction</td>
<td>95</td>
</tr>
<tr>
<td>7.2 The $O(n \log \log n)$ Adaptive Strategy</td>
<td>100</td>
</tr>
<tr>
<td>7.3 Mastermind with Black and White Answer-Pegs</td>
<td>110</td>
</tr>
<tr>
<td>7.4 Non-Adaptive Strategies</td>
<td>116</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>127</td>
</tr>
<tr>
<td><strong>Curriculum Vitae</strong></td>
<td>141</td>
</tr>
</tbody>
</table>
Abstract

In this thesis we deal with games in the intersection of extremal combinatorics and random graph theory which gained a lot of attention over the past decades. Consider the following one-player game. Starting with an empty graph on \( n \) vertices the player, called Painter, is consecutively presented random edges which are inserted into the graph and which she needs to color immediately with one of \( r \) available colors. Her goal is to avoid a certain graph property, e.g. containing a monochromatic triangle, for as long as possible. How long can she typically survive in this game? And how does an optimal strategy for her look like? These so-called online Ramsey games were first studied by Friedgut, Kohayakawa, Rödl, Ruciński and Tetali in 2003 and a large body of research considering plenty of goals for Painter and plenty of variations of the game has been developed.

A particularly popular variant is given by the so-called Achlioptas processes named after Dimitris Achlioptas. Here, the player, called Chooser, is presented random pairs of possible edges, from each of which she needs to choose one to be inserted into the graph while the other is put back into the pool of non-edges. These processes were originally studied regarding the question if one can accelerate or delay the appearance of a linear-sized component, a so-called giant component, in comparison to
the classical random graph process by Erdős and Rényi, which simply inserts a random edge in every step without any choices to be made. The appearance of the giant component is also called phase transition.

Note that all these online games, where the player has to decide without any knowledge about future steps, have an offline counterpart, where she is presented all decisions (that is e.g. all random edges) at once. Here, the question is not how long she can typically survive, but how many decisions we can typically reveal to her such that she can still achieve her goal. Observe that any online strategy can only be as good as an optimal offline strategy. In this sense, offline settings provide natural benchmarks for the player in online games. It is thus essential to understand the underlying offline scenarios in order to evaluate the quality of online strategies.

Our first main result deals with offline Ramsey games where Painter’s goal is to avoid a monochromatic giant component. This goal was considered for the offline variant of Achlioptas processes by Bohman and Kim in 2006. Moreover, Bohman, Frieze and Wormald in 2004 solved the setting where the edges do not come in pairs and Chooser can instead select any half of them without further restrictions. We consider the two corresponding Ramsey-type settings: an unbalanced one where Painter is presented a random graph and needs to color its edges with \( r \geq 2 \) colors without creating a monochromatic giant component, and a balanced one where all edges are additionally partitioned into random sets of size \( r \) and Painter is restricted to use each of the \( r \) colors exactly once in every set. We show that, in contrast to the edge-selection scenarios where the thresholds of the two settings differ, they coincide in the Ramsey scenario. Moreover, they are identical to the threshold for \( r \)-orientability.

Our second main result is concerned with the phase transition of random graph process. In a Science paper from 2009 Achlioptas, D’Souza and Spencer conjectured that certain edge-selection strategies for Chooser create Achlioptas processes with discontinuous phase transitions. However, this was disproved by Riordan and Warnke in a 2011 Science paper where they show that every Achlioptas process has a continuous phase transition. We complement their results by providing three examples for random graph processes which are closely related to the classical one by Erdős and Rényi, but exhibit a discontinuous phase transition.

Our third main result concerns a generalization of an offline Ramsey graph avoidance game to hypergraphs. In the 1990’ies Rödl and Ru-
ciński solved this problem completely for the simple graph case. For an arbitrary graph $F$ they prove a threshold result for the property that the edges of a random graph can be colored with $r \geq 2$ colors without a monochromatic copy of $F$. For the case of hypergraphs Friedgut, Rödl and Schacht (2010) and, independently, Conlon and Gowers (2011) showed an upper bound for the threshold of the avoidance of an arbitrary hypergraph $F$. In this thesis we prove a matching lower bound for a large class of hypergraphs, including all complete hypergraphs. Moreover, we give examples of hypergraphs for which the upper bound from Friedgut et al. and Conlon and Gowers is not tight.

Our fourth main result deals with the well-known two-player game of Mastermind. Roughly speaking, one player, called Codemaker, generates a secret code of length $n$ over an alphabet of size $k$ and the other player, called Codebreaker, has to identify this code with as few questions as possible according to a fixed question & answer scheme. The number of questions needed to identify the code has been studied intensively in discrete mathematics, e.g. by Erdős and Rényi for $k = 2$ in 1963 and by Chvátal for $k \leq n^2$ in 1983. Despite many efforts the currently-best known bounds for the prominent case $k = n$ are a lower bound of $\Omega(n)$ questions and an upper bound of $O(n \log n)$. Among other results we improve the upper bound for this case to $O(n \log \log n)$. 

Eine besonders populäre Variante sind die so genannten \textit{Achlioptas-Prozesse}, die nach Dimitris Achlioptas benannt sind. Dabei werden dem Spieler, den wir hier \textit{Auswähler} nennen, fortlaufend zufällige Paare von möglichen Kanten präsentiert, von denen er jeweils eine auswählen muss, die dem Graphen hinzugefügt wird, während die andere in den Pool
Zusammenfassung


All diese Online-Spiele, in denen der Spieler sofort entscheiden muss ohne Vorwissen über in der Zukunft liegende Schritte, besitzen ein Offline-Pendant, bei dem ihm alle Entscheidungen (also z.B. alle zufälligen Kanten) auf einmal präsentiert werden. Hierbei ist die Frage nicht, wie lange der Spieler typischerweise überleben kann, sondern wie viele Entscheidungen wir ihm im Normalfall zeigen können, so dass er sein Ziel immernoch erreichen kann. Dabei kann jede Online-Strategie höchstens so gut sein wie eine optimale Offline-Strategie. Die Offline-Szenarien dienen also als eine Art natürliche Messlatte für Online-Spiele. Daher ist es von entscheidender Bedeutung die zugrunde liegenden Offline-Varianten zu verstehen, um eine Aussage über die Güte von Online-Strategien treffen zu können.


In unserem zweiten Hauptresultat betrachten wir den Phasenübergang von Zufallsgraphenprozessen. In 2009 haben Achlioptas, D’Souza und


In unserem vierten Hauptresultat geht es um das bekannte Gesellschaftsspiel Mastermind. Grob gesagt gibt es dort einen Spieler, den Codemaker, der ein geheimes Wort der Länge $n$ über einem Alphabet der Größe $k$ wählt, und den anderen Spieler, Codeknacker, der dieses Wort in einem vorgegebenen Frage-Antwort-Schema mit möglichst wenig Fragen herausfinden muss. Die Anzahl Fragen, die in diesem Spiel benötigt wird, um das Wort zu erraten, wurde in der diskreten Mathematik schon vielfach analysiert, z.B. 1963 von Erdős und Rényi für den Fall $k = 2$ und 1983 von Chvátal für $k \leq n^2$. Trotz vieler Bemühungen sind die besten derzeit bekannten Schranken für den Fall $k = n$ eine untere Schranke von $\Omega(n)$ und eine obere Schranke von $\mathcal{O}(n \log n)$. Neben anderen Resultaten verbessern wir hier die obere Schranke auf $\mathcal{O}(n \log \log n)$. 
Zusammenfassung
I want to start by expressing my sincere gratitude to my supervisor Angelika Steger. She gave me the opportunity to pursue research projects in her group at ETH Zurich. I really enjoyed the atmosphere in the group which, to a high degree due to her thoughtful guidance, I experienced as very open and inspiring. I am very thankful for the discussions with her about research, teaching and other topics and have always appreciated her open door and advice. Moreover, I am very grateful to her for encouraging me to work with other co-authors.

I want to thank Benjamin Doerr not only for agreeing to co-referee this thesis, but also for the collaboration that resulted from my stay in Saarbrücken. I greatly benefitted from working with him and really enjoyed my time in Saarland. Moreover, I do not want to forget to mention the good times at Poznan’s central square.

Furthermore, I want to thank all my co-authors: Reto Spöhel, Konstantinos Panagiotou, Yury Person, Torsten Mütze, Luca Gugelmann, Carola Doerr, and Anupam Prakash.

One of the most enjoyable parts of my PhD studies was the friendly atmosphere in the CSA and CADMO group. I want to thank all current and previous members who I met on my way: Thomas Rast, Florian Jug, Christoph Krautz, Ueli Peter, Nicla Bernasconi, Dan Hefetz, Jo-
hannes Lengler, Rajko Nenadov, Hafsteinn Einarsson, Emo Welzl, Peter Widmayer, Dominik Scheder, Yann Disser, Andrea Francke, Robin Moser, Andreas Razen, Vincent Kusters and, last but not least, Marianna Berger, Andrea Salow and Denise Spicher for helping me with administrative matters. Furthermore, I would like to thank the staff from Hotel Kreuz in Buchboden for four very nice research retreats.

My work was supported by a grant from the Swiss National Science Foundation, which I gratefully acknowledge.

In particular, I want to thank Reto for many helpful discussions, for being a great host in Saarbrücken and for teaching me that the first iteration of a write-up can usually be improved; Kosta for introducing me to “temple run”; Yury for being a great host for Luca and me in Berlin while everybody else fled from us; Torsten for letting me help him move; Luca for joining me on almost every conference trip; Carola for preparing me for Zurich with her grandmother’s recipe of cheese fondue and not being mad about tiny spots of red wine on her carpet; Thomas for being a great office mate and solving all sorts of computer problems by typing fast in a console; Flo (Wer?) for riding cows with me, and for showing me that even when I get older I can still make puns; Chrissie for awesome holidays in New Zealand and for being my Hot Pasta breakfast buddy; Ueli for being my workout and table tennis partner in Buchboden; Nicla for being a good host for “Bang!”; Dan for “shooting Nicla” (to avoid any confusion, this refers to the aforementioned game); Johannes for being the math encyclopedia; Rajko for giving me a cangaroo that supports the warm-up of my volleyball team; Hafi for keeping up the tradition of serving coffee during exams although it comes with unwanted side effects; Dominik for proofreading parts of this thesis, singing intervals in front of the coffee machine with me and for being a great roommate and so; Yann for having a strong opinion on everything; Andrea for selling me her bike and for organizing a beautiful two-day hike through the Alpstein; Robin for broadening my horizon concerning the Zurich nightlife; Andi Razen for joining me in this experience; and Vincent for his humor and for reviving the poker tradition.

Last, but not least I want to thank everyone outside of ETH who provided me with the necessary balance to the work hours: my family for being fifty-one and for offering me “chill palaces” in Kiel, Aachen, Ratisbon and Hamburg at any time; everyone from the Ireland and Croatia crew for great holidays; especially Heike for not being rude to me in Cesenatico for 8 hours and for calling this work a bunch of mind farts.
(I am just waiting for you to finish your thesis!); Nadine and Simon for being perfect roommates; everyone from ETH Bigband for three awesome tours, countless drinks at Hot Pasta and many live experiences of random processes involving 20 musicians (cf. [ETHB12]); in particular Christoph Eck for giving us this opportunity and for his moving speech at the apéro after my defense; everyone from Herren 1 and Damen 1 of VBC Oerlikon and everyone from ASVZ volleyball for providing a great counterpart to sitting at the desk all day long.
Playing games has always been a central part of humans’ social interaction. Researchers from various subjects have contributed to the scientific analysis of a great variety of games and have made game theory a fascinating and vivid research area. Their motivation is diverse and can stem from economical, political, psychological, biological or pure mathematical interest. Classical game theory had an extensive development in the 1950ies based on work of von Neumann and Morgenstern [MvN44] and Nash [Nas51] although the origins of the field date back at least to the late 1920ies [vN28]. These authors are mainly concerned with so-called games of imperfect information. In these games players do not have full information about the current state of the game or the other player’s moves. This is for example the case in most card games where each player’s hand cards are hidden from the other players or for games where players move simultaneously. One contribution of this thesis is an
The main contribution of this thesis is studying various aspects of games in combination with random graph theory. Before stating our results in Section 1.1 we briefly introduce the underlying concepts. For a more detailed exposition of the background and related results we refer to Chapter 3.

The study of random graphs was initiated by the seminal work of Paul Erdős and Alfréd Rényi in the late 1950ies \cite{ER59,ER60}. It is motivated by the question how a ‘typical’ graph with a given density looks like. More precisely, Erdős and Rényi consider, for given parameters $n$ and $m = m(n)$ a graph $G(n,m)$ chosen uniformly at random from all graphs on $n$ vertices with $m$ edges. In the context of this thesis it is convenient to consider random graphs from the following process perspective. Starting with an empty graph on $n$ vertices we insert random edges one by one into the graph. It is not hard to see that the graph after $m$ edge insertions is distributed as $G(n,m)$. How long does the process ‘typically’ take until it has a given property, e.g. is connected or contains a triangle? As it turns out, many natural graph properties exhibit a so-called threshold behavior. Roughly speaking, this means that there exists a critical time $m_0 = m_0(n)$ in the process such that before $m_0$ the graph of the process is unlikely to have the property while after $m_0$ it has the property with high probability. One of the first results of this type \cite{ER60} states that the component structure in the random graph process undergoes a phase transition at $n/2$. Around this point a connected component of linear size, a so-called giant component, emerges.

Motivated by the question if one can delay or accelerate this phase transition, the following one-player-game that introduces a freedom of choice into the random graph process gained a lot of attention over the last decade. Starting with an empty graph the player, called Chooser, is presented a pair of possible edges in every step from which she immediately has to select one to be included in the graph while the other is put back into the pool of possible edges. Such a process is called Achlioptas process after Dimitris Achlioptas and it has been shown that using appropriate edge-selection strategies Chooser can indeed accelerate or delay the phase transition by a constant factor \cite{BF01,BK06b,SW07}. Furthermore, these processes became of great interest also to the physics community when Achlioptas, D’Souza and Spencer \cite{ADS09} provided
numerical evidence that certain strategies for Chooser to delay the giant component cause a \textit{discontinuous phase transition}. They conjectured that at the critical time when a linear-size component first appears it ‘almost instantly’ (in a sublinear number of steps) accumulates a constant fraction of the vertices, in contrast to the \textit{continuous} phase transition of the classical random graph process where the largest component grows ‘smoothly’. However, the conjecture was recently disproven by Riordan and Warnke [RW11].

Another way to introduce freedom into the random graph process is motivated by Ramsey theory. Starting with an empty graph we insert random edges one by one into the graph and the player, called \textit{Painter}, needs to color them immediately on insertion with one of $r$ available colors. Her goal is to avoid a given property, e.g. containing a monochromatic copy of a fixed given subgraph, for as long as possible or to achieve it as quickly as possible. What is the typical duration of the game if Painter plays optimally? These games are called \textit{Ramsey games} and were first studied by Friedgut et al. [FKR+03] who show that if Painter’s goal is to avoid a monochromatic triangle the duration of the game (with Painter playing optimally) has a threshold at $N_0(n) = n^{4/3}$. More precisely, a simple greedy strategy ensures Painter to survive $o(n^{4/3})$ steps with high probability, while every strategy fails with high probability to survive $\omega(n^{4/3})$ steps. There is a large body of research around these games and we provide an overview of the state of the art in Section 3.4. Here, we only mention that the corresponding offline variant in which all edges of a random graph $G(n,m)$ are presented to Painter at once has been solved in full generality by Rödl and Ruciński [RR93, RR95]. For every graph $F$ and every number $r \geq 2$ of colors they determined a threshold $m_0 = m_0(F,n,r)$ such that for every $m = o(m_0)$ there exists, with high probability, an edge-coloring of $G(n,m)$ with $r$ colors without a monochromatic copy of $F$, while for every $m = \omega(m_0)$ such an edge-coloring does not exist with high probability. Note that the analysis of offline games is essentail for the study of online settings since it provides suitable benchmarks. The strength of any online strategy should always be measured by a comparison to its offline counterpart.
1.1 Our Results

1.1.1 Offline Ramsey Games with Giants

As mentioned before Chooser can delay or accelerate the appearance of a linear-sized component in an Achlioptas process. An offline version, in which she is presented all pairs of edges at once, was solved by Bohman and Kim [BK06a]. In [BFW04] Bohman, Frieze and Wormald prove a threshold result for the variant in which Chooser is not presented pairs of edges but simply all possible edges from which she needs to select half. Their results show that the thresholds of both settings differ. Hence, loosening the restriction to select from predefined pairs substantially enhances Chooser’s power to avoid a giant component.

Our first main result concerns the corresponding Ramsey-type settings. That is, we consider the unbalanced and balanced offline Ramsey game in which Painter’s goal is to avoid a monochromatic giant component for as long as possible. In the former Painter is presented all edges of a random graph at once and she needs to find a coloring with a given number $r \geq 2$ of colors without a monochromatic giant component. In the latter one all edges are additionally partitioned into sets of size $r$ and Painter is restricted to use each of the $r$ colors exactly once in every set. We show that, in contrast to the edge-selection scenarios, the thresholds for both settings coincide. Moreover, they are identical to the threshold for $r$-orientability, where a graph is called $r$-orientable if there exists an orientation of the edges such that every vertex has in-degree at most $r$. This result is joint work with Reto Spöhel and Angelika Steger. We present the details in Chapter 4. The unbalanced setting was independently solved by Bohman et al. [BFK+11].

1.1.2 The Phase Transition of Random Graph Processes

Our second main result is related to the results of Riordan and Warnke [RW11] who show that regardless of Chooser’s edge-selection strategy Achlioptas processes have a continuous phase transition. We complement their results by providing three examples for random graph processes which are closely related to the classical one by Erdős and Rényi, but exhibit a discontinuous phase transition. All three start with the empty graph on
1.1. Our Results

$n$ vertices and, depending on the process, we connect in every step (i) one vertex chosen randomly from all vertices, and one chosen randomly from a restricted set of vertices, (ii) two components chosen randomly from the set of all components, or (iii) a randomly chosen vertex and a randomly chosen component.

This result is joint work with Konstantinos Panagiotou, Reto Spöhel and Angelika Steger. We present the details in Chapter 5.

1.1.3 Ramsey Properties of Random Hypergraphs

As mentioned above Rödl and Rucínski studied offline Ramsey properties of random graphs [RR93, RR95]. In this thesis we consider a generalization of this setting to random hypergraphs. This was first considered in [RR98] and recently Friedgut, Rödl and Schacht [FRS10] and independently Conlon and Gowers [CG] settled a conjecture on an upper bound for the threshold of the avoidance of an arbitrary hypergraph $F$. That is, they show that dense enough random hypergraphs do not allow for an $r$-hyperedge-coloring without a monochromatic copy of $F$.

Our third main result is a matching lower bound for a large class of hypergraphs $F$, that is, we show that for sparse enough random hypergraphs we can find an $r$-hyperedge-coloring without a monochromatic copy of $F$. Moreover, we give examples of hypergraphs for which the upper bound in [FRS10, CG] is not tight. However, we are not able to solve the problem in full generality and there are still many hypergraphs for which the threshold behavior is left open.

This result is joint work with Luca Gugelmann, Yury Person and Angelika Steger. We present the details in Chapter 6.

1.1.4 Playing Mastermind with Many Colors

Mastermind is a well-known two-player boardgame. One player, called Codemaker, generates a secret code of a given length $n$ over a given alphabet of size $k$ and the other player, called Codebreaker, has the goal to identify the secret code as quickly as possible. For this, he consecutively queries codes (of length $n$ and over the same alphabet) each of which is immediately answered by two integers, one denoting how many coordinates of the query coincide with the secret code, and the other denoting
how many coordinates could additionally coincide if Codebreaker used
the same characters in an optimal permutation. The game ends as soon
as Codebreaker identifies the secret code. A commercial version with
$k = 6$ and $n = 4$ where the first integer of the answer is denoted by
black answer-pegs and the second by white answer-pegs was launched in
the early 1970ies. The game has relations to information-theoretic ques-
tions and query complexity. Hence, many researchers have analyzed the
duration of the game for different code lengths and alphabet sizes when
Codebreaker uses an optimal strategy [ER63b, Knu77, Chv83, Goo09b],
some of the work even dating before Mastermind was publicly avail-
able as a boardgame. It turns out that if $k$ is significantly smaller than
$n$ a random guessing strategy is asymptotically optimal. However, for
the prominent case when $k = n$ the previously best-known results show
that Codebreaker needs at least $\Omega(n)$ queries and can succeed within
$O(n \log n)$.

Our fourth main result improves the upper bound for this case to
$O(n \log \log n)$. For this we analyze a strategy for Codebreaker which
only uses black pegs and which, in contrast to e.g. random guessing,
is adaptive, that is, it uses the answers to previous queries in order to
decide for the next query. Furthermore, we prove that an optimal non-
adaptive strategy, that is, one that simply goes through a predefined list
of queries until it can rule out all but one secret code, requires $\Theta(n \log n)$
steps. This shows that in the case $k = n$ adaptive strategies are more
powerful than non-adaptive ones, in contrast to when $k$ is significantly
smaller than $n$. We also improve the best previously-known bounds for
Mastermind with black and white answer-pegs for a wide range of $k$.

These results are joint work with Benjamin and Carola Doerr and Reto
Spöhel. We present the details in Chapter 7.

1.2 Organization of This Thesis

In Chapter 2 we provide a reference of notations that we will use through-
out this thesis. We then present the mathematical background and the
motivation for our results, as well as related work in Chapter 3. Chap-
ters 4 to 7 are each devoted to one of our four main results. They
contain their own brief introduction which presents the main aspects
that are important for the respective result.
In this chapter we introduce some basic notation which we will use throughout this thesis.

2.1 Graphs and Hypergraphs

Basic Notation

For a graphs (hypergraphs) $F, G$ we use the following notation.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V(G)$</td>
<td>vertex-set of $G$</td>
</tr>
<tr>
<td>$E(G)$</td>
<td>edge-set (hyperedge-set) of $G$</td>
</tr>
<tr>
<td>$v(G)$</td>
<td>number of vertices of $G$, $v(G) =</td>
</tr>
</tbody>
</table>
### Chapter 2. Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e(G)$</td>
<td>number of edges (hyperedges) of $G$, $e(G) =</td>
</tr>
<tr>
<td>$N(v), N_G(v)$</td>
<td>neighborhood of a vertex $v \in V(G)$, $N_G(v) = {u \in V(G) : {u, v} \in E(G)}$ for simple graphs</td>
</tr>
<tr>
<td>$\deg(v), \deg_G(v)$</td>
<td>degree of a vertex $v$ in $G$, $\deg_G(v) =</td>
</tr>
<tr>
<td>$\delta(G)$</td>
<td>minimum degree of $G$, $\delta(G) = \min_{v \in V(G)} \deg_G(v)$</td>
</tr>
<tr>
<td>$\Delta(G)$</td>
<td>maximum degree of $G$, $\Delta(G) = \max_{v \in V(G)} \deg_G(v)$</td>
</tr>
<tr>
<td>$\chi(G)$</td>
<td>(weak) chromatic number of $G$, i.e., the minimum number of colors such that the vertices of $G$ can be colored without creating a monochromatic hyperedge</td>
</tr>
<tr>
<td>$\text{lk}(v), \text{lk}_G(v)$</td>
<td>link of $v$, i.e., the $k-1$-uniform hypergraph induced by all hyperedges incident to $v$ where $v$ is removed, $\text{lk}_G(v) = (V, E)$ where $V = N_G(v)$ and $E = {e \setminus v : e \in E(G) \land v \in e}$</td>
</tr>
<tr>
<td>$G \to (F)^c_r$</td>
<td>$G$ is $F$-Ramsey for $r$ colors, i.e., every $r$-edge-coloring of $G$ contains a monochromatic copy of $F$</td>
</tr>
<tr>
<td>$G \to (F)^v_r$</td>
<td>$G$ is $F$-vertex-Ramsey for $r$ colors, i.e., every $r$-vertex-coloring of $G$ contains a monochromatic copy of $F$</td>
</tr>
</tbody>
</table>

### Distinguished Graphs and Hypergraphs

We use the following symbols for special graphs and hypergraphs.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_\ell$</td>
<td>complete graph on $\ell$ vertices</td>
</tr>
<tr>
<td>$P_\ell$</td>
<td>paths with $\ell$ edges</td>
</tr>
<tr>
<td>$C_\ell$</td>
<td>cycle on $\ell$ vertices</td>
</tr>
<tr>
<td>$S_\ell$</td>
<td>star with $\ell$ rays</td>
</tr>
<tr>
<td>$S^\odot_\ell$</td>
<td>sunshine graph with a cycle of length $\ell$ and $\ell$ rays</td>
</tr>
<tr>
<td>$K^{(k)}_\ell$</td>
<td>complete $k$-uniform hypergraph on $\ell$ vertices</td>
</tr>
<tr>
<td>$P^{(k)}_{\ell,t}$</td>
<td>$t$-tight $k$-uniform path with $\ell$ hyperedges</td>
</tr>
<tr>
<td>$C^{(k)}_{\ell,t}$</td>
<td>$t$-tight $k$-uniform cycle with $\ell$ hyperedges</td>
</tr>
</tbody>
</table>
2.2 Asymptotic Notation

We use the standard Landau symbols $O, \Omega, \Theta, o, \omega$ to describe the asymptotic behavior of functions. Furthermore we use the following abbreviations and symbols.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_{\ell,t}^{(k)} )</td>
<td>( k )-uniform star with ( \ell ) hyperedges and center of size ( t )</td>
</tr>
<tr>
<td>( S_{\ell,t}^{\geq k} )</td>
<td>( k )-uniform sunshine hypergraph with ( t )-tight cycle of ( \ell ) hyperedges and ( \ell ) rays</td>
</tr>
<tr>
<td>( G(n,p) )</td>
<td>random graph on ( n ) vertices with edge probability ( p )</td>
</tr>
<tr>
<td>( G(n,m) )</td>
<td>random graph on ( n ) vertices with ( m ) edges</td>
</tr>
<tr>
<td>( H^{(k)}(n,p) )</td>
<td>random ( k )-uniform hypergraph on ( n ) vertices with hyperedge probability ( p )</td>
</tr>
</tbody>
</table>

### Density Measures

Throughout the thesis we use the following density measures for graphs (hypergraphs) \( G \).

<table>
<thead>
<tr>
<th>Symbol &amp; definition</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d(G) = \frac{e(G)}{v(G)} )</td>
<td>(hyper)edge-density of ( G )</td>
</tr>
<tr>
<td>( m(G) = \max_{J \subseteq G} d(J) )</td>
<td>maximum (hyper)edge-density of ( G )</td>
</tr>
<tr>
<td>( d_k(G) = \frac{e(G)-1}{v(G)-k} )</td>
<td>( k )-(hyper)edge-density of ( G ) (only defined if ( v(G) &gt; k ))</td>
</tr>
<tr>
<td>( m_k(G) = \max_{J \subseteq G, v(G) &gt; k} d_k(J) )</td>
<td>maximum ( k )-(hyper)edge-density of ( G ), short ( k )-density of ( G ) (only defined if ( v(G) &gt; k ))</td>
</tr>
</tbody>
</table>

A (hyper)graph \( G \) is called balanced if \( d(G) = m(G) \) and strictly-balanced if additionally \( d(J) < d(G) \) for every \( J \subsetneq G \). Moreover, for every \( k \geq 2 \), \( G \) is \( k \)-balanced if \( d_k(G) = m_k(G) \) and strictly \( k \)-balanced if additionally \( d_k(J) < d_k(G) \) for every \( J \subsetneq G \).
## Chapter 2. Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.a.s., w.h.p.</td>
<td>asymptotically almost surely, with high probability, i.e., with probability tending to 1 as ( n ) tends to infinity</td>
</tr>
<tr>
<td>( f \ll g )</td>
<td>( f = o(g) )</td>
</tr>
<tr>
<td>( f \gg g )</td>
<td>( f = \omega(g) )</td>
</tr>
<tr>
<td>( f \asymp g )</td>
<td>( f = \Theta(g) )</td>
</tr>
</tbody>
</table>
Chapter 3

Background and Related Work

In this thesis we study game-type problems that combine extremal combinatorics with random graph theory. To illustrate the motivation behind our work as well as its relevance we give a brief history over previous and related results. We first provide introductions to the fields of random graphs, games in discrete mathematics and extremal combinatorics in Sections 3.1 to 3.3. In Section 3.4 we then give an overview of previous work in the intersection of these fields.

3.1 Random Graphs

The study of random graphs was initiated by the seminal work of Paul Erdős and Alfréd Rényi in the late 1950’s and 1960’s [ER59, ER60, ER61a, ER61b, ER63a, ER64, ER66, ER68] (see also [KR97]) and has
since then developed into a vivid research area with many beautiful and fascinating results. We refer to the monographs [JR00, Bol01] for an overview.

Roughly speaking, the field deals with the question how a ‘typical’ graph on \( n \) vertices with \( m \) edges looks like. More precisely, let \( G(n, m) \) denote a graph chosen uniformly at random among all graphs on \( n \) vertices with \( m \) edges. What is the probability of \( G(n, m) \) to have a certain property, e.g. to be connected, Hamiltonian or 3-colorable? The main attention is drawn to the asymptotic behavior of this probability. That is, what is the limit of this probability if the number of vertices \( n \) goes to infinity and the number of edges \( m = m(n) \) is a function depending on \( n \)?

It turns out that many natural graph properties \( \mathcal{P} \) have a so called \textit{threshold function} \( m_0 = m_0(n) \) such that

\[
\lim_{n \to \infty} \Pr [G(n, m) \in \mathcal{P}] = \begin{cases} 
0, & \text{if } m \ll m_0, \\
1, & \text{if } m \gg m_0.
\end{cases}
\]

In such a case one speaks of a \textit{weak threshold}. Furthermore, a threshold is called \textit{coarse} if for every constant \( c > 0 \) and \( m = cm_0 \) we have

\[
0 < \lim_{n \to \infty} \Pr [G(n, m) \in \mathcal{P}] < 1.
\]

A property \( \mathcal{P} \) has a \textit{sharp threshold}, if there exists \( m_0(n) \) such that for every constant \( \varepsilon > 0 \) we have

\[
\lim_{n \to \infty} \Pr [G(n, m) \in \mathcal{P}] = \begin{cases} 
0, & \text{if } m \leq (1 - \varepsilon)m_0, \\
1, & \text{if } m \geq (1 + \varepsilon)m_0.
\end{cases}
\]

Naturally, in both cases the roles of 0 and 1 can be exchanged. It is known [BT87] that all monotone graph properties have a threshold function, where a graph property is called monotone if it is closed under edge-insertion.

One of the first results of this form can be found in [ER60]. It states that the component structure of a random graph undergoes a drastic change around \( n/2 \) edges. More precisely, the property of containing a connected component of linear size has a sharp threshold at \( n/2 \) (see also Figure 3.1). This result builds the fundament of our studies in Chapters 4 and 5, see also Section 3.4.3. For every graph \( G \) let \( L_1(G) \) denote the size of its largest connected component.

**Theorem 3.1** ([ER60]). Let \( m_0 = n/2 \). Then for every \( \varepsilon > 0 \) we have
3.1. Random Graphs  

Figure 3.1: A simulation of the evolution of the largest component in a random graph with 20 million vertices

- if $m \leq (1 - \varepsilon)m_0$, then a.a.s. $L_1(G(n,m)) = \mathcal{O}(\log n)$, and
- if $m \geq (1 + \varepsilon)m_0$, then a.a.s. $L_1(G(n,m)) = \Omega(n)$, and all other components have $\mathcal{O}(\log n)$ vertices.

Observe that the theorem shows even more than just a sharp threshold behavior since it even bounds the size of the largest component by $\mathcal{O}(\log n)$ for $m \leq (1 - \varepsilon)m_0$. This result can be viewed from a random graph process perspective. Starting with the empty graph on $n$ vertices, we add a single edge chosen uniformly at random from all non-edges in every step. It is not hard to see that the graph after inserting $m$ edges is distributed as $G(n,m)$. In this context Theorem 3.1 states that, asymptotically, a linear-sized component (a so called ‘giant component’) appears around the time when we have inserted $n/2$ edges. We say that the random graph has a phase transition at $n/2$. Recently, a very simple and elegant proof of Theorem 3.1 was found by Michael Krivelevich and Benny Sudakov [KS]. The pioneering work of Erdős and Rényi has invoked a large body of research from which we want to mention two main branches here. A lot of work has been devoted to a better understanding of the behavior in the ‘threshold window’, that is, for functions $m$ that satisfy $m - n/2 = o(n)$. For an overview we refer to chapter 5 of [JLR00]. Another branch investigated how introducing a freedom of choice into the random graph process allows for a delay
or acceleration in the appearance of a giant component. The problem considered in Chapter 4 is motivated by this branch. We treat it in more detail in Section 3.4.3.

Another fundamental result in random graph theory which is related to our results in Chapter 6 is the so-called small subgraphs theorem by Béla Bollobás. Given a fixed graph \( F \), what is the probability that \( G(n, m) \) contains a copy of \( F \)? For special classes of graphs \( F \) a complete answer in form of a threshold result was already given in [ER60]. Later, Bollobás provided the generalization to arbitrary graphs \( F \).

**Theorem 3.2** ([Bol81]). Let \( F \) be a fixed non-empty graph. Then we have

\[
\lim_{n \to \infty} \Pr [G(n, m) \text{ contains a copy of } F] = \begin{cases} 0 & \text{if } m \ll n^{2-1/m(F)}, \\ 1 & \text{if } m \gg n^{2-1/m(F)}, \end{cases}
\]

where

\[
m(F) = \max_{J \subseteq F} d(J) \quad \text{and} \quad d(J) = \frac{e(J)}{v(J)} \text{ for every graph } J.
\]

Here, \( m(F) \) and \( d(F) \) denote the (maximum) edge-density of \( F \). There is a comparatively simple proof of this theorem using a first and second moment method argument. In Chapter 6 we use an extension of Theorem 3.2 to hypergraphs which can be shown by a straightforward generalization of this argument. We give the details in Chapter 6.

As a final remark in this short introduction to random graph theory, we point out that the \( G(n, m) \) random graph model has a major drawback. Note that the presence of two different edges is not independent. In many analyses this is a nasty technical obstacle. Hence, many proofs make use of the independent random graph model \( G(n, p) \) which denotes a graph on \( n \) vertices in which each of the \( \binom{n}{2} \) possible edges is present with probability \( p = p(n) \), independently of all other edges. Observe that if we set \( p = 2m/n^2 \), then we expect that \( G(n, p) \) has \( \binom{n}{2}p \approx m \) edges. One of the fundamental results in random graph theory establishes a strong connection between both models if the relation \( p = 2m/n^2 \) is satisfied [Luc90]. In particular, thresholds of monotone graph properties are equivalent in both models (with respect to this relation).
3.2 Games in Discrete Mathematics

3.2.1 Perfect Information Games

The study of games in discrete mathematics has generated a large body of research on *perfect information* games in which at every time all players have full information on the state of the current game and the other player’s moves. Such games can in principal be solved by exhaustive enumeration of all possible combinations of legal moves, typical examples are Tic-Tac-Toe, Nim, Checkers or Chess. It is fairly easy to see that, when played optimally, Tic-Tac-Toe always ends in a draw (see [Mun10] for an illustrative proof), and recently even the game of Checkers with roughly $5 \cdot 10^{20}$ possible positions was proved to be a draw-game [SBB+07], after 18 years of continuous computer calculations. However, we are to this day still unable to understand the ‘combinatorial chaos’ of Chess in full detail. A class of perfect information games that was studied intensely over the last decades is given by the so-called *positional games*. They consist of a set of elements, called *board*, from which two players take elements in turns. Furthermore, there is a family subsets of the board, the so-called *winning sets* and there are different ways to determine the winner of the game. Such games were first studied in two seminal papers by Hales and Jewett [HJ63] and Erdős and Selfridge [ES73]. They consider the variant of *strong* positional games in which the first player to claim a winning set wins the game. (Note that Tic-Tac-Toe is an example for such a game, where the board is the $3 \times 3$ square and the winning sets are all rows, columns and diagonals.) Their work was later complemented and extended by many researchers with József Beck playing a key role [Bec81, Bec82, Bec08]. It turns out that probabilistic tools, in particular the *probabilistic method* [Erd59, Erd61, AS08], are very powerful to get a grip on the combinatorial chaos of these games.

An important variant of positional games are the so-called *Maker-Breaker games*, in which the first player, called *Maker*, wins if she succeeds to claim a winning set and otherwise the second player, called *Breaker* wins. These games have particularly been studied on complete graphs, that is, when the board of the game consists of the edges (or vertices) of a complete graph and the family of winning sets is formed by all subsets of edges (or vertices) that induce a graph with a certain property. One of the earliest works by Lehman [Leh64] shows that Maker can easily
create a spanning subgraph. In fact, one can see that for most graph properties Maker can easily win the game if the complete graph which forms the board is large enough. One way to keep things interesting even for large boards is to enhance Breaker’s power by allowing him to take more than one element in one move. Such games are called biased positional games and the question of how much power Breaker requires to win the game has been studied for various families of winning sets, e.g. spanning subgraphs, Hamiltonian subgraphs, subgraphs containing a clique of fixed size, see [CE78, Bec85, Bec02, BL00, BL01, BP05].

Another way to keep the games interesting, which is of particular interest in the context of this thesis, is to make the board sparser by playing on a random graph $G(n,p)$ instead of the complete graph. This approach was first pursued by Stojaković and Szabó [SS05] and further developed in [BSFHK12, MS]. For a couple of graph properties they provide an answer to the question by how much we need to reduce the board (that is, how small we have to make $p$) in order to prevent Maker from winning. In fact, there is a surprising correlation for many games between the two variants of increasing Breaker’s power and decreasing the board size. Roughly speaking, when playing with the ‘threshold bias’ $b_0 = b_0(n)$ at which a Maker’s win ‘turns’ into a Breaker’s win, then Maker claims roughly $\frac{1}{b_0} \binom{n}{2} = \Theta(n^2/b_0)$ edges; and for many games the ‘threshold probability’ $p_0 = p_0(n)$ at which the outcome of the game on the board $G(n,p)$ ‘turns’ is at $p_0 = 1/b_0$ such that the expected number of edges in $G(n,p_0)$ is of order $n^2/b_0$.

### 3.2.2 Mastermind

A particular game that we study in this thesis is the well-known boardgame Mastermind, see Section 1.1.4 for the rules. It is a game of imperfect information which has been studied extensively by discrete mathematicians and theoretical computer scientists. It has attracted a lot of attention due to its relation to fundamental complexity and information-theoretic questions, and its relevance is illustrated well by the large number of results on various aspects of the game.

One line of research, which we address in Chapter 7, investigates the question how many queries Codebreaker needs to identify Codemaker’s secret code if he employs an optimal strategy [ER63b, Knu77, Chv83, Goo09b], see Section 7.1.2 for an overview of previous work regarding this aspect.
A number of results exists on the computational complexity of evaluating given guesses and answers. Stuckman and Zhang [SZ06] showed that it is NP-hard to decide whether there exists a secret code that is consistent with a given sequence of queries and black- and white-peg answers. This result was extended to black-peg Mastermind by Goodrich [Goo09b]. More recently, Viglietta [Vig12] showed that both hardness results apply also to the setting with only $k = 2$ colors. In addition, he proved that counting the number of consistent secret codes is \#P-complete.

Another intensively studied question in the literature concerns the computation of (explicit) optimal winning strategies for small values of $n$ and $k$. The foundation for these works was laid by Knuth’s famous paper [Knu77] for the case with $n = 4$ positions and $k = 6$ colors. His strategy is worst-case optimal. Koyama and Lai [KL93] studied the average-case difficulty of Mastermind. They gave a strategy that solves Mastermind in an expected number of about 4.34 guesses if the secret string is sampled uniformly at random from all $6^4$ possibilities, and they showed that this strategy is optimal. Today, a number of worst-case and average-case optimal winning strategies for different (small) values of $n$ and $k$ are known—both for the black- and white-peg version of the game [God04, JP09] and for the black-peg version [JP11]. Non-adaptive strategies for specific values of $n$ and $k$ were studied in [God03].

In the field of computational intelligence, Mastermind is used as a benchmark problem. For several heuristics, among them genetic and evolutionary algorithms, it has been studied how well they play Mastermind [KC03, TK03, BGL09, MCM11a, MCM11b].

Trying to understand the intrinsic difficulty of a problem for such heuristics, Droste, Jansen, and Wegener [DJW06] suggested to use a query complexity variant (called black-box complexity). For the so-called one-max test-function class, an easy benchmark problem in the field of evolutionary computation, the black-box complexity problem is just the Mastermind problem for two colors. This inspired, among others, the result [DW12] showing that a memory-restricted version of Mastermind (using only two rows of the board) can still be solved in $O(n/\log n)$ guesses when the number of colors is constant.

Several privacy problems have been modeled via Mastermind. Goodrich [Goo09a] used black-peg Mastermind to study the extent of private genomic data leaked by comparing DNA-sequences (even when using protocols only revealing the degree of similarity). Focardi and Luccio [FL10]
showed that certain API-level attacks on user PIN data can be seen as an extended Mastermind game.

### 3.3 Extremal Combinatorics

As the name suggests the field of *extremal combinatorics* deals with questions of the form: how do extreme combinatorial objects with a given property look like? Typically one is interested in how large/small or dense/sparse such an object can be. Dating back to 1928 one of the earliest results in this field is due to the German mathematician Emanuel Sperner and accordingly called *Sperner’s Theorem* [Spe28]. It states that given an \( n \)-element set \( M \), every family \( F \) of subsets of \( M \) in which no member is subset of another member (i.e. \( F \) is an antichain with respect to inclusion) satisfies \(|F| \leq \binom{n}{n/2}\) where equality holds if and only if \( F \) is the family of all subsets of \( M \) of cardinality \( \lceil n/2 \rceil \) or \( \lfloor n/2 \rfloor \).

The subfield of extremal combinatorics that will be most relevant for this thesis is *Ramsey Theory*. Its name is derived from the British mathematician Frank Plumpton Ramsey who laid the foundation with his trend-setting work from the late 1920ies, shortly before he died at the age of 26. Roughly speaking, this subfield studies under which conditions order must appear. Typical results show how large an object must be so that no matter how we partition it into a given number of pieces, at least one piece will always have a specific given structure. Although there are earlier works with results of this form (e.g. [Sch16]) the fundamental theorem of the field is *Ramsey’s Theorem* which in its (almost literally) original form reads as follows.

**Theorem 3.3** (Ramsey’s Theorem, [Ram30]). Given any positive integers \( r, n, \) and \( \mu \) we can find an \( m_0 \) such that, if \( m \geq m_0 \) and the subsets of size \( r \) of any set \( \Gamma_m \) of size \( m \) are divided in any manner into \( \mu \) mutually exclusive classes \( C_i \) \((i = 1, 2, \ldots, \mu)\), then \( \Gamma_m \) must contain a subset \( \Delta_n \) of size \( n \) such that all subsets of size \( r \) of \( \Delta_n \) belong to the same \( C_i \).

The theorem is often introduced in the context of edge-colorings of graphs with 2 colors. It then states that given a positive integer \( k \) we can find \( m \) such that every edge-coloring of the complete graph \( K_m \) on \( m \) vertices with 2 colors, say red and blue, contains an entirely red or an
entirely blue complete subgraph $K_k$ on $k$ vertices. Observe that this is a special case of Theorem 3.3 with $r = 2$, $n = k$ and $\mu = 2$. The minimum $m$ which satisfies this property is called Ramsey number of $k$, denoted by $R(k)$. It is known that $R(1) = 1$, $R(2) = 2$, $R(3) = 6$, $R(4) = 18$, but already $R(5)$ is unknown, the best currently-known bounds being $43 \leq R(5) \leq 49$.\footnote{The following anecdote of Paul Erdős is often used to express the difficulty to calculate these numbers. “Suppose aliens invade the earth and threaten to obliterate it in a year’s time unless human beings can find the Ramsey number of five. We could marshal the world’s best minds and fastest computers, and within a year we could probably calculate the value. If the aliens demanded the Ramsey number for six, however, we would have no choice but to launch a preemptive attack.” [GS90]} For later reference note that the statement can easily be extended to hyperedge-colorings of $r$-uniform hypergraphs with any fixed number of colors. (This simply corresponds to the case of general $r$ and $\mu$ in Theorem 3.3.)

One of the biggest open questions in Ramsey Theory is the asymptotic behavior of $R(k)$ for $k$ going to infinity. It has been shown in [ES35, Erd47] that
\[
\sqrt{2}^k \leq R(k) \leq 4^k
\]
and the arguments behind these results are not too involved. However, despite the effort of many researchers [Spe75, GR87, Tho88, Con09] it is still unknown if there exists $\varepsilon > 0$ such that $R(k) \geq (\sqrt{2} + \varepsilon)^k$ or $R(k) \leq (4 - \varepsilon)^k$ for $k$ large enough.

There are plenty of beautiful results in the spirit of Theorem 3.3, see e.g. [GRS90] for an overview. Here, we only mention the work of Schur [Sch16] concerning solutions of the equation $x + y = z$ in one of the ‘pieces’, its generalization by Rado [Rad33] to systems of linear equations, and the work of van-der-Waerden [vdW27] regarding arithmetic progressions.

Of particular interest for this thesis is the following Ramsey-theoretic question. It generalizes the setup of the aforementioned edge-coloring special case of Ramsey’s Theorem from complete graphs to arbitrary graphs. More precisely, given two graphs $G$ and $F$, and a number $r \geq 2$ of colors. Does every edge-coloring of $G$ with $r$ colors contain a monochromatic copy of $F$? If this is the case, then $G$ is called $F$-Ramsey for $r$ colors and we write
\[
G \rightarrow (F)^e_r
\]
where the superscript $e$ denotes that we color the edges of $G$. (We
write \( v \) if we color its vertices instead.) We write \( G \not\rightarrow (F)_r^c \) if there exists an \( r \)-edge-coloring of \( G \) without a monochromatic copy of \( F \). In this notation, Theorem 3.3 implies that for every \( k \geq 2 \) there exists an integer \( m \) such that \( K_m \not\rightarrow (K_k)_r^2 \).

### 3.4 Ramsey Games on Random Graphs and Achlioptas Processes

The study of games on random graphs has been a very active research area over the past decades. Here, we outline the development of Ramsey type games and related variants and give an overview on the state of the art.

The games considered in this section are *one player games*. Furthermore, all games start with an empty graph on \( n \) vertices into which edges are inserted according to a given random graph process and according to the choices made by the player. Her goal is to avoid a given graph property \( \mathcal{P} \) for as long as possible or embrace \( \mathcal{P} \) as quickly as possible. What is the typical duration of such a game if the player employs an optimal strategy? Properties that have been of great interest are for example ‘containing a copy of a fixed subgraph’, ‘containing a linear-sized component’, or ‘being Hamiltonian’. We distinguish between the following types of games.

- **Ramsey games.** Here, the player, called *Painter*, is presented a new edge in every step which is chosen uniformly at random from all non-edges and which she needs to color immediately and irrevocably with one of \( r \geq 2 \) available colors. A variant are the so-called **vertex Ramsey games** in which Painter is presented vertices (together with the random edges to the previously revealed vertices) which she needs to color immediately. Another variant are **balanced Ramsey games**, in which Painter is presented \( r \) random edges (or vertices) in every step and she needs to color all of them immediately using each of the \( r \) colors exactly once. In Ramsey games the player’s goal can be **symmetric**, e.g. ‘avoid a monochromatic copy of a fixed subgraph’, or **asymmetric**, e.g. ‘avoid a blue triangle and a red path of length 4’.

- **Achlioptas Processes.** Here, in every step the player, called *Chooser*, is presented a pair of (or in general \( r \geq 2 \)) possible edges chosen
uniformly at random from all non-edges, one of which she needs to choose immediately and irrevocably to insert into the graph, while the other is (are) put back into the pool of non-edges. Throughout this thesis we speak of Achlioptas processes (in plural) to denote the class of all Achlioptas processes that can be obtained by the different edge selection strategies of Chooser. If her strategy is fixed we speak of an Achlioptas process (in singular).

For all these online games in which the player needs to act immediately, irrevocably and without any knowledge about future steps there is also a corresponding offline variant. In these variants all decisions (e.g. all edges) are presented to the player at once and the question is not how long she can survive, but if she can achieve her goal with an optimal strategy for a given number of steps that we present to her at once. Observe that any upper bound on the number of steps that Painter can typically handle in the offline setting immediately implies an upper bound on the typical duration of the corresponding online game. In this sense offline settings provide an appropriate benchmark that any online strategy should be compared to. For a complete understanding of an online game it is thus essential to study the underlying offline problem.

3.4.1 Offline Ramsey Games with Small Subgraphs

The study of offline Ramsey games where Painter’s goal is to avoid a monochromatic copy of a given subgraph $F$ was initiated by Łuczak, Ruciński, and Voigt \cite{LRV92} who mainly consider the vertex variant of this game. They found a threshold function for the property $G(n, p) \rightarrow (F)^v_r$ for arbitrary graphs $F$ and also established one in the edge-coloring case for the property $G(n, p) \rightarrow (K_3)^e_2$. Thereupon, in a series of papers Rödl and Ruciński showed lower bounds for the density threshold of $G(n, p) \rightarrow (F)^e_r$ \cite{RR93}, then extended the known result about triangles to an arbitrary number of colors \cite{RR94} and finally solved the edge-coloring problem in full generality \cite{RR95}. It was added later by Friedgut and Krivelevich \cite{FK00} that the path of length 3 has an exceptional threshold behavior in the case of 2 colors. Formally, the complete result reads as follows.

**Theorem 3.4** (\cite{RR93, RR95, FK00}). For all integers $r \geq 2$ and for every graph $F$ which is not a forest of stars and, in the case $r = 2$, paths of length 3 there exist constants $c = c(F, r) > 0$ and $C = C(F, r) > 0$
such that
\[
\lim_{n \to \infty} \Pr\left[ G(n, p) \to (F)_c^c \right] = \begin{cases} 0 & \text{if } p < cn^{-1/m_2(F)} \\ 1 & \text{if } p > Cn^{-1/m_2(F)} \end{cases},
\]
where
\[
m_2(F) = \max_{H \subseteq F, v(H) \geq 3} d_2(H)
\]
and
\[
d_2(H) = \frac{e(H) - 1}{v(H) - 2}
\]
for graphs $H$ with $v(H) \geq 3$.

Here, $m_2(F)$ and $d_2(F)$ denote for every graph $F$ with at least 3 vertices the (maximum) 2-edge-density, where we shortly call $m_2(F)$ the 2-density of $F$. Note that under the assumption $m_2(F) = d_2(F)$ we have that $p = n^{-1/m_2(F)}$ is the density where we expect that every edge is in roughly a constant number of copies of $F$. This observation can be used to provide an intuitive understanding of the bounds of Theorem 3.4. If $c$ is very small, then the number of copies of $F$ is small enough that they are so scattered that a coloring without a monochromatic copy of $F$ can be found. If on the other hand $C$ is big then these copies a.a.s. overlap so heavily that every coloring has to induce at least one monochromatic copy of $F$ (in fact, as Rödl and Ruciński show, they induce “many” copies).

Extending this work, Friedgut and Krivelevich [FK00] proved the thresholds of Theorem 3.4 to be sharp for a large class of graph (most trees and the special case when $F$ is what they call strongly strictly balanced), and later Friedgut, Rödl, Ruciński and Tetali [FRRT06] showed that this is also the case if $F$ is a triangle.

Furthermore, the work of Rödl and Ruciński was the starting point for a series of papers on a variety of offline Ramsey games involving small subgraphs. Kohayakawa and Kreuter [KK97] proved threshold results for an asymmetric version in which the goal is for given parameters $\ell_1, \ldots, \ell_r$ to color the edges of $G(n, p)$ with $r$ colors without a cycle of length $\ell_i$ in color $i$. Later, Marciniszyn, Skokan, Spöhel and Steger [MSSS09] solved the asymmetric problem for avoiding cliques of size $\ell_i$ in color $i$. Their proof of the 1-statement relies on the so-called KLR-conjecture (which is now confirmed [ST, BMS] but was not at the time of publication of the work) and there is a recent alternative proof.
of the 1-statement for the case of 2 colors by Kohayakawa, Schacht and Spöhel [KSS] that does not rely on it. The corresponding asymmetric vertex Ramsey game was solved by Kreuter [Kre96] in full generality, that is, for an arbitrary list \( G_1, G_2, \ldots, G_r \) of graphs where Painter’s goal is to avoid a copy of \( G_i \) in color class \( i \).

One of the most intriguing problems in random Ramsey theory today is the generalization of Theorem 3.4 to \( k \)-uniform hypergraphs. Denote by \( H^{(k)}(n, p) \) a \( k \)-uniform random hypergraph in which each of the \( \binom{n}{k} \) possible hyperedges is present with probability \( p \) independently of all other hyperedges. Ramsey properties in random hypergraphs were first considered in [RR98] in which Rödl and Ruciński prove a threshold for the vertex case \( H^{(k)}(n, p) \to (F)^v_r \) for any \( r \geq 2 \) and arbitrary \( k \)-uniform hypergraphs \( F \), as well as for the hyperedge case \( H^{(3)}(n, p) \to (K_4^{(3)})^e_r \), where \( K_4^{(3)} \) denotes the complete \( 3 \)-uniform hypergraph on 4 vertices. Moreover, they conjecture a general upper bound and mention that corresponding lower bounds might be obtained similar to [RR93]. In [RRS07] they, together with Schacht, prove a threshold result for \( H^{(k)}(n, p) \to (F)^e_r \) in the case when \( F \) is a \( k \)-partite \( k \)-uniform hypergraph. Recently, Friedgut, Rödl and Schacht [FRS10], and independently Conlon and Gowers [CG], confirmed the conjectured 1-statement from [RR98] for arbitrary hypergraphs and arbitrary \( r \).

**Theorem 3.5** ([FRS10, CG]). Let \( F \) be a \( k \)-uniform hypergraph with maximum degree at least 2 and let \( r \geq 2 \). Then there exists a constant \( C > 0 \) such that for \( p \geq C n^{-1/m_k(F)} \) we have

\[
\lim_{n \to \infty} \Pr \left[ H^{(k)}(n, p) \to (F)^e_r \right] = 1 ,
\]

where

\[
m_k(F) = \max_{H \subseteq F, \ \nu(H) > k} d_k(H)
\]

and

\[
d_k(H) = \frac{e(H) - 1}{\nu(H) - k}
\]

for hypergraphs \( H \) with \( \nu(H) > k \).

Here, \( m_k(F) \) and \( d_k(F) \) denote for every hypergraph \( F \) with at least \( k + 1 \) vertices the so-called (maximum) \( k \)-hyperedge-density, where we shortly call \( m_k(F) \) the \( k \)-density of \( F \). The question if a corresponding
0-statement to Theorem 3.5 exists is one of the main open problems in this field.

**Conjecture 3.6 ([RR98]).** For every $k$-uniform hypergraph $F$ with at least $k + 1$ vertices and every $r \geq 2$ there exists a constant $c > 0$ such that for $p \leq cn^{-1/m_k(F)}$ we have

$$\lim_{n \to \infty} \Pr \left[ H^{(k)}(n, p) \to (F)^e_r \right] = 0.$$  

Observe that these bounds on $p$ can be motivated similarly to the ones in Theorem 3.4. If $p$ is of order $n^{-1/m_k(F)}$ then, under the assumption that $m_k(F) = d_k(F)$ we expect a fixed hyperedge in $H^{(k)}(n, p)$ to be contained in a constant number of copies of $F$. In [FRS10] the authors already indicate that Conjecture 3.6 might not hold for all hypergraphs: “We believe that the matching 0-statement also holds for ‘most’ hypergraphs $F$.” Indeed, we show in Chapter 6 that there are hypergraphs $F$ for which the threshold is not of order $n^{-1/m_k(F)}$. However, we also show a 0-statement that matches the bounds from Theorem 3.5 for a large class of hypergraphs, including all complete hypergraphs.

In a recent paper by Allen, Hladky, Böttcher and Piguet [ABHP13] the authors suggest another generalization of Theorem 3.4. Observe that the setting of the theorem can be seen as the question of how many edges we need to remove randomly from the complete graph on $n$ vertices to allow an $r$-edge-coloring without a monochromatic copy of the fixed forbidden graph $F$. In contrast to this Allen et al. suggest to not remove any edges from the complete graph, but to randomly select copies of $F$ that we do not care about if they appear monochromatically. The question they asked is how many copies of $F$ we need to select to allow a coloring in which no ‘dangerous’ copy of $F$ is monochromatic. More formally, given a graph $F$ and a number $r \geq 2$ of colors, we define a *restriction set* $\mathcal{R}_F$ to be a subset of all copies of $F$ in the complete graph on $n$ vertices. Moreover, we define a *random restriction set* $\mathcal{R}_F(n, q)$ by including every such copy of $F$ independently with probability $q$. For what values of $q = q(n)$ can one color the edges of $K_n$ such that no copy of $F$ from the restriction set is monochromatic? In joint work with Gugelmann, Person and Steger [GPST12] we considered the combination of this question with the usual randomization of $G(n, p)$ and showed the following result. By $G \xrightarrow{\mathcal{R}_F} (F)^e_r$ we denote the property that every $r$-edge-coloring of $G$ contains a monochromatic copy of $F$ that is also in the restriction set $\mathcal{R}_F$. Moreover, we say that a graph $G$ is
2-balanced if \( m_2(G) = \frac{e(G) - 1}{v(G) - 2} \) and call it strictly 2-balanced if moreover \( \frac{e(J) - 1}{v(J) - 2} < m_2(G) \) for every \( J \not\subseteq G \).

Theorem 3.7 \((\text{GPST12})\). Let \( F \) be a strictly 2-balanced graph with at least 3 edges. Let \( r \geq 2 \) be a fixed integer. There exist constants \( c = c(F, r) > 0 \) and \( C = C(F, r) > 0 \) such that

\[
\lim_{n \to \infty} \Pr \left[ G(n, p) \xrightarrow{\mathcal{R}_F(n, q)} (F)^e_r \right] = \begin{cases} 0, & \text{if } n^{v(F)} p^{e(F)} q \leq cn^2 p, \\ 1, & \text{if } n^{v(F)} p^{e(F)} q \geq Cn^2 p. \end{cases}
\]

Again, the intuition behind the order of magnitude of the threshold can be explained as follows. If \( p \) and \( q \) are such that \( n^{v(F)} p^{e(F)} q \) is of order \( n^2 p \), then we expect every edge in \( G(n, p) \) to be contained in a constant number of ‘dangerous’ copies of \( F \), where a copy of \( F \) is dangerous if it is in \( \mathcal{R}_F(n, q) \).

### 3.4.2 Online Ramsey Games with Small Subgraphs

Let us now turn to online Ramsey games where Painter’s goal is to avoid a given subgraph for as long as possible. The study of these games was initiated in \( \text{FKR}^+03 \) by Friedgut et al. Recall that in online Ramsey games Painter starts on an empty graph on \( n \) vertices and is presented the edges of a random graph one by one and has to color them immediately and irrevocably. Friedgut et al. showed that if the forbidden graph is the triangle and Painter is allowed to use 2 colors, then she can with high probability succeed for any number of steps that is asymptotically less than \( n^{4/3} \) just by coloring greedily, i.e., by assigning every edge color blue if this does not close a blue triangle in which case she uses red. Moreover, they showed that every strategy fails with high probability to avoid monochromatic triangles for any number of steps that is asymptotically larger than \( n^{4/3} \). Their proof uses a technique called two-round exposure which has become standard for proving upper bounds in online Ramsey games: they allow Painter to color the first half of the edges of the random graph in an offline-fashion, that is, the first half of the edges is revealed at once. They show that Painter is forced to create many ‘dangers’ one of which will be ‘hit’ by the second half of the edges and force her to loose.

The analysis of greedy strategies for online Ramsey games has been generalized by Marciniszyn et al. \( \text{MSS09a} \) to arbitrary graphs \( F \) and
an arbitrary number of colors, where for more than 2 colors the greedy strategy simply uses in each step the minimum color that does not create a monochromatic copy of $F$. Furthermore, matching upper bounds were shown in [MSS09b] for the case of 2 colors and a large class of forbidden graphs $F$, including cliques and cycles of arbitrary size, using an extension of the two-round exposure argument presented in [FKR+03]. For all these graphs the thresholds in the online setting are strictly smaller than in the offline setting (cf. Theorem 3.4). However, the lower bound from the greedy strategy approaches the offline threshold for increasing $r$. Recently, Belfrage, Mütze and Spöhel [BMS12] presented a new technique to prove upper bounds for online Ramsey games which yields optimal bounds if $F$ is a forest. They establish a surprising connection between one-player probabilistic games and deterministic two-player games where the random graph process is in some sense replaced by an adversary who has to follow certain restrictions.

While these results settle the analysis of online Ramsey avoidance games for 2 colors and many forbidden graphs $F$, it turns out that the game is significantly harder to analyze for 3 or more colors. In fact, after some progress in [BB10] using the aforementioned result from [BMS12], the case of $F$ being a triangle was solved just recently for an arbitrary number of colors by Noever [Noe12].

The corresponding vertex Ramsey game for arbitrary graphs $F$ was solved by Mütze, Rast and Spöhel [MRS11]. Furthermore, the balanced variant of the (edge-coloring) Ramsey game was first studied in [MMS07] by Marciniszyn, Mitsche and Stojaković. They solve the problem for 2 colors and for the case when $F$ is a cycle. These results were generalized by Prakash, Spöhel and myself [PST09] who solve both the balanced edge-coloring and vertex-coloring Ramsey game for an arbitrary number of colors and a large class of graphs, including cycles and cliques of arbitrary sizes. The thresholds for the balanced game are strictly smaller the ones for the unbalanced game, although the corresponding offline problems have identical thresholds [KSS10]. Hence, in contrast to the offline setting there is a substantial difference between the balanced and unbalanced scenario in online Ramsey avoidance games. However, as in the unbalanced case the thresholds of the balanced game approach the offline threshold for increasing $r$. 


3.4.3 Achlioptas Processes

The motivation to consider Achlioptas processes was the question if one can accelerate or delay the appearance of the giant component by introducing a freedom of choice into the classical random graph process, cf. Theorem 3.1 and Figure 3.1. One of the earliest results that demonstrated the power of the freedom of choice paradigm is the work by Azar et al. on load balancing [ABKU94]. Recall that in Achlioptas processes the player, called Chooser, is presented a pair of possible edges in every step which is drawn uniformly at random from all non-edges in the current graph. She immediately has to select one of them to be included as an edge into the graph, while the other is put back into the pool of non-edges.

The question whether one can delay the appearance of the giant component was first answered positively by Bohman and Frieze [BF01]. They showed that a simple strategy (choose the first edge if it connects two isolated vertices, otherwise choose the second edge) delays its appearance for roughly $0.535n$ steps. Later, Flaxman, Gamarnik and Sorkin [FGS05] and Bohman and Kravitz [BK06b] proved that the giant component can also be embraced with a simple strategy: the latter work shows that choosing the first edge if and only if it connects two non-isolated vertices creates a linear-sized component within $0.385n$ steps.

In [FGS05] it is also shown that Chooser needs at least $0.2507n$ steps to create a giant (which is strictly more than the trivial $0.25n$ lower bound below which even choosing both edges presented in every step does not yield a giant component). Moreover, Chooser cannot avoid the giant component for more than $0.965n$ steps as shown by Bohman, Frieze and Wormald [BFW04]. The currently best-known bounds are due to Spencer and Wormald [SW07] who analyze more complex strategies for Chooser and show that she can create a giant component within $0.334n$ steps and avoid it for at least $0.829n$ steps.

Some of the Achlioptas processes that were considered in this line of research became of great interest for another aspect. Based on the results of computer simulations it was conjectured by Achlioptas, Spencer and D’Souza [ADS09] that certain edge selection strategies lead to a so-called discontinuous phase transition. Roughly speaking, this means that at the critical time when the transition occurs the largest component ‘almost instantly’ (in a sublinear number of steps) accumulates a constant fraction of the vertices. This is in contrast to the continuous
phase transition of the classical random graph process where the largest component grows ‘smoothly’. After many efforts, also from the physics community, to settle this conjecture \cite{CKK10,CKP+09,DM10,RF10}, it was first claimed in \cite{dCDGM10} and then proved rigorously by Riordan and Warnke \cite{RW11} that the phase transition of any Achlioptas process is continuous. In Chapter 5 we complement these results by proving discontinuous phase transitions for a variety of processes all of which are closely related to the classical Erdős-Rényi random graph process.

Let us now consider the offline variant of Achlioptas processes. That is, Chooser is presented a given number of random pairs of possible edges at once and her goal is to avoid a giant component, or to create one. It was shown in \cite{BK06b} that creating a giant component has a sharp threshold at $0.25n$ edge pairs, i.e., for any $c > 0.25$ it is possible to choose one edge from each of $cn$ randomly chosen edge-pairs such that the resulting graph contains a giant component, while for $c < 0.25$ this is clearly not possible since the graph formed by all presented edges does not contain one. The avoidance of the giant component was studied by Bohman and Kim \cite{BK06a} who show that a sharp threshold exists at roughly $0.978n$ edge-pairs. Observe that this threshold is strictly larger than the upper bound of $0.965n$ for the online variant. Hence, there exist $0.965 < c < 0.978$ such that no strategy for Chooser in the online game can avoid a giant component with high probability for $cn$ steps, while an optimal offline strategy that sees all $cn$ edge-pairs in advance can. The question what happens if we loosen the restriction to select one edge from each pair, but instead simply present $2cn$ edges for some constant $c > 0$ and ask Chooser to select $cn$ edges out of these, was studied in \cite{BFW04}. They show that the threshold for this variant is at roughly $0.979n$ (which is strictly larger than the one of the more restricted offline Achlioptas setting). In Chapter 4 we study the two corresponding offline Ramsey games for a given number $r$ of colors: in one Painter is presented $rcn$ random edges which she has to color with $r$ colors, while in the other she is given $rcn$ random edges partitioned into sets of size $r$ and she needs to use each of the $r$ colors exactly once in each set. The first setting was independently solved by Bohman, Frieze, Krivelevich, Loh and Sudakov \cite{BFK+11}. It turns out that in contrast to the Achlioptas setting the thresholds are identical for both Ramsey games.

The study of Achlioptas processes, online and offline, has not only fo-
cused on questions concerning the giant component. Here, we only mention that Krivelevich, Lubetzky and Sudakov [KLS10] investigated the creation of a Hamiltonian cycle, where they consider generalized Achlioptas processes in which Chooser is presented \( r \geq 2 \) possible edges in every step instead of 2. Furthermore, the goal of avoiding a fixed subgraph, which has generated a large body of research in the context of Ramsey games, has been solved completely for the online setting by Mütze, Spöhel and myself [MST11], disproving a conjecture of Krivelevich, Loh and Sudakov [KLS09] who solved the problem for the special cases if \( F \) is a clique, cycle or complete bipartite graph. The corresponding offline question was considered by Krivelevich, Spöhel and Steger [KSS10] who show that for most graphs \( F \) the threshold for avoiding a (monochromatic) copy of \( F \) is the same in the offline Ramsey game (cf. Theorem 3.4), the offline balanced Ramsey game and the offline Achlioptas setting.
Chapter 4

Offline Ramsey Games with Giants

In this chapter we present our results about two offline Ramsey games in which Painter’s goal is to avoid a monochromatic component of linear size. This is joint work with Reto Spöhel and Angelika Steger, which has been published in the Electronic Journal of Combinatorics.

4.1 Introduction

As mentioned in the previous Chapter the classical random graph process has a phase transition at \( n/2 \) [ER60]. Around the point when the average degree in the graph is 1 the component structure changes drastically and a linear-sized ‘giant component’ emerges, cf. Theorem 3.1. In Section 3.4.3 we outlined how Achlioptas processes introduce a freedom of choice into this process. Recall that in Achlioptas processes two ran-
dom possible edges are sampled at each step, and exactly one of them has to be selected immediately and irrevocably to be included in the graph. It turns out that in this online scenario the appearance of the giant component can be both accelerated and delayed by a constant factor if appropriate edge selection strategies are used [BF01, BFW04, BK06b, FGS05, SW07].

Motivated by this line of research, Bohman and Kim [BK06a] studied a similar question in an offline setting. Given $cn$ pairs of random edges, can we select one edge from each pair to obtain a graph with $cn$ edges that does not contain a giant component? If we assume that all edges are sampled without replacement, then drawing $cn$ random edge pairs is equivalent to drawing $2cn$ random edges and partitioning these into pairs uniformly at random. More generally, we denote for any fixed integer $r \geq 2$ by $G^r(n, m)$ a random $r$-matched graph obtained by generating a random graph $G(n, m)$ and partitioning its edge set into sets of size $r$ uniformly at random (where we assume w.l.o.g. that $m$ is divisible by $r$). Throughout, we call these $r$-sets and denote them by $E_j$, $1 \leq j \leq m/r$. We say that a subgraph is an Achlioptas subgraph of $G^r(n, m)$ if it contains exactly one edge from each $r$-set.

In this terminology, the result of Bohman and Kim reads as follows.

**Theorem 4.1 ([BK06a]).** There exists an analytically computable constant $c_1 \approx 0.9768$ such that for any constant $c > 0$ the following holds.

- If $c < c_1$, then a.a.s. $G^2(n, 2cn)$ has an Achlioptas subgraph (with $cn$ edges) all components of which have $O(n^{1-\varepsilon})$ vertices, where $\varepsilon > 0$ is a constant depending on $c$.

- If $c > c_1$, then a.a.s. every Achlioptas subgraph of $G^2(n, 2cn)$ contains a component with $\Theta(n)$ vertices.

A similar but less restrictive scenario was investigated earlier by Bohman, Frieze, and Wormald [BFW04]. Given an ordinary random graph $G(n, 2cn)$, can we select $cn$ edges such that the resulting graph does not contain a giant component? As it turns out, the threshold for this property is slightly higher than the one stated in Theorem 4.1.

**Theorem 4.2 ([BFW04]).** There exists an analytically computable constant $c_2 \approx 0.9793$ such that for any constant $c > 0$ the following holds.

- If $c < c_2$, then a.a.s. $G(n, 2cn)$ has a subgraph with $cn$ edges all components of which have $O(1)$ vertices.
4.1. Introduction

- If $c > c_2$, then a.a.s. every subgraph of $G(n, 2cn)$ with $cn$ edges contains a component with $\Theta(n)$ vertices.

4.1.1 Our Results

In this chapter we consider analogous questions for two Ramsey-type settings. In the first we are given a random graph $G(n, m)$ and are required to color its edges with a fixed number $r$ of available colors. Our goal is to avoid a monochromatic giant component in each of the $r$ color classes. Note that, by the pigeon-hole principle, Theorem 4.2 yields an upper bound for the case $r = 2$.

The second setting we investigate is more restrictive and motivated by Theorem 4.1. We call an $r$-edge-coloring of the random $r$-matched graph $G^r(n, m)$ valid if every color is used for exactly one edge from every $r$-set. Note that this implies in particular that every color class has the same size. Our goal now is to find a valid $r$-edge-coloring of $G^r(n, m)$ such that no color class contains a giant component. Note that, again by the pigeon-hole principle, Theorem 4.1 yields an upper bound for the case $r = 2$ in this more restricted setting that is slightly better than the bound given by Theorem 4.2 for the first setting.

As it turns out (see Theorem 4.3 below), the thresholds of these two Ramsey settings coincide, in contrast to what happens for the edge-selection scenarios studied in Theorem 4.1 and Theorem 4.2. In order to state our main result, we need to introduce one more notion. For any integer $r \geq 2$, a graph $G$ is called $r$-orientable if there exists an orientation of its edges such that the in-degree of every vertex is at most $r$. Cain, Sanders, and Wormald [CSW07] and independently Fernholz and Ramachandran [FR07] showed that the property that $G_{n,m}$ is $r$-orientable has a sharp threshold at $m = r c^*_r n$ for some analytically computable constant $c^*_r$. In [CSW07], the following numerical values are stated: $c^*_2 \approx 0.897$, $c^*_3 \approx 0.959$, $c^*_4 \approx 0.980$, and $c^*_5 \approx 0.989$.

Our result is the following.

**Theorem 4.3.** For every integer $r \geq 2$, let $c^*_r$ denote the constant which determines the threshold for $r$-orientability of the random graph $G(n, rcn)$. Then for any constant $c > 0$ the following holds.

- If $c < c^*_r$, then a.a.s. there exists a valid $r$-edge-coloring of $G^r(n, rcn)$ in which all monochromatic components have $O(n^{1-\varepsilon})$ vertices,
where \( \varepsilon > 0 \) is a constant depending on \( c \) and \( r \).

- If \( c > c^*_r \), then a.a.s. every \( r \)-edge-coloring of \( G(n, rcn) \) contains a monochromatic component with \( \Theta(n) \) vertices.

Note that the first statement of the theorem is about the restricted setting and the second statement is about the unrestricted setting. Together they imply that the thresholds of both scenarios coincide.

The threshold for the unrestricted setting was also derived independently from us by Bohman, Frieze, Krivelevich, Loh, and Sudakov [BFK+11]. They also investigated the corresponding online setting, where random edges appear one by one and have to be colored immediately with one of \( r \) available colors, and the version of the problem where the goal is to create monochromatic giants instead of avoiding them.

Let us conclude this introduction by stating an open question we would like to see answered: What is the threshold for the property that the vertices of a random graph can be colored in such a way that no color class induces a linear-sized component? It is not hard to see that for any \( c > 1.5 \), a random 3-vertex-coloring of \( G(n, cn) \) yields a giant component a.a.s. On the other hand, Achlioptas and Molloy [AM97] showed that a.a.s. \( G(n, cn) \) can be properly 3-colored as long as \( c < 1.923 \), thus providing a significantly better lower bound than a random vertex-coloring.

### 4.2 Proof of Main Result

#### 4.2.1 Preliminaries

We first state two rather technical lemmas which we will use in our proofs. The following density lemma can be seen as a generalisation of Lemma 2 in [BF10]. Roughly speaking, it shows that in a random graph with constant average degree all subgraphs of edge-density larger than 1 are large.

**Lemma 4.4.** Let \( c > 0 \). For every \( \varepsilon > 0 \) there exists \( \delta = \delta(c, \varepsilon) > 0 \) such that a.a.s. the random graph \( G := G(n, cn) \) has the property that for every \( S \subseteq V(G) \) for which \( G[S] \) contains more than \( (1 + \varepsilon)|S| \) edges we have \( |S| \geq \delta n \).

**Proof.** We will show by a union bound argument that with high probability every subset \( S \subseteq V(G) \) with \( |S| < \delta n \) (we will determine \( \delta \) at the
end of the proof) has less than \((1 + \varepsilon)|S|\) edges. Observe that this property is monotone. We can thus employ the independent random graph model \(G' := G(n, p)\) in the remainder of the proof (see e.g. Theorem 2.2 in [Bol01]), where \(p = 2c/n\).

We denote for every \(4 \leq s \leq \delta n\) by \(E_s\) the event that there exists \(S \subseteq V(G')\) with \(|S| = s\) such that \(G'[S]\) is connected and \(e(G'[S]) \geq (1 + \varepsilon)s\). Moreover, let \(E = \bigcup_{s=4}^{\delta n} E_s\). Observe that if there is such a subset \(S \subseteq V(G)\) with \(G'[S]\) being disconnected, then considering the densest component of \(G'[S]\) still implies that \(E\) occurs. Hence, it suffices to show \(Pr[E] = o(1)\).

With foresight let us first consider the case \(4 \leq s \leq \log_{2ce}(n)/3\). We show that in \(G'\) with high probability no subset of size \(s\) induces a connected subgraph with even \(s + 1\) edges. Note that there are \(\binom{n}{s}\) subsets of size \(s\) and \(s^{s-2}\) possible spanning trees on such a vertex set. Moreover, there are at most \(\binom{s^2}{2}\) possibilities to choose two additional edges. Hence, we have

\[
Pr[E_s] \leq \binom{n}{s} s^{s-2} \left(\binom{s}{2}\right) p^{s+1} \\
\leq \left(\frac{ne}{s}\right)^s s^{s-2} s^4 \left(\frac{2c}{n}\right)^{s+1} \\
= \frac{cs^2}{4n} (2ce)^s \\
\leq \frac{cs^2}{4n} n^{1/3} = o(n^{-1/2}) .
\]

It remains to consider \(\log_{2ce}(n)/3 < s \leq \delta n\) (where \(\delta\) is still to be determined). Similar to our argument for the previous case we have

\[
Pr[E_s] \leq \binom{n}{s} s^{s-2} \left(\frac{s}{\varepsilon s}\right) p^{s(1+\varepsilon)} \\
\leq \left(\frac{ne}{s}\right)^s s^{s-2} \left(\frac{se}{2\varepsilon}\right) \varepsilon s \left(\frac{2c}{n}\right)^{s(1+\varepsilon)} \\
= \frac{1}{s^2} \left(2(ce)^{1+\varepsilon} \varepsilon^{-\varepsilon} \frac{s}{n}\right)^s \\
\leq \frac{1}{s^2} \left(2(ce)^{1+\varepsilon} \varepsilon^{-\varepsilon} \delta\right)^s \\
= \frac{1}{s^2}.
\]
where the last inequality follows for \( \delta = \delta(c, \varepsilon) = (2(ce)^{1+\varepsilon\varepsilon-\varepsilon})^{-1} \).

Combining both cases we now obtain

\[
\Pr[\mathcal{E}] \leq \sum_{s=4}^{\delta n} \Pr[\mathcal{E}_s] = \sum_{s=4}^{[\log_{2c}(n)/3]} \Pr[\mathcal{E}_s] + \sum_{s=[\log_{2c}(n)/3]+1}^{\delta n} \Pr[\mathcal{E}_s] \\
\leq o(n^{-1/2}) + \sum_{s=[\log_{2c}(n)/3]+1}^{\delta n} \frac{1}{s^2} = o(1) .
\]

The following lemma conveniently sums up the arguments given in Section 4 of [BK06a]. Roughly speaking, it states that large trees and unicyclic components break into sublinear pieces if we randomly remove a constant fraction of the edges.

**Lemma 4.5 ([BK06a]).** Let \( c > 0 \). For every \( \delta > 0 \) there exists \( \varepsilon = \varepsilon(c, \delta) > 0 \) such that a.a.s. the random graph \( G := G(n, cn) \) has the property that for every subgraph \( H \subseteq G \) in which all components are trees or unicyclic, a.a.s. the graph obtained by removing \( \min\{\delta n, e(H)\} \) edges uniformly at random from \( H \) has only components with \( O(n^{1-\varepsilon}) \) vertices.

4.2.2 Upper Bound Proof for the Unrestricted Setting

Let \( c > c^*_r \). We need to show that a.a.s. every \( r \)-edge-coloring of \( G(n, rcn) \) contains a monochromatic component of linear size.

In [CSW07, FR07] it was shown that \( m = rc^*_r n \) – the threshold for \( r \)-orientability of \( G(n, m) \) – coincides with the threshold for the property that the \((r + 1)\)-core of \( G(n, m) \) has average degree at most \( 2r \) (the \( k \)-core of a graph \( G \) is the maximal subgraph of \( G \) with minimum degree at least \( k \)). For some details, see the introduction of [FR07]. It is known [PSW96] that a.a.s. the \((r + 1)\)-core of \( G(n, rc^*_r n) \) has linear size. By a standard two-round exposure argument this implies that for \( c > c^*_r \), there exists \( \varepsilon > 0 \) such that a.a.s. the \((r + 1)\)-core \( C_{r+1} \) of \( G(n, rcn) \) has average degree at least \((1 + \varepsilon)2r\), i.e., satisfies

\[
|E(C_{r+1})| \geq (1 + \varepsilon)r|V(C_{r+1})| .
\]

(For explicit statements about the number of vertices and edges in cores of random graphs see [CW06, Fou03].) By averaging, a.a.s. in every \( r \)-edge-coloring of \( G(n, rcn) \) there exists a monochromatic subgraph \( H \subseteq G \)
4.2 Proof of Main Result

4.2.3 Lower Bound Proof for the Restricted Setting

Let $c < c_r^*$. We need to show that a.a.s. we can find a valid $r$-edge-coloring of $G^r(n, rcn)$ in which every monochromatic component is of size $O(n^{1-\varepsilon})$ for some constant $\varepsilon$ which only depends on $c$ and $r$.

Similarly to the proof of Theorem 4.1 in [BK06a], we generate $G^r(n, rcn)$ by first generating a slightly denser random $r$-matched graph $G^+$ and then removing a few $r$-sets uniformly at random. Let $\delta = \delta(c, r) > 0$ such that $c + \delta < c_r^*$, and let $G^+ := G^r(n, r(c + \delta)n)$ be a random $r$-matched graph with $r$-sets $E_1, \ldots, E_{(c+\delta)n}$. We let $G$ denote the $r$-matched graph obtained by removing $\delta n$ $r$-sets chosen uniformly at random from $G^+$. Note that, by symmetry, $G$ is distributed exactly as $G^r(n, rcn)$. Thus if we can show that a.a.s. there is a valid $r$-edge-coloring of $G^+$ in which all monochromatic components are trees or unicyclic, the lower bound part of Theorem 4.3 immediately follows by applying Lemma 4.5 (with $c \leftarrow r(c + \delta)$ and $\delta \leftarrow \delta$) in each color class and taking a union bound over all $r$ color classes.

Recall that $m = rc_r^* n$ is the threshold for $r$-orientability of the random graph $G_{n,m}$. Hence, by our choice of $c$, a.a.s. there exists an orientation...
of the underlying unmatched graph of $G^+$ in which every vertex has in-degree at most $r$. Conditioning on this and considering a fixed such orientation $d$, we now look for a partition of $G^+$ into $r$ edge-disjoint Achlioptas subgraphs such that in every color the in-degree of every vertex is at most 1. Note that every edge $e \in E(G^+)$ is oriented towards exactly one vertex and is contained in exactly one of the $r$-sets $E_1, \ldots, E_{(c+\delta)n}$. This naturally defines a bipartite multigraph $B$ with parts $V(G^+)$ and $\{E_1, \ldots, E_{(c+\delta)n}\}$ in which every edge of $G^+$ induces an edge connecting the vertex it is oriented towards with the $r$-set it is contained in (see Figure 4.1 for an example). Clearly, every $r$-set $E_j$ has degree exactly $r$ in $B$, and, since $d$ is an $r$-orientation of $G^+$, every vertex $v \in V(G^+)$ has degree at most $r$ in $B$. Hence, the maximum degree in $B$ is $r$. It is well known (as Hall’s theorem implies that an $r$-regular bipartite (multi)graph can be partitioned into $r$ perfect matchings) that the chromatic index of a bipartite graph equals its maximum degree. Hence there exists a proper $r$-edge-coloring $c_B : E(B) \to \{1, \ldots, r\}$ of $B$. Since $c_B$ is proper, the subsets $P_1^+, \ldots, P_r^+ \subseteq E(G^+)$ corresponding to the color classes of $c_B$ form a partition of $G^+$ into $r$ edge-disjoint Achlioptas subgraphs such that the in-degree of every vertex in each of the parts is at most 1. This implies that for $1 \leq i \leq r$ every component in $P_i^+$ is a tree or unicyclic since a connected graph with two cycles has edge-density strictly larger than 1 and is thus not 1-orientable. Thus we have found a valid $r$-edge-coloring of $G^+$ (with color classes $P_1^+, \ldots, P_r^+$) in which every monochromatic component is a tree or unicyclic, and Lemma 4.5 can be used to complete the proof as outlined above. \qed
Explosive Percolation in Erdős-Rényi-Like Random Graph Processes

In this chapter we consider phase transitions of several random graph processes which are closely related to the classical Erdős-Rényi random graph process. The findings in this chapter are joint work with Konstantinos Panagiotou, Reto Spöhel and Angelika Steger and have been published in Combinatorics, Probability and Computing.
5.1 Introduction

The previous chapter was concerned with the question if one can delay the appearance of the giant component in Ramsey-type variants of the classical random graph process. Instead of studying when the phase transition appears we turn our attention to the behavior of several random graph processes at the transition in this chapter. Recall that in the classical Erdős–Rényi random graph process a linear-sized component appears around \( n/2 \), cf. Theorem 3.1. This phase transition has been studied in great detail (for a survey see chapter 5 in [JLR00]). For every graph \( G \) let \( L_1(G) \) denote the size of its largest component. It is known that \( L_1(G(n,cn)) \) a.a.s. satisfies \( L_1(G(n,cn)) = (f(c) + o(1))n \) for a continuous function \( f(c) \) with \( f(c) = 0 \) for every \( c < 0.5 \) and \( \lim_{c \to 0.5^+} f(c) = 0 \). Thus, the phase transition in the Erdős–Rényi process is continuous\(^1\) (see Figure 5.1). Over the last decade, Achlioptas processes (see Section 3.4 for a precise definition) with various edge selection strategies were studied, mostly motivated by the question if one can accelerate or delay the appearance of the giant component, see e.g. [BF01, SW07].

In a paper in Science from 2009 [ADS09] Achlioptas, D’Souza and Spencer provided strong numerical evidence that the min-product rule (select the edge that minimizes the product of the component sizes of

\[^1\text{In the literature such a phase transition is also called second order while discontinuous phase transitions are called first order.}\]
the endpoints) and min-sum rule (select the edge that minimizes the respective sum) exhibit discontinuous phase transitions (see Figure 5.1), in contrast to a variety of closely related edge-selection rules, in particular the ones analyzed in [SW07]. A discontinuous phase transitions essentially means that for some constant $d > 0$ the size of the largest component a.a.s. ‘jumps’ from $o(n)$ to $dn$ within $o(n)$ edge insertions, that is, at the phase transition a constant fraction of the vertices is accumulated into a single giant component within a sublinear number of steps. This phenomenon is also called explosive percolation. Since it is of great interest, in particular in physics, a series of papers has been devoted to understanding the phase transition of the min-product and min-sum rule (see e.g. [CKK10, CKP+09, DM10, RF10]), most of the arguments not being rigorous but supported by computer simulations.

Countering the numerical evidence it was claimed in [dCDGM10] that the transition is actually continuous. In a recent Science paper [RW11] (see also [RW12]) Riordan and Warnke indeed confirmed this claim with a rigorous proof. In fact, their argument shows continuous phase transitions for an even larger class of processes. Loosely speaking, a random graph process $(G(t))_{t \geq 0}$ is called an $\ell$-vertex rule if $G(0)$ is empty, and $G(t)$ is obtained from $G(t - 1)$ by drawing a set $V_t$ of $\ell$ vertices uniformly at random and adding one or more edges within $V_t$. Informally, this states that graph processes that operate locally in every step (i.e., on a vertex set of constant size) cannot exhibit explosive percolation. Note that Achlioptas processes are essentially 4-vertex rules.

In this chapter we study the characteristics of random graph processes that can cause a discontinuous phase transition. Before stating our results in the forthcoming sections, let us give an informal picture of processes that exhibit explosive percolation. Note that a process, which inserts one edge per step, can only have a discontinuous transition if at some point in time the components with size $\omega(1)$ and $o(n)$ occupy a constant fraction of the vertices (cf. e.g. [RW12]). Employing the terminology of [FL09] these components form a type of ‘powder keg’ that ‘explodes’ at the phase transition. Intuitively speaking, such a powder keg is formed if the process keeps the sizes of the largest components close together and prevents a single component from growing too large before the phase transition occurs. To some extent, the min-product and min-sum rule try to approximate this effect by favoring the construction of smaller components. However, as discussed above, this does not lead to a discontinuous phase transition. In this chapter we consider different
approaches to form a powder-keg.

5.2 Our Results

We introduce three variants of the Erdős–Rényi process and prove discontinuous phase transitions for all of them.

5.2.1 The Half-Restricted Process

First, we introduce a graph process which we call the half-restricted process. The idea is to connect two vertices in every step, but to restrict one of them to be within smaller components.

The half-restricted process has a parameter $0 < \beta \leq 1$ and starts with the empty graph $H_{\beta}(0) = H_{n,\beta}(0)$ on the vertex set $[n]$. In every step $t \geq 1$ we draw an unrestricted vertex $u \in [n]$ uniformly at random and, independently, a restricted vertex $v$ which is drawn uniformly at random from a restricted vertex set $R_{\beta}(H_{\beta}(t-1))$ defined as follows. For every graph $G$ we denote by $R_{\beta}(G)$ the $\lfloor \beta n \rfloor$ vertices in smallest components. Precisely, let $v_1, v_2, \ldots, v_n$ be the vertices of $G$ sorted in increasing order according to the sizes of the components they are contained in (where vertices with the same component size are sorted according to their labels). Then $R_{\beta}(G) := \{v_1, v_2, \ldots, v_{\lfloor \beta n \rfloor}\}$. We obtain $H_{\beta}(t)$ from $H_{\beta}(t-1)$ by inserting an edge between $u$ and $v$ if $u \neq v$ and the edge is not already present (otherwise we do nothing). Note that for $\beta < 1$ the half-restricted process is not an $\ell$-vertex rule.

For a half-restricted process let $\alpha_t$ be the random variable that denotes the maximum size of all components that the restricted vertex can be drawn from in step $t$. Hence, $\alpha_t$ denotes the size of the component of $v_{\lfloor \beta n \rfloor}$ for an ordering $v_1, v_2, \ldots, v_n$ of the vertices according to their component sizes in $H_{\beta}(t-1)$. Clearly, $\alpha_t$ is increasing in $t$. For every positive integer $k$ let $T_k$ denote the last step in which $\alpha_t$ is still below $k$, that is,

$$T_k := \max\{t : \alpha_t < k\} .$$

We show that for any parameter $\beta < 1$ the half-restricted process exhibits explosive percolation. Thus, even though Figure 5.1 suggests that the phase transitions of the min-product (or min-sum) rule and that of
the half-restricted process behave similarly, their mathematical structure is fundamentally different. More precisely, we show that around the step in the half-restricted process when the number of vertices in components of constant size drops below $\beta n$ a giant component of size almost $(1 - \beta)n$ is created within a sublinear number of steps.

**Theorem 5.1.** Let $0 < \beta < 1$. For every $K = K(n)$ with $K \geq (\ln n)^{1.02}$, every $C = C(n)$ with $C = \omega(1)$ and $C = o(\ln K)$, and every $\varepsilon > 0$ there exists a constant $c = c(\beta, \varepsilon)$ such that a.a.s.

1. $L_1(\mathcal{H}_\beta(T_C)) \leq K$, and
2. $L_1(\mathcal{H}_\beta(T_C + n/c\sqrt{C})) \geq (1 - \varepsilon)(1 - \beta)n$.

Note that for $K = (\ln n)^{1.02}$ and $C = (\ln \ln n)^{0.99}$ the theorem states that, a.a.s., the number of steps from the first appearance of a component of size $(\ln n)^{1.02}$ to the appearance of a component of size $(1 - \varepsilon)(1 - \beta)n$ is $O(n/(\ln \ln n)^{0.49}) = o(n)$.

### 5.2.2 The Component Process

In this subsection we introduce a graph process which we call the component process. We start with the empty graph $C(0)$. In the $t$-th step (for every integer $t \geq 1$) we obtain the graph $C(t)$ from $C(t - 1)$ by drawing a pair of components uniformly at random from the set of all components in $C(t - 1)$, and inserting an arbitrary edge between them. In this way, after $n - 1$ steps we obtain a tree $C(n - 1)$ which connects all $n$ vertices, and the process then stops. This process is closely related to what is called Smoluchowski’s coagulation equation with constant kernel in the physics literature, see e.g. [Ald99, DT00, KP94].

**Theorem 5.2.** For every $\varepsilon > 0$ there exists a constant $K = K(\varepsilon) > 0$ such that a.a.s. $L_1(C((1 - \varepsilon)n)) \leq K \ln n$.

Observe that the graph of the component process is connected after $n - 1$ steps. Hence, Theorem 5.2 implies that the component process exhibits a discontinuous phase transition.

### 5.2.3 The Mixed Process

We also consider a mixture of the Erdős–Rényi and the component processes, which we call the mixed process. In this process, we start with
the empty graph $\mathcal{M}(0)$ and in the $t$-th step (for every integer $t \geq 1$) we obtain the graph $\mathcal{M}(t)$ from $\mathcal{M}(t-1)$ as follows. We draw a vertex $v$ uniformly at random from the set of all vertices and a component uniformly at random from the set of all components in $\mathcal{M}(t-1)$ except the one that $v$ is contained in. We then insert an arbitrary edge between them.

**Theorem 5.3.** For every $\varepsilon > 0$ there exists a constant $K = K(\varepsilon) > 0$ such that a.a.s. $L_1(\mathcal{M}((1-\varepsilon)n)) \leq K \ln n$.

Observe that the graph of the mixed process is connected after $n-1$ steps. Hence, Theorem 5.3 implies that the mixed process exhibits a discontinuous phase transition.

### 5.3 Proof for the Half-Restricted Process

In this section we prove Theorem 5.1. In our proof we need a technical lemma introduced in the following.

Let $N$ be a positive integer, and for every $0 \leq i < N$ let $X_i \sim \text{Geom}(\frac{N-i}{N})$ be a geometrically distributed random variable with parameter $\frac{N-i}{N}$ and set $X(a,b) := \sum_{i=a}^{b} X_i$ for every $0 \leq a < b < N$. All subsequent statements and arguments are about sums of geometrically distributed random variables, but we will use that they have the following combinatorial interpretation in a coupon collector scenario with $N$ coupons. (In a coupon collector scenario, we have a number of different coupons and repeatedly draw one uniformly at random with replacement. We are interested in how often we have to draw until we have seen every coupon.) Observe that $X_i$ is distributed like the number of coupons we draw while holding exactly $i$ different coupons, waiting for the $(i+1)$st, and thus $X(a,b)$ can be viewed as the number of coupons we draw while holding at least $a$ and at most $b$ different coupons.

Note that

$$\mathbb{E}[X(a,b)] = \sum_{i=a}^{b} \frac{N}{N-i} = N \sum_{i=a}^{N-a} \frac{1}{i} = N(H_{N-a} - H_{N-b-1}) , \quad (5.1)$$

where $H_n = \sum_{i=1}^{n} 1/i$ denotes the $n$-th harmonic number for every $n \geq 1$. It is well-known that

$$H_n = (1 + o(1)) \ln n . \quad (5.2)$$
Lemma 5.4. Let \( k = k(N) = \omega(1) \) and \( s = s(N) \) with \( s = o(N \ln k) \). Then for \( N \) large enough

\[
\Pr[X(N - k, N - 2) \leq s] \leq e^{-k^{0.99}} .
\]

Proof. First note that by using (5.1), (5.2) and \( k = \omega(1) \) we have

\[
\mathbb{E}[X(N - k, N - 2)] = N(H_k - H_1) = (1 + o(1))N \ln k .
\]

Thus, \( s = o(\mathbb{E}[X(N - k, N - 2)]) \). Consider a coupon collector scenario with \( N \) different coupons of which we have already seen \( N - k \). For each of the remaining \( k \) coupons let \( Y_j \) (where \( 1 \leq j \leq k \)) denote the indicator random variable for the event that it is not drawn within the next \( s \) trials. By the comments preceding this lemma, the probability of the event \( X(N - k, N - 2) \leq s \) equals the probability that at most one of these coupons is not drawn within the \( s \) trials. Hence, for \( Y := \sum_{j=1}^{k} Y_j \) we observe that

\[
\Pr[X(N - k, N - 2) \leq s] = \Pr[Y \leq 1] .
\] (5.3)

Using the identity \( 1 - x \geq e^{-2x} \), which holds for all \( 0 \leq x \leq 1/2 \), we have for every \( 1 \leq j \leq k \) that

\[
\mathbb{E}[Y_j] = \Pr[Y_j = 1] = \left(1 - \frac{1}{N}\right)^s \geq e^{-\frac{2s}{N}} ,
\]

and thus, for \( N \) large enough,

\[
\mathbb{E}[Y] \geq ke^{-\frac{2s}{N}} = k^{1 - \frac{2s}{N \ln k}} \geq 8k^{0.99} ,
\]

where we used \( s = o(N \ln k) \) in the last step. One can check that the random variables \( Y_1, Y_2, \ldots, Y_k \) are negatively associated (see e.g. Chapter 3 in [DP09]). Hence, we can apply Chernoff bounds to \( Y \) and obtain for \( N \) large enough that

\[
\Pr[Y \leq 1] \leq \Pr\left[Y \leq \left(1 - \frac{1}{2}\right) \mathbb{E}[Y]\right] \leq e^{-\mathbb{E}[Y]/8} \leq e^{-k^{0.99}} ,
\]

which together with (5.3) finishes the proof. \( \square \)

We now turn to the proof of Theorem 5.1.
Proof of Theorem 5.1. We fix \( \beta < 1, K = K(n) \) with \( K \geq (\ln n)^{1.02} \), \( C = C(n) \) with \( C = \omega(1) \) and \( C = o(\ln K) \), and \( \varepsilon > 0 \). To simplify notation we write \( \mathcal{H}(t) \) instead of \( \mathcal{H}_\beta(t) \).

We first address (i). We need to show that at the step when the restricted vertex can be in a component of size \( C \) for the first time, there is a.a.s. no component of size larger than \( K \). The main idea is that a large component can only form if the unrestricted vertex is drawn from this component so often that this is unlikely to happen within \( T_C \) steps.

We first note that it is not hard to show that \( T_C \leq 4n \) a.a.s. (see e.g. Lemma 3 and the remark following its proof in [RW12]). Let \( E^* \) denote this event.

Note that up to step \( T_C \) two components of size at least \( C \) can never be merged by an edge since the restricted vertex in every step is drawn from vertices in components of size less than \( C \). Hence, we can easily keep track of the components of size at least \( C \) and call them chunks. Let \( A_1, A_2, \ldots \) denote all chunks in order of appearance during the process, where a chunk keeps its label if merged with another component, and the new component is not inserted into the list. Clearly, there can be at most \( n/C \) chunks. For every \( 1 \leq i \leq n/C \) we denote by \( E_i \) the event that chunk \( A_i \) has size larger than \( K \) in \( \mathcal{H}(T_C) \). We will show that \( \Pr[E_i \cap E^*] \leq 1/n \) for every \( 1 \leq i \leq n \). By applying a union bound this implies

\[
\Pr[\mathcal{H}(T_C) \text{ contains a comp. of size } > K] = \Pr \left[ \bigcup_{i=1}^{n/C} E_i \cap E^* \right] + \Pr[\overline{E}^*] \\
\leq \frac{n}{C} \cdot \frac{1}{n} + o(1) = o(1) .
\]

(5.4)

It remains to bound \( \Pr[E_i \cap E^*] \) for every \( 1 \leq i \leq n/C \). Let \( 1 \leq i \leq n/C \) be fixed for the remainder of the proof. Clearly, a chunk has size at most \( 2C \) when it appears. Moreover, since for every \( t \leq T_C \) the restricted vertex is drawn from vertices in components of size smaller than \( C \), the chunk \( A_i \) can grow by at most \( C \) in every step. Hence, the chunk has size at most \( (j + 1) \cdot C \) before a vertex from the chunk is drawn for the \( j \)th time. Hence, a vertex from chunk \( A_i \) needs to be drawn in at least \( K/C - 1 \) steps after its appearance for \( E_i \) to occur. Let \( X_2, X_3, \ldots, X_{K/C} \) denote the number of steps between steps in which we draw a vertex from \( A_i \). That is, \( X_2 \) is the number of steps from the appearance of
chunk $A_i$ until a vertex from $A_i$ is drawn for the first time, $X_3$ is the time from that step until a vertex from $A_i$ is drawn for the second time, and so on. Furthermore, let $X := \sum_{j=2}^{K/C} X_j$. Then,

$$\Pr[\mathcal{E}_i \cap \mathcal{E}^*] \leq \Pr[\{X \leq T_C\} \cap \mathcal{E}^*]. \quad (5.5)$$

Recall that for time steps $t \leq T_C$ only the unrestricted vertex in every step can be in $A_i$. In the period $t \leq T_C$ we thus have that $X_j$ conditioned on $X_1, X_2, \ldots, X_{j-1}$ is geometrically distributed with parameter at most $jC/n$. Therefore we can bound $\Pr[\{X \leq T_C\} \cap \mathcal{E}^*]$ as follows. Let $Y_j$ be an independent random variable with $Y_j \sim \text{Geom}(jC/n)$, and set $Y := \sum_{j=2}^{K/C} Y_j$. Then

$$\Pr[\{X \leq T_C\} \cap \mathcal{E}^*] \leq \Pr[\{Y \leq T_C\} \cap \mathcal{E}^*] \leq \Pr[Y \leq 4n]. \quad (5.6)$$

Here, we use Lemma 5.4 with $N = n/C$, $k = K/C$ and $s = 4n$. Concerning the prerequisites of the lemma, we have $k = \omega(1)$ since $C = o(K)$. Furthermore, we have $s = o(N \ln k)$ since $C = o(\ln K)$ and thus

$$N \ln k = \frac{n}{C} \cdot \ln \left(\frac{K}{C}\right) = (1 - o(1)) \frac{n \ln K}{C} = \omega(n).$$

Hence, Lemma 5.4 gives us for large enough $n$ that

$$\Pr[Y \leq 4n] \leq e^{-(K/C)^{0.99}}.$$ 

Since $K \geq (\ln n)^{1.02}$ and $C = o(\ln K)$ we have

$$\Pr[Y \leq T_C] \leq e^{-\ln n} = \frac{1}{n}.$$ 

Using (5.4), (5.5) and (5.6) this settles the proof of (i), and it remains to prove (ii), i.e., we have to show that for an appropriate constant $c = c(\beta, \varepsilon)$, which we will specify later, we have that $\mathcal{H}(T_C + n/c\sqrt{C})$ contains a component of size $(1 - \varepsilon)(1 - \beta)n$ with high probability.

We set $a := n/(2c\sqrt{C})$ and split the proof into two parts. In the first part (the first $a$ additional steps after $T_C$) we collect a suitable amount of vertices in components of size at least $C$, and in the second part (the remaining $a$ steps) we actually build a giant component on these vertices.

Consider steps $T_C + 1$ to $T_C + a$ in the graph process. For every $t \geq 1$ let $U(t)$ denote the set of vertices in components of size at least $C$ in
\( \mathcal{H}(t) \). Note that by definition of \( T_C \) we have \(|U(T_C + 1)| \geq (1 - \beta)n\). We now show that with high probability we have

\[
|U(T_C + a)| \geq (1 - \beta)n + \frac{1 - \beta}{8c\sqrt{C}}n. \tag{5.7}
\]

Clearly, this holds if \(|U(T_C + a)| \geq (1 - \beta/2)n\) and we thus assume \(|U(T_C + i)| \leq (1 - \beta/2)n\) for every \(1 \leq i \leq a\) in the remainder.

For every \(1 \leq i \leq a\) let \( X_i := |U(T_C + i)| - |U(T_C + i - 1)| \) denote the number of vertices added to the components of size at least \( C \) in the \( i \)th additional step. Furthermore, let \( X := \sum_{i=1}^{a} X_i \). We prove a lower bound on the probability that \( X_i \) contributes at least 1 vertex. Clearly, \( X_i \geq 1 \) if the unrestricted vertex is drawn from components of size at least \( C \), which by definition of \( T_C \) happens with probability at least \( 1 - \beta \), and if the restricted vertex is drawn from components of size smaller than \( C \), which happens with probability at least \( \frac{(\beta/2)n}{\beta n} = 1/2 \) since we assume \(|U(T_C + i)| \leq (1 - \beta/2)n\). Hence, for every \(1 \leq i \leq a\) we have \( \Pr[X_i \geq 1] \geq (1 - \beta)/2 \) and thus also \( \mathbb{E}[X] \geq a(1 - \beta)/2 = (1 - \beta)n/(4c\sqrt{C}) \). Using Chernoff bounds we obtain

\[
\Pr \left[ X < \frac{1}{2} \cdot \frac{(1 - \beta)n}{4c\sqrt{C}} \right] \leq e^{-\Theta(\frac{n}{\sqrt{C}})} = o(1),
\]

which establishes (5.7).

We now look at the second half of additional steps, i.e., steps \( T_C + a + 1 \) to \( T_C + 2a \), and show that a.a.s. in these steps a sufficiently large component is created within \( U := U(T_C + a) \). Note that \( U \) is a fixed set of vertices which does not change from step to step.

Let \( \mathcal{E} \) denote the event that \( \mathcal{H}(T_C + 2a) \) has no component of size \((1 - \varepsilon)(1 - \beta)n\) conditioned on (5.7). We now show that \( \Pr[\mathcal{E}] = o(1) \), which clearly finishes the proof. Observe that \( \mathcal{E} \) can only occur if for every \( a + 1 \leq i \leq 2a \) we have that the largest component in \( \mathcal{H}(T_C + i) \) has size less than \((1 - \varepsilon)(1 - \beta)n\). In the following we bound the probability for this to happen.

We call a step successful if it connects two components in \( U \). Since every component in \( U \) has size at least \( C \) we have that \( n/C \) successful steps will connect all components in \( U \) such that \( U \) forms one giant component of size at least \((1 - \beta)n\). We now compute the probability to have a successful step conditioned on that we do not have a component of size \((1 - \varepsilon)(1 - \beta)n\). For a successful step, the restricted vertex needs
5.4. Proof for the Component Process

In this section we prove Theorem 5.2. Our proof relies on the following lemma, which bounds the probability that the component process contains a component of size exactly $k$ after exactly $t$ steps.

**Lemma 5.5.** For every $\varepsilon > 0$, $k \geq 1$ and $1 \leq t \leq (1-\varepsilon)n$ the probability that $C(t)$ contains a component of size exactly $k$ is for $n$ large enough at most

$$ne^{-(k-1)\varepsilon + 13k^2/(\varepsilon n)}.$$

**Proof.** Let $\varepsilon > 0$, let $k \geq 1$ be fixed and let $1 \leq t \leq (1-\varepsilon)n$. We use a union bound argument. We first count the number of possible ways to create a component of size $k$ within $t$ steps. First, there are $\binom{n}{k}$ ways to choose the vertices of the component. Let us fix such a set $S$ of $k$ vertices. We now estimate the number of ways that a component on $S$ can be created. Recall that at any time of the process the graph generated by the component process is a forest. In particular, if $S$ is a component after $t$ steps then $S$ forms a tree and there are exactly $k-1$ edges within $S$. In the beginning, $S$ consists of $k$ separate singleton vertices to be in $U$ and the unrestricted vertex needs to be drawn from a different component in $U$. This happens with probability at least

$$\frac{|U| - (1-\beta)n}{\beta n} \cdot \frac{|U| - (1-\varepsilon)(1-\beta)n}{n} \geq \frac{1-\beta}{8\beta c\sqrt{C}} \cdot \varepsilon (1-\beta) =: p,$$

where we used (5.7) to bound the first factor. Thus, setting $c := (1-\beta)\sqrt{\varepsilon/(32\beta)}$ the number $S$ of successful steps satisfies

$$\mathbb{E}[S] \geq ap = \frac{n}{2c\sqrt{C}} \cdot \frac{\varepsilon (1-\beta)^2}{8\beta c\sqrt{C}} \geq \frac{2n}{C},$$

and by Chernoff bounds

$$\Pr[S \leq \frac{n}{C}] \leq \Pr[S \leq \frac{1}{2}\mathbb{E}[S]] \leq e^{-\mathbb{E}[S]/8} = e^{-\Omega(n^2)} = o(1).$$

Hence, we a.a.s. have at least $n/C$ successful steps conditioned on that we do not have a component of size $(1-\varepsilon)(1-\beta)n$. Together with the considerations before (5.8) this implies that $\Pr[\mathcal{E}] = o(1)$. \qed

5.4 Proof for the Component Process

In this section we prove Theorem 5.2. Our proof relies on the following lemma, which bounds the probability that the component process contains a component of size exactly $k$ after exactly $t$ steps.

**Lemma 5.5.** For every $\varepsilon > 0$, $k \geq 1$ and $1 \leq t \leq (1-\varepsilon)n$ the probability that $C(t)$ contains a component of size exactly $k$ is for $n$ large enough at most

$$ne^{-(k-1)\varepsilon + 13k^2/(\varepsilon n)}.$$

**Proof.** Let $\varepsilon > 0$, let $k \geq 1$ be fixed and let $1 \leq t \leq (1-\varepsilon)n$. We use a union bound argument. We first count the number of possible ways to create a component of size $k$ within $t$ steps. First, there are $\binom{n}{k}$ ways to choose the vertices of the component. Let us fix such a set $S$ of $k$ vertices. We now estimate the number of ways that a component on $S$ can be created. Recall that at any time of the process the graph generated by the component process is a forest. In particular, if $S$ is a component after $t$ steps then $S$ forms a tree and there are exactly $k-1$ edges within $S$. In the beginning, $S$ consists of $k$ separate singleton vertices to be in $U$ and the unrestricted vertex needs to be drawn from a different component in $U$. This happens with probability at least

$$\frac{|U| - (1-\beta)n}{\beta n} \cdot \frac{|U| - (1-\varepsilon)(1-\beta)n}{n} \geq \frac{1-\beta}{8\beta c\sqrt{C}} \cdot \varepsilon (1-\beta) =: p,$$

where we used (5.7) to bound the first factor. Thus, setting $c := (1-\beta)\sqrt{\varepsilon/(32\beta)}$ the number $S$ of successful steps satisfies

$$\mathbb{E}[S] \geq ap = \frac{n}{2c\sqrt{C}} \cdot \frac{\varepsilon (1-\beta)^2}{8\beta c\sqrt{C}} \geq \frac{2n}{C},$$

and by Chernoff bounds

$$\Pr[S \leq \frac{n}{C}] \leq \Pr[S \leq \frac{1}{2}\mathbb{E}[S]] \leq e^{-\mathbb{E}[S]/8} = e^{-\Omega(n^2)} = o(1).$$

Hence, we a.a.s. have at least $n/C$ successful steps conditioned on that we do not have a component of size $(1-\varepsilon)(1-\beta)n$. Together with the considerations before (5.8) this implies that $\Pr[\mathcal{E}] = o(1)$. \qed
components and there are \( \binom{k}{2} \) ways to connect two of them. In general, after inserting \( i \) edges within \( S \), there are \( k-i \) components in \( S \) and thus \( \binom{k-i}{2} \) ways to connect two of them in the component process. Hence, the number of ways to create a component on \( S \) is
\[
\binom{k}{2} \binom{k-1}{2} \cdots \binom{2}{2} = \frac{k!(k-1)!}{2^{k-1}}.
\]

It now remains to choose the steps \( 1 \leq t_1 < t_2 < \cdots < t_{k-1} \leq t \) in which we insert an edge in \( S \). Observe that this can be done in \( \binom{t}{k-1} \leq \frac{t^{k-1}}{(k-1)!} \) ways. Altogether, the number of ways to create a component of size \( k \) is at most
\[
n^k \cdot \frac{k!(k-1)!}{2^{k-1}} \cdot \frac{t^{k-1}}{(k-1)!} = n^k 2^{-\binom{k-1}{2} t^{k-1}}.
\]  

Having fixed \( S \) and the order and steps in which we connect components in \( S \), let us estimate the probability of the event that \( S \) forms a component in this way. Set \( t_0 = 0 \) and \( t_k = t+1 \). For every \( 1 \leq i \leq k-1 \) the probability to choose the fixed component pair in step \( t_i \) is \( \frac{1}{n-t_i+1} \) since there are \( n-t_i + 1 \) components to choose from in step \( t_i \). Moreover, in every step \( j \notin \{t_1, \ldots, t_{k-1}\} \) two components outside of \( S \) need to be chosen, which for every \( t_i < j < t_{i+1} \) happens with probability \( \frac{(n-j+1-(k-i))}{(n-j+1)} \leq 1 - (k-i)/(n-j+1) \leq \exp(-2(k-i)/(n-j+1)) \). Hence, the probability that between steps \( t_i \) and \( t_{i+1} \) only components outside of \( S \) are chosen is bounded by \( \exp(-2(k-i) \cdot (H_{n-t_i} - H_{n-t_{i+1}})) \). Altogether, we obtain that the probability to create a component on \( S \) in a fixed way is at most
\[
\left( \prod_{i=1}^{k-1} \frac{1}{n-t_i+1} \right) \cdot \left( \prod_{i=0}^{k-1} e^{-2(k-i) \cdot (H_{n-t_i} - H_{n-t_{i+1}})} \right)
\leq 2^{k-1} \exp \left( -2 \sum_{i=1}^{k-1} \ln(n-t_i-1) 
- 2 \sum_{i=0}^{k-1} (k-i)(H_{n-t_i-1} - H_{n-t_{i+1}-2}) + \frac{6k^2}{n-t_k-1} \right),
\]  

where the term \( 6k^2/(n-t_k-1) \) accounts for replacing \( H_{n-t_{i+1}} \) by \( H_{n-t_{i+1}-2} \). It is easy to see that
\[
H_a - H_b \geq \ln \left( \frac{a}{b+1} \right) \quad \text{for all } a, b \in \mathbb{N}, a > b.
\]
Thus, using \( t_0 = 0 \) we deduce for the second sum in (5.10) that
\[
-2 \sum_{i=0}^{k-1} (k - i)(H_{n-t_i-1} - H_{n-t_{i+1}-2}) \\
\leq -2 \sum_{i=0}^{k-1} (k - i) \cdot (\ln(n - t_i - 1) - \ln(n - t_{i+1} - 1)) \\
= -2k \ln(n - 1) + 2 \sum_{i=1}^{k-1} \ln(n - t_i - 1) + 2 \ln(n - t_k - 1) .
\]
(5.11)

Using \( n - t_k - 1 = n - t - 2 \geq \varepsilon n/2 \) for \( n \) large enough we thus obtain an upper bound for (5.10) of
\[
2^{k-1} \exp \left( -2k \ln(n - 1) + \frac{12k^2}{\varepsilon n} + 2 \ln(\varepsilon n/2) \right) \leq 2^{k-1} n^{-(2k-1)e^{13k^2}/(\varepsilon n)} . \]
(5.12)

Putting together (5.9) and (5.12) and using \( t \leq (1 - \varepsilon)n \) we obtain that the probability that \( C(t) \) contains a component of size exactly \( k \) is at most
\[
n^k 2^{-(k-1)} t^{k-1} \cdot 2^{k-1} n^{-(2k-1)e^{13k^2}/(\varepsilon n)} \leq n e^{-(k-1)\varepsilon + 13k^2/(\varepsilon n)} . \]

**Proof of Theorem 5.2.** We use a union bound argument. Let \( \varepsilon > 0 \) and \( K = 3/\varepsilon \). Clearly, the component process cannot contain a component of size larger than \( K \ln n \) after exactly \( (1 - \varepsilon)n \) steps if it does not contain a component of size between \( K \ln n + 1 \) and \( 2K \ln n \) at any time \( t \in \{1, 2, \ldots, (1 - \varepsilon)n\} \). For every \( k \geq 1 \) and \( 1 \leq t \leq (1 - \varepsilon)n \) let \( \mathcal{E}(k, t) \) denote the event that the component process contains a component of size exactly \( k \) after exactly \( t \) steps. Then we have by Lemma 5.5 that
\[
\Pr \left[ \bigcup_{K \ln n + 1 \leq k \leq 2K \ln n} \mathcal{E}(k, t) \right] \leq \sum_{k=K \ln n + 1}^{2K \ln n} \sum_{t=1}^{(1-\varepsilon)n} n e^{-(k-1)\varepsilon + 6k^2/(\varepsilon n)} \\
\leq (1 - \varepsilon)K \ln n \cdot n^{2-K\varepsilon + o(1)} = o(1) . \]
\[\square\]
5.5 Proof for the Mixed Process

In this section we prove Theorem 5.3. The proof proceeds similarly to the proof for the component process (cf. Section 5.4). It relies on a lemma which bounds the probability that the mixed process contains a component of size exactly \( k \) after exactly \( t \) steps.

**Lemma 5.6.** For every \( \varepsilon > 0 \), \( k \geq 1 \) and \( 1 \leq t \leq (1 - \varepsilon)n \) the probability that \( \mathcal{M}(t) \) contains a component of size exactly \( k \) is for \( n \) large enough at most
\[
ne^{1-(k-1)\varepsilon^2/2+13k^2/(\varepsilon n)}.
\]

**Proof.** Let \( \varepsilon > 0 \), let \( k \geq 1 \) be fixed and let \( 1 \leq t \leq (1 - \varepsilon)n \). Similar to the component process (cf. Lemma 5.5) we apply a union bound argument. We first count the number of possible ways to create a component of size \( k \) within \( t \) steps. As for the component process there are \( \binom{n}{k} \leq \frac{n^k}{k!} \) ways to choose the vertex set \( S \) of such a component, and \( \binom{t}{k-1} \leq \frac{t^{k-1}}{(k-1)!} \) possibilities to choose the steps \( 1 \leq t_1 < t_2 < \cdots < t_{k-1} \leq t \) in which we connect two components in \( S \). (Note that the mixed process also satisfies that for every \( i \geq 0 \) the graph \( \mathcal{M}(i) \) is a forest.)

Observe that just before step \( t_i \) the set \( S \) consists of \( k - (i - 1) \) components two of which are connected in step \( t_i \). Now, for the choice of the vertex in step \( t_i \) we have \( k \) possibilities and for the choice of the component we have \( k - (i - 1) - 1 = k - i \) possibilities (all components in \( S \) are fine except the one that the chosen vertex is contained in). Hence, after fixing the set \( S \) and the steps \( t_1, \ldots, t_{k-1} \) there are
\[
k(k-1) \cdot k(k-2) \cdots k \cdot 1 = k^{k-1}(k-1)!
\]
possibilities to create a component on \( S \). Using \( k! \geq (k/e)^k \) this yields altogether at most
\[
\frac{n^k}{k!} k^{k-1}(k-1)! \cdot \frac{t^{k-1}}{(k-1)!} \leq (ne)^k e^{(k-1)\ln t}
\]
ways to create a component of size exactly \( k \) in exactly \( t \) steps.

Having fixed \( S \) and the steps and way in which we connect components in \( S \), let us estimate the probability of this event. Set \( t_0 = 0 \) and \( t_k = t + 1 \). First observe that before step \( j \) the graph has exactly \( n - j + 1 \) components, and the component drawn in that step is drawn
from a set of \( n - j \) components, namely all \( n - j + 1 \) components of the
graph except the one that contains the vertex drawn in that step. Thus,
for every \( 1 \leq i \leq k - 1 \) the probability to choose the fixed vertex and
component in step \( t_i \) is \( 1/n \cdot 1/(n-t_i) \). (If you read this and send an email
to thomahen at gmail dot com with the subject ‘powder keg exploded’,
you will be greatly rewarded.) Moreover, in every step \( t_i < j < t_{i+1} \) we
need to choose a vertex outside of \( S \) together with a component outside
of \( S \). This happens with probability \( (1 - \frac{k}{n}) \cdot (1 - \frac{k-i}{n-j}) \).
Altogether, we obtain (similarly to (5.10)) that the probability to create a component
on \( S \) in a fixed way is at most

\[
\prod_{i=1}^{k-1} \left( \frac{1}{n} \cdot \frac{1}{n-t_i} \right) \cdot \prod_{i=0}^{k-1} \prod_{j=t_{i+1}}^{t_i+1} \left( 1 - \frac{k}{n} \right) \left( 1 - \frac{k-j}{n-j} \right)
\leq n^{-(k-1)} \exp \left( - \sum_{i=1}^{k-1} \ln(n-t_i) - \frac{k}{n} (t - (k - 1)) \right.
onumber
\]

\[
- \sum_{i=0}^{k-1} (k - i)(H_{n-t_{i+1}} - H_{n-t_i}) + \left( \frac{2k^2}{n-t_k} - 1 \right) \right). \quad (5.14)
\]

Estimating the last sum in the exponent similar to (5.10) in the previous
section (cf. (5.11), the only difference is a missing factor of 2) we obtain
that (5.14) is at most

\[
n^{-(k-1)} \exp \left( - \frac{k}{n} (t - (k - 1)) - k \ln(n - 1) + \ln(\varepsilon n/2) + \frac{4k^2}{\varepsilon n} \right) \nonumber
\]

\[
\leq n^{-(k-1)} e^{-kt/n-(k-1) \ln n + 6k^2 / (\varepsilon n)} . \quad (5.15)
\]

Combining (5.13) and (5.15), and using \( t \leq (1-\varepsilon)n \) and \( 1-x \leq e^{-x-x^2/2} \)
we obtain that the probability that \( M(t) \) contains a component of size
exactly \( k \) is at most

\[
(n \varepsilon k)^{k-1} e^{(k-1) \ln t} \cdot n^{-(k-1)} e^{-kt/n-(k-1) \ln n + 6k^2 / (\varepsilon n)}
= n e^{k+(k-1) \ln(1-(k-t)/n)-kt/n+6k^2 / (\varepsilon n)}
\leq n e^{k-(k-1)(1+\ln(\frac{n-t}{2n^2})) + 6k^2 / (\varepsilon n)}
\leq n e^{1-(k-1)\varepsilon^2/2 + 6k^2 / (\varepsilon n)} . \quad \square
\]
Proof of Theorem 5.3. The proof can be done similarly to the one of Theorem 5.2 if we set $K = 5/\varepsilon^2$ instead of $3/\varepsilon$. □
In this chapter we present our results for an offline Ramsey game in random hypergraphs where Painter’s goal is to avoid a given fixed sub-hypergraph. This is joint work with Luca Gugelmann, Yury Person and Angelika Steger.

6.1 Introduction

As outlined in Section 3.4.1 Ramsey properties of random graphs were studied intensely in the beginning of the 1990ies. The problem of avoiding an arbitrary given graph $F$ in edge-colorings of random graphs was solved in full generality by Rödl and Ruciński with a small addition of 55
Friedgut and Krivelevich, cf. Theorem 3.4.

Note that stars and, in the case of 2 colors also paths of length 3, are excluded from the theorem’s statement. These two graphs have an exceptional threshold behavior. In the case of a star $S_\ell$ with $\ell$ rays it can be shown that the threshold is determined by the appearance of a star $S_{\ell,r}^* = S_{r(\ell-1)+1}$ with $r(\ell-1)+1$ rays. Clearly, the edges of $S_{\ell,r}^*$ cannot be colored with $r$ colors without a monochromatic copy of $S_\ell$, see Section 6.5.1 for more details. For paths $P_3$ with three edges one can show that the 0-statement only holds if $p \ll \frac{n^{-1/m_2(P_3)}}{2}$ (instead of $p \leq cn^{-1}$ as in the statement of Theorem 3.4). The reason is given by the so-called sunshine graphs $S_{\ell}^\solar$ for odd $\ell \geq 5$ (see Figure 6.1) which cannot be edge-colored with 2 colors without a monochromatic $P_3$ and which are sparse enough to be present in $G(n,p)$ for $p$ at the desired threshold. This special case was found by Friedgut and Krivelevich [FK00]. We give more details in Section 6.5.2. Observe that for both exceptional structures the threshold is defined by a finite ‘counterexample’ formed by a sparse graph which cannot be colored without a monochromatic copy of $F$.

A generalization of the 1-statement of Theorem 3.4 to $k$-uniform hypergraphs was first shown for special cases [RR98, RRS07]. A general result for the avoidance of arbitrary $k$-uniform hypergraphs was recently proved by Friedgut, Rödl and Schacht [FRS10] and, independently, by Conlon and Gowers [CG], cf. Theorem 3.5.

Recall that the intuition behind the condition $p \geq Cn^{-1/m_k(F)}$ in Theorem 3.5 is the following. We denote by $H^{(k)}(n,p)$ a random $k$-uniform hypergraph, that is, a hypergraph on $n$ vertices in which each of the $\binom{n}{k}$ possible hyperedges is present independently with probability $p$. Under the assumption $m_k(F) = d_k(F)$ we expect on a fixed hyperedge in

![Figure 6.1: The sunshine graph $S_5^\solar$ of edge-density 1 whose edges cannot be 2-colored without a monochromatic path of length 3. (Observe that a similar construction with a cycle of any given odd length also has this property.)](image)
6.1. Introduction

\(H^{(k)}(n,p)\) order of \(n^{v(F) - k} p^{e(F) - 1} = \Theta(1)\) copies of \(F\). Hence, if \(C\) is large we expect many copies of \(F\) on a fixed hyperedge and these copies overlap so heavily that we cannot find a hyperedge-coloring without a monochromatic copy of \(F\).

6.1.1 Our Results

In this chapter we study the corresponding 0-statement, that is, the question for which values of \(p\) we have

\[
\lim_{n \to \infty} \Pr \left[ H^{(k)}(n,p) \to (F)^c_r \right] = 0.
\]

It was conjectured in [FRS10] that for 'most' hypergraphs \(F\) such a 0-statement should hold if \(p \leq \frac{cn^{-1/m_k(F)}}{m_k(F)}\) for a suitably small constant \(c > 0\), thus providing a matching 0-statement to Theorem 3.5. In this chapter we provide such a 0-statement for a large class of hypergraphs. On the other hand, we also give examples of hypergraphs where the threshold is not at \(n^{-1/m_k(F)}\). Before we give the formal statements of our results we motivate their main ingredients.

Our approach to obtain a 0-statement extends ideas from [MSSS09] and [GPST12]. It is based on a procedure to find a 'valid' hyperedge-coloring, i.e., one without monochromatic copies of \(F\). In order to show that this procedure is successful for \(p \leq \frac{cn^{-1/m_k(F)}}{m_k(F)}\) it is crucial that \(F\) is 'structurally well-behaved'. More precisely, we rely on the fact that \(F\) is strictly \(k\)-balanced, that is, for every proper subhypergraph \(J \subseteq F\) we have \(d_k(J) < m_k(F)\). If this is not the case, then we replace \(F\) by a strictly \(k\)-balanced subhypergraph \(F' \subseteq F\) which satisfies \(m_k(F') = m_k(F)\) and use the procedure to find a hyperedge-coloring without monochromatic copies of \(F'\). (It is not hard to see that such a subhypergraph always exists, e.g. by considering the set of all subhypergraphs \(J \subseteq F\) that satisfy \(d_k(J) = m_k(F)\).

As mentioned above there are two exceptions in the case of simple graphs which both stem from finite 'counterexamples'. We first formalize the notion of such counterexamples. From Ramsey's theorem (cf. Theorem 3.3) we know that for all hypergraphs \(F\) there exist hypergraphs \(H\) that cannot be colored without generating a monochromatic copy of \(F\). For example, all sufficiently large complete hypergraphs have this property. In many cases these counterexamples are irrelevant in our context, as they are too dense to appear in the random hypergraph at
Chapter 6. Ramsey Properties of Random Hypergraphs

the critical density. Crucial, however, are those hypergraphs that cannot be colored without avoiding a monochromatic copy of $F$ and which are sparse enough so that they appear in $H^{(k)}(n, p)$ with positive probability at the conjectured threshold. Regarding the appearance of a fixed subgraph in a random hypergraph it is known that for every hypergraph $H$ we have

$$\lim_{n \to \infty} \Pr \left[ H \subseteq H^{(k)}(n, p) \right] = \begin{cases} 0 & \text{if } p \ll n^{-1/m(H)}, \\ c' = c'(c) > 0 & \text{if } p = cn^{-1/m(H)}, \\ 1 & \text{if } p \gg n^{-1/m(H)} \end{cases} \quad (6.1)$$

where $m(H) = \max_{J \subseteq H} e(J)/v(J)$ denotes the maximum hyperedge-density of $H$. This is a generalization of a classical theorem by Bollobás on small subgraphs [Bol81], cf. Theorem 3.2. A proof of the 0-statement and 1-statement for the case of $k$-uniform hypergraphs can be found in the appendix of this chapter. Observe that for the two exceptions in the simple graph case we have $m(S^*_\ell, r) = r(\ell-1)+1 < 1 = m_2(S_\ell)$ for every $k, r \geq 2$ and $m(S) = 1 = m_2(P_3)$.

Combining the two preceding paragraphs our coloring procedure requires that $F$ contains a strictly $k$-balanced subhypergraph $F'$ for which there are no counterexamples. This motivates the following definition.

**Definition 6.1.** Let $F$ be a $k$-uniform hypergraph and $r \geq 2$. We say that $F$ is Ramsey-density-obeying for $r$ colors if there exists a strictly $k$-balanced subhypergraph $F'$ of $F$ with $m_k(F') = m_k(F)$ such that all hypergraphs $H$ satisfy:

$$m(H) \leq m_k(F) = m_k(F') \implies H \not\rightarrow (F')^e_r.$$

Our main result states that for all Ramsey-density-obeying hypergraphs, there is a 0-statement which matches the 1-statement from Theorem 3.5.

**Theorem 6.2.** Let $k \geq 3$ and $r \geq 2$ be integers and $F$ be a $k$-uniform hypergraph which is Ramsey-density-obeying for $r$ colors. Then there exists $c = c(F, r) > 0$ such that for $p \leq cn^{-1/m_k(F)}$ we have

$$\lim_{n \to \infty} \Pr[H^{(k)}(n, p) \rightarrow (F)^e_r] = 0.$$

Furthermore we prove that a large class of hypergraphs, including in particular all $k$-uniform complete hypergraphs $K^{(k)}_\ell$ on $\ell > k$ vertices are Ramsey-density-obeying.
Corollary 6.3. Let \( k \geq 3 \) and \( r \geq 2 \). For complete \( k \)-uniform hypergraphs \( K_{\ell}^{(k)} \) on \( \ell > k \) vertices there exists constants \( c = c(\ell, k, r) > 0 \) and \( C = C(\ell, k, r) > 0 \) such that

\[
\lim_{n \to \infty} \Pr[H^{(k)}(n, p) \to (K_{\ell}^{(k)})^c_r] = \begin{cases} 
0 & \text{if } p \leq cn^{-1/m_{k}(K_{\ell}^{(k)})} \\
1 & \text{if } p \geq Cn^{-1/m_{k}(K_{\ell}^{(k)})}.
\end{cases}
\]

It is not hard to see that if \( F \) is strictly \( k \)-balanced and not Ramsey-density-obeying for \( r \geq 2 \) colors, then one cannot prove a 0-statement that matches the bounds of Theorem 3.5. In fact, by combining (6.1) and Theorem 6.2 one easily obtains that in this case the threshold for the property \( H^{(k)}(n, p) \to (F)^c_r \) is at \( p = n^{-1/m(H)} \) where \( H \) is a counterexample that minimizes \( m(H) \). Unfortunately, in contrast to the simple graph case, there exist large families of hypergraphs that are not strictly \( k \)-balanced and not Ramsey-density-obeying, cf. Section 6.5 for some examples.

6.1.2 Organization of This Chapter

Section 6.2 is devoted to the proof of our main result, Theorem 6.2. In Section 6.4 we show for a large class of hypergraphs that they are Ramsey-density-obeying, while discussing the threshold behaviour of a few exceptional (not Ramsey-density-obeying) structures in Section 6.5. We conclude with a section on open problems that we would like to see solved.

6.2 Proof of Main Result

Throughout the remainder of this chapter, we fix the uniformity \( k \geq 2 \), the number \( r \geq 2 \) of colors and consider \( F \) to be a fixed \( k \)-uniform hypergraph. Since \( F \) is Ramsey-density-obeying there exists \( F' \subset F \) such that \( m_k(F') = m_k(F) \), \( F' \) is strictly \( k \)-balanced and every hypergraph \( H \) satisfies \( m(H) \leq m_k(F') \Rightarrow H \not\rightarrow (F')^c_r \). We assume w.l.o.g. that \( F' = F \) (otherwise we replace \( F \) by \( F' \)). Moreover, we assume that \( F \) has at least 3 hyperedges. This requirement is motivated by the fact that the case of 1 hyperedge is trivial and all strictly \( k \)-balanced hypergraphs on 2 hyperedges are not Ramsey-density-obeying, cf. Section 6.5.
Recall that we need to show that for \( p \leq cn^{-1/m_k(F)} \) the random hypergraph \( H^{(k)}(n,p) \) a.a.s. has the property that there exists an \( r \)-hyperedge-coloring of \( H^{(k)}(n,p) \) that does not contain a monochromatic copy of \( F \). We call such a coloring valid. In the remainder we explain how to construct a valid coloring and show that this construction succeeds with high probability.

**Open and Closed Edges/Substructures and F-Cores**

As a first step we identify a set of hyperedges that we can color easily. Let \( H \) be a hypergraph. Let \( e \in E(H) \) be a hyperedge for which every two copies \( F_1, F_2 \subseteq H \) of \( F \) that contain \( e \) intersect in at least two hyperedges, i.e., \( |E(F_1) \cap E(F_2)| \geq 2 \). We call such hyperedges open and all other hyperedges closed. Note that open hyperedges are easy to color in the following sense. If there exists a valid coloring for \( H - e \) (the hypergraph obtained from \( H \) by removing \( e \)), then we can extend it to one for \( H \). Assume that this is not possible, that is, \( e \) completes a monochromatic copy of \( F \). Then since \( r \geq 2 \) the hyperedge \( e \) completes both a copy \( F_1 \) of \( F \) which is entirely colored with one color, say red, (except for the hyperedge \( e \) which is just to be colored) and a copy \( F_2 \) of \( F \) which is entirely colored with another color, say blue. Clearly, \( E(F_1) \cap E(F_2) = \{e\} \) which is a contradiction to \( e \) being open.

It is easy to see that successively removing open hyperedges yields the maximum subhypergraph of \( H \) (where maximum is with respect to subgraph inclusion) in which every hyperedge \( e \) is contained in at least two copies of \( F \) whose edge sets intersect exactly in \( e \). We call this subhypergraph the F-core of \( H \). By the above argument, it suffices to find a valid coloring for the F-core of \( H^{(k)}(n,p) \).

We say that a subhypergraph \( H' \) of the F-core of \( H \) is F-closed if every copy of \( F \) from the F-core of \( H \) is either contained in \( H' \) or edge-disjoint with \( H' \). It is easy to see that the hyperedges of the F-core can be partitioned into minimal F-closed subhypergraphs where minimal is with respect to subgraph inclusion. Furthermore, each such subhypergraph can be colored separately in order to find a valid coloring of the F-core.

Theorem 6.2 is an immediate consequence of the following lemma which states that with high probability every F-closed subhypergraph in the F-core of \( H^{(k)}(n,p) \) has constant size.
Lemma 6.4. Let $F$ be a strictly $k$-balanced $k$-uniform hypergraph with $e(F) \geq 3$. There exists $c = c(F) > 0$ and $L = L(F) > 0$ such that if $p \leq cn^{-1/m_k(F)}$ then a.a.s. every minimal $F$-closed subhypergraph of the $F$-core of $H^{(k)}(n,p)$ has size at most $L$.

With this lemma at hand the proof of Theorem 6.2 is straightforward.

Proof (of Theorem 6.2). Recall that we assume w.l.o.g. that $F$ is strictly $k$-balanced and that every hypergraph $H$ satisfies $m(H) \leq m_k(F) \Rightarrow H \not\rightarrow (F)_r^c$. Choose $c = c(F)$ and $L = L(F)$ according to Lemma 6.4. Then, by Lemma 6.4 $H^{(k)}(n,p)$ a.a.s. has the property that every minimal $F$-closed subhypergraph $H$ of the $F$-core of $H^{(k)}(n,p)$ is of size at most $L$. Since $p \leq cn^{-1/m_k(F)}$ it follows via a standard small subgraphs argument that a.a.s. every such $H$ satisfies $m(H) \leq m_k(F)$ and by our assumptions on $F$ thus also $H \not\rightarrow (F)_r^c$. Conditioning on this property we can deterministically find a valid $r$-hyperedge-coloring for every such $H$. Moreover, the union of the valid colorings of all minimal $F$-closed subhypergraphs yields a valid coloring of the $F$-core of $H^{(k)}(n,p)$. As explained above this coloring can be extended to a valid coloring of $H^{(k)}(n,p)$.

6.3 The Grow Sequences Approach

This section is devoted to the proof of Lemma 6.4. In order to prove it we first define a process that generates minimal $F$-closed structures iteratively starting from a single copy of $F$.

6.3.1 Grow Sequences

Let $H$ be a hypergraph. In the following we describe a procedure that yields a sequence of copies of $F$ which, starting from an empty hypergraph, constructs an $F$-closed subhypergraph of $H$. Let $F_1$ be a copy of $F$ from the $F$-core of $H$. Now, for every $\ell \geq 1$, we let $F_{\ell+1}$ be a copy of $F$ from the $F$-core such that $F_{\ell+1}$ intersects $\bigcup_{i=1}^{\ell} F_i$ in at least one hyperedge and such that $F_{\ell+1}$ introduces at least one new hyperedge. If no such copy exists in the $F$-core of $H$ the sequence ends after the $\ell$-th step and we set $S := (F_1, F_2, \ldots, F_{\ell})$ and $H(S) := \bigcup_{i=1}^{\ell} F_i$ as the hypergraph of $S$. We call such a sequence a grow sequence, and say that
$S$ is *contained* in or *appears* in $H$, meaning that $H(S)$ is a subgraph of $H$. Observe that for every minimal $F$-closed subhypergraph $H' \subseteq H$ there exists a grow sequence $S$ such that $H(S) = H'$. It thus suffices to show that there exists a constant $L$ such that every grow sequence that is contained in $H^{(k)}(n, p)$ has length at most $L$.

**Step Types of Grow Sequences.**

Let $S = (F_1, F_2, \ldots, F_\ell)$ be a grow sequence and let $S_i = (F_1, F_2, \ldots, F_i)$ denote the prefix including the first $i$ steps of $S$ for every $i \leq \ell$. We now view $S$ from the perspective of building a hypergraph step by step from the empty hypergraph $H(S_0)$ over $H(S_1)$ and so on up to $H(S_\ell) = H(S)$. We split the steps of this process into different types (see also Figure 6.2). We call step one the *first step*. We call a step $i \geq 2$ *regular* if the intersection of $F_i$ with $H(S_{i-1})$ is exactly one edge and $k$ vertices, and *degenerate* otherwise. Moreover, we call a hyperedge or vertex *degenerate* in $H(S_i)$ if it is contained in a copy $F_i'$ of a degenerate step $i' \leq i$. Otherwise we call a hyperedge or vertex *non-degenerate*. Note that a non-degenerate hyperedge or vertex in $H(S_i)$ may become degenerate in $H(S_i)$ if step $i$ is degenerate. However, once a hyperedge or vertex is degenerate in $H(S_i)$ it stays degenerate in $H(S_{i'})$ for every $i' \geq i$.

For regular steps we introduce a further categorization. We call a regular step $i$ *open* if it attaches a copy $F_i$ of $F$ to a hyperedge that is open and non-degenerate in $H(S_i - 1)$. Otherwise we call a regular step *closed* otherwise.

Before we continue, we give an intuitive reasoning of how we will use the distinction between regular and degenerate steps. Let $v_i$ denote the number of new vertices added in step $i$, i.e., $v_i := v(H(S_i)) - v(H(S_{i-1}))$ and let similarly $e_i$ denote the number of new hyperedges. It is not hard to see that for every regular step we have $v_i - e_i/m_k(F) = 0$, and that for every degenerate step $v_i - e_i/m_k(F) < 0$ since $F$ is strictly $k$-balanced. As $F$ is finite, this implies the existence of a constant $\delta = \delta(F) > 0$ such that every degenerate step satisfies

$$v_i - e_i/m_k(F) < -\delta . \quad (6.2)$$

Our aim is to use a first moment method argument to show that a.a.s. grow sequences $S$ of length more than $L$ do not appear. In order to do
6.3. The Grow Sequences Approach

Figure 6.2: Illustration of the different step types of a grow sequence for $k = 4$. A regular step is open if the hyperedge in the intersection is open and non-degenerate, and closed otherwise.

so we need to count grow sequences and estimate their probability to appear in $H^{(k)}(n, p)$. Observe that for every step of a grow sequence we essentially have to choose $v_i$ new vertices (at most $n^{v_i}$ ways) and require the presence of $e_i$ new hyperedges (which happens with probability $p^{e_i}$).

From the discussion above we see that every regular step contributes a value of $e^{e_i}$ (as $v_i - e_i/m_k(F) = 0$ and $p = cn^{-1/m_k(F)}$), while degenerate steps introduce a factor of $n^{-c}$. That is, we should expect that we neither have, say, more than $C_1 \log n$ regular steps nor more than, say, $C_2$ degenerate steps. Together with a few structural properties of grow sequences this observation suffices to prove Lemma 6.4.

Note, however, that in the above argument we are somehow sloppy as we ignored the choices for the intersection with the preceding copies of $F$ in the grow sequence. Intuitively, this is no problem as long as the grow sequence is ‘short’ and the number of such choices remains ‘small’. However, it turns out that for regular steps this choice actually hinders our first moment method argument. In the following we thus restrict grow sequences to special types which still generate all $F$-closed subhypergraphs of the $F$-core, but which are easier to count.
Canonical Grow Sequences.

Let $H$ be an $F$-closed subhypergraph of an $F$-core. We now describe a unique canonical way to construct a grow sequence $S$ with $H(S) = H$. For this, fix an arbitrary ordering of the vertices of $H$. Note that this vertex ordering also naturally induces an ordering on the hyperedges of $H$. Using these orderings we can say for two vertex or edge sets $X$ and $Y$ which of the two is lexicographically smaller.

Now, let $\mathcal{F}(H)$ denote the set of all copies of $F$ in $H$. We set $F_1$ to be the lexicographically smallest copy in $\mathcal{F}(H)$ and $S_1 = (F_1)$. Now, for every $i \geq 1$, if $H(S_i)$ contains an open non-degenerate hyperedge then we let $e$ denote the unique lexicographically smallest such hyperedge. Since $H$ is $F$-closed, there must be at least one copy of $F$ in $\mathcal{F}(H)$ that contains $e$ and is not yet contained in $H(S_i)$. We set $F_{i+1}$ to be the lexicographically smallest such copy and set $S_{i+1} = (F_1, F_2, \ldots, F_{i+1})$. Otherwise, if $H(S_i)$ only contains hyperedges which are closed or degenerate, then we choose $F_{i+1}$ to be the lexicographically smallest copy in $\mathcal{F}(H)$ which is not edge-disjoint from $H(S_i)$ and not yet contained in $H(S_i)$, and again set $S_{i+1} = (F_1, F_2, \ldots, F_{i+1})$.

We call a grow sequence canonical if it follows the above procedure. Then, every $F$-closed subhypergraph has exactly one corresponding canonical grow sequence. Hence, in order to prove Lemma 6.4 it suffices to show that $H^{(k)}(n, p)$ a.a.s. does not contain a canonical grow sequence of length more than $L$.

6.3.2 Structural Properties of Canonical Grow Sequences

In the following we derive structural properties of canonical grow sequences. Let

$$\gamma = \gamma(F) = \max\{|e_1 \cap e_2| : e_1, e_2 \in E(H) \land e_1 \neq e_2\}$$

denote the maximum number of vertices in an intersection of two hyperedges of $F$. It is not hard to see that a regular step of a grow sequence does not create a pair $e_1, e_2$ of hyperedge for which $|e_1 \cap e_2| > \gamma$. This immediately implies the following observation.

Observation 6.5. Let $S$ be a grow sequence of length $\ell$ and $e_1, e_2 \in E(H(S_i))$ for some $i \leq \ell$ such that $|e_1 \cap e_2| > \gamma$. Then at least one of $e_1$ and $e_2$ must be degenerate in $H(S_i)$. 
The following construction is illustrated in Figure 6.3. We now show that by attaching a copy of $F$ to a given hypergraph $G$ as in a regular step, i.e., we attach a copy $F_e$ of $F$ to $G$ such that $F_e$ and $G$ intersect exactly in a hyperedge $e$, then the only new copies of $F$ that are created consist of all hyperedges of $F_e - e$ and an additional hyperedge from $G$ that intersects the hyperedge $e$ in more than $\gamma$ vertices. (Note that in the simple graph case ($k = 2$), this implies that besides the copy of $F$ that is attached no new copies of $F$ are generated.) For a hypergraph $G$ and a hyperedge $e$ we denote by $G + e$ the hypergraph obtained by adding $e$ to $E(G)$ and the vertices incident to $e$ to $V(G)$ if necessary.

**Claim 6.6.** Let $G$ be a hypergraph and let $e \in E(G)$ and let $F$ be a strictly $k$-balanced hypergraph with $e(F) \geq 3$. Let $G^+ := G \cup F_e$ denote the graph obtained from attaching a copy $F_e$ of $F$ to the hyperedge $e$ such that $v(G^+) - v(G) = v(F) - k$ (see also Figure 6.3). Then we have for every copy $\tilde{F}$ of $F$ in $G^+$ that contains at least one hyperedge from $E(F_e - e)$ that

$$\tilde{F} = F_e - e + \tilde{e},$$

where $\tilde{e} \in E(G)$ and $|\tilde{e} \cap e| > \gamma$.

**Proof.** Let $\tilde{F} \subseteq G^+$ be a copy of $F$ that contains at least one hyperedge from $E(F_e) \setminus \{e\}$. Note that if $\tilde{F} = F_e$ then the claimed conditions are clearly satisfied with $\tilde{e} = e$. Hence, assume $\tilde{F} \neq F_e$. We first show that $E(F_e) \setminus \{e\} \subseteq E(\tilde{F})$. For the sake of a contradiction assume that this is not the case. Clearly, $\tilde{F}$ contains at least one hyperedge from the ‘new’ hyperedges of $F_e$, i.e., from $E(F_e) \setminus \{e\}$, and at least one
hyperedge from the ‘old’ edges $E(G) \setminus \{e\}$. Hence, the hypergraphs $\tilde{F}_{\text{new}} = F[V(F_e)]$ (the subgraph of $\tilde{F}$ induced by the vertex set $V(F_e)$) and $\tilde{F}_{\text{old}} = \tilde{F}[V(G)]$ are both non-empty and satisfy $\tilde{F}_{\text{new}} \neq F$ and $\tilde{F}_{\text{old}} \neq F$.

**Case 1:** $e \in E(\tilde{F})$. Then, clearly $e \in E(\tilde{F}_{\text{new}})$ and $e \in E(\tilde{F}_{\text{old}})$ and we thus have

$$m_k(F) = m_k(\tilde{F}) = \frac{e(\tilde{F}) - 1}{v(\tilde{F}) - k} = \frac{e(\tilde{F}_{\text{new}}) - 1 + e(\tilde{F}_{\text{old}}) - 1}{v(\tilde{F}_{\text{new}}) - k + v(\tilde{F}_{\text{old}}) - k} < m_k(F),$$

where the last inequality follows from the fact that $F$ is strictly $k$-balanced and that $\tilde{F}_{\text{new}} \neq F$ and $\tilde{F}_{\text{old}} \neq F$.\(^1\)

**Case 2:** $e \notin E(\tilde{F})$. Recall that we assume $E(F_e) \setminus \{e\} \not\subseteq E(\tilde{F})$. This clearly implies $e(\tilde{F}_{\text{old}}) \geq 2$. Let $\tilde{F}_{\text{new}}^+ = \tilde{F}_{\text{new}} + e$. Note that $\tilde{F}_{\text{new}}^+$ has one more hyperedge than $\tilde{F}_{\text{new}}$ and possibly up to $k-1$ more vertices than $\tilde{F}_{\text{new}}$. Moreover, it is easy to see that

$$e(\tilde{F}) = e(\tilde{F}_{\text{old}}) + e(\tilde{F}_{\text{new}}^+) - 1 \quad \text{and} \quad v(\tilde{F}) = v(\tilde{F}_{\text{old}}) + v(\tilde{F}_{\text{new}}^+) - k.$$

We thus have

$$m_k(F) = m_k(\tilde{F}) = \frac{e(\tilde{F}) - 1}{v(\tilde{F}) - k} \leq \frac{e(\tilde{F}_{\text{old}}) - 1 + e(\tilde{F}_{\text{new}}^+) - 1}{v(\tilde{F}_{\text{old}}) - k + v(\tilde{F}_{\text{new}}^+) - k} < m_k(F),$$

where the last inequality follows from the fact that $F$ is strictly $k$-balanced and that $\tilde{F}_{\text{old}}$ is a subgraph of $\tilde{F}$ and $\tilde{F}_{\text{new}}^+$ is a subgraph of $F_e$ and thus both are subgraphs of copies of $F$ and, crucially, $\tilde{F}_{\text{new}}^+ \neq F$ since $e(\tilde{F}_{\text{old}}) \geq 2$. Clearly, equation (6.3) is again a contradiction.

We have thus established that $F = F_e - e + \tilde{e}$ for a hyperedge $\tilde{e} \in E(G) \setminus \{e\}$ and it remains to show that $|\tilde{e} \cap e| > \gamma$. Assume that $|\tilde{e} \cap e| \leq \gamma$. Then let $e_1, e_2 \in E(F)$ such that $|e_1 \cap e_2| = \gamma$. And let $\tilde{F}_{\text{old}}^+ = \tilde{F}_{\text{old}} + e$, and $F_{1,2}$ denote the hypergraph with hyperedge set $\{e_1, e_2\}$ and vertex set $e_1 \cup e_2$. As before, let $\tilde{F}_{\text{new}} = \tilde{F}[V(F_e)] = F_e - e$. Since $\tilde{F}_{\text{old}}^+$ consists only of the hyperedges $\tilde{e}$ and $e$ and has at least as many vertices as $F_{1,2}$ we have $m_k(\tilde{F}_{\text{old}}^+) \leq m_k(F_{1,2}) < m_k(F)$ since $F$ is strictly $k$-balanced. Then we have, similar to (6.3), that

$$m_k(F) = m_k(\tilde{F}) = \frac{e(\tilde{F}) - 1}{v(\tilde{F}) - k} = \frac{e(\tilde{F}_{\text{old}}^+) - 1 + e(\tilde{F}_{\text{new}}) - 1}{v(\tilde{F}_{\text{old}}^+) - k + v(\tilde{F}_{\text{new}}) - k} < m_k(F),$$

\(^1\)Here we use that if $a/b < x$ and $c/d < x$, then $(a+c)/(b+d) < x$; we will tacitly use this observation repeatedly in the remainder of the chapter.
since both $\tilde{F}_{\text{old}}$ and $\tilde{F}_{\text{new}}$ have at least two edges (for $\tilde{F}_{\text{new}}$ recall that $E(F) \setminus \{e\} \subseteq E(\tilde{F}_{\text{new}})$ and that $e(F) \geq 3$) and a $k$-density of less than $m_k(F)$. Clearly, this is a contradiction. 

With this claim at hand we can show that every regular step of a canonical grow sequence increases the number of open non-degenerate hyperedges.

**Claim 6.7.** Let $S = (F_1, F_2, \ldots, F_\ell)$ be a canonical grow sequence. If step $i$ is regular, then $H(S_i)$ contains at least $e(F) - 2$ more open non-degenerate hyperedges than $H(S_{i-1})$.

Note that this claim essentially states that the $e(F) - 1$ new hyperedges added in a regular step $i$ are open in $H(S_i)$. If the copy of $F$ added in step $i$ is added to an open hyperedge of $H(S_i)$ this edge now becomes closed, giving the worst case bound of $e(F) - 2$. As $e(F) \geq 3$ this implies that a regular step $i$ cannot make $H(S_i)$ closed and a closed regular step has to be preceded by a degenerate step. The following corollary is an immediate consequence of this.

**Corollary 6.8.** For every canonical grow sequence $S = (F_1, \ldots, F_\ell)$ we have that the number of closed regular steps is bounded by the number of degenerate steps. Moreover, this is also true in every prefix sequence $S_i$ with $i \leq \ell$.

**Proof of 6.7** We denote by $e \in E(F_i) \cap E(H(S_{i-1}))$ the unique hyperedge in the intersection of $F_i$ with $H(S_{i-1})$. We show that

(i) all ‘new’ hyperedges of $F_i$ are open and non-degenerate in $H(S_i)$, and that

(ii) no open non-degenerate hyperedge except $e$ is closed in step $i$.

It is easy to see that $e$ is closed in $H(S_i)$ (if it was not already closed in $H(S_{i-1})$ and the claim thus follows immediately from (i) and (ii).

We first show (i). This follows easily from the fact that we assume that $F$ contains at least three hyperedges. As Claim 6.6 implies that every copy of $F$ that contains a new hyperedge, actually contains all new hyperedges we cannot have two copies of $F$ that only intersect in a single new hyperedge.
For (ii) we show that no open non-degenerate hyperedge except $e$ forms a copy of $F$ with the newly added hyperedges of step $i$. Let $e_{\text{old}} \in E(H(S_{i-1}) - e)$ and assume that there exists a copy $\tilde{F} \subseteq H(S_i)$ of $F$ that contains $e_{\text{old}}$ and a hyperedge from $E(F_i - e)$. By Claim 6.6 we know that $\tilde{F} = F_i - e + e_{\text{old}}$ and that $|e_{\text{old}} \cap e| > \gamma$. It follows from Observation 6.5 that $e_{\text{old}}$ or $e$ must be a degenerate hyperedge in $H(S_{i-1})$. If $e_{\text{old}}$ is degenerate, then we are done. Otherwise $e_{\text{old}}$ is non-degenerate and thus $e$ must be degenerate. Then, since $S$ is canonical, $H(S_{i-1})$ does not contain any open non-degenerate hyperedges and thus $e_{\text{old}}$ must be closed.

The main technical difficulty in our proof of Lemma 6.4 is that we need to show that ‘long’ grow sequences must have a certain number of degenerate steps. For this we need to show that a single degenerate step can only ‘close’ a bounded number of open non-degenerate hyperedges. In order to prove this we need that for $p \leq cn^{-1/m_k(F)}$ we have that $H^{(k)}(n,p)$ does not contain substructures in which many copies of $F$ overlap in more than $k$ vertices.

**Definition 6.9 (Lump).** Let $F$ be a strictly $k$-balanced $k$-uniform hypergraph. A lump of size $T$ is a hypergraph $\bigcup_{i \in [T]} F_i$, where $F_1, \ldots, F_T$ are pairwise different copies of $F$ such that $|\cap_{i \in [T]} V(F_i)| \geq k + 1$.

We first give an intuition why $H^{(k)}(n,p)$ does a.a.s. not contain large lumps. Let $T \geq 1$ and $\tilde{F} = \bigcup_{i \in [T]} F_i$ be a clutter. Then every copy $F_i$ intersects the preceding copies $F_1, \ldots, F_{i-1}$ in at least $k + 1$ vertices and thus in a subhypergraph of $F$ with a ‘non-trivial’ $k$-edge-density of less than $m_k(F)$ where by ‘non-trivial’ we refer to the denominator being non-zero. Hence the ratio of hyperedges to vertices that are introduced by $F_i$ is larger than $m_k(F)$. Specifically, we show that there exists a constant $\alpha = \alpha(F,k)$ such that every $F_i$ accounts for a factor of $n^{-\alpha}$ for the expected number of copies of $\tilde{F}$ in $H^{(k)}(n,p)$. Hence, if we choose $T$ large enough, this expectation tends to 0.

**Claim 6.10.** There exists a constant $T = T(F,k)$ such that a.a.s. $H^{(k)}(n,p)$ does not contain a lump of size $T$ if $p \leq cn^{-1/m_k(F)}$.

**Proof.** Let $\alpha = \alpha(F,k)$ denote the normalized difference between the $k$-density of $F$ itself and a densest proper subhypergraph of $F$. That is,

$$\alpha = \alpha(F,k) = \min_{U \subseteq V(F), v(U) \geq k+1} \left\{ \frac{m_k(F) - d_k(U)}{m_k(F)} \right\}.$$ (6.4)
Since $F$ is strictly $k$-balanced we have $\alpha > 0$. Let $T \geq 1$ be a constant and let $\hat{F} = \cup_{i \in [T]} F_i$ be a lump of size $T$. For every $i \geq 2$ let $F_i^\cap = F_i \cap \bigcup_{j=1}^{i-1} F_j$ denote the intersection of $F_i$ with the preceding copies of $F$. W.l.o.g. we assume that $F_i^\cap \neq F$ for every $i \geq 2$. If this is not the case, then we simply remove the copies $F_i$ for which $F_i^\cap \cong F$. Note that this removes at most $(Tv(F))^v(F) = \mathcal{O}(1)$ copies. Furthermore, note that since $T$ is constant there are only a constant number of different lumps $\hat{F}$, as for each $i$ there are only a constant number of different choices for $F_i^\cap$. Hence, it suffices to show that for a given lump the expected number of copies $\mathbb{E}[X_{\hat{F}}]$ of $\hat{F}$ in $H^{(k)}(n, p)$ is $o(1)$. This can easily be seen as follows:

$$
\mathbb{E}[X_{\hat{F}}] \leq n^{v(F)} \prod_{i=2}^{T} n^{v(F) - v(F_i^\cap)} p^{e(F) - e(F_i^\cap)} \\
\leq c^e(\hat{F}) n^{v(F)} \prod_{i=2}^{T} n^{v(F) - k - (v(F_i^\cap) - k)} n^{\frac{e(F) - (e(F_i^\cap) - 1)}{m_k(F)}} \\
= c^e(\hat{F}) n^{v(F)} \prod_{i=2}^{T} n^{- (v(F_i^\cap) - k) \frac{m_k(F) - d_k(F_i^\cap)}{m_k(F)}} \\
= \mathcal{O}(n^{v(F) - (T - 1)\alpha})
$$

where we used that $F_i^\cap \neq F$ and $v(F_i^\cap) - k \geq 1$ for every $i \geq 2$. Clearly, this is $o(1)$ if we chose $T > v(F)/\alpha + 1$. \hfill $\square$

**Claim 6.11.** For every strictly $k$-balanced hypergraph $F$ there exists a function $D : \mathbb{N} \to \mathbb{N}$ such that the following holds. Let $S = (F_1, F_2, \ldots, F_\ell)$ be a canonical grow sequence for which $H(S)$ does not contain a lump of size $T$, where $T$ is chosen according to Claim [6.10]. If step $j$ is the $i$-th degenerate step of $S$ then at most $D(i)$ open non-degenerate hyperedges from $S_{j-1}$ are closed in $S_j$.

**Proof.** Let step $j$ be the $i$-th degenerate step of the canonical grow sequence $S = (F_1, F_2, \ldots, F_\ell)$. To prove the claim we show that any open non-degenerate hyperedge closed in step $j$ must be contained in a copy of $F$ which satisfies certain additional properties.

Recall that a vertex in $H(S_j)$ is *degenerate* if it is contained in some degenerate copy $F_j'$ for some $j' \leq j$. We denote with $V_d$ the set of all such vertices in $H(S_j)$. Clearly $V_d$ contains at most $i \cdot v(F)$ vertices.
We call a vertex semi-degenerate (see also Figure 6.4) if it is degenerate or if it is contained in a copy $F_a$ of $F$ of a regular step $a \leq j$ for which (at step $j$) $V(F_a - f_a)$ intersects $V_d$, where $f_a$ denotes the hyperedge in the intersection of $F_a$ with $H(S_{a-1})$. In other words a vertex $v$ is semi-degenerate if it is degenerate itself or if at some point up to step $j$ it became “close” to a degenerate vertex. Note that since we have at most $i \cdot v(F)$ degenerate vertices, the number of semi-degenerate vertices is bounded by $i \cdot v(F)^2$.

Let $\mathcal{F}_{sd}$ denote the set of copies of $F$ in $H(S_j)$ that contain at least $k + 1$ semi-degenerate vertices. We show below that all non-degenerate hyperedges which are open in $H(S_{j-1})$ and closed in step $j$ must be closed by – and thus also contained in – a copy of $F$ in $\mathcal{F}_{sd}$. The claim then follows by the following argument. By assumption $H(S)$ does not contain a lump of size $T = T(F)$ and the number of semi-degenerate vertices is bounded by a constant, therefore the number of copies of $F$ in $\mathcal{F}_{sd}$ is also bounded by some constant $C_1$ (depending on $i$, $F$ and $T = T(F)$). These copies can span (and thus close) at most $C_2 = e(F) \cdot C_1$ hyperedges.

Let $\mathcal{F}_0$ denote the set of all copies of $F$ in $H(S_j)$ that are not contained in $\mathcal{F}_{sd}$. It remains to show that no copy of $F$ in $\mathcal{F}_0$ closes an open non-degenerate hyperedge in step $j$. For this we show that each $\hat{F} \in \mathcal{F}_0$ must have the following form: $\hat{F} = F_a - f_a + \hat{f}$, where $a < j$ is some regular step attaching $F_a$ to the hyperedge $f_a$ and $\hat{f}$ is a degenerate or
closed hyperedge in \(H(S_{a-1})\).

With these properties at hand we can now conclude the proof of the claim as follows. Assume there exists an open non-degenerate hyperedge \(e\) in \(H(S_{j-1})\) and two copies \(F', F'' \in \mathcal{F}_0\) such that \(F'\) and \(F''\) intersect exactly in the hyperedge \(e\). By the above we have \(F' = F_{a'} - f_{a'} + f'\) and \(F'' = F_{a''} - f_{a''} + f''\) where \(a'\) and \(a''\) are regular steps and \(f', f''\) are closed or degenerate. We have \(a' \neq a''\), as otherwise \(F_{a'} - f_{a'} \subseteq F' \cap F''\) and since \(F\) contains at least 3 hyperedges this contradicts the fact that \(F' \cap F''\) consists of only the hyperedge \(e\). As \(a'\) and \(a''\) are regular steps \(F_{a'} - f_{a'}\) and \(F_{a''} - f_{a''}\) must be disjoint, but then the open non-degenerate hyperedge \(e\) must be equal to \(f'\) or \(f''\), which contradicts the fact that both are either closed or degenerate.

It remains to prove the aforementioned structural results for copies \(\hat{F}\) in \(\mathcal{F}_0\). Observe that by the definition of \(\mathcal{F}_0\) we know that \(\hat{F}\) contains either no or exactly one degenerate hyperedge.

We first consider the case that \(\hat{F} \in \mathcal{F}_0\) contains no degenerate hyperedge. Let \(a = a(\hat{F}) < j\) denote the first step in which \(\hat{F}\) is contained in \(H(S_a)\) (i.e., \(\hat{F} \notin H(S_{a-1})\)). As \(\hat{F}\) does not contain a degenerate hyperedge, we know that step \(a\) is regular, i.e., the copy \(F_a\) was attached to a single hyperedge \(f_a\). Moreover, Claim 6.6 implies that \(\hat{F} = F_a - f_a + \hat{f}\); for a hyperedge \(\hat{f}\) in \(H(S_{a-1})\) such that \(|f \cap f_a| \geq \gamma\). We now show that \(\hat{f}\) is closed in \(H(S_{a-1})\). Note that by Observation 6.5 at least one of the two hyperedges is degenerate in \(H(S_a)\). As by assumption \(\hat{f}\) is not degenerate, \(f_a\) must be. Hence, in step \(a\) of the grow sequence \(S\) a copy of \(F\) is attached to a degenerate hyperedge. Since \(S\) is canonical we thus have that \(H(S_{a-1})\) does not contain open non-degenerate hyperedges. Since \(\hat{f}\) is in \(H(S_{a-1})\) it follows that \(\hat{f}\) is closed.

Next we consider the case that \(\hat{F} \in \mathcal{F}_0\) contains exactly one degenerate hyperedge, say \(f_d\). Let \(a = a(\hat{F})\) denote the step in which the last hyperedge of \(\hat{F} - f_d\) was added, i.e., \(\hat{F} - f_d\) is a subhypergraph of \(H(S_a)\) but not of \(H(S_{a-1})\). As \(\hat{F} - f_d\) does not contain degenerate hyperedges, \(a\) is a regular step. Moreover, since \(\hat{F} \notin \mathcal{F}_{sd}\) we know that \(f_d\) cannot contain a vertex from \(V(F_a) \setminus V(H(S_{a-1}))\), as otherwise all vertices of \(F_a\) would be semi-degenerate and \(|V(\hat{F}) \cap (V(F_a) \cup f_d)| \geq k + 1\). Hence, we can apply Claim 6.6 for \(G \leftarrow H(S_{a-1}) + f_d\) and \(F_e \leftarrow F_a\) to deduce that \(\hat{F} = F_a - f_a + f_d\) and \(|f_d \cap f_a| > \gamma\) where, as before, \(f_a\) denotes the hyperedge in \(H(S_{a-1})\) to which the copy \(F_a\) is attached to. \(\square\)
We are now ready to prove the key property of canonical grow sequence which can roughly be summarized as follows: the longer a canonical grow sequence is the more degenerate steps it has to contain.

**Claim 6.12.** For every $d \geq 1$ there exists a constant $\ell_{\max}(d, F)$ such that every canonical grow sequence $S$ with at most $d$ degenerate steps and for which $H(S)$ does not contain a lump of size $T$, where $T$ is chosen according to Claim 6.10 has length at most $\ell_{\max}(d)$.

**Proof.** Let $d \geq 1$ be a constant and let $S$ be a canonical grow sequence of length $\ell$ with at most $d$ degenerate steps. Then $S$ contains at least $\ell - d$ regular steps each of which increases the number of open hyperedges by at least $e(F) - 2$ by Claim 6.7. Moreover, since $H(S)$ does not contain a lump of size $T$ we obtain by Claim 6.11 that every degenerate step decreases the number of open hyperedges by at most $D_{\text{max}} := \max_{i \leq d} D(i)$ where $D(i)$ is chosen according to Claim 6.11. Hence, since $H(S)$ is $F$-closed we have $(\ell - d)(e(F) - 2) \leq dD_{\text{max}}$ and thus $\ell \leq \frac{dD_{\text{max}}}{e(F) - 2} + d$. \hfill $\square$

### 6.3.3 Proof of Lemma 6.4

With the properties of canonical grow sequences from the previous section at hand we are now ready to prove Lemma 6.4. As argued at the beginning of this section it suffices to show that there exists a constant $L = L(F)$ such that a.a.s. the random hypergraph $H^{(k)}(n, p)$ does not contain a canonical grow sequence of length more than $L$. Note that it follows from Claim 6.10 that if $p \leq cn^{-1/m_k(F)}$ then a.a.s. the hypergraph $H^{(k)}(n, p)$ does not contain a lump of size $T$, for an appropriately chosen constant $T$. It thus suffices to consider canonical grow sequences for which the corresponding hypergraph does not contain a lump of size $T$.

Let $S$ denote the set of all canonical grow sequences of length more than $L = L(F)$ (we will fix this constant later) for which the corresponding hypergraph does not contain a lump of size $T$. Note that if we can show for a sequence $S \in S$ of length $\ell \geq L$ that a prefix sequence $S_i$ consisting of the first $i$ steps of $S$ for $i \leq \ell$ is not contained in $H^{(k)}(n, p)$ then $S$ does not appear as well. Hence, it suffices to find a suitable set $\text{Pre}(S)$ with the two properties that all sequences of $S$ have a prefix sequence in $\text{Pre}(S)$ and such that a.a.s. no sequence from $\text{Pre}(S)$ appears in $H^{(k)}(n, p)$. 

To achieve this we define $\text{Pre}(S)$ as the set of the following prefixes. Let $d_{\text{max}} = d_{\text{max}}(F)$ be a constant which we will determine later. For each $S \in \mathcal{S}$ we include the prefix sequence of $S$ containing either all steps up to (and including) the $d_{\text{max}}$-th degenerate step or all steps up to the log $n$-th step, if the index of the $d_{\text{max}}$-th degenerate step is larger than log $n$. Note that this is well-defined as we can force any sequence $S \in \mathcal{S}$ to contain at least $d_{\text{max}}$ degenerate steps by choosing $L$ large enough, cf. Claim 6.12.

The key intuition is that prefixes $S \in \text{Pre}(S)$ containing ‘many’ degenerate steps give rise to a very dense hypergraph $H(S)$, which correspondingly is unlikely to appear. On the other hand prefixes $S \in \text{Pre}(S)$ with ‘few’ degenerate steps must contain many regular steps. The corresponding hypergraph $H(S)$ will then be very large and also unlikely to appear.

We define $\text{Pre}^{d_{\text{max}}}(S)$ as the set containing all prefix sequences from $\text{Pre}(S)$ with exactly $d_{\text{max}}$ degenerate steps and $\text{Pre}^{\log n}(S)$ as the set of those with length exactly log $n$. Clearly, every prefix sequence in $\text{Pre}(S)$ is in at least one of the two subsets. We consider the subsets $\text{Pre}^{\log n}(S)$ and $\text{Pre}^{d_{\text{max}}}(S)$ separately.

### Sequences with log $n$ Steps

We start with prefix sequences in $\text{Pre}^{\log n}(S)$. By Corollary 6.8 we have for every $S \in \text{Pre}^{\log n}(S)$ that the number of regular steps in which a copy of $F$ is attached to a closed hyperedge is bounded by the number of degenerate steps. In the following, we call these regular steps *closed* and all other regular steps *open*. We bound the number of elements in $\text{Pre}^{\log n}(S)$, i.e., we bound the number of all sequences of steps of length log $n$ which contain at most $m := 2d_{\text{max}}$ steps that are not open and regular. To do so we first fix the number of steps that are not open regular and their position in the sequence. For this we have at most

$$m(\log n)^m$$  \hspace{1cm} (6.5)

choices. Moreover, we have at most $n^{v(F)}$ choices for the copy of a step that is not open regular. As there are at most $m$ of these we have in total at most

$$n^{v(F)m}$$ \hspace{1cm} (6.6)

different choices for these steps. All remaining steps are open regular steps. Recall that since all sequences in $S$ are canonical we have for...
every open regular step that there is no choice for the \( k \) vertices in which the copy of \( F \) added in that step intersects the preceding copies of \( F \). Hence, for each of the at most \( \log n \) open regular steps we only need to choose \( v(F) - k \) new vertices and the role of the \( k \) vertices from the intersection in the copy of \( F \). Thus, every open regular step gives at most
\[
v(F)^k n^{v(F) - k}
\]
choices. The very first step is special and we model it by choosing \( k \) vertices as the starting hyperedge (at most \( n^k \) choices), and another \( v(F)^k n^{v(F) - k} \) choices as in an open regular step. Together with (6.5), (6.6) and (6.7) we therefore have that the number of elements in \( \text{Pre}^{\log n}(S) \) is at most
\[
m(\log n)^m \cdot n^{v(F)m} \cdot n^k \cdot (v(F)^k n^{v(F) - k})^{\log n}.
\]
For a fixed sequence in \( \text{Pre}^{\log n}(S) \) the probability that it appears in \( H^{(k)}(n, p) \) is bounded from above by the probability that the first \( \log n - m \) open regular steps appear. Each such step requires \( e(F) - 1 \) new hyperedges to be present in \( H^{(k)}(n, p) \), so this probability is at most \( (p^{e(F) - 1})^{\log n - m} \). Using \( n^{v(F) - k} p^{e(F) - 1} \leq \tilde{c} \) for \( \tilde{c} = ce(F)^{-1} \) we can now deduce that the number \( X_{\text{Pre}^{\log n}(S)} \) of sequences from \( \text{Pre}^{\log n}(S) \) that appear in \( H^{(k)}(n, p) \) satisfies
\[
\mathbb{E}[X_{\text{Pre}^{\log n}(S)}] \leq m(\log n)^m \cdot n^{2v(F)m + k} \cdot v(F)^k \cdot \tilde{c}^{\log n - m}
= m(\log n)^m \cdot n^{2v(F)m + k + k \log(v(F)) - \log(1/\tilde{c})} \cdot \tilde{c}^{-m}.
\]
As \( m \) is a constant depending only on \( F \) we can choose \( \tilde{c} = \tilde{c}(F) \) such that the above expectation is \( o(1) \).

**Sequences with \( d_{\max} \) Degenerate Steps**

It remains to consider the prefix sequences in \( \text{Pre}^{d_{\max}}(S) \), i.e., those which contain exactly \( d_{\max} \) degenerate steps and at most \( \log n \) steps in total. We partition these sequences further into sets \( \text{Pre}^{d_{\max}}(V, E, \ell) \) which contain all sequences from \( \text{Pre}^{d_{\max}}(S) \) of length exactly \( \ell \) for which the total number of new vertices and new hyperedges added in the \( d_{\max} \) degenerate steps are exactly \( V \) and \( E \) respectively. Clearly, this set can
only be non-empty if \( V \leq d_{\max}v(F) \), \( E \leq d_{\max}e(F) \), and \( \ell \leq \log n \). Hence, the total number of subsets that we need to consider is bounded by
\[
d_{\max}v(F) \cdot d_{\max}e(F) \cdot \log n = \mathcal{O}(\log n).
\] (6.8)

Recall that for every degenerate step the numbers \( v_{\text{new}} \) of new vertices and \( e_{\text{new}} \) of new hyperedges satisfy \( v_{\text{new}} - e_{\text{new}}/m_{k}(F) < -\delta \) where \( \delta > 0 \), cf. (6.2). Therefore the set \( \text{Pre}^{d_{\max}}(V,E,\ell) \) can only be non-empty if
\[
V - E/m_{k}(F) < -\delta d_{\max}.
\] (6.9)

We now derive a bound on the number of sequences contained in a set \( \text{Pre}^{d_{\max}}(V,E,\ell) \). Similarly to (6.5) we have for \( m = 2d_{\max} \) that there are at most
\[
m(\ell)^m = o(n)
\] (6.10)
choices for the step configuration, i.e., the number of steps that are not open and regular and their positions in the sequence. Moreover, we again model the first step by choosing a starting hyperedge (at most \( n^{k} \) choices) and by counting the remaining choices similar to an open regular step.

Considering all degenerate steps at once we need to choose a total of \( d_{\max}v(F) \) vertices for them, \( V \) of which are new vertices and not from the previously seen ones, resulting in at most \( n^{V} \) choices, and at most \( d_{\max}v(F) - V \leq d_{\max}v(F) \) of which are chosen from the previously seen ones, giving at most \( (v(F)\ell)^{d_{\max}v(F)} \) choices. Having fixed the new vertices and old vertices it remains to choose for every degenerate step which vertices of the copy of \( F \) are from the old and which from the new vertices. This gives at most another \( 2^{v(F)d_{\max}} \leq (v(F)\ell)^{v(F)d_{\max}} \) choices. In total the number of choices for the vertices of degenerate steps is bounded by
\[
n^{V}(v(F)\ell)^{2v(F)d_{\max}} = n^{V} \cdot o(n).
\] (6.11)

All other \( \ell - d_{\max} \) steps are regular steps. Similar to (6.7) each open regular step gives rise to at most \( (v(F)^{k}n_{v(F)}^{v(F) - k}) \) choices. For the at most \( d_{\max} \) closed regular steps we additionally have to select \( k \) vertices that in contrast to open regular steps are not predetermined. This accounts at most for another \( (v(F)\ell)^{kd_{\max}} \) choices. In total all regular steps give rise to at most
\[
(v(F)\ell)^{kd_{\max}} \cdot (v(F)^{k}n_{v(F)}^{v(F) - k})^{\ell - d_{\max}} = o(n) \cdot n^{k \log v(F)} \cdot n^{(v(F) - k)(\ell - d_{\max})}
\] (6.12)
choices, where we used $v(F)^{k(\ell - d_{\text{max}})} \leq n^k \log v(F)$ which holds since \( \ell \leq \log n \). Combining (6.10), (6.11), (6.12) and the \( n^k \) choices for the starting hyperedge, we get that the number of sequences in the set $\text{Pre}^{d_{\text{max}}}(V, E, \ell)$ is at most

$$n^{k+k \log v(F)+(v(F)-k)(\ell-d_{\text{max}})+V+1} \quad (6.13)$$

for \( n \) large enough where the +1 in the exponent takes care of all \( o(n) \) terms. It is easy to see that a prefix sequence from the set $\text{Pre}^{d_{\text{max}}}(V, E, \ell)$ appears in $H^{(k)}(n, p)$ with probability

$$p(p^{e(F)-1})^{\ell-d_{\text{max}}} \cdot p^E. \quad (6.14)$$

We now combine (6.13) and (6.14) with $n^{v(F)-k}p^{e(F)-1} \leq c^{e(F)-1} \leq 1$ which holds if we choose \( c \leq 1 \). With this we obtain that for every subset $\text{Pre}^{d_{\text{max}}}(V, E, \ell)$ the number $X_{\text{Pre}^{d_{\text{max}}}(V, E, \ell)}$ of sequences that are present in $H^{(k)}(n, p)$ satisfies

$$\mathbb{E}[X_{\text{Pre}^{d_{\text{max}}}(V, E, \ell)}] \leq n^{k+k \log v(F)+(v(F)-k)(\ell-d_{\text{max}})+V+1} \cdot p(p^{e(F)-1})^{\ell-d_{\text{max}}} \cdot p^E \leq n^{k+k \log v(F)+1} \cdot n^V \cdot p^E$$

$$\leq n^{k+k \log v(F)+1+V-E/m_k(F)} \leq n^{k+k \log v(F)+1-\delta d_{\text{max}}}. \quad (6.9)$$

Since this bound is independent of $V, E$ and $\ell$, we obtain with (6.8) that we have for large enough $n$ that

$$\mathbb{E}[X_{\text{Pre}^{d_{\text{max}}}(S)}] = O(\log n) \cdot n^{k+k \log v(F)+1-\delta d_{\text{max}}}. \quad (6.15)$$

Choosing $d_{\text{max}} = d_{\text{max}}(F)$ large enough and $L = L(F)$ such that every canonical grow sequence with at least $L$ steps has at least $d_{\text{max}}$ degenerate steps (cf. Claim 6.12) we have that the expectation in (6.15) is $o(1)$. \qed

### 6.4 A Large Class of Ramsey-Density-Obeying Hypergraphs

In this section we show for a large class of hypergraphs, in particular including all complete $k$-uniform hypergraphs $K_{\ell}^{(k)}$ on $\ell > k$ vertices.
Together with Theorem 6.2 it immediately follows that a 0-statement asymptotically matching the bounds of Theorem 3.5 exists for all such hypergraphs.

We denote by $\chi(H)$ the weak chromatic number of $H$, that is, the minimum number $t$ of colors such that the vertices of $H$ can be colored with $t$ colors without creating a monochromatic hyperedge.

**Theorem 6.13.** Let $k \geq 3$ and $r \geq 2$ and $F$ be a $k$-uniform hypergraph. If $F$ contains a strictly $k$-balanced hypergraph $F'$ with $m_k(F) = m_k(F')$ such that

1. $F' = K^{(k)}_\ell$ for some $\ell > k$, or
2. $\chi(F') \geq k + 1$, or
3. $m(F') \geq 2 + \frac{2k-1}{v(F')-2k}$, or
4. $\lfloor km_k(F') \rfloor \leq r(\delta(F') - 1)$ or $\lfloor km_k(F') \rfloor < (\chi(F') - 1)^r$,

then $F$ is Ramsey-density-obeying for $r$ colors.

Before we get to the proof let us first describe examples for hypergraphs of each of the 4 classes and illustrate the relationships between them. Observe that large complete hypergraphs have large chromatic number and also high density. Hence, for large values of $\ell$ we have that [2] and [3] cover [1]. However, for small values of $\ell$ we can only use [1] to prove that complete hypergraphs are Ramsey-density-obeying. Furthermore, for $k \geq 4$ it is easy to construct hypergraphs of chromatic
number 2 with arbitrarily high density by fixing two vertices that are
contained in every hyperedge. For these hypergraphs only (3) can be ap-
plied. Examples for hypergraphs that can only be shown to be Ramsey-
density-obeying by (4) (see also Figure 6.5) are the Fano plane and
many $t$-tight cycles, where a $t$-tight cycle $C_{t,t}^{(k)}$ with $\ell$ hyperedges is a
$k$-uniform hypergraph whose $\ell(k-t)$ vertices can be cyclically ordered
in such a way that the hyperedges are segments of that ordering and
every two consecutive hyperedges intersect in exactly $t$ vertices. For the
Fano plane, denoted by $H_{FP}$, observe that it is strictly $k$-balanced and
that we have $m(H_{FP}) = 1, m_3(H_{FP}) = 3/2$ and $\chi(H_{FP}) = 3$. Hence,
neither (2), nor (3) are applicable. For $t$-tight cycles observe that they
are also strictly $k$-balanced and satisfy $1/k \leq m(C_{t,t}^{(k)}) \leq 1$ and have
chromatic number 2 such that (2), (3) and the second criterion of (4)
cannot be used. However, we have

$$\left\lfloor km_k(C_{t,t}^{(k)}) \right\rfloor = \left\lfloor \frac{k}{k-t + \frac{t}{t-1}} \right\rfloor$$

and $\delta(C_{t,t}^{(k)}) = \left\lfloor \frac{k}{k-t} \right\rfloor$ such that for $t \geq k/2$ and large enough $\ell$ or large
enough $r$ the first criterion of (4) applies. Finally, we mention that we
neither know an example for a hypergraph that is only covered by (2) nor
a proof for the statement that (1), (3) and (4) make this part obsolete.

The remainder of this section is devoted to the proof of Theorem 6.13.

To improve readability we assume that $F' = F$. Now, let $H$ be a
$k$-uniform hypergraph with $m(H) \leq m_k(F)$. We need to show that
$H \nrightarrow (F)^c_r$, that is, there exists a valid hyperedge-coloring of $H$ with
$r$ colors. For (1) we additionally show that $K_{t,t}^{(k)}$ is strictly $k$-balanced.
We treat each of the four classes of hypergraphs in its own subsection.
For (1) - (3) we show how to find a valid hyperedge-coloring of $H$ with
2 colors, and only in the case of (4) we use all $r$ colors.

6.4.1 Complete Hypergraphs

In this subsection we deal with the case $F = K_{t,t}^{(k)}$ for some $\ell > k$. We
first show that every complete hypergraph is strictly $k$-balanced.

**Lemma 6.14.** Let $\ell > k \geq 3$. The complete $k$-uniform hypergraph on
$\ell$ vertices is strictly $k$-balanced.
6.4. A Large Class of Ramsey-Density-Obeying Hypergraphs

Proof. Let $J \subseteq K^{(k)}_{\ell}$. If $v(J) = v(K^{(k)}_{\ell})$ then clearly $e(J) < e(F)$ and thus also $d_k(J) < d_k(K^{(k)}_{\ell})$. Moreover, if $v(J) < v(K^{(k)}_{\ell})$ then we have

$$d_k(J) = \frac{e(J) - 1}{v(J) - k} \leq \frac{\left(\frac{v(J)}{2}\right) - 1}{v(J) - k} < \frac{\left(\frac{v(K^{(k)}_{\ell})}{2}\right) - 1}{v(K^{(k)}_{\ell}) - k} = d_k(K^{(k)}_{\ell}).$$

In the remainder we show how to find a valid hyperedge-coloring with 2 colors for $H$. The idea is to show that we can find an ordering $v_1, v_2, \ldots, v_{v(H)}$ of the vertices of $H$ such that every vertex $v_i$ has only ‘few’ neighbors in the set $\{v_1, v_2, \ldots, v_{i-1}\}$. If we then start with the empty hypergraph and add the vertices of $H$ one by one in this ordering, we can show that no vertex introduces enough hyperedges to enforce a monochromatic copy of $F$.

Formally, we prove a lower bound on the number of hyperedges that are required to enforce a monochromatic clique on $\ell-1$ vertices in a $(k-1)$-uniform hypergraph. Let $r_e(F)$ denote the size-Ramsey number of $F$, that is

$$r_e(F) = \min\{e(G) : G \rightarrow (F)^e_{2}\}$$

Lemma 6.15. Let $\ell > k \geq 3$. Then we have

$$r_e(K^{(k-1)}_{\ell-1}) \geq km_k(F).$$

With this claim at hand we are ready to prove Theorem 6.13 for cliques.

Proof (of Theorem 6.13). We first argue how to find a vertex ordering as mentioned above. As we have $m(H) \leq m_k(F)$ every subgraph $H' \subseteq H$ satisfies $e(H')/v(H') \leq m_k(F)$. Hence, the average degree of every subgraph is at most $km_k(F)$ and furthermore, every subgraph contains a vertex of degree at most $km_k(F)$. This is also known as degeneracy of $H$, i.e., we just proved an upper bound of $km_k(F)$ on the degeneracy of $H$. We now define a sequence $v_1, v_2, \ldots, v_{v(H)}$ inductively in reversed order. Having fixed the vertices $v_{i+1}, v_{i+2}, \ldots, v_{v(H)}$ for any $i \leq v(H)$ we can find $v_i$ as follows. By the above argument $H[V(H)\backslash\{v_{i+1}, \ldots, v_{v(H)}\}]$ contains a vertex of degree at most $km_k(F)$. Let $v_i$ be an arbitrary such vertex. Clearly, this sequence has the property that for every $i \leq v(H)$ the degree of $v_i$ to $\{v_1, v_2, \ldots, v_{i-1}\}$ is at most $km_k(F)$.

We now show that when we start with the empty hypergraph and introduce the vertices (together with their incident hyperedges) step by
step according to this ordering, then for every \( 1 \leq i \leq v(H) \) we can avoid creating a monochromatic copy of \( F \) in step \( i \). For this, let \( E_i \) denote the set of hyperedges that are introduced by adding vertex \( v_i \). Let \( H_i^{(k-1)} \) denote the link of \( v_i \), i.e., the \( (k-1) \)-uniform hypergraph whose hyperedge-set is obtained by removing \( v_i \) from every hyperedge in \( E_i \). Then, we have \( e(H_i^{(k-1)}) \leq km_k(F) \). By Lemma 6.15 we can thus find a 2-hyperedge-coloring of \( H_i^{(k-1)} \) without a monochromatic copy of \( K_{\ell-1}^{(k-1)} \). Since every monochromatic copy of \( K_{\ell}^{(k)} \) that contains \( v_i \) induces a monochromatic \( K_{\ell-1}^{(k-1)} \) in the link of \( v_i \), this coloring corresponds to a coloring of \( E_i \) that does not create a monochromatic copy of \( K_{\ell}^{(k)} \).

It remains to bound the size-Ramsey number of \( K_{\ell-1}^{(k-1)} \). We denote by \( r(\cdot) \) the classical Ramsey number, i.e. the minimum number \( m \) such that \( K_m \rightarrow (K_\ell)_2^e \). We make use of the following theorem by Erdős, Faudree, Rousseau and Schelp [EFRS78].

**Theorem 6.16** ([EFRS78]). For all integers \( \ell \) it holds that \( r_e(K_\ell) = \left( r(K_\ell) \right)^2 \).

*Proof (of Lemma 6.15).* We split the proof into two steps. We first show by a combinatorial argument that for every \( \ell > k \geq 2 \) we have

\[
r_e(K_\ell^{(k)}) \geq \binom{r(K_{\ell-k+2}) + k - 2}{k}.
\]  

(6.16)

We then show that for every \( \ell > k \geq 2 \) we have

\[
\binom{r(K_{\ell-k+2}) + k - 3}{k-1} \geq km_k(K_\ell^{(k)}),
\]

(6.17)

which concludes the proof.

We show by induction on \( k \) that (6.16) holds for every \( \ell > k \geq 2 \). The base case \( k = 2 \) immediately follows from Theorem 6.16. Now let \( H^* \) be a \( (k+1) \)-uniform hypergraph with \( H^* \rightarrow (K_\ell^{(k+1)})^e_2 \) and \( e(H^*) = r_e(K_\ell^{(k+1)}) \). In order to prove (6.16) we need to show that

\[
e(H^*) \geq \binom{r(K_{\ell-k+1}) + k - 1}{k+1}.
\]
We first claim that
\[ \delta(H^*) \geq r_e(K_{(k)}) . \] (6.18)

Note that otherwise there exists \( v \in V(H^*) \) such that the link of \( v \) is a \( k \)-uniform hypergraph with less than \( r_e(K_{(k)}) \) hyperedges, that is we can color the hyperedges of the link of \( v \) without a monochromatic copy of \( K_{(k)}^{(k)} \). Moreover since \( e(H^*) = r_e(K_{(k+1)}) \) it follows that \( H^* - v \not\rightarrow (K_{(k)}^{(k+1)})^2 \). Hence, there exists a valid hyperedge-coloring of \( H^* - v \), i.e., one without a monochromatic copy of \( K_{(k+1)}^{(k+1)} \). Furthermore, using the above hyperedge-coloring of the link of \( v \) we can extend this coloring of \( H^* - v \) to a valid coloring of \( H^* \). Hence, (6.18) holds and with the induction hypothesis we obtain
\[ \delta(H^*) \geq r_e(K_{(k)}) \geq \left( \frac{r(K_{\ell-k+2}) + k - 2}{k} \right) = \left( \frac{r(K_{\ell-k+1}) + k - 2}{k} \right) . \]

It is not hard to see that this implies that the neighborhood of every vertex must have size at least \( r(K_{\ell-k+1}) + k - 2 \). From this we infer
\[ v(H^*) \geq r(K_{\ell-k+1}) + k - 2 + 1 = r(K_{\ell-k+1}) + k - 1 , \]
which implies
\[ e(H^*) \geq \frac{v(H^*)}{k+1} \left( \frac{r(K_{\ell-k+1}) + k - 2}{k} \right) \geq \left( \frac{r(K_{\ell-k+1}) + k - 1}{k+1} \right) . \]

This establishes (6.16) and it remains to prove (6.17) for every \( \ell > k \geq 3 \).

Case 1. \( \ell = k + 1 \). In this case we use \( r(K_3) = 6 \) and \( m_k(K_{(k)}) = k \) to obtain
\[ \left( \frac{r(K_{\ell-k+2}) + k - 3}{k-1} \right) = \left( \frac{k+3}{k-1} \right) = \left( \frac{k+3}{4} \right) > k^2 = km_k(K_{(k)}^{(k)}) . \]

Case 2. \( \ell \geq k + 2 \). First, observe that \( r(K_{\ell-k+2}) > (\ell-k+1)^2 \) since the following 2-edge-coloring of the complete simple graph on \( (\ell-k+1)^2 \) vertices does not contain a monochromatic copy of \( K_{\ell-k+2} \). Partition the vertices into \( \ell - k + 1 \) groups of \( \ell - k + 1 \) vertices; then color all edges that run inside one group with one color and all edges that run between two groups with the other color. Clearly, this coloring does not contain a monochromatic copy of \( K_{\ell-k+2} \). Together with \( \ell - k \geq 2 \) we
obtain
\[
\begin{align*}
\binom{r(K_{\ell-k+2}) + k - 3}{k-1} &> \binom{(\ell - k + 1)^2 + k - 3}{k-1} \\
&\geq \binom{3(\ell - k + 1) + k - 3}{k-1} \\
&= \binom{\ell + 2(\ell - k)}{k-1} \\
&\geq \binom{\ell + 4}{k-1} = \frac{\ell + 4}{k-1} \binom{\ell}{k-2} > \frac{\ell + 4}{k-1} \binom{\ell}{k-2}.
\end{align*}
\]
Moreover, we have
\[
\begin{align*}
\binom{\ell}{k-2} \binom{\ell + 4}{k-1} &= \binom{\ell}{k-1} \frac{\ell + 4}{\ell - k + 2} > \frac{\ell - k + 1}{\ell - k} \\
&= \frac{k}{\ell - k} > \frac{k}{\ell - k} = m_k(K_{\ell}^{(k)}).
\end{align*}
\]

6.4.2 Hypergraphs with Large Chromatic Number

Here we show Theorem 6.13 for k-uniform hypergraphs \( F \) with \( \chi(F) \geq k + 1 \).

In the following \( \delta(F) \) denotes the minimum vertex degree of \( F \), i.e. the minimum number of hyperedges that any vertex is contained in. We first show that \( \chi(F) \geq k + 1 \) implies that every hypergraph \( H \) with \( m(H) \leq m_k(F) \) is not vertex Ramsey for \( F \) with \( k \) colors.

**Lemma 6.17.** Let \( F \) and \( H \) be \( k \)-uniform hypergraphs, \( k \geq 2 \). If \( F \) is strictly \( k \)-balanced, \( \chi(F) \geq k + 1 \) and \( m(H) \leq m_k(F) \), then we have \( H \not\rightarrow (F)^v_k \).

**Proof.** We need to show that there exists a \( k \)-coloring of the vertices of \( H \) that does not contain a monochromatic copy of \( F \). First observe that \( v(F) > k \) since otherwise \( F \) has chromatic number 2. Moreover, it is not hard to see that \( \delta(F) > m_k(F) \). If this is not the case, then there exists \( v \in V(F) \) with \( \deg(v) \leq m_k(F) = \frac{e(F)-1}{v(F)-k} \) and the graph
obtained by deleting $v$ from $F$ satisfies
\[ d_k(F - v) \geq \frac{e(F) - 1 - \frac{e(F) - 1}{v(F) - k - 1}}{v(F) - k - 1} = \frac{e(F) - 1}{v(F) - k} = m_k(F), \]
contradicting the strict $k$-balancedness of $F$.

As in the proof of Theorem 6.13 (1) we have that $m(H) \leq m_k(F)$ implies that $H$ is $km_k(F)$-degenerate. That is, every subhypergraph of $H$ contains a vertex of degree at most $km_k(F)$ and there exists an ordering $v_1, v_2, \ldots, v_{v(H)}$ of the vertices of $H$ such that for every $i \leq v(H)$ the degree of $v_i$ to \{v_1, v_2, \ldots, v_{i-1}\} is at most $km_k(F) < k\delta(F)$. Inserting the vertices of $H$ one by one in this ordering we can now iteratively construct a vertex-coloring of $H$ with $k$ colors that does not create a monochromatic copy of $F$. Consider the insertion of vertex $v_i$, $i \leq v(H)$. By construction, $v_i$ is contained in less than $k\delta(F)$ hyperedges in $H[\{v_1, v_2, \ldots, v_i\}]$. By the pigeonhole-principle there exists a color $1 \leq c \leq k$ such that by assigning color $c$ to $v_i$ we obtain that $v_i$ is contained in less than $\delta(F)$ monochromatic hyperedges, thereby avoiding the creation of a monochromatic copy of $F$ in this insertion.

Proof (of Theorem 6.13 (2)). Lemma 6.17 asserts a vertex-coloring $c : V(H) \to [k]$ with $k$ colors such that no copy of $F$ is monochromatic.

We now color the hyperedges of $H$ with 2 colors as follows. If for $e \in E(H)$ we have $|c(e)| = 1$, i.e. $e$ is monochromatic under $c$, then we color $e$ red, otherwise we color $e$ blue. We claim that this hyperedge-coloring, denoted by $\hat{c} : E(H) \to \{\text{red}, \text{blue}\}$ of $H$ does not contain monochromatic copies of $F$. Since every strictly $k$-balanced hypergraph is connected, a red copy of $F$ in $\hat{c}$ implies that there is also a monochromatic copy of $F$ in the vertex-coloring $c$, which is a contradiction. Now assume, that $\hat{c}$ induces a blue copy of $F$. By construction of $\hat{c}$ no hyperedge of that copy of $F$ is monochromatic in $c$. Since $c$ uses only $k$ colors, this contradicts $\chi(F) \geq k + 1$. We conclude that $H \not\rightarrow (F)_2^\epsilon$.

### 6.4.3 Hypergraphs of High Density

In this subsection we prove part (3) from Theorem 6.13. The proof is in some sense a generalization of the one from Theorem 6 in [RR93]. It is split into two parts. In one part we prove that every hypergraph $F$ that satisfies $m(F) \geq 2 + \frac{2k-1}{v(F) - 2k}$ also satisfies $m_k(F) \leq 2([m(F)] - 1)$,
and in the other part we show that this implies that we can color the hyperedges of every hypergraph $H$ that satisfies $m(H) \leq m_k(F)$ without a monochromatic copy of $F$. We first give the proof of the second part and then address the (technical) first part.

**Lemma 6.18.** Let $F$ be a strictly $k$-balanced hypergraph with $m_k(F) \leq 2(\lceil m(F) \rceil - 1)$, and $H$ be a hypergraph with $m(H) \leq m_k(H)$. Then $H \not\rightarrow (F)^2$.

**Proof.** We generalize the Nash-Williams Theorem [NW61][NW64] for hypergraphs. More precisely, we show that we can partition $H$ into at most $s = 2(\lceil m(F) \rceil - 1)$ subhypergraphs of maximum hyperedge-density at most 1 each. For this, let $B = (L \cup R, E_B)$ be the following bipartite graph. Let $L = E(H)$ and $R$ consists of $s$ copies $v_1, \ldots, v_s$ of every vertex $v \in V(H)$, i.e., $|R| = s \cdot v(H)$ and for every hyperedge $e \in L$ and every copy $v_i \in R$ of the vertex $v \in V(H)$ we have $\{e, v_i\} \in E_B$ if and only if $v \in e$. Now, let $A \subseteq L$ and let $H_A$ denote the hypergraph formed by the hyperedges in $A$. Then we have that the neighborhood of $A$ in the graph $B$ satisfies

$$|N_B(A)| = s|V(H_A)| \geq \frac{s|A|}{m(H)} \geq |A|,$$

where we used in the last step that $m(H) \leq m_k(F) \leq s$. By Hall’s theorem we can derive that $B$ contains a matching of size $|L| = |E(H)|$. We now define a partition of $H$ into $s$ subhypergraphs $H_1, \ldots, H_s$ as follows. Let $H_i$ denote the hypergraph formed by the hyperedges that are matched to an $i$-th copy of some vertex $v \in V(H)$. Clearly, this is a partition of the hyperedges, and moreover every hypergraph $H_i$ satisfies $m(H_i) \leq 1$ since the matching in $B$ describes an injective mapping from the hyperedge set to the vertex set and thus every subhypergraph contains at least as many vertices as hyperedges.

We now color the hyperedges of $H_1, H_2, \ldots, H_{s/2}$ blue and the edges in $H_{s/2+1}, \ldots, H_s$ red. Then the blue subhypergraph has maximum hyperedge-density at most $s/2 = \lceil m(F) \rceil - 1 < m(F)$ and thus cannot contain a monochromatic copy of $F$. The same argument applies to the red subhypergraph.

In order to prove point (3) in Theorem 6.13 it suffices to prove the following technical lemma.\qed
Lemma 6.19. Let $F$ be a strictly $k$-balanced hypergraph on at least $2k+1$ vertices with $m(F) \geq 2 + \frac{2^k - 1}{v(F) - 2k}$. Then $m_k(F) \leq 2\lceil m(F) \rceil - 1$.

Proof. We have

\[ m_k(F) = \frac{e(F) - 1}{v(F) - k} \]
\[ = \frac{e(F)}{v(F)} \cdot \frac{v(F) - k}{v(F) - k} - \frac{1}{v(F) - k} \]
\[ \leq m(F) \cdot \frac{v(F)}{v(F) - k} - \frac{1}{v(F) - k} \]
\[ = 2m(F) - \left(2 - \frac{v(F)}{v(F) - k}\right)m(F) - \frac{1}{v(F) - k} \]
\[ \leq 2m(F) - \frac{v(F) - 2k}{v(F) - k} \cdot \frac{2v(F) - 2k - 1}{v(F) - k} - \frac{1}{v(F) - k} \]
\[ = 2m(F) - 2 \leq 2\lceil m(F) \rceil - 1. \]

6.4.4 Further Ramsey-Density Obeying Hypergraphs

This subsection is devoted to the proof of Theorem 6.13 (4). We first consider the case that $\lfloor km_k(F) \rfloor \leq r(\delta(F) - 1)$.

As in the proof of part (1) we use that $m(H) \leq m_k(F)$ implies that $H$ is $km_k(F)$-degenerate. Hence, there exists an ordering $v_1, v_2, \ldots, v_{v(H)}$ of the vertices of $H$ such that for every $1 \leq i \leq v(H)$ the degree of $v_i$ into $\{v_1, v_2, \ldots, v_{i-1}\}$ is at most $km_k(F)$. Since degrees are integers this degree is in fact bounded by $\lfloor km_k(F) \rfloor \leq r(\delta(F) - 1)$. We now argue how to obtain a coloring of the hyperedges of $H$ with $r$ colors without creating a monochromatic copy of $F$. For this, we insert the vertices of $H$ in this ordering one by one into an initially empty hypergraph and show that we can avoid creating a monochromatic copy of $F$ in every insertion. Assume that the hyperedges of $H[\{v_1, v_2, \ldots, v_{i-1}\}]$ are colored appropriately and consider the insertion of vertex $v_i$ together with all incident hyperedges. Since its degree is at most $r(\delta(F) - 1)$ we can color these hyperedges such that each of the $r$ colors is used at most $\delta(F) - 1$ times. This immediately implies that we did not create a monochromatic copy of $F$ with the insertion of $v_i$.

We now turn to the case that $\lfloor km_k(F) \rfloor < (\chi(F) - 1)^r$. As in the preceeding paragraph we use an ordering $v_1, v_2, \ldots, v_{v(H)}$ such that for
Chapter 6. Ramsey Properties of Random Hypergraphs

Every \(1 \leq i \leq v(H)\) the degree of \(v_i\) into \(\{v_1, v_2, \ldots, v_{i-1}\}\) is at most \([km_k(F)] \leq (\chi(F) - 1)^r - 1\). Hence, we can greedily obtain a proper vertex-coloring of \(H\) with \((\chi(F) - 1)^r\) colors, which we denote by \(c : V(H) \rightarrow \{1, 2, \ldots, \chi(F) - 1\}^r\). Observe that we represent each color by a vector of length \(r\) with entries in \(\{1, 2, \ldots, \chi(F) - 1\}\). We now show how to partition the hyperedges of \(H\) into \(r\) sets \(E_1, E_2, \ldots, E_r\) such that each of these parts forms a hypergraph of chromatic number at most \(\chi(F) - 1\). For this, let \(E_i\) denote the set of hyperedges \(e\) for which the color of all its \(k\) vertices agree on the first \(i - 1\) coordinates, but there exist \(u, v \in e\) such that \(c(u)\) and \(c(v)\) differ in the \(i\)-th coordinate. In other words, \(i\) is the minimum coordinate to certify that \(e\) is not monochromatic. Since \(c\) is proper, every hyperedge belongs to exactly one part \(E_1, E_2, \ldots, E_r\). Moreover, it is easy to see that for every \(1 \leq i \leq r\) the hypergraph \(H_i := (V(H), E_i)\) admits a vertex-coloring with \(\chi(F) - 1\) colors by assigning every vertex \(v \in V(H)\) the entry of the \(i\)-th coordinate of \(c(v)\). This implies that \(H_i\) has chromatic number less than \(\chi(F)\) and thus does not contain a copy of \(F\). Now, coloring all hyperedges in \(E_i\) with color \(i\) for every \(1 \leq i \leq r\) yields a hyperedge-coloring of \(H\) with \(r\) colors which does not contain a monochromatic copy of \(F\). This completes the proof of part (4) of Theorem 6.13.

6.5 Exceptional structures

In this section we present a few examples for hypergraphs whose threshold behavior does not go along the statement of Theorem 6.2. Recall that in the case of simple graphs there are only two exceptions for which the threshold behavior differs from the general case: star forests and the path of length 3 in the case of 2 colors, cf. our remarks in the introduction of this chapter. We first show that both can be generalized to the setup of \(k\)-uniform hypergraphs. Unfortunately, it turns out that there are even more hypergraphs with a different threshold behavior as we illustrate in section 6.5.3.

6.5.1 Star Forests

We first consider a generalization of stars to hypergraphs, see also Figure 6.6. We say that the \(k\)-uniform hypergraph \(S\) is a \(t\)-star, \(t \in [k - 1]\), if all hyperedges of \(S\) intersect pairwise in the same set of \(t\) vertices but...
are otherwise disjoint. These vertices are called the center of the star \( S \). Let \( S_{\ell,t}^{(k)} \) denote a \( k \)-uniform \( t \)-star with \( \ell \) hyperedges. It is easy to see that

\[
m(S_{\ell,t}^{(k)}) = \frac{\ell}{(k-t)\ell + t} = \frac{1}{k-t + \frac{t}{\ell}} \quad \text{and} \quad m_k(S_{\ell,t}^{(k)}) = \frac{1}{k-t}.
\]

Observe that Theorem 3.5 implies that for every \( p \geq Cn^{t-k} \) for a large enough constant \( C > 0 \) we a.a.s. have \( H^{(k)}(n,p) \rightarrow (S_{\ell,t}^{(k)})^e_r \). However, as for stars in the simple graph case it follows by the pigeonhole principle that \( S_{(\ell-1)+1,t}^{(k)} \rightarrow (S_{\ell,t}^{(k)})^e_r \). It is well known (see also Theorem 6.20 in the appendix of this chapter) that the appearance of \( S_{(\ell-1)+1,t}^{(k)} \) in \( H^{(k)}(n,p) \) has a coarse threshold at

\[
p = n^{-1/m(S_{r(\ell-1)+1,t}^{(k)})} = n^{t-k-\frac{(\ell-1)r+1}{t}}.
\]

Hence, we have a.a.s. that \( H^{(k)}(n,p) \rightarrow (S_{\ell,t}^{(k)})^e_r \) even for every \( p \gg n^{t-k-\frac{(\ell-1)r+1}{t}} \). This shows that there cannot be a 0-statement that matches the bounds of Theorem 3.5. In fact, one can show that \( S_{r(\ell-1)+1,t}^{(k)} \) is the sparsest counterexample. Applying our proof of Theorem 6.2 together with the fact that \( H^{(k)}(n,p) \) does a.a.s. not contain any constant-size counterexamples for \( p \ll n^{t-k-\frac{(\ell-1)r+1}{t}} \) we obtain that the property \( H^{(k)}(n,p) \rightarrow (S_{\ell,t}^{(k)})^e_r \) has a coarse threshold at \( p = n^{t-k-\frac{(\ell-1)r+1}{t}} \).

More generally, if \( F \) is a vertex-disjoint collection of stars \( (S_{\ell_i,t_i}^{(k)})_i \) (i.e., a star forest), then the threshold probability for \( H^{(k)}(n,p) \rightarrow (F)^e_r \) is \( p = n^w \), where we let \( w = \max_i \left( t_i - k - \frac{t_i}{(\ell_i-1)r+1} \right) \).

### 6.5.2 Hyperpaths of Length 3 and Sunshine Hypergraphs

The second exception for simple graphs is the case of two colors and a path \( P_3 \) of 3 edges. For this graph there exists an infinite family of counterexamples, the so-called (odd) sunshine graphs \( S_{2\ell+1}^\oplus \) for \( \ell \geq 2 \), defined as the graph consisting of an odd cycle of length \( 2\ell + 1 \geq 5 \) where every vertex of the cycle is connected to a pendant edge (ray). It is easy to see that these graphs satisfy \( S_{2\ell+1}^\oplus \rightarrow (P_3)_r^e \). Moreover, observe that \( m(S_{2\ell+1}^\oplus) = 1 = m_2(P_3) \). It is known that for \( p = o(1/n) \),
we have a.a.s. that $G(n, p)$ is a forest. Hence, it does not contain any sunshine graph and one can show that $G(n, p)$ does indeed not contain any counterexample that forces us to create a monochromatic $P_3$. On the other hand, it is known that there is a constant $C > 0$, such that for $p \geq C/n$, $G(n, p)$ contains a.a.s. an element from the family of odd sunshine graphs. Altogether, one can show that the threshold behavior for avoiding a monochromatic $P_3$ with 2 colors is as follows. There exists a constant $C > 0$ such that

$$
\lim_{n \to \infty} \Pr \left[ G(n, p) \to (P_3)_{c(n)}^\delta \right] = \begin{cases} 
0 & \text{if } p \ll n^{-1}, \text{ and} \\
1 & \text{if } p \geq Cn^{-1}, \text{ and} \\
c(p) & \text{if } p = c'n^{-1} \text{ for a constant } 0 < c' < C.
\end{cases}
$$

In the following we describe generalizations of paths and sunshine graphs to hypergraphs. We denote by $P_{\ell, t}^{(k)}$ the $k$-uniform $t$-tight path with $\ell$ hyperedges, that is, the hypergraph on the vertex set $[(k-t)\ell + t]$ with hyperedges $e_1, \ldots, e_{\ell}$ such that $e_i = \{(k-t)(i-1)+1, \ldots, (k-t)(i-1)+k\}$. Note that two consecutive hyperedges of $P_{\ell, t}^{(k)}$ intersect in exactly $t$ vertices. Recall that $C_{\ell, t}^{(k)}$ denotes a $k$-uniform $t$-tight cycle with $\ell$ hyperedges (see Figure 6.5), that is, the hypergraph on the vertex set $[\ell(k-t)]$ which corresponds to a path $P_{\ell, t}^{(k)}$ 'wrapped around' by identifying the last $t$ vertices of $P_{\ell, t}^{(k)}$ with the first $t$ vertices (in increasing order). It is
easy to check that
\[ m_k(P_{\ell,t}^{(k)}) = \frac{\ell - 1}{(k - t)\ell + t - k} = \frac{1}{k - t} \]
and
\[ m(C_{\ell,t}^{(k)}) = \frac{\ell}{\ell(k - t)} = \frac{1}{k - t}. \]

Next we define the sunshine hypergraph \( S_{\ell,t}^{\odot(k)} \) (see also Figure 6.6) as the \( t \)-tight cycle \( C_{\ell,t}^{(k)} \) with \( \ell \) additional hyperedges each containing distinct intersections \( e_i \cap e_{i+1} \) or \( e_t \cap e_1 \), respectively, and \( k - t \) additional new vertices. We therefore have \( v(S_{\ell,t}^{\odot(k)}) = 2(k - t)\ell \) and \( e(S_{\ell,t}^{\odot(k)}) = 2\ell \). One easily checks that
\[ m(S_{\ell,t}^{\odot(k)}) = \frac{1}{k - t}. \]

Observe that \( P_{3,1}^{(2)} \) and \( S_{2\ell+1,1}^{\odot(2)} \) correspond to the simple graphs \( P_3 \) and the sunshine graph \( S_{2\ell+1}^{\odot} \). As in the graph case it is not hard to see that \( S_{2\ell+1,t}^{\odot(k)} \rightarrow (P_{3,t}^{(k)})^c_2 \). It is known that for every \( k, t \) and every constant \( c > 0 \) and \( p = cn^{t-k} \) we have for every \( \ell \) that
\[ \lim_{n \to \infty} \Pr [H^{(k)}(n,p) \text{ contains a copy of } S_{2\ell+1,t}^{\odot(k)}] = c' \]
for a constant \( c' = c'(t, k, \ell, c) > 0 \). Hence, the 0-statement for the property \( H^{(k)}(n,p) \rightarrow (P_{3,t}^{(k)})^c_2 \) can possibly hold only if \( p \ll n^{t-k} \).

This is however not the end of the story. Observe that for every \( t > k/2 \) that path \( P_{3,t}^{(k)} \) contains at least one vertex which is contained in all three hyperedges. We now show how we can exploit this observation to construct even sparser counterexamples. We first consider the kase \( k = 3 \) and \( t = 2 \). Let \( H^* \) denote the \( k \)-uniform hypergraph that consists of a vertex whose link forms the sunshine graph \( S_5^{\odot} \). Formally, \( V(H^*) = V(S_5^{\odot}) \cup \{w\} \) and \( E(H^*) = \{\{u,v,w\} : \{u,v\} \in E(S_5^{\odot})\} \). We write \( H^* = \{w\} \vdash S_5^{\odot} \) for this construction. It is not hard to see that \( H^* \rightarrow (P_{3,2}^{(3)})^c_2 \). Moreover, we have
\[ m(H^*) = \frac{10}{11} < 1 = m_3(P_{3,2}^{(3)}). \]
Hence, in this case the appearance of $H^*$ in $H^{(3)}(n,p)$ improves on the 1-statement from Theorem 3.5 (which holds for every $p \geq Cn^{-1}$) since it implies that even for every $p \gg n^{-11/10}$ we have

$$\lim_{n \to \infty} \Pr \left[ H^{(3)}(n,p) \to (P_{3,2}^{(3)})\right] = 1.$$ 

Moreover, we can generalize this construction to arbitrary $k, t$ with $t > k/2$ and improve on the 1-statement of Theorem 3.5 which for $F = P_{3,t}^{(k)}$ is shown for every $p \geq Cn^{t-k}$. More precisely, let $s$ denote the number of vertices that are contained in all 3 hyperedges of $P_{3,t}^{(k)}$, that is, $s = 2t - k$. Setting $H^* := \{w_1, w_2, \ldots, w_s\} \cup S^{\otimes (k-s)}_{5,k-t}$, one easily checks that $H^* \to (P_{3,t}^{(k)})^{e_2}$ and

$$m(H^*) = \frac{10}{10(k-t) + s} < \frac{1}{k-t} = m(S^{\otimes (k)}_{5,t}) = m_k(P_{3,t}^{(k)}).$$

Hence, we have a.a.s. that $H^{(k)}(n,p) \to (P_{3,t}^{(k)})$ for every $p \gg n^{-1/m(H^*)} = n^{t-k-s/10}$. It is this construction that motivates the following section.

### 6.5.3 Exceptional Structures with High Uniformity

In this subsection we show how to use the above construction to obtain hypergraphs of high uniformity which do not have a 0-statement matching the bounds of Theorem 3.5. For every hypergraph $H$ and $t \geq 1$ we denote by $H^{r+t}$ the $(k+t)$-uniform hypergraph $\{w_1, w_2, \ldots, w_t\} \cup H$ for some distinct vertices $w_1, w_2, \ldots, w_t$, that is, the hyperedges of $H^{r+t}$ are all of the form $\{w_1, w_2, \ldots, w_t\} \cup e$ for some $e \in E(H)$. Observe that for large enough $t$ we have that $H^{r+t}$ is strictly balanced since for every $J \subsetneq H^{r+t}$ we have $e(J) < e(H)$ and thus also

$$\frac{e(J)}{v(J) + t} < \frac{e(H)}{v(H) + t}.$$

This implies that $m(H^{r+t}) = \frac{e(H)}{v(H) + t}$ for large enough $t$.

Now, let $r \geq 2$ be fixed, let $F$ be an arbitrary (not necessarily balanced or $k$-balanced) $k$-uniform hypergraph and let $H$ be an arbitrary $k$-uniform hypergraph such that $H \to (F)^{e_r}$. We claim that for large
6.6. Open Problems

The smallest hypergraph, for which we do not know the threshold is $K_4^{(3)} \setminus e$, the 3-uniform hypergraph with 3 hyperedges and 4 vertices. All criteria from Theorem 6.13 fail and none of the constructions from Section 6.5 yield a counterexample for this hypergraph. Another small hypergraph for which we would like to see a solution is the 3-uniform hypergraph with vertex set \{1, 2, 3, 4, 5\} and hyperedges \{1, 2, 3\}, \{1, 2, 4\} and \{3, 4, 5\}. We call it the mushroom hypergraph (where the hyperedge \{3, 4, 5\} forms the hat of the mushroom). Notice that this hypergraph is neither strictly 3-balanced nor Ramsey-density-obeying.
Finally, we note that for all hypergraphs that we mention as exceptional structures the intersection over all hyperedges is non-empty. We leave it as an open problem whether it is possible to find exceptional structures which are not of this form. Observe that the mushroom hypergraph is a possible candidate.

Appendix

Theorem 6.20. Let \( H \) be a non-empty hypergraph. Then we have

\[
\lim_{n \to \infty} \Pr[H \subseteq H^{(k)}(n,p)] = \begin{cases} 
0 & \text{if } p \ll n^{-1/m(H)}, \\
1 & \text{if } p \gg n^{-1/m(H)}.
\end{cases}
\]

Proof. We use a first and second moment method argument. The same argument can be used to prove the standard small subgraphs theorem for simple graph, cf. Theorem 3.2.

We first consider the case \( p \ll n^{-1/m(H)} \) and show that for such \( p \) the random hypergraph \( H^{(k)}(n,p) \) a.a.s. contains no copy of \( H \). For this, let \( H' \subseteq H \) with \( d(H') = m(H) \) and such that \( H' \) is strictly balanced. Clearly, \( H' \) exists since the set of all subhypergraphs \( J \subseteq H \) with \( d(J) = m(H) \) must contain such a hypergraph. Clearly, if \( H^{(k)}(n,p) \) does not contain a copy of \( H' \), then it does also not contain a copy of \( H \).

Let \( X_{H'} \) denote the number of copies of \( H' \) in \( H^{(k)}(n,p) \). Observe that we can count the number of possible such copies by first choosing \( v(H') \) vertices (\( \binom{n}{v(H')} \) choices) and then the role of each of these \( v(H') \) vertices in the copy of \( H' \) \((v(H')!/\text{aut}(H')) \) choices, where \( \text{aut}(H') \) denotes the number of automorphisms of \( H' \). Furthermore, each such copy is present in \( H^{(k)}(n,p) \) with probability \( p^{e(H')} \). Hence, we have

\[
\mathbb{E}[X_{H'}] = \binom{n}{v(H')} \frac{v(H')!}{\text{aut}(H')} \cdot p^{e(H')} = (1 - o(1)) \frac{n^{v(H')} p^{e(H')}}{\text{aut}(H')} = \Theta \left( n^{v(H')} p^{e(H')} \right),
\]

which for \( p \ll n^{-1/m(H)} = n^{-v(H')/e(H')} \) immediately implies

\[
\mathbb{E}[X_{H'}] = o(1) .
\]
Hence, by the first moment method we have \( \Pr[H' \subseteq H^{(k)}(n,p)] = o(1) \) if \( p \ll n^{-1/m(H)} \).

Having settled the 0-statement it remains to prove the 1-statement. For this, we use the second moment method. First note that similar to (6.19) we have

\[
\mathbb{E}[X_H] = \Theta \left( n^{v(H)} p^{e(H)} \right),
\]

which for \( p \gg n^{-1/m(H)} \geq n^{-v(H)/e(H)} \) immediately implies \( \mathbb{E}[X_H] = \omega(1) \). We now bound the variance of \( X_H \). For this, let \( \mathcal{H} \) denote the set of all possible copies of \( H \) on \( n \) vertices. Note that for every \( H_1, H_2 \in \mathcal{H} \) with \( e(H_1) \cap e(H_2) = \emptyset \) the presence of \( H_1 \) and \( H_2 \) in \( H^{(k)}(n,p) \) are independent. Hence, for such \( H_1, H_2 \) we have

\[
\Pr[H_1 \subseteq H^{(k)}(n,p) \land H_2 \subseteq H^{(k)}(n,p)] - \Pr[H_1 \subseteq H^{(k)}(n,p)] \Pr[H_2 \subseteq H^{(k)}(n,p)] = 0.
\]

With this, we obtain

\[
\text{Var}[X_H] = \sum_{H_1, H_2 \in \mathcal{H}} \Pr[H_1 \subseteq H^{(k)}(n,p) \land H_2 \subseteq H^{(k)}(n,p)] - \Pr[H_1 \subseteq H^{(k)}(n,p)] \Pr[H_2 \subseteq H^{(k)}(n,p)] \\
\leq \sum_{J \subseteq H, \ H_1, H_2 \in \mathcal{H}, \ e(J) \geq 1} \sum_{H_1 \cap H_2 \geq J} \Pr[H_1 \subseteq H^{(k)}(n,p) \land H_2 \subseteq H^{(k)}(n,p)] \\
= \sum_{J \subseteq H, \ H_1, H_2 \in \mathcal{H}, \ e(J) \geq 1} \sum_{H_1 \cap H_2 \geq J} p^{2e(H) - e(J)}.
\]

(6.20)

Note that for fixed \( J \subseteq H \) with \( e(J) \geq 1 \) the number of possible copies of \( H_1, H_2 \) that intersect exactly in a copy of \( J \) is of order \( n^{2v(H) - v(J)} \). Furthermore, for \( p \gg n^{-1/m(H)} \) we have for every \( J \subseteq H \) that \( p^{-e(J)} \ll n^{v(J)} \) and hence \( n^{-v(J)} p^{-e(J)} = o(1) \). We can thus continue (6.20) with

\[
\text{Var}[X_H] = \sum_{J \subseteq H, \ e(J) \geq 1} \Theta \left( n^{2v(H)} p^{2e(H)} \right) \Theta \left( n^{-v(J)} p^{-e(J)} \right) \\
= \sum_{J \subseteq H, \ e(J) \geq 1} o \left( n^{2v(H)} p^{2e(H)} \right) \\
= o(n^{2v(H)} p^{2e(H)}) = o(\mathbb{E}[X_H]^2).
\]
By the second moment method it then follows that the probability that $H^{(k)}(n,p)$ contains a copy of $H$ is $1 - o(1)$ if $p \gg n^{-1/m(H)}$. \qed
In this chapter we present our findings on the well-known game of Mastermind with many colors. This is joint work with Benjamin and Carola Doerr and Reto Spöhel. It has been presented at the Symposium on Discrete Algorithms 2013 in New Orleans and submitted to the Journal of the ACM.

7.1 Introduction

Mastermind (see Section 7.1.1 for the rules) and other guessing games like liar games [Pel02, Spe94] have attracted the attention of computer scientists not only because of their playful nature, but more importantly because of their relation to fundamental complexity and information-theoretic questions. In fact, Mastermind with two colors was first an-
analyzed by Erdős and Rényi [ER63b] in 1963, several years before the release of Mastermind as a commercial boardgame.

Since then, intensive research by various scientific communities produced a plethora of results on various aspects of the Mastermind game (cf. Section 3.2.2). In a famous 1983 paper, Chvátal [Chv83] determined, precisely up to constant factors, the asymptotic number of queries needed on a board of size $n$ for all numbers $k$ of colors with $k \leq n^{1-\varepsilon}$, $\varepsilon > 0$ a constant. Interestingly, a very simple guessing strategy suffices, namely asking random guesses until the answers uniquely determine the secret code.

Surprisingly, for larger numbers of colors, no sharp bounds exist. In particular for the natural case of $n$ positions and $k = n$ colors, Chvátal’s bounds $O(n \log n)$ and $\Omega(n)$ from 1983 are still the best known asymptotic results.

We almost close this gap open for roughly 30 years and prove that Codebreaker can solve the $k = n$ game using only $O(n \log \log n)$ guesses. This bound, as Chvátal’s, even holds for black-pegs only Mastermind. When also white answer-pegs are used, we obtain a similar improvement from the previous-best $O(n \log n)$ bound to $O(n \log \log n)$ for all $n \leq k \leq n^2 \log \log n$.

### 7.1.1 Mastermind

Mastermind is a two-player board game invented in the seventies by the Israeli telecommunication expert Mordechai Meirowitz. The first player, called Codemaker here, privately chooses a color combination of four pegs. Each peg can be chosen from a set of six colors. The goal of the second player, Codebreaker, is to identify this secret code. To do so, he guesses arbitrary length-4 color combinations. For each such guess he receives information of how close his guess is to Codemaker’s secret code. Codebreaker’s aim is to use as few guesses as possible.

Besides the original 4-position 6-color Mastermind game, various versions with other numbers of positions or colors are commercially available. The scientific community, naturally, often regards a generalized version with $n$ positions and $k$ colors (according to Chvátal [Chv83], this was first suggested by Pierre Duchet). For a precise description of this game, let us denote by $[k]$ the set $\{1, \ldots, k\}$ of positive integers not exceeding $k$. At the start of the game, Codemaker chooses a secret
code $z \in [k]^n$. In each round, Codebreaker guesses a string $x \in [k]^n$. Codemaker replies with the numbers $\text{eq}(z, x) := |\{i \in [n] \mid z_i = x_i\}|$ of positions in which her and Codebreaker’s string coincide, and with $\pi(z, x)$, the number of additional pegs having the right color, but being in the wrong position. Formally, $\pi(z, x) := \max_{\rho \in S_n} |\{i \in [n] \mid z_i = x_{\rho(i)}\}| - \text{eq}(z, x)$, where $S_n$ denotes the set of all permutations of the set $[n]$. In the original game, $\text{eq}(z, x)$ is indicated by black answer-pegs, and $\pi(z, x)$ is indicated by white answer-pegs. Based on this and all previous answers, Codebreaker may choose his next guess. He “wins” the game if his guess equals Codemaker’s secret code.

We should note that often, and partially also in this chapter, a black-pegs only variant is studied, in which Codemaker reveals $\text{eq}(z, x)$ but not $\pi(z, x)$. This is justified both by several applications (cf. Section 3.2.2) and by the insight that, in particular for small numbers of colors, the white answer-pegs do not significantly improve Codebreaker’s situation (see Section 7.3).
7.1.2 Previous Results

Mastermind has been studied intensively in the mathematics and computer science literature. For the original 4-position 6-color version, Knuth [Knu77] has given a deterministic strategy that wins the game in at most five guesses. He also showed that no deterministic strategy has a 4-round guarantee.

The generalized $n$-position $k$-color version was investigated by Chvátal [Chv83]. He noted that a simple information-theoretic argument (attributed to Pierre Duchet) provides a lower bound of $\Omega(n \log k/ \log n)$ for any $k = k(n)$.

Extending the result [ER63b] of Erdős and Rényi from $k = 2$ to larger numbers of colors, he then showed that for any fixed $\varepsilon > 0$, $n$ sufficiently large and $k \leq n^{1-\varepsilon}$, repeatedly asking random guesses until all but the secret code are excluded by the answers is an optimal Codebreaker strategy (up to constant factors). More specifically, using the probabilistic method and random guesses, he showed the existence of a deterministic non-adaptive strategy for Codebreaker, that is, a set of $(2 + \varepsilon)n^{1+2\log k/\log(n/k)}$ guesses such that the answers uniquely determine any secret code Codemaker might have chosen (here and in the remainder, $\log n$ denotes the binary logarithm of $n$). These bounds hold even in the black-pegs only version of the game.

For larger values of $k$, the situation is less understood. Note that the information-theoretic lower bound is $\Omega(n)$ for any number $k = n^\alpha$, $\alpha > 0$ a constant, of colors. For $k$ between $n$ and $n^2$, Chvátal presented a deterministic adaptive strategy using $2n \log k + 4n$ guesses. For $k = n$, this strategy does not need white answer-pegs. Chvátal’s result has been improved subsequently. Chen, Cunha, and Homer [CCH96] showed that for any $k \geq n$, $2n \lceil \log n \rceil + 2n + \lceil k/n \rceil + 2$ guesses suffice. Goodrich [Goo99b] proved an upper bound of $n \lceil \log k \rceil + \lceil (2-1/k)n \rceil + k$ for the number of guesses needed to win the Mastermind game with an arbitrary number $k$ of colors and black answer-pegs only. This was again improved by Jäger and Peczarski [JP11], who showed an upper bound of $n \lceil \log n \rceil - n + k + 1$ for the case $k > n$ and $n \lceil \log k \rceil + k$ for the case $k \leq n$. Note that for the case of $k = n$ colors and positions, all these results give the same asymptotic bound of $\mathcal{O}(n \log n)$. 
7.1.3 Our Contribution

The results above show that Mastermind is well understood for $k \leq n^{1-\varepsilon}$, where we know the correct number of queries apart from constant factors. In addition, a simple non-adaptive guessing strategy suffices to find the secret code, namely casting random guesses until the code is determined by the answers.

On the other hand, for $k = n$ and larger, the situation is less clear. The best known upper bound, which is $\mathcal{O}(n)$ (and tight) for $k = n^\alpha$, $0 < \alpha < 1$ a constant, suddenly increases to $\mathcal{O}(n \log n)$ for $k = n$, while the information-theoretic lower bound remains at $\Omega(n)$.

In this chapter, we prove that indeed there is a change of behavior around $k = n$. We show that, for $k = \Theta(n)$, the random guessing strategy, and, in fact, any other non-adaptive strategy, cannot find the secret code with an expected number of less than $\Theta(n \log n)$ guesses. This can be proven via an entropy compression argument as used by Moser [Mos09], cf. Theorem 7.1. For general $k$, our new lower bound for non-adaptive strategies is $\Omega(n \log(k) / \max\{\log(n/k), 1\})$. We also show that this lower bound is tight (up to constant factors). In fact, for $k \leq n$, $\mathcal{O}(n \log(k) / \max\{\log(n/k), 1\})$ random guesses suffice to determine the secret code. That is, we extend Chvátal’s result from $k \leq n^{1-\varepsilon}$, $\varepsilon > 0$ a constant, to all $k \leq n$.

The main contribution of this chapter is a (necessarily adaptive) strategy that for $k = n$ finds the secret code with only $\mathcal{O}(n \log \log n)$ queries. This reduces the $\Theta(\log n)$ gap between the previous-best upper and the lower bound to $\Theta(\log \log n)$. Like the previous strategies for $k \leq n$, our new one does not use white answer-pegs. Our strategy also improves the current best bounds for other values of $k$ in the vicinity of $n$; see Theorem 7.1 below for the precise result.

The central part of our guessing strategy is setting up suitable coin-weighing problems, solving them, and using the solution to rule out the possible occurrence of some colors at some positions. By a result of Grebinski and Kucherov [GK00], these coin weighing problems can be solved by relatively few independent random weighings.

While our strategy thus is guided by probabilistic considerations, it can be derandomized to obtain a deterministic $\mathcal{O}(n \log \log n)$ strategy for black-peg Mastermind with $k = n$ colors. Moreover, appealing to an algorithmic result of Bshouty [Bsh09] instead of Grebinski and Kucherov’s
result, we obtain a strategy that can be realized as a deterministic polynomial-time codebreaking algorithm.

We also improve the current-best bounds for Mastermind with black and white answer-pegs, which stand at $\mathcal{O}(n \log n)$ for $n \leq k \leq n^2 \log \log n$. For these $k$, we prove that $\mathcal{O}(n \log \log n)$ guesses suffice. We point out that this improvement is not an immediate consequence of our $\mathcal{O}(n \log \log n)$ bound for $k = n$ black-peg Mastermind. Reducing the number of colors from $k$ to $n$ is a non-trivial sub-problem as well. For example, when $k \geq n^{1+\varepsilon}$, Chvátal’s strategy for the game with black and white answer-pegs also uses $\Theta(n \log n)$ guesses to reduce the number of colors from $k$ to $n$, before employing a black-peg strategy to finally determine the secret code.

### 7.1.4 Organization of this Chapter

We describe and analyze our $\mathcal{O}(n \log \log n)$ strategy for $k = n$ colors in Section 7.2. In Section 7.3 we present a strong connection between the black-pegs only and the classic (black and white pegs) version of Mastermind. This yields, in particular, the claimed bound of $\mathcal{O}(n \log \log n)$ for the classic version with $n \leq k \leq n^2 \log \log n$ colors. In Section 7.4 we analyze non-adaptive strategies. We prove a lower bound via entropy compression and show that it is tight for $k \leq n$ by extending Chavatal’s analysis of random guessing to all $k \leq n$.

### 7.2 The $\mathcal{O}(n \log \log n)$ Adaptive Strategy

In this section we present the main contribution of this chapter, a black-pegs only strategy that solves Mastermind with $k = n$ colors in $\mathcal{O}(n \log \log n)$ queries. We state our results for an arbitrary number $k = k(n)$ of colors; they improve upon the previously known bounds for all $k = o(n \log n)$ with $k \geq n^{1-\varepsilon}$ for every fixed $\varepsilon > 0$.

**Theorem 7.1.** For Mastermind with $n$ positions and $k = k(n)$ colors, the following holds.

- If $k = \Omega(n)$ then there exists a randomized winning strategy that uses black pegs only and needs an expected number of $\mathcal{O}(n \log \log n + k)$ guesses.
7.2. The $O(n \log \log n)$ Adaptive Strategy

- If $k = o(n)$ then there exists a randomized winning strategy that uses black pegs only and needs an expected number of $O\left(n \log \left(\frac{\log n}{\log(n/k)}\right)\right)$ guesses.

The $O$-notation in Theorem 7.1 only hides absolute constants. Note that, setting $k = n^{1-\delta}$, $\delta = \delta(n)$, the bound for $k = o(n)$ translates to $O(n \log (\delta^{-1}))$.

We describe our strategy and prove Theorem 7.1 in Sections 7.2.1, 7.2.2, 7.2.3. We discuss the derandomization of our strategy in Section 7.2.4.

7.2.1 Main Ideas

Our goal in this section is to give an informal sketch of our main ideas, and to outline how the $O(n \log \log n)$ bound for $k = n$ arises. For the sake of clarity, we nevertheless present our ideas in the general setting—it will be useful to distinguish between $k$ and $n$ notationally. As justified in Section 7.2.2 below, we assume that $k \leq n$ and that both $k$ and $n$ are powers of two.

A simple but crucial observation is that when we query a string $x \in [k]^n$ and the answer $\text{eq}(z, x)$ is 0 (recall that $z$ denotes Codemaker’s secret color code), then we know that all queried colors are wrong for their respective positions; i.e., we have $z_i \neq x_i$ for all $i \in [n]$. To make use of this observation, we maintain, for each position $i$, a set $C_i \subseteq [k]$ of colors that we still consider possible at position $i$. Throughout our strategy we reduce these sets successively, and once $|C_i| = 1$ for all $i \in [n]$ we have identified the secret code $z$. Variants of this idea have been used by several previous authors [Chv83, Goo09b].

Our strategy proceeds in phases. In each phase we reduce the size of all sets $C_i$ by a factor of two. Thus, before the $j$th phase we will have $|C_i| \leq k/2^{j-1}$ for all $i \in [n]$. Consider now the beginning of the $j$th phase, and assume that all sets $C_i$ have size exactly $k' := k/2^{j-1}$. Imagine we query a random string $r$ sampled uniformly from $C_1 \times \cdots \times C_n$. The expected value of $\text{eq}(z, r)$ is $n/k'$, and the probability that $\text{eq}(z, r) = 0$ is $(1-1/k')^n \leq e^{-n/k'}$. If $k'$ is significantly smaller than $n$, this probability is very small, and we will not see enough 0-answers to exploit the simple observation we made above. However, if we group the $n$ positions into $m := 4n/k'$ blocks of equal size $k'/4$, the expected contribution of each such block is $1/4$, and the probability that a fixed such block contributes
0 to \( \text{eq}(z,r) \) is \((1 - 1/k')^{k'/4} \approx e^{-1/4} \), i.e., constant. We will refer to blocks that contribute 0 to \( \text{eq}(z,r) \) as 0-blocks in the following. For a random query we expect a constant fraction of all \( m \) blocks to be 0-blocks. If we can identify which blocks these are, we can rule out a color at each position of each such block and make progress towards our goal.

As it turns out, the identification of the 0-blocks can be reduced to a coin-weighing problem that has been studied by several authors; see [GK00, Bsh09] and references therein. Specifically, we are given \( m \) coins of unknown integer weights and a spring scale. We can use the spring scale to determine the total weight of an arbitrary subset of coins in one weighing. Our goal is to identify the weight of every coin with as few weighings as possible.

In our setup, the ‘coins’ are the blocks we introduced above, and the ‘weight’ of each block is its contribution to \( \text{eq}(z,r) \). To simulate weighings of subsets of coins by Mastermind queries, we use ‘dummy colors’ for some positions, i.e., colors that we already know to be wrong at these positions. Using these, we can simulate the weighing of a subset of coins (=blocks) by copying the entries of the random query \( r \) in blocks that correspond to coins we wish to include in our subset, and by using dummy colors for the entries of all other blocks.

Note that the total weight of our ‘coins’ is \( \text{eq}(z,r) \). Typically this value will be close to its expectation \( n/k' \), and therefore of the same order of magnitude as the number of blocks \( m \). It follows from a coin-weighing result by Grebinski and Kucherov [GK00] that \( \mathcal{O}(m/\log m) \) random queries (of the described block form, simulating the weighing of a random subset of coins) suffice to determine the contribution of each block to \( \text{eq}(z,r) \) with some positive probability. As observed before, typically a constant fraction of all blocks contribute 0 to \( \text{eq}(z,r) \), and therefore we may exclude a color at a constant fraction of all \( n \) positions at this point.

Repeating this procedure of querying a random string \( r \) and using additional ‘random coin-weighing queries’ to identify the 0-blocks eventually reduces the sizes of the sets \( C_i \) below \( k'/2 \), at which point the phase ends. In total this requires \( \Theta(k') \) rounds in which everything works out as sketched, corresponding to a total number of \( \Theta(k' \cdot (m/\log m)) = \Theta(n/\log(4n/k')) \) queries for the entire phase.
7.2. The \( \mathcal{O}(n \log \log n) \) Adaptive Strategy

Summing over all phases, this suggests that for \( k = n \) a total number of

\[
\sum_{j=1}^{\log k} \mathcal{O} \left( \frac{n}{\log(4n/ (\frac{k}{2^j} - 1))} \right)^{k=n} \mathcal{O}(n) \sum_{j=1}^{\log n} \frac{1}{j+1} = \mathcal{O}(n \log \log n)
\]

queries suffice to determine the secret code \( z \), as claimed in Theorem 7.1 for \( k = n \).

We remark that our precise strategy, Algorithm 1, slightly deviates from this description. This is due to a technical issue with our argument once the number \( k' \) of remaining colors drops below \( C \log n \) for some \( C > 0 \). Specifically, beyond this point the error bound we derive for a fixed position is not strong enough to beat a union bound over all \( n \) positions. To avoid this issue, we stop our color reduction scheme before \( k' \) becomes that small (for simplicity as soon as \( k' \) is less than \( \sqrt{n} \)), and solve the remaining Mastermind problem by asking random queries from the remaining set \( C_1 \times \cdots \times C_n \), as originally proposed by Erdős and Rényi [ER63b] and Chvátal [Chv83].

7.2.2 Precise Description of Codebreaker’s Strategy

Assumptions on \( n \) and \( k \), Dummy Colors

Let us now give a precise description of our strategy. We begin by determining a dummy color for each position, i.e., a color that we know to be wrong at that particular position. For this we simply query the \( n + 1 \) many strings \((1,1,\ldots,1), (2,1,1,\ldots,1), \ldots, (2,2,\ldots,2) \in [k]^n\). Processing the answers to these queries in order, it is not hard to determine the location of all 1’s and 2’s in Codemaker’s secret string \( z \). In particular, this provides us with a dummy color for each position.

Next we argue that for the main part of our argument we may assume that \( n \) and \( k \) are powers of two. To see this for \( n \), note that we can simply extend Codemaker’s secret string in an arbitrary way such that its length is the smallest power of two larger than \( n \), and pretend we are trying to determine this extended string. To get the answers to our queries in this extended setting, we just need to add the contribution of the self-made extension part (which we determine ourselves) to the answers Codemaker provides for the original string. As the extension changes \( n \) at most by a factor of two, our claimed asymptotic bounds are unaffected by this.
To argue that we may also assume \( k \) to be a power of two, we make use of the dummy colors we already determined for the original value of \( k \). Similar to the previous argument, we increase \( k \) to the next power of two and consider the game with this larger number of colors. To get the answers to our queries in this extended setting from Codemaker (who still is in the original setting), it suffices to replace every occurrence of a color that is not in the original color set with the dummy color at the respective position.

We may and will also assume that \( k \leq n \). If \( k > n \) we can trivially reduce the number of colors to \( n \) by making \( k \) monochromatic queries. With this observation the first part of Theorem 7.1 follows immediately from the \( O(n \log \log n) \) bound we prove for the case \( k = n \).

### Eliminating Colors with Coin-Weighing Queries

With these technicalities out of the way, we can focus on the main part of our strategy. As sketched above, our strategy operates in phases, where in the \( j \)th phase we reduce the sizes of the sets \( C_i \) from \( k/2^{j-1} \) to \( k/2^j \). For technical reasons, we do not allow the sizes of \( C_i \) to drop below \( k/2^j \) during phase \( j \); i.e., once we have \( |C_i| = k/2^j \) for some position \( i \in [n] \), we no longer remove colors from \( C_i \) at that position and ignore any information that would allow us to do so.

Each phase is divided into a large number of rounds, where a round consists of querying a random string \( r \) and subsequently identifying the 0-blocks (blocks that contribute 0 to \( \text{eq}(z,r) \)) by the coin-weighing argument outlined above.

To simplify the analysis, the random string \( r \) is sampled from the same distribution throughout the entire phase. Specifically, at the beginning of phase \( j \) we define the set \( \mathcal{R}_j := C_1 \times \cdots \times C_n \), and sample the random string \( r \) uniformly at random from \( \mathcal{R}_j \) in each round of phase \( j \). Note that we do not adjust \( \mathcal{R}_j \) during phase \( j \); information about excluded colors we gain during phase \( j \) will only be used in the definition of the set \( \mathcal{R}_{j+1} \) in phase \( j + 1 \).

We now introduce the formal setup for the coin-weighing argument. As before we let \( k' := k/2^{j-1} \) and partition the \( n \) positions into \( m := 4n/k' \) blocks of size \( k'/4 \). More formally, for every \( s \in [m] \) we let \( B_s := \{(s-1)k'/4 + 1, \ldots, sk'/4 \} \) denote the indices of block \( s \), and we denote by \( v_s := |\{i \in B_s : z_i = r_i\}| \) the contribution of block \( B_s \) to
7.2. The $\mathcal{O}(n \log \log n)$ Adaptive Strategy

eq($z, r$). (Note that $\sum_{s \in [m]} v_s = eq(z, r)$.) As indicated above we wish to identify the 0-blocks, that is, the indices $s \in [m]$ for which $v_s = 0$.

For $y \in \{0, 1\}^m$, define $r_y$ as the query that is identical to $r$ on the blocks $B_s$ for which $y_s = 1$, and identical to the string of dummy colors on all other blocks. Thus $eq(z, r_y) = \sum_{s \in [m], y_s = 1} v_s$. With this observation, identifying the values $v_s$ from a set of queries of form $r_y$ is equivalent to a coin-weighing problem in which we have $m$ coins with positive integer weights that sum up to $eq(z, r)$: Querying $r_y$ in the Mastermind game provides exactly the information we obtain from weighing the set of coins indicated by $y$.

We will only bother with the coin-weighing if the initial random query of the round satisfies $eq(z, r) \leq m/2$. (Recall that the expected value of $eq(z, r)$ is $m/4$.) If this is the case, we query an appropriate number $f(m)$ of strings of form $r_y$, with $y \in \{0, 1\}^m$ sampled uniformly at random (u.a.r.) and independently. The function $f(m)$ is implicit in the proof of the coin-weighing result of [GK00]; it is in $\Theta(m / \log m)$ and guarantees that the coin-weighing succeeds with probability at least $1/2$. Thus with probability at least $1/2$, these queries determine all values $v_s$ and, in particular, identify all 0-blocks. Note that the inequality $eq(z, r) \leq m/2$ also guarantees that at least half of the $m$ blocks are 0-blocks.

We say that a round is successful if $eq(z, r) \leq m/2$ and if the coin-weighing successfully identifies all 0-blocks. In each successful round, we update the sets $C_i$ as outlined above; i.e., for each position $i$ that is in a 0-block and for which $|C_i| > k'/2$ we set $C_i := C_i \setminus \{r_i\}$. Note that it might happen that $r_i$ is a color that was already removed from $C_i$ in an earlier round of the current phase, in which case $C_i$ remains unchanged. If a round is unsuccessful we do nothing and continue with the next round.

This completes the description of our strategy for a given phase. We abandon this color reduction scheme once $k'$ is less than $\sqrt{n}$. At this point, we simply ask queries sampled uniformly and independently at random from the current set $\mathcal{R} = C_1 \times \cdots \times C_n$. We do so until the answers uniquely determine the secret code $z$. It follows from Chvátal’s result [Chv83] that the expected number of queries needed for this is $O(n \log k' / \log(n/k')) = O(n)$.

This concludes the description of our strategy. It is summarized in Algorithm [1]. Correctness is immediate from our discussion, and it remains
to bound the expected number of queries the strategy makes.

**Algorithm 1:** Playing Mastermind with many colors

1. Determine a dummy color for each position;
2. foreach \( i \in [n] \) do \( C_i \leftarrow [k] \);
3. \( j \leftarrow 0 \) and \( k' \leftarrow k \);
4. while \( k' > \sqrt{n} \) do
   5. \( j \leftarrow j + 1, k' \leftarrow k/2^{j-1}, R_j \leftarrow C_1 \times \cdots \times C_n, \) and \( m \leftarrow 4n/k' \);
   repeat
   6. Select a string \( r \) u.a.r. from \( R_j \) and query \( eq(z, r) \);
   7. if \( eq(z, r) \leq m/2 \) then
      8. for \( i = 1, \ldots, f(m) \) /* \( f(m) = \Theta(m/\log m) \) */ do
         9. Sample \( y \) u.a.r. from \( \{0, 1\}^m \) and query \( eq(z, r_y) \);
      10. if these \( f(m) \) queries determine the 0-blocks of \( r \) then
         11. foreach \( i \in [n] \) do
             12. if \( i \) is in a 0-block and \(|C_i| > k'/2\) then
                 13. \( C_i \leftarrow C_i \setminus \{r_i\} \);
         until \( \forall i \in [n] : |C_i| = k'/2 \);
   14. \( R \leftarrow C_1 \times \cdots \times C_n \);
15. Select strings \( r \) independently and u.a.r. from \( R \) and query \( eq(z, r) \)
16. until \( z \) is determined;

### 7.2.3 Proof of Theorem 7.1

We begin by bounding the expected number of rounds in the \( j \)th phase.

**Claim 7.2.** The expected number of rounds required to complete phase \( j \) is \( \mathcal{O}(k^j) = \mathcal{O}(k/2^j) \).

**Proof.** We first show that a round is successful with probability at least \( 1/4 \). Recall that \( eq(z, r) \) has an expected value of \( n/k' = m/4 \). Thus, by Markov’s inequality, we have \( eq(z, r) \leq m/2 \) with probability at least \( 1/2 \). Moreover, as already mentioned, the proof of the coin-weighing result by Grebinski and Kucherov [GK00] implies that our \( f(m) = \Theta(m/\log m) \) random coin-weighing queries identify all 0-blocks with probability at least \( 1/2 \). Thus, in total the probability for a successful round is at least \( 1/2 \cdot 1/2 = 1/4 \).
We continue by showing that the probability that a successful round decreases the number of available colors for a fixed position, say position 1, is at least $1/4$. Note that this happens if $r \in \mathcal{R}_j$ satisfies the following two conditions:

(i) $v_1 = 0$, i.e., block $B_1$ is a 0-block with respect to $r$, and

(ii) $r_1 \in C_1$, i.e., the color $r_1$ has not been excluded from $C_1$ in a previous round of phase $j$.

For (i) recall that in a successful round at least $m/2$ of the $m$ blocks are 0-blocks. It follows by symmetry that $B_1$ is a 0-block with probability at least $1/2$. Moreover, conditional on (i), $r_1$ is sampled uniformly at random from the $k' - 1$ colors that are different from $z_1$ and were in $C_1$ at the beginning of the round. Thus the probability that $r_1$ is in the current set $C_1$ is $|C_1|/(k' - 1)$, which is at least $1/2$ because we do not allow $|C_1|$ to drop below $k'/2$. We conclude that, conditional on a successful round, the random query $r$ decreases $|C_1|$ by one with probability at least $1/2 \cdot 1/2 = 1/4$.

Thus, in total, the probability that a round decreases $|C_1|$ by one is at least $1/4 \cdot 1/4 = 1/16$ throughout our strategy. It follows that the probability that after $t$ successful rounds in phase $j$ we still have $|C_1| > k'/2$ is bounded by the probability that in $t$ independent Bernoulli trials with success probability $1/16$ we observe fewer than $k'/2$ successes. If $t/16 \geq k'$, by Chernoff bounds this probability is bounded by $e^{-ct}$ for some absolute constant $c > 0$.

Let us now denote the number of rounds phase $j$ takes by the random variable $T$. By a union bound, the probability that $T \geq t$, i.e., that after $t$ steps at one of the positions $i \in [n]$ we still have $|C_i| > k'/2$, is bounded by $ne^{-ct}$ for $t \geq 16k'$. It follows that

$$\mathbb{E}[T] = \sum_{t \geq 1} \Pr[T \geq t] \leq 16k' + n \sum_{t > 16k'} e^{-ct} = 16k' + ne^{-\Omega(k')} = \mathcal{O}(k'),$$

where the last step is due to $k' \geq \sqrt{n} = \omega(\log n)$.

With Claim 7.2 in hand, we can bound the total number of queries required throughout our strategy by a straightforward calculation.

Proof (of Theorem 7.1). Recall that for each phase $j$ we have $m = \Theta(n/k') = \Theta(n/(k/2^{j-1}))$ and that $f(m) = \Theta(m/\log m)$. Thus by
Claim 7.2, the expected number of queries our strategy asks in phase $j$ is bounded by

$$O(k') \cdot (1 + f(m)) = O\left(\frac{n}{\log(n/k^{2j})}\right) = O\left(\frac{n}{\log(n/k) + j}\right).$$

It follows that throughout the main part of our strategy we ask an expected number of queries of at most

$$O(n \log(n/k^{2j}) + j) = O\left(n \log\left(\frac{\log n}{\log(n/k)}\right)\right).$$

(This calculation is for $k < n$; as observed before, for $k = n$ a very similar calculation yields a bound of $O(n \log \log n)$.) As the number of queries for determining the dummy colors and for wrapping up at the end is only $O(n)$, Theorem 7.1 follows.

### 7.2.4 Derandomization

The strategy we presented in the previous section can be derandomized and implemented as a polynomial-time algorithm.

**Theorem 7.3.** The bounds stated in Theorem 7.1 can be achieved by a deterministic winning strategy. Furthermore, this winning strategy can be realized in polynomial time.

**Proof.** The main loop of the algorithm described above uses randomization in two places: for generating the random string $r$ of each round (line 7 in Algorithm 1), and for generating the $f(m)$ many random coin-weighing queries $r_y$ used to identify the 0-blocks of $r$ if $eq(z, r) \leq m/2$ (line 10).

The derandomization of the coin-weighing algorithm is already given in the work of Grebinski and Kucherov [GK00]. They showed that a set of $f'(m) = \Theta(m/\log m)$ random coin-weighing queries $y^1, \ldots, y^{f'(m)}$, sampled from $\{0,1\}^m$ independently and uniformly at random, has, with some positive probability, the property that it distinguishes any two distinct coin-weighing instances in the following sense: For any
two distinct vectors \(v, w\) with non-negative integer entries such that 
\[
\sum_{s \in [m]} v_s \leq m/2 \quad \text{and} \quad \sum_{s \in [m]} w_s \leq m/2,
\]
there exists an index \(j \in [f'(m)]\) for which 
\[
\sum_{s \in [m], y_j^s = 1} v_s \neq \sum_{s \in [m], y_j^s = 1} w_s.
\]
It follows by the probabilistic method that, deterministically, there is a set \(D \subseteq \{0, 1\}^m\) of size at most \(f'(m)\) such that the answers to the corresponding coin-weighing queries identify every possible coin-weighing instance. Hence we can replace the \(f(m)\) random coin-weighing queries of each round by the \(f'(m)\) coin-weighing queries corresponding to the fixed set \(D\).

It remains to derandomize the choice of \(r\) in each round. As before we consider \(m := 4n/k'\) blocks of size \(k'/4\), where \(k'\) is the size of the sets \(C_i\) at the beginning of a phase. To make sure that a constant fraction of all queries in a phase satisfy \(eq(z, r) \leq m/2\) (compare line 8 of Algorithm 1), we ask a set of \(k'\) queries such that, for each position \(i \in [n]\), every color in \(C_i\) is used at position \(i\) in exactly one of these queries. (If all sets \(C_i\) are equal, this can be achieved by simply asking \(k'\) monochromatic queries.) The sum of all returned scores must be exactly \(n\), and therefore we cannot get a score of more than \(m/2 = 2n/k'\) for more than \(k'/2\) queries. In this way we ensure that for at least \(k - k'/2 = k'/2\) queries we get a score of at most \(m/2\).

As in the randomized version of our strategy, in each of these \(k'/2\) queries at least half of the blocks must be 0-blocks. We can identify those by the derandomized coin-weighing discussed above. Consider now a fixed block. As it has size \(k'/4\), it can be a non-0-block in at most \(k'/4\) queries. Thus it is a 0-block in at least \(k'/2 - k'/4 = k'/4\) of the queries.

To summarize, we have shown that by asking \(k'\) queries of the above form we get at least \(k'/2\) queries of score at most \(m/2\). For each of them we identify the 0-blocks by coin-weighing queries. This allows us to exclude at least \(k'/4\) colors at each position. I.e., as in the randomized version of our strategy we can reduce the number of colors by a constant factor using only \(O(k' \cdot m / \log m) = O(n / \log(4n/k))\) queries. By similar calculations as before, the same asymptotic bounds follow.

We abandon the color reduction scheme when \(k'\) is a constant. At this point, we can solve the remaining problem in time \(O(n)\) by repeatedly using the argument we used to determine the dummy colors in Section 7.2.2.

Note that all of the above can easily be implemented in polynomial time if we can solve the coin-weighing subproblems in polynomial time. An
algorithm for doing the latter is given in the work of Bshouty [Bsh09]. Using this algorithm as a building block, we obtain a deterministic polynomial-time strategy for Codebreaker that achieves the bounds stated in Theorem 7.1.

7.3 Mastermind with Black and White Answer-Pegs

In this section, we analyze the Mastermind game in the classic version with both black and white answer-peggs. Interestingly, there is a strong general connection between the two versions. Roughly speaking, we can use a strategy for the $k = n$ black-peg game to learn which colors actually occur in the secret code of a black/white-peg game with $n$ positions and $n^2$ colors. Having thus reduced the number of relevant colors to at most $n$, Codebreaker can again use a $k = n$ black-peg strategy (ignoring the white answer-peggs) to finally determine the secret code.

More precisely, for all $k, n \in \mathbb{N}$ let us denote by $b(n, k)$ the minimum (taken over all strategies) maximum (taken over all secret codes) expected number of queries needed to find the secret code in a black-peg Mastermind game with $k$ colors and $n$ positions. Similarly, denote by $bw(n, k)$ the corresponding number for the game with black and white answer-peggs. Then we show the following.

**Theorem 7.4.** For all $k, n \in \mathbb{N}$ with $k \geq n$,

$$bw(n, k) = \Theta\left(\frac{k}{n} + b(n, n)\right).$$

Combining this with Theorem 7.1 we obtain a bound of $\mathcal{O}(n \log \log n)$ for black/white Mastermind with $n \leq k \leq n^2 \log \log n$ colors, improving all previous bounds in that range.

For the case $k \leq n$ it is not hard to see that $bw(n, k) = \Theta(b(n, k))$, see Corollary 7.6 below. Together with Theorem 7.4 this shows that to understand black/white-peg Mastermind for all $n$ and $k$, it suffices to understand black-peg Mastermind for all $n$ and $k$.

Before proving Theorem 7.4 let us derive a few simple preliminary results on the relation of the two versions of the game.
Lemma 7.5. For all \( n, k \),
\[
bw(n, k) \geq b(n, k) - k + 1.
\]

Proof. We show that we can simulate a strategy in the black/white Mastermind game by one receiving only black-pegs answers and using \( k - 1 \) more guesses. Fix a strategy for black/white Mastermind. Our black-peg strategy first asks \( k - 1 \) monochromatic queries. This tells us how often each of the \( k \) color arises in the secret code. From now on, we can play the strategy for the black/white game. While we only receive black answer-pegs, we can compute the number of white pegs we would have gotten in the black/white game from the just obtained information on how often each color occurs in the code. With this information available, we can indeed play as in the given strategy for black/white Mastermind.

Lemma 7.5 will be used to prove that the \( b(n, n) \) term in the statement of Theorem 7.4 cannot be avoided. As a corollary, it yields that white answer-pegs are not extremely helpful when \( k = \mathcal{O}(n) \).

Corollary 7.6. For all \( k \leq n \),
\[
bw(n, k) = \Theta(b(n, k)).
\]

Proof. Obviously, \( bw(n, k) \leq b(n, k) \) for all \( n, k \). If \( k = o(n) \), then the information theoretic lower bound \( b(n, k) = \Omega(n \log k / \log n) \) is of larger order than \( k \), hence the lemma above shows the claim. For \( k = \Theta(n) \), note first that both \( b(n, k) \) and \( bw(n, k) \) are in \( \Omega(n) \) due to the information theoretic argument. If \( b(n, k) = \mathcal{O}(n) \), there is nothing to show. If \( b(n, k) = \omega(n) \), we again invoke Lemma 7.5.

In the remainder of this section, we prove Theorem 7.4. To describe the upper bound, let us fix the following notation. Let \( C \) be the set of all available colors and \( k = |C| \). Denote by \( z \in C^n \) the secret code chosen by Codemaker. Denote by \( C^* := \{ z_i \mid i \in [n] \} \) the (unknown) set of colors in \( z \).

Codebreaker’s strategy leading to the bound of Theorem 7.4 consists of roughly these three steps.

1. Codebreaker first asks roughly \( k/n \) guesses containing all colors. Only colors in a guess receiving a positive answer can be part of the
secret code, so this reduces the number of colors to be regarded to at most $n^2$. Also, Codebreaker can learn from the answers the cardinality $n'$ of $C^*$, that is, the number of distinct colors in the secret code.

(2) By asking an expected number of $\Theta(n')$ (dependent) random queries, Codebreaker learns $n'$ disjoint sets of colors of size at most $n$ such that each color of $C^*$ is contained in exactly one of these sets. Denote by $k'$ the cardinality of a largest of these sets.

(3) Given such a family of sets, Codebreaker can learn $C^*$ with an expected number of $b(n', k')$ queries by simulating an optimal black-peg Mastermind strategy. Once $C^*$ is known, an expected number of $b(n, n')$ queries determine the secret code, using an optimal black-peg strategy for $n'$ colors.

Each of these steps is made precise in the following. Before doing so, we remark that after a single query Codebreaker may detect $|C^* \cap X|$ for any set $X$ of at most $n$ colors via a single Mastermind query to be answered by black and white pegs.

**Lemma 7.7.** For an arbitrary set $X$ of at most $n$ colors, let $\text{col}(X) := |C^* \cap X|$, the number of colors of $X$ occurring in the secret code. After a single initial query, Codebreaker can learn $\text{col}(X)$ for any $X$ via a single Mastermind query to be answered by black and white pegs.

*Proof.* As the single initial query, Codebreaker may ask $(1, \ldots, 1)$, the code consisting of color 1 only. Denote by $b$ the number of black pegs received (there cannot be a white answer-peg). This is the number of occurrences of color 1 in the secret code.

Let $X \subseteq C$, $\nu := |X| \leq n$. To learn $\text{col}(X)$, Codebreaker extends $X$ to a multiset of $n$ colors by adding the color 1 exactly $n - \nu$ times and guesses a code arbitrarily composed of this multiset of colors. Let $y$ be the total number of (black and white) answer-pegs received. Then $\text{col}(X) = y - \min\{n - \nu, b\}$, if $1 \notin X$ or $b = 0$, and $\text{col}(X) = y - \min\{n - \nu, b - 1\}$ otherwise. \hfill \Box

To ease the language, we shall call a query determining $\text{col}(X)$ a *color query*. We now show that using roughly $k/n$ color queries, Codebreaker can learn the number $|C^*|$ of different colors occurring in the secret code and exclude all but $n|C^*|$ colors.

**Lemma 7.8.** With $\lceil k/n \rceil$ color queries, Codebreaker can learn both $|C^*|$ and a superset $C_0$ of $C^*$ consisting of at most $n|C^*|$ colors.
7.3. Mastermind with Black and White Answer-Pegs

Proof. Let \( X_1, \ldots, X_{[k/n]} \) be a partition of \( C \) into sets of cardinality at most \( n \). By asking the corresponding \( [k/n] \) color queries, Codebreaker immediately learns \( |C^*| := \sum_{i=1}^{[k/n]} \text{col}(X_i) \). Also, \( C_0 := \bigcup \{ X_i \mid \text{col}(X_i) > 0 \} \) is the desired superset.

\[ \]

**Lemma 7.9.** Assume that Codebreaker knows the number \( n' = |C^*| \) of different colors in \( z \) as well as a set \( C_0 \supseteq C^* \) of colors such that \( |C_0| \leq n|C^*| \).

Then with an expected number of \( \Theta(n') \) color queries, Codebreaker can find a family \( C_1, \ldots, C_{n'} \) of disjoint subsets of \( C_0 \), each of size at most \( \lceil |C_0|/n' \rceil \leq n \), such that \( C^* \subseteq C_1 \cup \ldots \cup C_{n'} \) and \( |C^* \cap C_i| = 1 \) for all \( i \in [n'] \).

Proof. Roughly speaking, Codebreaker’s strategy is to ask color queries having an expected answer of one. With constant probability, such a query contains exactly one color from \( C^* \). Below is a precise formulation of this strategy.

**Algorithm 2:** Codebreaker’s strategy

1. \( \text{while } n' > 0 \text{ do} \)
2. \( \quad k' \leftarrow \lceil |C_0|/n' \rceil; \)
3. \( \quad \text{Let } C_{n'} \text{ be a random subset of } C \text{ with } |C_{n'}| = k'; \)
4. \( \quad \text{Ask the color query } C_{n'}; \)
5. \( \quad \text{if } \text{col}(C_{n'}) = 1 \text{ then} \)
6. \( \quad \quad C_0 \leftarrow C_0 \setminus C_{n'}; \)
7. \( \quad \quad n' \leftarrow n' - 1; \)

For the analysis, note first that the value of \( k' \) during the application of the above strategy does not increase. In particular, all sets \( C_i \) defined and queried have cardinality at most \( \lceil |C_0|/n' \rceil \leq n \). It is also clear that the above strategy constructs a sequence of disjoint \( C_i \) and that for each color occurring in \( z \) there is exactly one \( C_i \) containing this color.

It remains to prove the estimate on the expected number of queries. To this aim, we first note that throughout a run of this strategy, \( n' \) is the number of colors of \( C^* \) left in \( C_0 \). Hence the event “\( \text{col}(C_{n'}) = 1 \)” occurs
with probability
\[
\frac{n'k'(|C_0| - n') \ldots (|C_0| - n' - k' + 2)}{|C_0| \ldots (|C_0| - k' + 1)} \\
\geq \frac{(|C_0| - n') \ldots (|C_0| - n' - k' + 2)}{(|C_0| - 1) \ldots (|C_0| - k' + 1)} \\
\geq \left( \frac{|C_0| - n' - k' + 2}{|C_0| - k' + 1} \right)^{k'-1} \\
= \left( 1 - \frac{n' - 1}{|C_0| - k' + 1} \right)^{k'-1} \\
\geq \left( 1 - \frac{n' - 1}{|C_0| - (|C_0|/n')} \right)^{k'-1} \\
\geq \left( 1 - \frac{|C_0|/(k' - 1)}{|C_0| - (|C_0|/n')} \right)^{k'-1} \\
\geq \left( 1 - \frac{1}{(k' - 1)(1 - 1/n')} \right)^{k'-1},
\]
which is bounded from below by a constant (the later estimates assume \(n' \geq 2\); for \(n' = 1\) the second term of the sequence of inequalities already is one).

Consequently, with constant probability the randomly chosen \(C_{n'}\) satisfies “\(\text{col}(C_{n'}) = 1\)”. Hence after an expected constant number of iterations of the while-loop, such a \(C_{n'}\) will be found. Since each such success reduces the value of \(n'\) by one, a total expected number of \(\Theta(|C^*|)\) iterations suffices to find the desired family of sets \((C_i)_{i \in [n']}\).

Given a family of sets as just constructed, Codebreaker can simulate a black-peg strategy to determine \(C^*\).

**Lemma 7.10.** Let \(C_1, \ldots, C_{n'}\) be a family of disjoint subsets of \(C\) such that \(C^* \subseteq C_1 \cup \ldots \cup C_{n'}\) and \(|C^* \cap C_i| = 1\) for all \(i \in [n']\). Assume that \(k' := \max\{|C_i| \mid i \in [n']\} \leq n\). Then Codebreaker can detect \(C^*\) using an expected number of \(b(n', k')\) color queries.

**Proof.** Let \(z' \in C_1 \times \ldots \times C_{n'}\) be the unique such string consisting of colors in \(C^*\) only. Note that in black-peg Mastermind, the particular
sets of colors used at each position are irrelevant. Hence there is a strategy for Codebreaker to detect $z'$ using an expected number of $b(n',k')$ guesses from $C_1 \times \ldots \times C_{n'}$ and receiving black-peg answers only.

We now show that for each such query, there is a corresponding color query in the $(n,k)$ black/white Mastermind game giving the same answer. Hence we may simulate the black-peg game searching for $z'$ by such color queries. Since $z'$ contains all colors of $C^*$ and no other colors, once found, it reveals the set of colors occurring in the original secret code $z$.

Let $y' \in C_1 \times \ldots \times C_{n'}$ be a query in the black-peg Mastermind game searching for $z'$. For each position $i \in [n']$, we have $z'_i = y'_i$ if and only if $y'_i \in C_i$ is the unique color from $C_i$ that is in $C^*$. As moreover the sets $(C_i)_{i \in [n']}$ are disjoint, we have $\eq(z',y') = \col(\{y'_1, \ldots, y'_{n'}\})$, and we can obtain this value (i.e., the black-peg answer for the guess $y'$ relative to $z'$) by a color query relative to $z$.

Note that if our only goal is to find out $C^*$, then for $k \ll n^2$ we can be more efficient by asking more color queries in Lemma 7.8, leading to a smaller set $C_0$, to smaller sets $C_i$ in Lemma 7.9, and thus to a smaller $k'$ value in Lemma 7.10. Since this will not affect the asymptotic bound for the total numbers of queries used in the black/white-peg game, we omit the details.

Proof (of Theorem 7.4). The upper bound follows easily from applying Lemmas 7.7 to 7.10, which show that Codebreaker can detect the set $C^*$ of colors arising in the secret code $z$ with an expected number of $1 + \ceil{k/n} + \mathcal{O}(n) + b(n,n)$ guesses. Since $|C^*| \leq n$, he can now use a strategy for black-peg Mastermind and determine $z$ with another expected number of $b(n,n)$ guesses. Note that $b(n,n) = \Omega(n)$, so this proves the upper bound.

We argue that this upper bound is optimal apart from constant factors. Assume first that the secret code is a random monochromatic string (Codemaker may even announce this). Fix a (possibly randomized) strategy for Codebreaker. With probability at least 1/2, this strategy does not use the particular color in any of the first $k/(2n)$ guesses. It then also did not guess the correct code. Hence the expected number of queries necessary to find the code is at least $k/(4n)$.

We finally show that for $k \geq n$, also the $b(n,n)$ term cannot be avoided. By the information theoretic argument, there is nothing to
show if \( b(n, n) = \Theta(n) \). Hence assume \( b(n, n) = \omega(n) \). We will show \( bw(n, k) + n + 1 \geq bw(n, n) \). The claim then follows from \( bw(n, n) = \Theta(b(n, n)) \) (Corollary 7.6).

We show that we can solve the \( k = n \) color Mastermind game by asking \( n+1 \) preliminary queries and then simulating a strategy for black/white Mastermind with \( n \) positions and \( k \geq n \). As in Section 7.2.2, we use \( n+1 \) queries to learn for each position whether it has color 1 or not. We then simulate a given strategy for \( k > n \) colors as follows. In a \( k \)-color query, replace all colors greater than \( n \) by color 1. Since we know the positions of the pegs in color 1, we can reduce the answers by the contribution of these additional 1-pegs in the query. This gives the answer we would have gotten in reply to the original query (since the secret code does not contain colors higher than \( n \)). Consequently, we can now simulate the \( k \)-color strategy in an \( n \)-color Mastermind game.

\[ \square \]

### 7.4 Non-Adaptive Strategies

When analyzing the performance of non-adaptive strategies, it is not very meaningful to ask for the number of queries needed until the secret code is queried for the first time. Instead we ask for the number of queries needed to identify it.

In their work on the 2-color black-peg version of Mastermind, Erdős and Rényi [ER63b] showed that random guessing needs, with high probability, \((2 + o(1))n/\log n\) queries to identify the secret code, and that this is in fact best possible among non-adaptive winning strategies. The upper bound was derandomized by Lindström [Lin64, Lin65] and, independently, by Cantor and Mills [CM66]. That is, for 2-color black-peg Mastermind a deterministic non-adaptive winning strategy using \((2 + o(1))n/\log n\) guesses exists, and no non-adaptive strategy can do better.

For adaptive strategies, only a weaker lower bound of \((1 + o(1))n/\log n\) is known. This bound results from the information-theoretic argument mentioned in Section 7.1.2. It remains a major open problem whether there exists an adaptive strategy that achieves this bound. In fact, it is not even known whether adaptive strategies can outperform the random guessing strategy by \textit{any} constant factor.

Here in this section we prove that for Mastermind with \( k = \Theta(n) \) colors,
adaptive strategies are indeed more powerful than non-adaptive ones, and outperform them even in order of magnitude. More precisely, we show that any non-adaptive strategy needs $\Omega(n \log n)$ guesses. Since we know from Section 7.2 that adaptively we can achieve a bound of $O(n \log \log n)$, this separates the performance of non-adaptive strategies from that of adaptive ones. Our result answers a question left open in [God03].

The $\Omega(n \log n)$ bound for non-adaptive strategies is tight. As we will show in Theorem 7.13 below, there exists a deterministic non-adaptive strategy that achieves the bound up to constant factors.

### 7.4.1 Lower Bound for Non-Adaptive Strategies

For the formal statement of the bound, we use the following notation. A deterministic non-adaptive strategy is a fixed ordering $x^1, x^2, \ldots, x^{k^n}$ of all possible guesses, i.e., the elements of $[k]^n$. A randomized non-adaptive strategy is a probability distribution over such orderings. For a given secret code $z \in [k]^n$, we ask for the smallest index $j$ such that the queries $x^1, \ldots, x^j$ together with their answers $eq(z, x^1), \ldots, eq(z, x^j)$ uniquely determine $z$.

Mastermind with non-adaptive strategies is also referred to as static Mastermind [God03].

**Theorem 7.11.** For any (randomized or deterministic) non-adaptive strategy for black-peg Mastermind with $n$ positions and $k$ colors, the expected number of queries needed to determine a secret code $z$ sampled uniformly at random from $[k]^n$ is $\Omega\left(\frac{n \log k}{\max\{\log(n/k), 1\}}\right)$.

Theorem 7.11 shows, in particular, that for any non-adaptive strategy there exists a secret code $z \in [k]^n$ which can only be identified after $\Omega\left(n \log k/\max\{\log(n/k), 1\}\right)$ queries. For $k \geq n$, this is an improvement of $\Theta(\log n)$ over the information-theoretic lower bound mentioned in the introduction. For the case $k = \Theta(n)$ Theorem 7.11 gives a lower bound of $\Omega(n \log n)$ guesses for every non-adaptive strategy, showing that adaptive strategies are indeed more powerful than non-adaptive ones in this regime (recall Theorem 7.1).

To give an intuition for the correctness of Theorem 7.11 note that for a uniformly chosen secret code $z \in [k]^n$, for any single fixed guess $x$ of a non-adaptive strategy the answer $eq(z, x)$ is binomially distributed.
with parameters \( n \) and \( 1/k \). That is, \( \text{eq}(z, x) \) will typically be within the interval \( n/k \pm \mathcal{O}(\sqrt{n/k}) \). Hence, we can typically encode the answer using \( \log(\mathcal{O}(\sqrt{n/k})) = \mathcal{O}(\log(n/k)) \) bits. Or, stated differently, our ‘information gain’ is usually \( \mathcal{O}(\log(n/k)) \) bits. Since the secret code ‘holds \( n \log k \) bits of information’, we would expect that we have to make \( \Omega(n \log k/\log(n/k)) \) guesses.

To turn this intuition into a formal proof, we recall the notion of entropy: For a discrete random variable \( Z \) over a domain \( D \), the entropy of \( Z \) is defined by \( H(Z) := -\sum_{z \in D} \Pr[Z = z] \log(\Pr[Z = z]) \). Intuitively speaking, the entropy measures the amount of information that the random variable \( Z \) carries. If \( Z \) for example corresponds to a random coin toss with \( \Pr[\text{‘heads’}] = \Pr[\text{‘tails’}] = 1/2 \), then \( Z \) carries 1 bit of information. However, a biased coin toss with \( \Pr[\text{‘heads’}] = 2/3 \) carries less (roughly 0.918 bits of) information since we know that the outcome of heads is more likely. In our proof we use the following properties of the entropy, which can easily be seen to hold for any two random variables \( Z, Y \) over domains \( D_Z, D_Y \).

(E1) If \( Z \) is determined by the outcome of \( Y \), i.e., \( Z = f(Y) \) for a deterministic function \( f \), then we have \( H(Z) \leq H(Y) \).

(E2) We have \( H((Z, Y)) \leq H(Z) + H(Y) \).

The inequality in (E2) holds with equality if and only if the two variables \( Z \) and \( Y \) are independent.

Proof (of Theorem 7.11). Below we show that there is a time \( s = \Omega\left(\frac{n \log k}{\max\{\log(n/k), 1\}}\right) \) such that any deterministic strategy at any time earlier than \( s \) determines less than half of the secret codes. Consequently, any deterministic strategy needs an expected time of at least \( s/2 \) to determine a secret chosen uniformly at random. Since any randomized strategy is a convex combination of deterministic ones, this latter statement also holds for randomized strategies.

Let \( S = (x^1, x^2, \ldots) \) denote a deterministic strategy of Codebreaker. We first show a lower bound on the number of guesses that are needed to identify at least half of all possible secret codes. For \( j = 1, \ldots, k^n \), let \( A_j = A_j(S) \subseteq [k]^n \) denote the set of codes that can be uniquely determined from the answers to the queries \( x^1, \ldots, x^j \). Let \( s \) be the smallest index for which \( |A_s| \geq k^n/2 \).
Consider a code $Z \in [k]^n$ sampled uniformly at random, and set $Y_i := \text{eq}(Z, x^i)$, $1 \leq i \leq s$. Moreover, let

$$
\tilde{Z} = \begin{cases} 
Z & \text{if } Z \in A_s, \\
\text{‘fail’} & \text{if } Z \notin A_s.
\end{cases}
$$

By our definitions, the sequence $Y := (Y_1, Y_2, \ldots, Y_s)$ determines $\tilde{Z}$, and hence by (E1) we have

$$
H(\tilde{Z}) \leq H(Y). 
$$

Moreover, we have

$$
H(\tilde{Z}) = -\sum_{z \in A_s} \Pr[\tilde{Z} = z] \log(\Pr[\tilde{Z} = z]) - \Pr[\tilde{Z} = \text{‘fail’}] \log(\Pr[\tilde{Z} = \text{‘fail’}]) \\
\geq -\sum_{z \in A_s} \Pr[Z = z] \log(\Pr[Z = z]) \\
= \frac{|A_s|}{k^n} \log(k^n) \\
\geq \frac{1}{2} n \log k. 
$$

We now derive an upper bound on $H(Y)$. For every $i$, $Y_i$ is binomially distributed with parameters $n$ and $1/k$. Therefore, its entropy is (see, e.g., [JS99])

$$
H(Y_i) = \frac{1}{2} \log \left( 2\pi e \frac{n}{k} \left( 1 - \frac{1}{k} \right) \right) + \frac{1}{2} + O\left( \frac{1}{n} \right) = O(\max\{\log(n/k), 1\}).
$$

We thus obtain

$$
H(Y) \overset{(E2)}{\leq} \sum_{i=1}^{s} H(Y_i) = sH(Y_1) = sO(\max\{\log(n/k), 1\}). 
$$

Combining (7.2), (7.3), and (7.4), we obtain

$$
s = \Omega \left( \frac{n \log k}{\max\{\log(n/k), 1\}} \right).
$$

Since, by definition of $s$, at least half of all secret codes in $[k]^n$ can only be identified by the strategy $S$ after at least $s$ guesses, it follows that the expected number of queries needed to identify a uniformly chosen secret code is at least $s/2$. □
7.4.2 Upper Bound for Non-Adaptive Strategies

We first show that for \( k = \Theta(n) \) a random guessing strategy asymptotically achieves the lower bound from Theorem 7.11. Afterwards, we will show that one can also derandomize this.

Lemma 7.12. For black-peg Mastermind with \( n \) positions and \( k = \Theta(n) \) colors, the random guessing strategy needs an expected number of \( O(n \log n) \) queries to determine an arbitrary fixed code \( z \in [k]^n \). Furthermore, for a large enough constant \( C \), \( Cn \log n \) queries suffice with probability \( 1 - o(1) \).

Proof. We can easily eliminate colors whenever we receive a 0-answer. For every position \( i \in [n] \) we need to eliminate \( k - 1 \) potential colors. This can be seen as having \( n \) parallel coupon collectors, each of which needs to collect \( k - 1 \) coupons.

The probability that for a random guess we get an answer of 0 is \((1 - 1/k)^n\), i.e., constant. Conditional on a 0-answer, the color excluded at each position is sampled uniformly from all \( k - 1 \) colors that are wrong at that particular position. Thus the probability that at least one of the \( k - 1 \) wrong colors at one fixed position is not eliminated by the first \( t \) 0-answers is bounded by \((k - 1)(1 - 1/(k - 1))^t \leq ke^{-t/k}\).

Let now \( T \) denote the random variable that counts the number of 0-answers needed to determine the secret code. By a union bound over all \( n \) positions, we have \( \Pr[T \geq t] \leq nke^{-t/k} = \Theta(n^2) \cdot e^{-\Theta(t/n)} \). It follows by routine calculations that \( \mathbb{E}[T] = O(n \log n) \) and \( \Pr[T \geq Cn \log n] = o(1) \) for \( C \) large enough. As a random query returns a value of 0 with constant probability, the same bounds also hold for the total number of queries needed.

We now consider deterministic non-adaptive strategies to identify the secret code. Chvátal [Chv83] proved that the bound given in Theorem 7.11 is tight if \( k \leq n^{1-\varepsilon} \), \( \varepsilon > 0 \) a constant. Here we extend his argument to every \( k \leq n \). It essentially shows that a set of \( O(n \log k \max\{\log(n/k), 1\}) \) random guesses with high probability identifies every secret code. Our proof is based on the probabilistic method and is thus non-constructive. It remains an open question to find an explicit non-adaptive polynomial-time strategy that achieves this bound.

Theorem 7.13. There exists \( n_0 \in \mathbb{N} \) and a constant \( C > 0 \) such that for every \( n \geq n_0 \) and \( k \leq n \) there exists a deterministic non-adaptive
strategy for black-peg Mastermind with \( n \) positions and \( k \) colors that uses at most \( C \frac{n \log k}{\max\{\log(n/k),1\}} \) queries.

Proof. The idea is to use a probabilistic method type of argument, i.e., we show that, for an appropriately chosen constant \( C > 0 \) and \( n \) large enough, a set of \( N = C \frac{n \log k}{\max\{\log(n/k),1\}} \) random guesses with positive probability identifies every possible secret code. (In fact, we will show that such a set of queries has this property with high probability.)

If a set \( X = \{x^{(i)} \mid i \in N\} \) of queries distinguishes any two possible secret codes \( z, z' \), then there must exist for each such pair \( z \neq z' \) a query \( x \in X \) with \( \text{eq}(z, x) \neq \text{eq}(z', x) \). In particular we must have \( |\{i \in I(z, z') : x_i = z_i\}| \neq |\{i \in I(z, z') : x_i = z_i'\}| \) for \( I(z, z') := \{i \in [n] : z_i \neq z_i'\} \). Based on this observation we define (similar to [Chv83]) a difference pattern to be a set of indices \( I \subseteq [n] \) together with two lists of colors \((c_i)_{i \in I}, (c_i')_{i \in I}\) such that \( c_i \neq c_i' \) for every \( i \in I \). For every two distinct secret codes \( z, z' \in [k]^n \) we define the difference pattern corresponding to \( z \) and \( z' \) to be the set \( I(z, z') := \{i \in [n] : z_i \neq z_i'\} \) together with the lists \((z_i)_{i \in I}\) and \((z_i')_{i \in I}\). We say that a query \( x \in [k]^n \) splits a difference pattern given by \( I, (c_i)_{i \in I}, \) and \((c_i')_{i \in I}\) if

\[
|\{i \in I : x_i = c_i\}| \neq |\{i \in I : x_i = c_i'\}|.
\]

It is now easy to see that if a set of \( N \) queries has the property that every possible difference pattern is split by at least one query from that set, then these \( N \) queries together with the answers deterministically identify Codebreaker’s secret code.

In the following we show that a set of \( N = C \frac{n \log k}{\max\{\log(n/k),1\}} \) random queries with probability at least \( 1 - 1/n \) has the property that it splits every difference pattern.

The size of a difference pattern \( I, (c_i)_{i \in I}, (c_i')_{i \in I} \) is the cardinality of \( I \). Note that for fixed \( k \), the probability that a particular difference pattern is not split by a randomly chosen query only depends on its size. Let \( p(d, k) \) denote this probability for a difference pattern of size \( d \). The probability that there exists a difference pattern that is not split by any of the \( N \) random queries is at most

\[
\sum_{d=1}^{n} \binom{n}{d} (k(k-1))^d (p(d, k))^N.
\]
In order to show that this probability is at most $1/n$ it thus suffices to prove that for every $d \in [n]$ we have
\[
\binom{n}{d} (k(k-1))^d (p(d,k))^N < n^{-2}.
\] (7.5)

We first take a closer look at $p(d,k)$. Observe that if a query $x$ does not split a fixed difference pattern $I$, $(c_i)_{i \in I}$, $(c'_i)_{i \in I}$, then $x_i$ must agree with $c_i$ on exactly half of the positions in $I' := \{i \in I \mid x_i \in \{c_i, c'_i\}\}$, and it must agree with $c'_i$ on the other positions in $I'$. In particular, the size of $I'$ must be even. More precisely, we have
\[
p(d,k) = \sum_{i=0}^{\lfloor d/2 \rfloor} \binom{d}{2i} \left( \frac{2i}{i} \right) \left( \frac{1}{k} \right)^{2i} \left( 1 - \frac{2}{k} \right)^{d-2i}.
\]

Note that $(\binom{2i}{i}) 2^{-2i} \leq 1/2$ for every $i \geq 1$, and $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$. Hence,
\[
p(d,k) \leq \left( 1 - \frac{2}{k} \right)^d + \frac{1}{2} \sum_{j=1}^{d} \binom{d}{j} \left( \frac{2}{k} \right)^j \left( 1 - \frac{2}{k} \right)^{d-j}
= 1 - \frac{1}{2} \left( 1 - \left( 1 - \frac{2}{k} \right)^d \right)
\leq \exp \left( -\frac{1}{2} \left( 1 - e^{-2d/k} \right) \right).
\]

It follows that
\[
\ln \frac{1}{p(d,k)} \geq \frac{1}{2} \left( 1 - e^{-2d/k} \right).
\] (7.6)

We now split the proof into two cases, $k \geq cn$ and $k < cn$ where $c$ is a sufficiently small constant. (We determine $c$ at the end of the proof.)

**Case 1.** $k \geq cn$. Observe that in this case $\log(n/k) \leq \log(1/c)$ and $\log k = \log n + \Theta(1)$. Hence, the bound claimed in Theorem 7.13 evaluates to $O(n \log n)$ in this case. It thus suffices to show that there exists a constant $C > 0$ such that $N = Cn \log n$ queries already identify every secret code with high probability.
We show $n^{5d}(p(d, k))^N < 1$ for every $d \in [n]$, which clearly implies (7.5). In fact, we show the equivalent inequality

$$\frac{N}{5d} \ln \frac{1}{p(d, k)} > \ln n. \tag{7.7}$$

Using (7.6) we obtain

$$\frac{N}{5d} \ln \frac{1}{p(d, k)} \geq N \frac{1 - e^{-2d/k}}{10d}. \tag{7.8}$$

Using that $d \mapsto (1 - e^{-2d/k})/d$ is a decreasing function in $d$ we can continue with

$$\frac{N}{5d} \ln \frac{1}{p(d, k)} \geq N \frac{(1 - e^{-2n/k})}{10n},$$

which is clearly larger than $\ln n$ for any $N > \frac{10}{(1-e^{-2}) \log e} n \log n$. Hence for such $N$ we have (7.7) which settles this case.

**Case 2.** $k < cn$. In this case we need to be more careful in our analysis since in our claimed bound the factor $\log(n/k)$ might be large and the factor $\log k$ might be substantially smaller than $\log n$.

In what follows, we regard only the case $k \geq 3$; the case $k = 2$ has already been solved, cf. [ER63b].

We first consider difference patterns of size $d \leq n \log k \log(n/k) \log n$. As in Case 1 we show that (7.7) holds for these patterns. Observe that (7.8) holds again in this case. Since the function $d \mapsto (1 - e^{-2d/k})/d$ is decreasing in $d$ and since $d \leq \frac{n \log k}{\log(n/k) \log n}$ we obtain

$$\frac{N}{5d} \ln \frac{1}{p(d, k)} \geq N \frac{(1 - e^{-2n/k})}{10n} \log(n/k) \log n. \tag{7.9}$$

Next we bound the exponent $\frac{n \log k}{k \log(n/k) \log n}$ in the previous expression. Note that the derivative of $\frac{n \log k}{k \log(n/k) \log n}$ with respect to $k$ is

$$\frac{n (\log n - \ln(2) log k \log(n/k))}{\ln(2) k^2 \log n \log^2(n/k)}. \tag{7.10}$$

We now show that this expression is less than 0 for $3 \leq k \leq n/4$. Indeed, observe that by setting $g(k) = \ln(2) log k \log(n/k)$ we have for $n$ large
enough that $g(3) = \ln(2) \log(3) \log(n/3) > 1.09 \log(n) - 3.3 > \log n$ and $g(n/4) = 2 \ln(2) \log(n/4) > \log n$. Moreover, observe that

$$g'(k) = \frac{\log n - \log(k^2)}{k}.$$ 

From this one easily sees that the function $g$ has a local maximum at $k = \sqrt{n}$ as its only extremal point in the interval in the interval $3 \leq k \leq n/4$. Hence $g(k) > \log n$ for every $3 \leq k \leq n/4$ and thus (7.10) is negative.

Hence, $\frac{n \log k}{k \log(n/k) \log n}$ is a decreasing function in $k$ and we have

$$\frac{n \log k}{k \log(n/k) \log n} \geq e^{\log(1/c)} \left(1 + \frac{\log e}{\log n}\right) \geq 1$$ for $n$ large enough. With this we can continue (7.9) with

$$\frac{N}{5d} \ln \frac{1}{p(d,k)} \geq \frac{1 - e^{-2}}{10} \frac{N \log(n/k) \log n}{n \log k},$$

which is certainly larger than $\ln n$ for any $N \geq \frac{10}{(1-e^{-2}) \log e \log(n/k)}$. This settles the case $k < cn$ for all $d \leq \frac{n \log k}{\log(n/k) \log n}$.

In the remainder of this proof we consider the case $k < cn$ and $d \geq \frac{n \log k}{\log(n/k) \log n}$. For such $d$ we establish the inequality $2^k k^{2n} (p(d,k))^N < n^{-2}$ which clearly implies (7.5). As done previously, we actually show the equivalent inequality

$$N \log \frac{1}{p(d,k)} > 2n \log k + n + 2 \log n. \quad (7.11)$$

First observe that $(\frac{2}{i})^{2-2i} \leq 1/\sqrt{i}$ for every $i \geq 1$. We denote by $\text{Bin}(n,p)$ a binomially distributed random variable with parameters $n$ and $p$. With this, we obtain

$$p(d,k) \leq \sum_{j=0}^{\lfloor d/k \rfloor} \binom{d}{j} \left(\frac{2}{k}\right)^j \left(1 - \frac{2}{k}\right)^{d-j}$$

$$+ \left(\frac{d}{k}\right)^{-1/2} \sum_{j=\lfloor d/k \rfloor + 1}^n \binom{d}{j} \left(\frac{2}{k}\right)^j \left(1 - \frac{2}{k}\right)^{d-j}$$

$$= \Pr\left[\text{Bin}\left(d, \frac{2}{k}\right) \leq \frac{d}{k}\right]$$

$$+ \left(\frac{d}{k}\right)^{-1/2} \sum_{j=\lfloor d/k \rfloor + 1}^n \binom{d}{j} \left(\frac{2}{k}\right)^j \left(1 - \frac{2}{k}\right)^{d-j}.$$
Using the Chernoff bound \( \Pr[\text{Bin}(n, p) \leq (1 - \delta)np] \leq \left( \frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right)^{np} \) we obtain
\[
\Pr \left[ \text{Bin} \left( d, \frac{2}{k} \right) \leq \frac{d}{k} \right] \leq \left( \frac{e^{-1/2}}{(1/2)^{1/2}} \right)^{2d/k} = \left( \frac{2}{e} \right)^{d/k}.
\]
Hence, we have
\[
p(d, k) \leq \left( \frac{2}{e} \right)^{d/k} + \left( \frac{d}{k} \right)^{-1/2}.
\]
It is not hard to see that the function
\[
f(k) = \frac{\left( \frac{2}{e} \right)^{d/k}}{\left( \frac{d}{k} \right)^{-1/2}}
\]
attains its maximum at \( k = 2(1 - \ln 2)d \) and that \( f(2(1 - \ln 2)d) \leq 1 \). Hence we have
\[
p(d, k) \leq 2 \left( \frac{d}{k} \right)^{-1/2} = \left( \frac{d}{4k} \right)^{-1/2}.
\]
With this we obtain
\[
N \log \frac{1}{p(d, k)} \geq N \left( \log d - \log k - 2 \right)
\geq N \left( \log n + \log \log k - \log \log(n/k) - \log \log n - \log k - 2 \right)
= N \left( \log(n/k) - \log \log(n/k) - \log \left( \frac{\log n}{\log k} \right) - 2 \right)
\geq N \frac{4}{4} \log(n/k)
\]
where the last inequality follows from \( \frac{1}{2} \log(n/k) - \log \log(n/k) - \log \left( \frac{\log n}{\log k} \right) - 2 \geq 0 \) for every \( k \leq cn \) for a sufficiently small constant \( c > 0 \) and \( n \) large enough. (In fact, this step imposes the most restrictive bound on \( c \), i.e., any \( c > 0 \) that, for \( n \) large enough, satisfies \( \frac{1}{2} \log(1/c) - \log \log(1/c) - \log(1 - \frac{\log e}{\log cn}) - 2 \geq 0 \) is appropriate for our proof.) Clearly \( N \frac{4}{4} \log(n/k) \) is larger than \( 2n \log k + n + 2 \log n \) for any \( N > 16 \frac{n \log(n/k)}{\log(n/k)} \) and \( n \) large enough. This implies (7.11) and thus settles this last case. \( \square \)


[Sch16] Issai Schur, Über die Kongruenz $x^m + y^m \equiv z^m \pmod{p}$, Jahresbericht der Deutschen Mathematiker-Vereinigung 25 (1916), 114–117.


Henning Thomas
born on June 2, 1985
in Aachen, Germany

1991 - 1994  Primary school in Mönkeberg, Germany
1994 - 2003  Heinrich-Heine-Schule Heikendorf, Germany
             Degree: Abitur
2003 - 2006  Studies of Computer Science
             Christian-Albrechts-Universität zu Kiel, Germany
             Degree: Bachelor of Science
2006 - 2008  Studies of Computer Science
             ETH Zurich, Switzerland
             Degree: Master of Science
2008        Internship
             Siemens R&D Spring House, PA, USA
since 2008   Ph.D. student at ETH Zurich, Switzerland