Karlsruhe Institute of Technology

Freeze Tag Cop Number of Graphs

Bachelor's Thesis of

Jonas Dalchow

At the Department of Informatics
Institute of Theoretical Informatics (ITI)

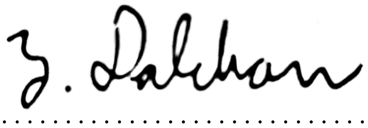
Reviewer:	PD Dr. Torsten Ueckerdt
Second reviewer:	T.T.-Prof. Dr. Thomas Bläsius
Advisor:	Miriam Goetze

27.05.2024 – 27.09.2024

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, 27.09.2024



. . . .

(Jonas Dalchow)

Abstract

Freeze Tag Cops and Robbers is a new variant of the well-studied pursuit and evasion game *Cops and Robbers*. In our variant, multiple cops play against multiple robbers, and for a graph G we are interested in the *freeze tag cop number* $c_r^{\mathrm{FT}}(G)$. It denotes the number of cops necessary to win against r robbers.
We determine lower and upper bounds on the *freeze tag cop number* for different families of graphs. We show that one cop always wins on paths, cycles, complete graphs, and interval graphs. For graphs with a universal vertex, two cops can always win, and there are graphs with a universal vertex for which two cops are necessary.
There is a constant c such that $c_r^{\mathrm{FT}}(T) < c \log r$ for all trees T, and this upper bound is tight. Furthermore, we prove that there is no function $f(r) \in o(r)$ such that $c_r^{\mathrm{FT}}(G) < f(r) + c(G)$ for all graphs G.

Zusammenfassung

Freeze Tag Cops and Robbers ist eine neue Variante des viel untersuchten Verfolgungsspiels *Cops and Robbers*. In unserer Variante spielen mehrere Cops gegen mehrere Robber und für einen Graphen G sind wir an der sogenannten *freeze tag cop number* $c_r^{\mathrm{FT}}(G)$ interessiert. Sie bezeichnet die Anzahl an benötigten Cops, um gegen r Robbers zu gewinnen.
Wir bestimmen untere und obere Schranken für die *freeze tag cop number* für unterschiedliche Familien von Graphen. Wir zeigen, dass ein Cop immer gewinnt auf Pfaden, Kreisen, vollständigen Graphen und Intervallgraphen. Auf Graphen mit einem universellen Knoten können zwei Cops immer gewinnen und es gibt Graphen mit universellem Knoten, für welche auch zwei Cops benötigt werden.
Es gibt eine Konstante c, sodass $c_r^{\mathrm{FT}}(T) < c \log r$ für alle Bäume T gilt. Diese Schranke ist fest. Außerdem beweisen wir, dass es keine Funktion $f(r) \in o(r)$ gibt, sodass $c_r^{\mathrm{FT}}(G) < f(r) + c(G)$ für alle Graphen G gilt.

Contents

1 Introduction

Cops and Robbers is a pursuit-evasion game that was first introduced in the early 1980s [AF84]. The game is played on connected, undirected graphs. In the game, one player controls a set of cops, and his goal is to catch a robber, who is controlled by the opposing player. At the beginning of the game, the first player places every cop on possibly distinct vertices. Then the other player places the robber on a vertex. From there on, the players take alternating turns. A turn consists of a player moving all the pieces he controls (all cops or the robber) to a vertex with a distance of at most one. This means every piece stays at its vertex or moves along an edge. The player controlling the cops wins if, at some point, one cop is on the same vertex as the robber. The other player wins if, the robber is able to evade the cops indefinitely. The minimum number of cops that are required to catch the robber, assuming perfect play from both parties, is the cop number of the graph, which is denoted by $c(G)$ for a graph G. The cop number has been well studied. It has been proven for example that for planar graphs $c(G) \leq 3$([AF84]). One of the most prominent open questions is whether Meynil's conjecture holds. The conjecture, that has not been proven yet, says that for a graph G with n vertices $c(G) \leq \sqrt{n}$.
Over time, many different variants of the game have been studied. In one of them, the robber can move at a speed of $R > 1$ for example ([FKL12]). Another variant of the game is called Tipsy Cop and Drunken Robber ([HIL24]). In this variant, the robber's moves are random, and the cop's moves are sometimes random and sometimes intentional. Here, the expected number of moves until the cop catches the robber is an interesting parameter.
In this thesis, we study a variant of the game, which we call Freeze Tag Cops and Robbers. In most variants, there is only a single robber, and adding more robbers would not make much sense because the cops could just catch the robbers one by one. We present a new variant of the game in which several cops play against several robbers, and a caught robber has a chance to become free again so that the cops have to catch the robber again.
In our variant, every robber can be either *free* or *frozen*. Initially, all robbers are free. A free robber becomes frozen if, at any point, he is on the same vertex as a cop. We say the cop freezes the robber. A frozen robber can become free again if, at any point, there is a free robber but no cop on the same vertex. We say that the robber frees the frozen one. The cops win if, at any point, all robbers are frozen. Otherwise, the robbers win. Analogously to the cop number, we define the *freeze tag cop number* $c_r^{\text{FT}}(G)$ as the minimum number of cops that are needed to win against r robbers on G. For $r = 1$, the cop number and the freeze tag cop number are equal, i.e. $c_1^{\text{FT}}(G) = c(G)$.
There are many different variants of the game, but for as far as we know, this variant of the game has not been studied yet. Our goal of this thesis is to determine lower and upper bounds of the freeze tag cop number for different families of graphs.

2 Simple Bounds

In this chapter, we will first show some general bounds on the freeze tag cop number that hold for all graphs. Then we will determine the freeze tag cop numbers on some graphs for which the problem is trivial in the original version of the game.

2.1 General bounds

To be able to freeze multiple robbers, there must be enough cops to freeze a single robber. Therefore, $c(G)$ provides a lower bound for $c_r^{\mathrm{FT}}(G)$ for every graph G. We will see that there are graphs for which both numbers are equal for every r.
A quite simple cop-winning strategy is to catch the robbers one by one, and after a robber is frozen, a cop stays on that vertex and guards the robber until the game is over. To freeze the r-th robber, $c(G)$ cops are needed to catch him, while $r - 1$ cops are guarding the other $r - 1$ frozen robbers. It follows that $c_r^{\mathrm{FT}}(G) \leq c(G) + r - 1$ for any graph G. We do not know whether there exists a graph G for which equality holds. Yet, in Chapter 4 we show that there are graphs for which $c_r^{\mathrm{FT}}(G)$ is linear in r.

Lemma 2.1: *For any graph G, $c(G) \leq c_r^{FT}(G) \leq c(G) + r - 1$.*

2.2 Paths

In the classical version of the game, on a path, one cop can move towards the robber until he catches him. Therefore, $c(P) = 1$ for any path P. It is quite easy to see, that one cop can also freeze an arbitrary number of robbers.

Lemma 2.2: *For any path P, $c_r^{FT}(P) = 1$.*

Proof. The cop can start on one end of the path and move towards the other end. The cop splits the graph into two components and in one of them all robbers are frozen. This component becomes larger and larger as the cop reaches the other end. When he reaches the other end, all the robbers on the path are frozen. ∎

2.3 Cycles

For a cycle of length greater than 3, one cop is not enough to catch a single robber. The robber can play in such a way that after each of his moves, the distance to the cop is 2. Thus, he is never caught and wins. With two cops, we can place one cop at a vertex and pretend that this vertex does not exist. Then we can catch the robber with the above-mentioned strategy for paths, see Lemma 2.2. Consider a cycle C_n on n vertices. Then $c(C_n) = 2$, for $n > 3$. Using the same strategy, two cops can win against an arbitrary number of robbers.

Theorem 2.3: *Two cops always win on cycles of length greater than 3, i.e. $c_r^{FT}(C_n) = 2$ for $n > 3$.*

We have now seen two classes of graphs where the same strategy from the original game can be applied to our version. These graphs provide examples where the cop number and the freeze tag cop number coincide, i.e. the lower bound in Lemma 2.1 is tight.

2.4 Complete Graphs

Complete graphs are another class of graphs for which determining the cop number is quite easy. A single cop can simply catch the robber on his first move, and therefore $c(K_n) = 1$. With more robbers, it is not that simple, but we can still show that one cop can win against an arbitrary number of robbers by using a different technique.

Theorem 2.4: *On a complete graph, one cop can win against any number of robbers, i.e.* $c_r^{FT}(K_n) = 1$.

Proof. Let $v_1, v_2, \dots, v_n$ be an arbitrary ordering of the vertices. We describe a winning strategy for the cop. The cop starts at vertex v_n. As long as there are free robbers remaining, the cop moves to v_i, where i is the largest integer, such that there is a free robber on v_i. Now we prove that all robbers are frozen after a finite number of turns. Let v_{p_k} be the vertex the cop lands on after his k-th turn. Let $f_k(v_i)$ be the number of frozen robbers on v_i after the cop finished his k-th turn. We define the potential A_k as

$$A_k = \sum_{i=1}^{n} f_k(v_i) \cdot r^i \tag{2.1}$$

Note, that is suffices to show that $A_{k+1} > A_k$ as the potential cannot grow indefinitely. Thus after finitely many turns, all robbers must be frozen. Assume there is still a free robber after the k-th turn of the cop. Consider now the state after the cop performed his $(k+1)$-th move. We know that in the last move, no robber ended his move on a vertex with index greater than p_{k+1}. Otherwise, the cop would have moved to that vertex. So all the robbers on vertices $v_{p_{k+1}} + 1, \dots, v_n$ are still frozen. This implies $f_k(v_i) = f_{k+1}(v_i)$ for $i \in \{p_{k+1} + 1, \dots, n\}$. Additionally, at least one robber was frozen on vertex $v_{p_{k+1}}$ and the robbers that were frozen on that vertex before are frozen again. Therefore, $f_{k+1}(v_{p_{k+1}}) \geq 1 + f_k(v_{p_{k+1}})$. It follows that:

$$
\begin{aligned}
A_k &= \sum_{i=1}^{n} f_k(v_i) \cdot r^i \\
&= \Big(\sum_{i=1}^{p_{k+1}-1} f_k(v_i) \cdot r^i \Big) + f_k(v_{p_{k+1}}) \cdot r^{p_{k+1}} + \sum_{i=p_{k+1}+1}^{n} f_k(v_i) \cdot r^i \\
&\leq \Big(\sum_{i=1}^{p_{k+1}-1} f_k(v_i) \cdot r^{p_{k+1}-1} \Big) + f_k(v_{p_{k+1}}) \cdot r^{p_{k+1}} + \sum_{i=p_{k+1}+1}^{n} f_k(v_i) \cdot r^i \\
&= \Big(\sum_{i=1}^{p_{k+1}-1} f_k(v_i) \Big) \cdot r^{p_{k+1}-1} + f_k(v_{p_{k+1}}) \cdot r^{p_{k+1}} + \sum_{i=p_{k+1}+1}^{n} f_{k+1}(v_i) \cdot r^i \\
&< r \cdot r^{p_{k+1}-1} + f_k(v_{p_{k+1}}) \cdot r^{p_{k+1}} + \sum_{i=p_{k+1}+1}^{n} f_{k+1}(v_i) \cdot r^i \\
&= (1 + f_k(v_{p_{k+1}})) \cdot r^{p_{k+1}} + \sum_{i=p_{k+1}+1}^{n} f_{k+1}(v_i) \cdot r^i \\
&\leq f_{k+1}(v_{p_{k+1}}) \cdot r^{p_{k+1}} + \sum_{i=p_{k+1}+1}^{n} f_{k+1}(v_i) \cdot r^i \\
&= \sum_{i=p_{k+1}}^{n} f_{k+1}(v_i) \cdot r^i \\
&\leq \sum_{i=1}^{n} f_{k+1}(v_i) \cdot r^i \\
&= A_{k+1}
\end{aligned}
\tag{2.2}
$$

We have hereby shown, that the potential increases in every step and all robbers are frozen after a finite number of turns. The cop wins. ■

2.5 Interval Graphs

Chordal graphs are graphs that do not contain an induced cycle of length larger than 3. For these graphs, it has been proven that one cop always wins against a robber([Hah07]). This is not the case in the freeze tag variant. Trees are chordal graphs and in Chapter 3 we will see that there are trees for which more cops are required to win against multiple robbers. Whether there is a better upper bound on the freeze tag cop number than the one mentioned in Lemma 2.1 for chordal graphs remains an open problem. Nevertheless, we can determine the freeze tag cop number for interval graphs, which are a subset of the chordal graphs. We will show, that one cop beats arbitrary many robbers, using a strategy similar to the one for complete graphs in Theorem 2.4.

Definition 2.5: *A graph is an interval graph if and only if we can assign to every vertex v_i an interval $[l_i, r_i]$ such that any two vertices are adjacent if and only if their intervals share a common point.*

Every set of intervals describes a graph and we call the set an *interval representation* of the graph.

As in the proof of Theorem 2.4, we will order the vertices and the cop plays in such a way, that the robbers who are on vertices with a higher index are frozen. We first observe that we can sort the vertices in such a way that if a robber from a smaller index frees a robber on a larger index, then the cop can move to that same vertex too.

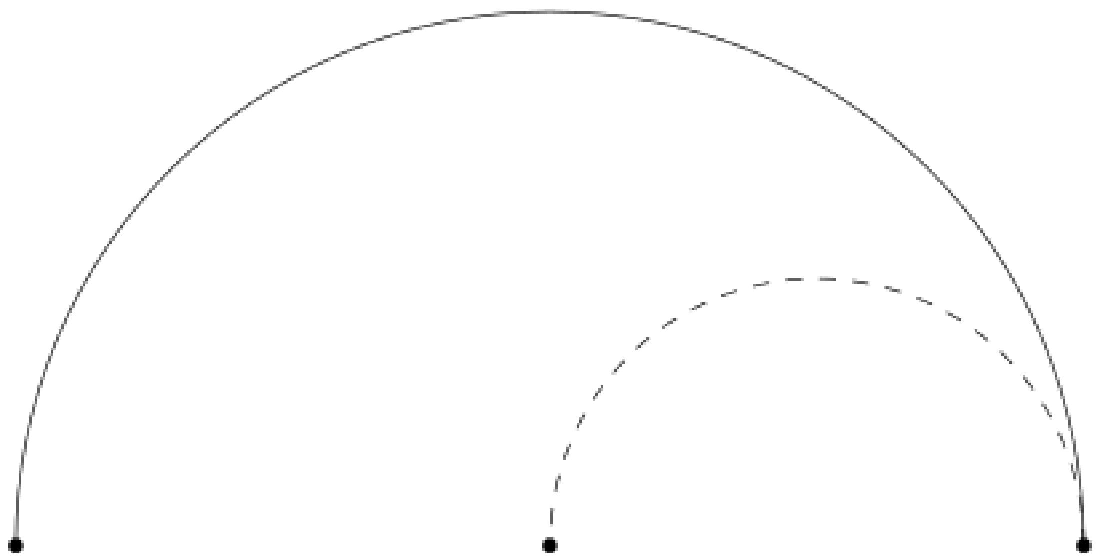


Figure 2.1: Forbidden pattern in interval graphs

Lemma 2.6: *If G is an interval graph, then there exists an ordering of the vertices $v_1, \dots, v_n$ that does not contain the pattern represented in Figure 2.1. This means for any vertices v_i, v_j, v_k with $i < j < k$, where v_i and v_k are adjacent, it follows that v_j and v_k are adjacent.*

Proof. Consider an interval representation of G and let $[l_a, r_a]$ denote the interval associated with vertex v_a for every a. We sort the intervals in ascending order of the right border. Now we need to show that if v_i and v_k are adjacent, then v_j and v_k are adjacent for $i < j < k$. If the edge $v_i v_k$ exists, the corresponding intervals intersect. Because of the way we ordered the intervals, we know $r_i \leq r_k$. So for the intervals to intersect, it must hold that $l_k \leq r_i$. Therefore, $l_k \leq r_i \leq r_j \leq r_k$ and v_j and v_k are adjacent because the intervals share r_j as a common point. ∎

Now we come to the main part of the proof.

Theorem 2.7: *On a connected interval graph, one cop wins against any number of robbers, i.e. for an interval graph G, $c_r^{FT}(G) = 1$.*

Proof. Let G be a connected interval graph. By Lemma 2.6, there exists an ordering $v_1, \dots v_n$ of the vertices such that for any indices $i < j < k$, if v_i and v_k are adjacent, then v_j and v_k are adjacent too. Next we describe a cop-winning strategy. We place the cop at v_n. On every turn, the cop wants to move from vertex v_i to v_{i-1}, if there is no free robber on a vertex v_k with $k \in \{i, \dots, n\}$. Otherwise, among all vertices with free robbers on them, the cop catches the robber on the vertex with the highest index in one move. We show that the cop can always reach that vertex in one move, and that the game ends after finitely many turns.

Let p_k be the index of the vertex the cop lands on after the k-th move. Let $f_k(v_i)$ be the number of frozen robbers on v_i after the cop finished his k-th turn. We define the potential A_k as

$$A_k = -\frac{p_k}{n+1} + \sum_{i=p_k}^{n} f_k(v_i) \cdot r^i \tag{2.3}$$

We claim that there is no free robber on the vertices $v_{p_k}, \ldots, v_n$ after the cop finishes his k-th move. We show, that by following our strategy, $A_k < A_{k+1}$, and because A_k can only take a finite number of different values, the potential cannot increase infinitely. Therefore, the cop wins.

We now look at the state of the game after the cop's $(k+1)$-th move.

First, let us assume that during the previous robber turn, one or more robbers moved to a vertex with an index greater than p_k. Let v_j be the vertex with the largest index where a robber moved to during the last robber turn. This will be the vertex where the cop moves next, i.e. $j = p_{k+1}$. Let v_l be the vertex the robber came from. We have $l < p_k < j$. Then v_{p_k} and v_j are adjacent according to Lemma 2.6. All the robbers on vertices $v_{p_{k+1}+1}, \ldots, v_n$ remain frozen. After the cop moves to $v_{p_{k+1}}$, there is then at least one more frozen robber at that vertex ($f_k(v_{p_{k+1}}) \leq f_{k+1}(v_{p_{k+1}}) - 1$). It remains to show that the potential increases, i.e. $A_{k+1} > A_k$. We have

$$\begin{aligned}
A_k &= -\frac{p_k}{n+1} + \sum_{i=p_k}^{n} f_k(v_i) \cdot r^i \\
&= -\frac{p_k}{n+1} + \left(\sum_{i=p_k}^{p_{k+1}-1} f_k(v_i) \cdot r^i \right) + f_k(v_{p_{k+1}}) \cdot r^{p_{k+1}} + \left(\sum_{i=p_{k+1}+1}^{n} f_k(v_i) \cdot r^i \right) \\
&\leq -\frac{p_k}{n+1} + \left(\sum_{i=p_k}^{p_{k+1}-1} f_k(v_i) \right) \cdot r^{p_{k+1}-1} + (f_{k+1}(v_{p_{k+1}}) - 1) \cdot r^{p_{k+1}} + \left(\sum_{i=p_{k+1}+1}^{n} f_{k+1}(v_i) \cdot r^i \right)
\end{aligned} \tag{2.4}$$

As at least one robber is frozen on $v_{p_{k+1}}$, at most $r-1$ robbers are on vertices with indices smaller than p_{k+1}, we obtain:

$$\begin{aligned}
A_k &\leq -\frac{p_k}{n+1} + (r-1) \cdot r^{p_{k+1}-1} + f_{k+1}(v_{p_{k+1}}) \cdot r^{p_{k+1}} - r^{p_{k+1}} + \left(\sum_{i=p_{k+1}+1}^{n} f_{k+1}(v_i) \cdot r^i \right) \\
&= -\frac{p_k}{n+1} - r^{p_{k+1}-1} + f_{k+1}(v_{p_{k+1}}) \cdot r^{p_{k+1}} + \sum_{i=p_{k+1}+1}^{n} f_{k+1}(v_i) \cdot r^i \\
&< -1 + \sum_{i=p_{k+1}}^{n} f_{k+1}(v_i) \cdot r^i \\
&< -\frac{p_{k+1}}{n} + \sum_{i=p_{k+1}}^{n} f_{k+1}(v_i) \cdot r^i \\
&= A_{k+1}
\end{aligned} \tag{2.5}$$

Otherwise, before the $(k+1)$-th cop move, there was no free robber on vertices $v_{p_k}, \dots v_n$. If the edge $v_{p_k} v_{p_k - 1}$ exists, then $p_{k+1} = p_k - 1$ and all the robbers on vertices $v_{p_k}, \dots v_n$ are still frozen. Thus, $f_k(v_i) = f_{k+1}(v_i)$ for $i \in p_k, \dots, n$. In this case:

$$
\begin{aligned}
A_k &= -\frac{p_k}{n+1} + \sum_{i=p_k}^{n} f_k(v_i) \cdot r^i \\
&= -\frac{p_{k+1}+1}{n+1} + \sum_{i=p_k}^{n} f_{k+1}(v_i) \cdot r^i \\
&\leq -\frac{p_{k+1}}{n+1} - \frac{1}{n+1} + \sum_{i=p_{k+1}}^{n} f_{k+1}(v_i) \cdot r^i \\
&< -\frac{p_{k+1}}{n+1} + \sum_{i=p_{k+1}}^{n} f_{k+1}(v_i) \cdot r^i \\
&= A_{k+1}
\end{aligned}
\tag{2.6}
$$

If the edge $v_{p_k} v_{p_k - 1}$ does not exist, then there must be an index $x > p_k$ such that v_x is adjacent to a vertex v_y with $y < p_k$ as G is connected. It follows that v_x and $v_{p_k - 1}$ are adjacent, see Lemma 2.6. For this specific case, we show that the potential increases after two moves. The cop moved from v_{p_k} to $v_x = v_{p_{k+1}}$. If, after the $(k+1)$-th move, one robber moves to a vertex with index j and $j > p_k$, then $p_{k+2} = j$. The corresponding edge exists according to Lemma 2.6. The potential increases in the same way as described in the other case above. If no robber moved to a vertex with a greater index than p_k, then $p_{k+2} = p_k - 1$ and the potential increases too. It follows that the game ends, and the cop wins after finitely many moves. ∎

2.6 Graphs with Universal Vertex

The last class of graphs we will look at in this chapter are graphs with a universal vertex, i.e. graphs that contain a vertex which is adjacent to all other vertices. The cop number for such graphs is 1 because the cop can start at the universal vertex and catch the robber in one move. We will show that the freeze tag cop number for such graphs is either 1 or 2. First, we show that a single cop always wins against up to three robbers.

Theorem 2.8: *On a graph G with a universal vertex, one cop always wins against three robbers, i.e. $c_3^{FT}(G) = 1$.*

Proof. Let u be the universal vertex in G and let the robbers be called r_1, r_2 and r_3. We describe a winning strategy for the cop. The cop starts at u. On his first turn, he freezes r_1 at a vertex v. On the next move, he goes back to u and on the move after that, he freezes the robber that is closest to v. Now at least two robbers are frozen because if one robber has freed r_1 then that robber is at v and the cop freezes both of them. The cop is at least as close to v as the last robber. If the last robber frees r_1, then the cop is at distance 1 to v and can move to v and win the game. Otherwise, he moves to u and then catches the last robber in the next move to win the game. ∎

It follows that $c_1^{\mathrm{FT}}(G) = 1$ and $c_2^{\mathrm{FT}}(G) = 1$ for a graph G with a universal vertex. Next, we will show that the bound proven is tight and that there is a graph with a universal vertex, where four robbers win against one cop. This is our first proof in which the robbers win. Those proofs are easier if we can make assumptions about the initial positions of all cops. We show that we can choose an arbitrary vertex and assume that all cops start at that vertex.

Lemma 2.9: *Let G be a graph and v be a vertex of G. Then c cops win against r robbers on G if and only if there is a winning strategy for the cops, where all cops are initially placed on v.*

Proof. Assuming that c cops win against r robbers, there is a winning strategy S for the cops that places the cops on the not necessarily different vertices $v_1, \ldots, v_c$. We now describe a winning strategy that places all cops on v. After the cops are placed, every cop c_i moves towards v_i. When every cop has reached his destination, the cops start moving according to the winning strategy S. The cops win because from then on, the game is the same as if the cops had started on vertices $v_1, \ldots, v_c$. The other direction holds trivially. ∎

Note that we could even choose the initial position for every cop and assume that they start there.

Theorem 2.10: *There exists a graph G with a universal vertex on which four robbers win against one cop, i.e. $c_4^{FT}(G) > 1$.*

Proof. We prove that four robbers win on a C_4 with one additional vertex v that is adjacent to all other vertices. Let u_1, u_2, u_3, u_4 be the vertices of C_4 in the order they appear on the cycle. We show that there are four states of the game (see Figure 2.2), and after one move of the cop and one move of the robbers, the game is in one of the four states again. We assume that the cop starts at v (see Lemma 2.9), the robbers start on the vertices u_1, u_2, u_3, u_4 and two of the robbers are already frozen, as shown in state 1. Now we look at each state separately and consider all possible moves the cop has.

- **State 1**: If the cop moves to a frozen robber, the robbers stand still, and we are in a state that is equivalent to state 2 (transition a in Figure 2.2).
  If the cop moves to a free robber,(we may assume that the cop moves to the vertex u_4), the free robber at u_3 frees the robber at u_2, and we are in state 3 (transition b in Figure 2.2).

- **State 2**: If the cop moves to u_2, the robbers stand still, and we are in a state that is equivalent to state 2 (transition c in Figure 2.2).
  If the cop moves to v, the robbers stand still, and we are in a state that is equivalent to state 1 (transition d in Figure 2.2).
  If the cop moves to u_4, the robber at u_3 moves to u_2, and we are in state 3 (transition e in Figure 2.2).

- **State 3**: If the cop moves to v, one robber moves from u_2 to u_3, and we are in a state that is equivalent to state 1 (transition f in Figure 2.2).
  If the cop moves to u_1, one robber moves from u_2 to u_3, and we are in a state that is equivalent to state 2 (transition g in Figure 2.2).
  If the cop moves to u_3, one robber moves from u_2 to u_1, and we are in state 4 (transition h in Figure 2.2).

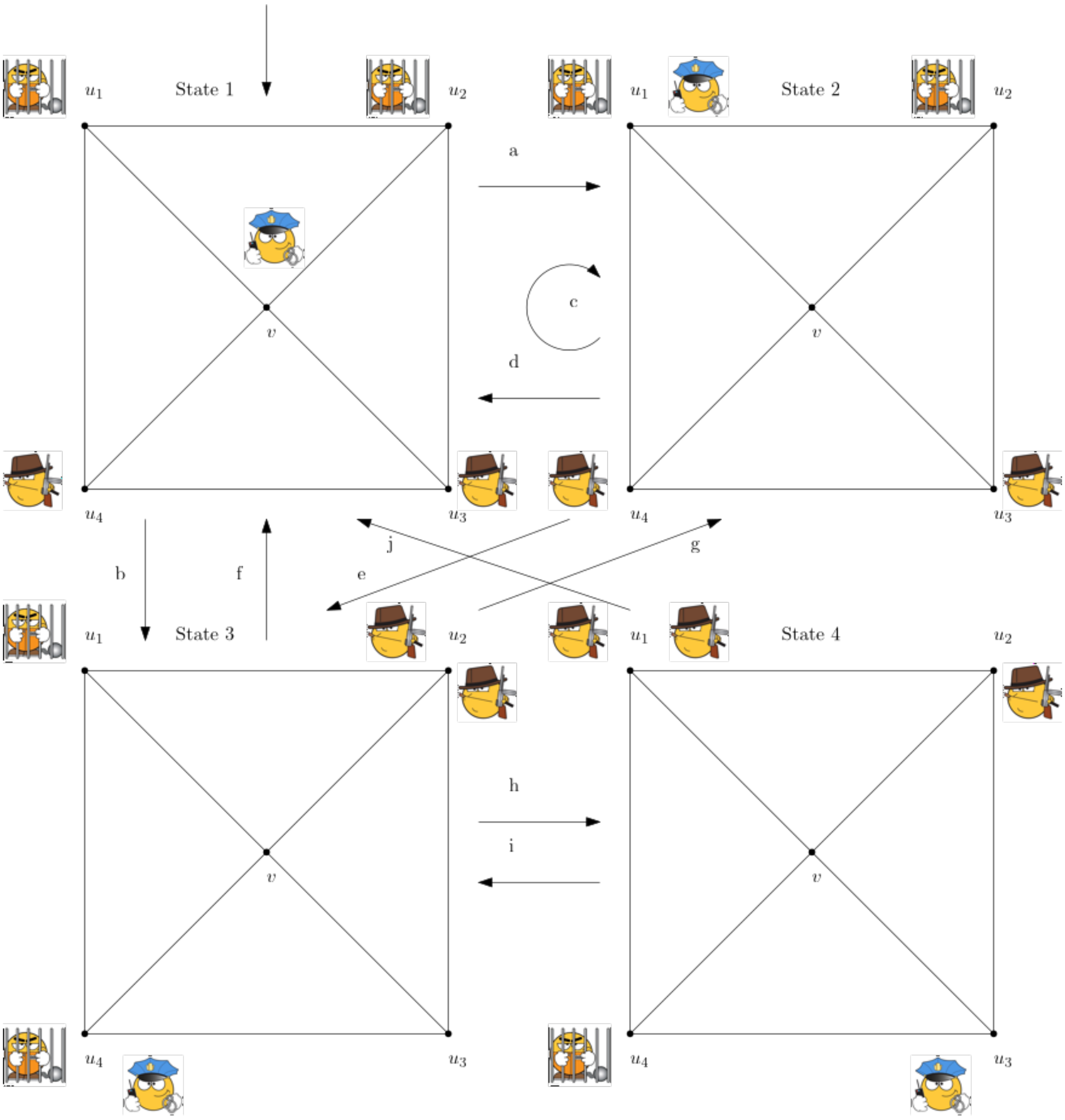


Figure 2.2: Sketch of all four states of the game.

- **State 4**: If the cop moves to u_2, one robber moves from u_1 to u_4, and we pretend that the other robber at u_1 is frozen. Then we are in a state that is equivalent to state 3 (transition i in Figure 2.2).
  If the cop moves to u_4, one robber moves from u_1 to u_2, and we pretend that the other robber at u_1 is frozen. Then we are in a state that is equivalent to state 3 (transition i in Figure 2.2).
  If the cop moves to v, the robber from u_2 moves to u_3 and one robber from u_1 moves to u_2. We pretend that the robber at u_3 is frozen, and we are in a state that is equivalent to state 1 (transition j in Figure 2.2).

We have hereby shown that no matter how the cop plays, we always end up in one of the four states, and the game goes on forever. ∎

Last but not least, we show two cops win against any number of robbers on a graph with a universal vertex using a similar strategy as in Theorem 2.4 again.

Theorem 2.11: *If G is a graph with a universal vertex, then two cops always win, i.e. for every r, $c_r^{FT}(G) \leq 2$.*

Proof. Let v be the universal vertex in G. Let G' be a complete graph on the set of vertices in $G - v$. For $u \in V(G)$, we denote the corresponding vertex in G' with u'. We know there is a winning strategy S on G' with a single cop, see Theorem 2.4. We now simulate S on G and will always keep one cop at v. If S places the cop on vertex u' in G', we place the two cops at v and u in G. For every robber in G we place the corresponding robber in G'. If a robber lands on v at any point, we can just remove the corresponding robber from G' because there is a cop at v at all times and the robber will be frozen forever. On every turn, S moves the cop from a vertex v_i' to a vertex v_j' and we move one cop from v to v_j and the other one from v_i to v. If a robber moves from v_i to v_j in G we let the robber in G' on v_i' move to v_j'. Following the same reasoning as the other proof, after a finite amount of turns, all robbers are frozen, and the cops win. ∎

3 Trees

In this chapter we investigate the probably most important family of graphs, trees. The game on trees is quite simple in the standard variant of the game. For all trees $c(G) = 1$ because the cop can just move towards the robber until he catches him. After every cop move the subtree containing the robber gets smaller, and the robber can only move within the subtree. So after a finite number of moves the cop catcher the robber. As we will see, the game is more complicated using our set of rules.

Let us first look at how many robbers a single cop can catch. A single cop can catch one robber as mentioned before, but it is not hard to see that one cop can also catch two robbers. After the cop froze the first robber, he can just move towards the second robber until he catches him. As before the subtree containing the second robber gets smaller every step and cop is always between the two robbers. Therefore, the second robber can never free the first robber and the cop always wins. We are also able to show that one cop wins against three robbers with a more complicated proof. In the proof we will often compare distances of vertices so let $d(a, b)$ denote the distance between the vertices a and b.

Theorem 3.1: *On a tree, one cop always wins against three robbers, i.e. for a tree T, $c_3^{FT}(T) = 1$.*

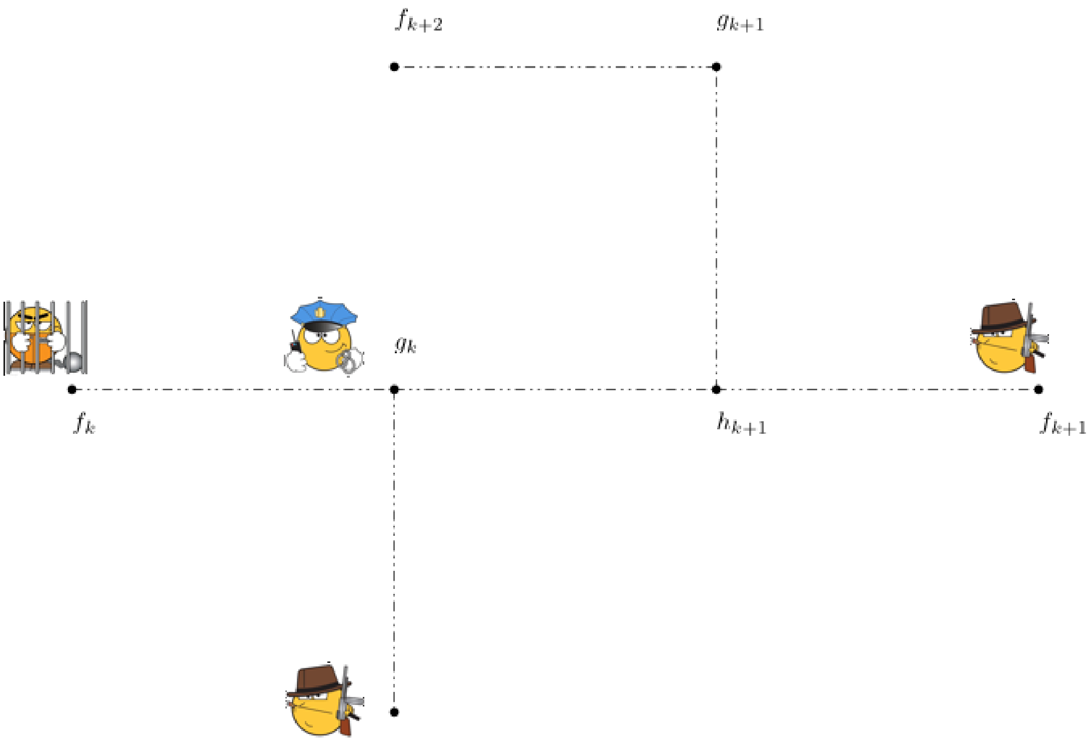


Figure 3.1: Sketch of the state of the game when the cop is at g_k

Proof. In the following, when talking about components, we refer to the components of $G - \{uv \mid u \in V(G)\}$, where v is the vertex the cop is on.
The winning cop-strategy, for which we prove its correctness, places the cop at an arbitrary vertex. Then, while there is still at least one free robber left, the cop chooses one of the free robbers (if none is left, he wins) and during the following turns, he moves to the component containing this robber until he freezes him.
If it happens at any point that there is only one free robber and the component containing that robber contains no frozen robber, then the cop wins after catching that robber.
Let f_k be the vertex on which the k-th freezing takes place. From there the cop moves towards the next targeted robber. We are now interested in the states of the game when for the first time, the two free robbers are in different components, and it is the cop's turn. Let g_k denote the vertex the cop is on. We know that g_k must be a vertex on the path between f_k and f_{k+1}. The first robber is still frozen because the cop always moved into the component containing both other robbers, and they had no way of getting around the cop. The two other robbers must be free, otherwise the cop can catch the last robber and wins as mentioned above.
We define a potential A_k as $A_k = d(f_k, g_k)$ and prove that after a finite number of turns all robbers are frozen by showing that $A_{k+1} > A_k$.
For $k > 1$ let h_k be the vertex on the path from f_{k-1} to f_k that minimizes $d(h_k, g_k)$, see Figure 3.1 for reference. In his next turns the cop moves along the paths from g_k to f_{k+1} and from f_{k+1} to g_{k+1}. That takes a total of $d(g_k, f_{k+1}) + d(f_{k+1}, g_{k+1})$ moves.
In the meantime, the free robber that is not chased needs at least one move to get to g_k and then $d(g_k, f_k)$ moves to free the frozen robber on f_k. Then one of the robbers has to reach g_{k+1} before the cop does, so that the two robbers can be in different components when the cop reaches g_{k+1}. So it must hold that:

$$
\begin{aligned}
& 1 + d(g_k, f_k) + d(f_k, g_{k+1}) < d(g_k, f_{k+1}) + d(f_{k+1}, g_{k+1}) \\
\iff\ & 1 + d(g_k, f_k) + d(f_k, h_{k+1}) + d(h_{k+1}, g_{k+1}) < d(g_k, f_{k+1}) + d(f_{k+1}, h_{k+1}) + d(h_{k+1}, g_{k+1}) \\
\iff\ & 1 + d(g_k, f_k) + d(f_k, h_{k+1}) < d(g_k, f_{k+1}) + d(f_{k+1}, h_{k+1}) \\
\iff\ & 1 + d(g_k, f_k) + d(f_k, f_{k+1}) - d(f_{k+1}, h_{k+1}) < d(f_k, f_{k+1}) - d(g_k, f_k) + d(f_{k+1}, h_{k+1}) \\
\iff\ & 1 + 2d(g_k, f_k) < 2d(f_{k+1}, h_{k+1}) \\
\implies\ & A_k = d(g_k, f_k) < d(f_{k+1}, h_{k+1}) \le d(f_{k+1}, h_{k+1}) + d(h_{k+1}, g_{k+1}) = d(f_{k+1}, g_{k+1}) = A_{k+1}
\end{aligned}
\tag{3.1}
$$

Because the potential cannot increase indefinitely the game must end after a finite number of turns and the cop wins. ∎

After this, the obvious follow-up question is: how about four robbers? Can a single cop always win against four robbers or are there trees on which four robbers win against one cop? We will show now that the latter is true and that there are trees on which a single cop is not enough to win against four robbers.

Theorem 3.2: *There exists a tree T on which four robbers win against a single cop, i.e. $c_4^{FT}(T) > 1$.*

Proof. The tree T is shown in Figure 3.2. Our set of vertices is $a, \dots, s$. We can assume that the single cop starts at a, see Lemma 2.9. Our robber-winning strategy places the robbers r_1, r_2, r_3 and r_4 on the vertices c, i, i and s respectively. We assume that r_4 is already frozen.
Let S be a cop-winning strategy that wins in as few moves as possible [1]. This implies that

[1] From this follows, that the game never reaches the same state twice.

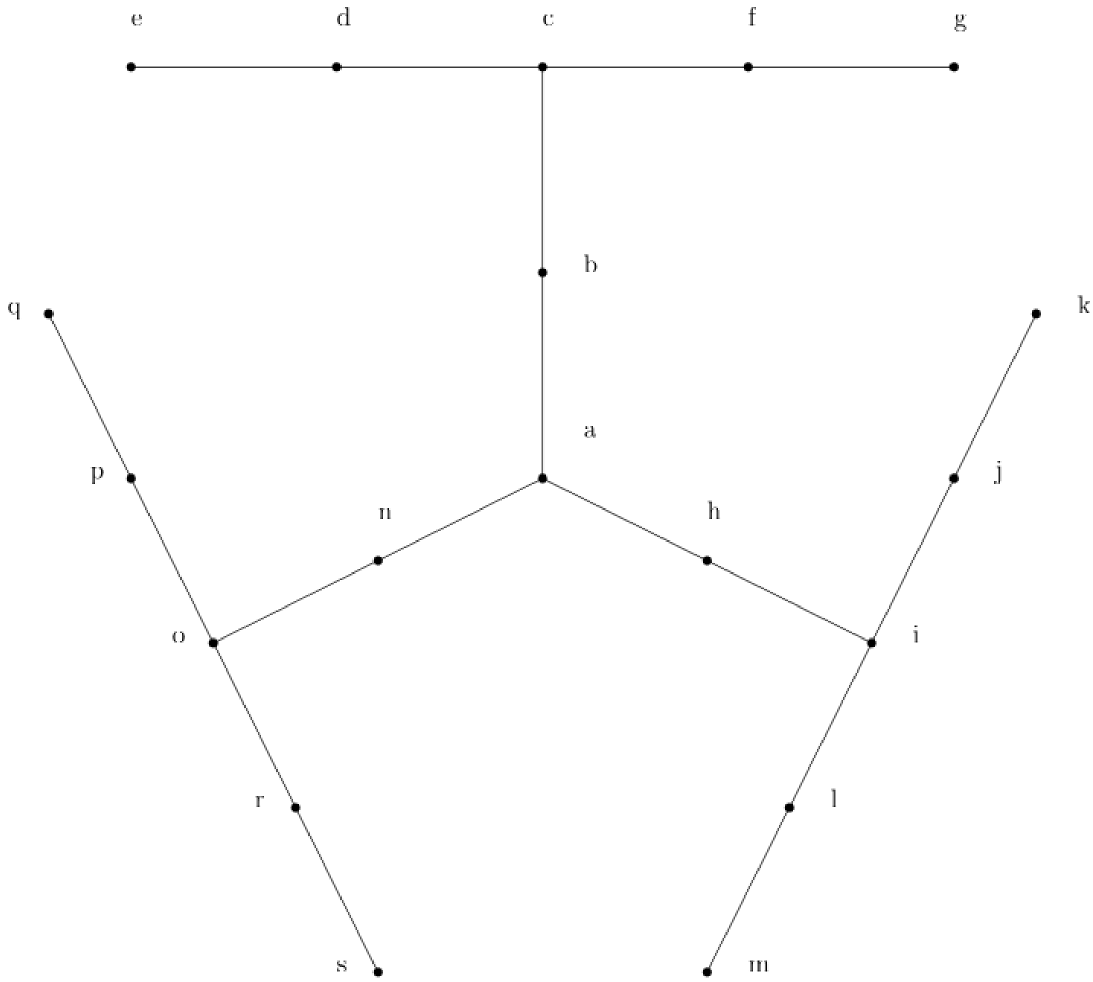


Figure 3.2: Tree T on which four robbers can win against a single cop.

the cop never moves back to the vertex he just came from except when he freezes a robber with that move, or he froze one the move before. Otherwise, all robbers can move back to the vertex they were on the move before. Then the game is in the same state as before, which is a contradiction to S winning as fast as possible.
We prove that S cannot exist because after a finite number of moves the game returns to a state that is equivalent to the initial state. The cop has to move towards one of the free robbers. He either moves towards r_1 or towards r_2 and r_3.

Case 1: the cop catches r_1
The robber r_1 moves along the path *cde* and lets himself get frozen at e. The cop will not turn around and has to move along the path *abcde* to catch the robber. Then the cop will move back along the path *edcba* because that is the direction where all the free robbers will be. In total this will take the cop eight moves. In the meantime r_2 stays at i and r_3 moves along the path *ihanors* in six moves and frees r_4. Then r_3 and r_4 move together along the path *sro* in two moves. Now the cop is at a, r_1 is frozen at e, r_2 is still at i and r_3 and r_4 are at o. That state is equivalent to the initial state.

actor	move sequence
cop	*abcdedcba*
r_1	*cdeeeeeee*
r_2	*iiiiiiiii*
r_3	*ihanorsro*
r_4	*sssssssro*

Figure 3.3: All 8 moves from every actor in Case 1.

Case 2: the cop catches r_2 or r_3
The robber r_2 moves along the path ijk and r_3 moves along the path ilm. The cop makes two moves towards both robbers along the path ahi. We can assume that the cop decides to catch r_2. His next two moves are then ijk. After that the cop moves back along the path $kjiha$ because that is the direction where all the free robbers will be. In the meantime r_3 moves along the path $mlihabc$ and does not get frozen because he passes i when the cop freezes r_2 at k. During the eight moves it takes the cop to get back to a, r_1 moves from c to s and frees r_4 and both robbers then move together along the path sro.

Now the cop is at a, r_2 is frozen at k, r_3 moved to c and r_3 and r_4 are at o. That state is equivalent to the initial state.

actor	move sequence
cop	*ahijkjiha*
r_1	*cbanorsro*
r_2	*ijkkkkkkk*
r_3	*ilmlihabc*
r_4	*sssssssro*

Figure 3.4: All 8 moves from every actor in Case 2.

In both cases we end up in an equivalent state after eight moves. This means the game does not end and the robbers win. ∎

Now that we have solved the problem for one cop, we investigate what happens if we add more cops to the game. We are interested in what the maximum number of robbers are that a fixed number of cops can beat on all trees. We will show that by adding just one cup, the cops can win against more than three times the number of robbers as before.

Theorem 3.3: *If for all trees T and a fixed r, $c_r^{FT}(T) \leq c$, then for all trees T' it holds that $c_{3r+2}^{FT}(T') \leq c + 1$.*

Proof. Let T be a tree. We prove that $c + 1$ cops have a winning strategy on T against $3r + 2$ robbers. In the following, when talking about components, we refer to the components of $G - \{uv \mid u \in V(G)\}$, where v is the vertex the cop c_1 is on.
Our cop-winning strategy starts with placing all cops on some vertex. Then we choose one component that contains at least $r + 1$ robbers and all cops move one step towards it. We repeat this process for as long as possible, but ignoring the component s we just came from. After the last step there are at most r robbers in every component excluding s. The total sum of robbers in all these components excluding s is at least $r + 1$. For the next moves c_1 stays

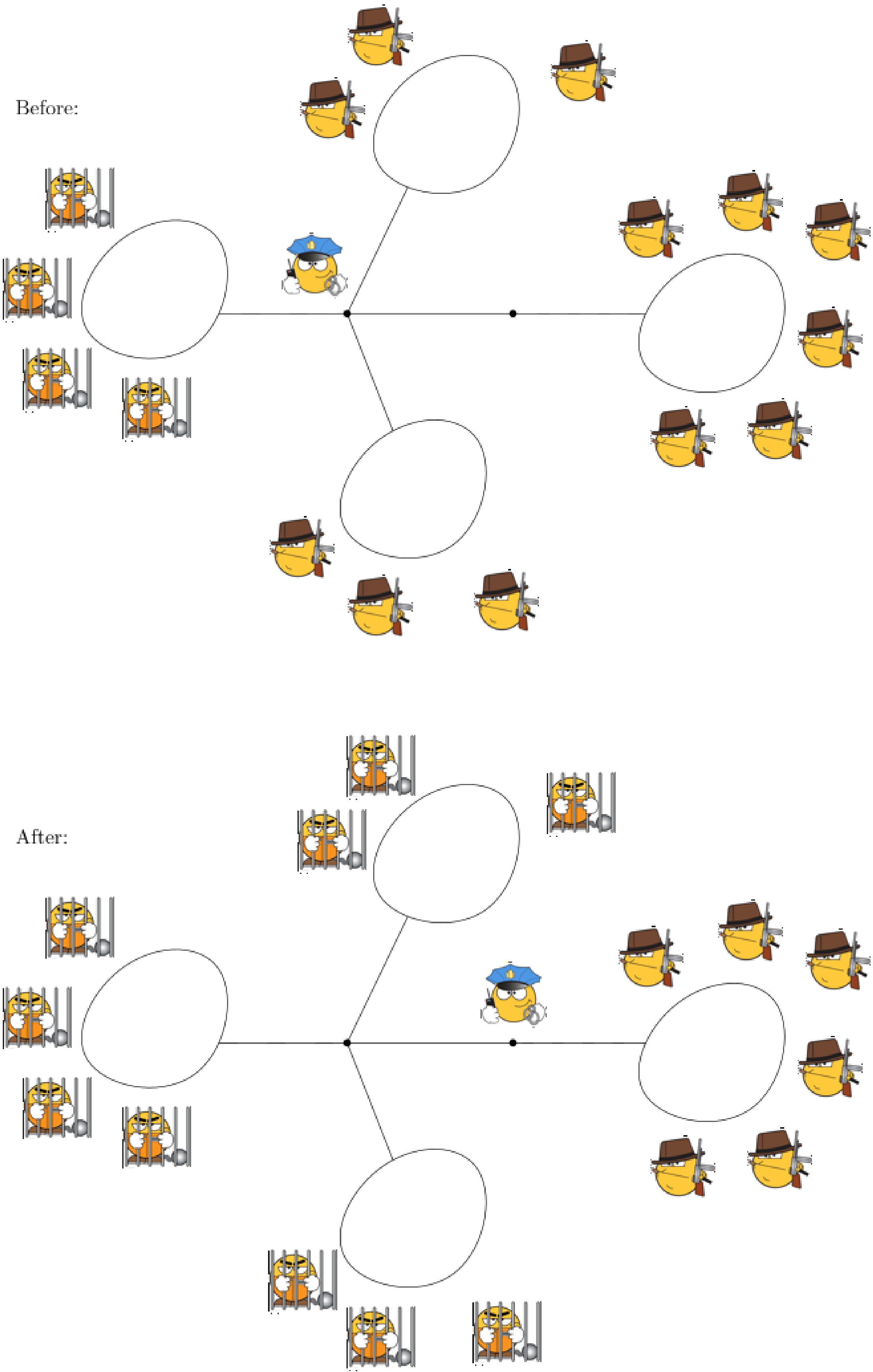


Figure 3.5: Sketch of the states of the game before and after one iteration

at the current vertex, while the other c cops freeze all robbers in all components except for s. This is possible because when a robber tries to reach a different component, he is frozen by c_1, and we know that $c_r^{\mathrm{FT}}(T) \leq c$. So the c cops can go from component to component and freeze all robbers in each component. Now all robbers in all components except for s are frozen. This component contains at most $2r + 1$ robbers.
Now c_1 moves one step into s. All the robbers in the component c_1 just came from are still frozen and the other components contain at most $2r + 1$ robbers in total. This implies that at most one component contains more than r robbers. Here we do the same strategy as above and let c_1 stay at the current vertex, while the other c cops freeze all robbers in the components containing at most r robbers.
These steps are then repeated until all cops are frozen. This happens after finitely many iterations because in every iteration, c_1 moves into a component that contains free robbers and the component c_1 came from contains only frozen robbers. One iteration can be seen in Figure 3.5. ■

We have thereby shown that for trees, the freeze tag cop number is at most logarithmic in the number of robbers. More specifically we can show:

Corollary 3.4: *Let T be a tree and $r \in \mathbb{N}_+$, then $c_r^{FT}(T) \leq \lceil \log_3(\frac{r+1}{4}) \rceil + 1$.*

Proof. We prove the corollary using induction over the number of robbers r. For $r \leq 3$, the claim holds as $c_r^{\mathrm{FT}}(T) = 1$ by Theorem 3.1. Assume the claim holds for all values smaller than r.
We now need to show that $c_r^{\mathrm{FT}}(T) \leq \lceil \log_3(\frac{r+1}{4}) \rceil + 1$. Let $r' = \lceil \frac{r-2}{3} \rceil$. Since $r \leq 3r' + 2$, it is also true that $c_r^{\mathrm{FT}}(T) \leq c_{3r'+2}^{\mathrm{FT}}(T)$. By induction hypothesis we know that $c_{r'}^{\mathrm{FT}}(T) \leq \lceil \log_3(\frac{r'+1}{4}) \rceil + 1$. Theorem 3.3 yields $c_{3r'+2}^{\mathrm{FT}}(T) \leq c_{r'}^{\mathrm{FT}}(T) + 1$. It follows that:

$$
\begin{aligned}
c_r^{\mathrm{FT}}(T) &\leq c_{3r'+2}^{\mathrm{FT}}(T) \leq c_{r'}^{\mathrm{FT}}(T) + 1 \\
&\leq \lceil \log_3(\frac{r'+1}{4}) \rceil + 1 + 1 \\
&= \lceil \log_3(\frac{(\frac{r-2}{3}) + 1}{4}) \rceil + \log_3(3) + 1 \\
&= \lceil \log_3(\frac{\frac{r+1}{3}}{4}) \rceil + \log_3(3) + 1 \\
&= \lceil \log_3(\frac{3 \cdot \frac{r+1}{3}}{4}) \rceil + 1 \\
&= \lceil \log_3(\frac{r+1}{4}) \rceil + 1
\end{aligned}
\tag{3.2}
$$

■

Similarly, let us assume that a tree exists on which c cops lose against r robbers. Then by tripling the number of robbers and adding a small constant number of robbers, we can construct a tree on which the robbers win with one more cop.

Theorem 3.5: *If for a some integer r there exists a tree T such that, $c_r^{FT}(T) > c$, then there exists a tree T' such that $c_{3r+6}^{FT}(T') > c + 1$.*

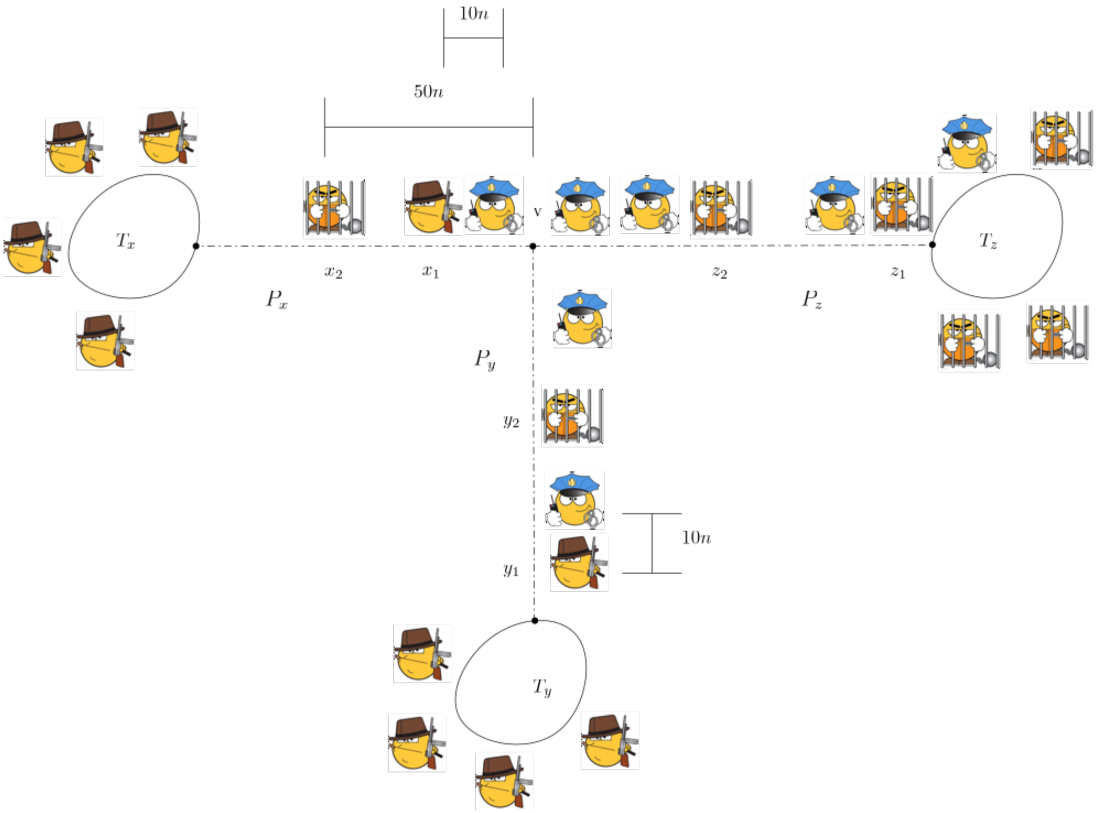


Figure 3.6: Sketch of state 1

Proof. Let T be a tree for which $c_r^{\mathrm{FT}}(T) > c$ holds and n be the number of vertices in T. We now construct a tree T' and prove that $3r + 6$ robbers win against $c + 1$ cops. We start with a vertex v and add three paths P_x, P_y, P_z, each consisting of $100n$ edges, which share v as an endpoint. For each of the paths, we introduce a copy of T which we denote by T_x, T_y and T_z respectively.
Identifying the endpoint of the paths different from v with a vertex in the corresponding copy of T yields the tree T'. We can assume that all cops are initially placed at v, see Lemma 2.9.

We place two robbers on P_x, one at a distance of $10n$ to v and the other one with a distance of $50n$. We call the first robber x_1 and the other one x_2. Analogously we place two robbers on P_y and call them y_1, y_2. On P_z we place the robber z_1 with distance $100n$ to v and z_2 with distance $50n$. We split the remaining $3r$ robbers in three groups X, Y, Z of size r. The robbers in X, Y, Z are placed in the trees T_x, T_y, T_z respectively.

In our strategy, we assume that the robbers in Z are frozen and that to win, the cops try to freeze all the robbers in Y before they try freezing X. The idea is, that when all the cops are moving towards Y, x_1 can free z_2 and move back. Then z_2 can free Z and go back to his vertex before the cops come back from freezing Y. This way all the robbers are where they need to be in time, and the cops are never able to freeze the three groups X, Y and Z at the same time.

We define three different states of the game and show that the game will always be in one of these three states. As at least one of the robbers is free in each of the three states, the game never ends, i.e. the robbers win. Sketches of the three states can be seen in Figure 3.6, Figure 3.7 and Figure 3.8.

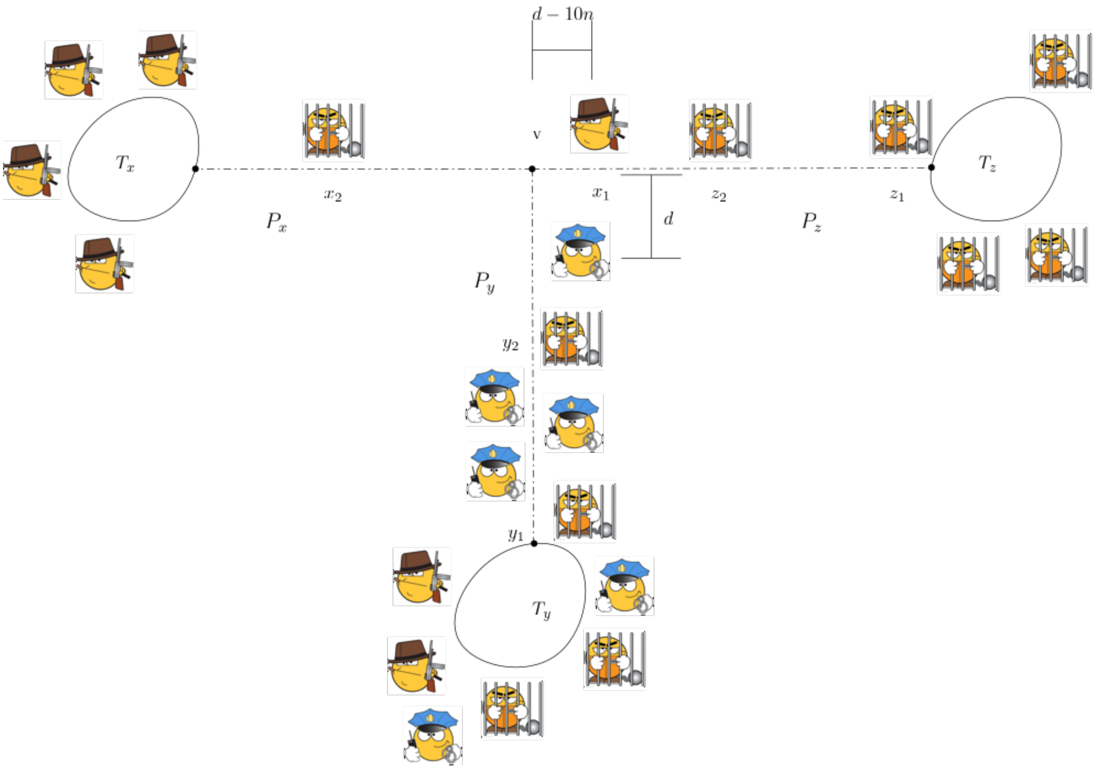


Figure 3.7: Sketch of state 2

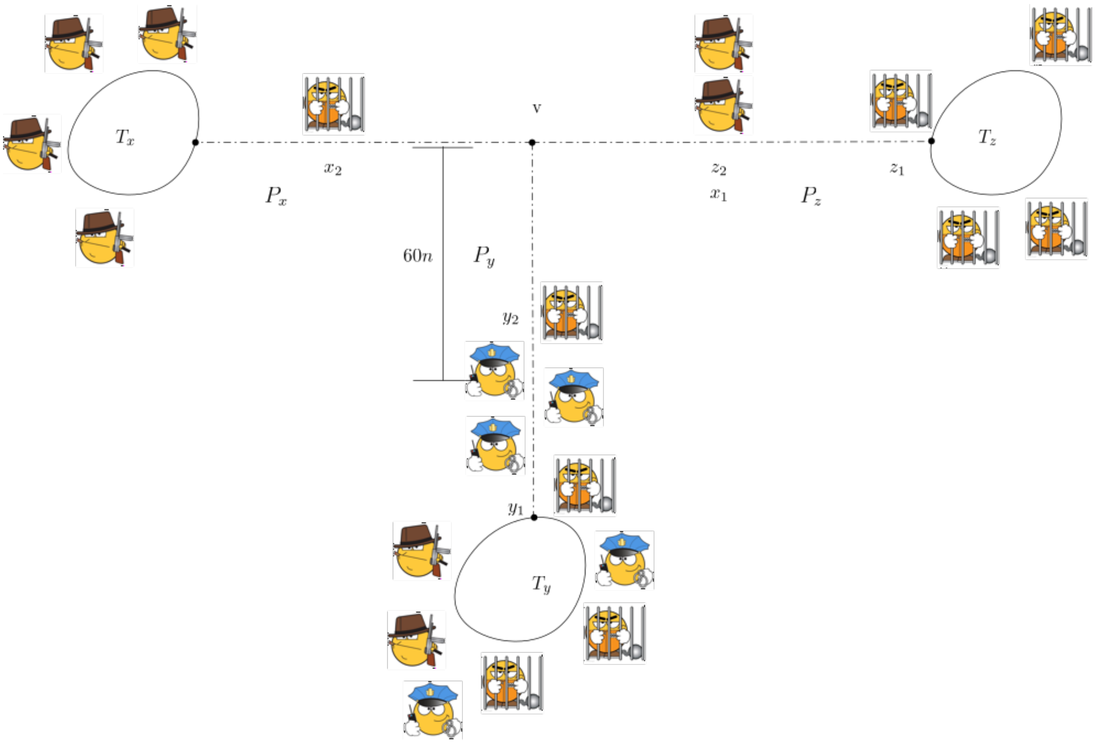


Figure 3.8: Sketch of state 3

- **State 1**: This is the initial state. To win the game, the cops have to move all cops into T_x or T_y to freeze all robbers in X and Y. State 1 describes the state, where the cops are split up and have not started to move towards T_x or T_y together. The following invariants hold in this state:
  1. Neither are all cops in $P_x \cup T_x$ and the closest cop has a distance to v greater than $10n$, nor are all cops in $P_y \cup T_y$ the closest cop has a distance of $10n$ to v.
  2. The robbers x_2, y_2, z_1 and z_2 are frozen on the vertices they were placed on initially.
  3. All robbers in Z are frozen.
  4. The robbers x_1 and y_1 are on the paths P_x and P_y respectively and are at a distance of at most $10n$ to the closest cop, or at v.
  5. x_1 and y_1 are either free, or frozen at the vertex with distance $100n$ to v on P_x or P_y respectively.

  For the invariant 2 and 3, we can just assume that the robbers are frozen no matter what the cops do. At least one robber in X, as well as in Y is free, because the robbers in X and Y play in such a way, that c cops are not enough to freeze all of them (possible because $c_r^{\mathrm{FT}}(T_x) > c$) and not all $c + 1$ robbers can be in T_x or T_y because of invariant 1. Invariants 4 and 5 hold because x_1 moves along P_x and keeps the distance always at $10n$ to the closest cop. The only exception is if x_1 reaches v or x_1 is at the vertex with distance $100n$ to v and potentially frozen. While frozen, x_1 is still at a distance of at most $10n$ to the closest cop. If the closest cop is at distance exactly $5n$ to the frozen x_1(the cop has to be in P_x), one free robber in X frees x_1 in at most n moves and x_1 plays as before. The robbers in Y and y_1 play the same way. If the cops violate invariant 1, we say that we are in state 2.

- **State 2**: We can assume, that all cops are in $P_y \cup T_y$. Here we are in the state, where the cops are on their way towards Y and x_1 can start moving towards z_2 to free him. The following invariants hold:
  1. All cops are in $P_y \cup T_y$ and the cop closest to v has a distance between $10n$ and $60n$ to v.
  2. The robbers x_2, y_2, z_1, z_2 and all robbers in Z are frozen on the vertices they were placed on initially.
  3. y_1 is at distance $10n$ to the closest cop, or frozen at the vertex with distance $100n$ to v.
  4. x_1 is in P_z and his distance to v is by $10n$ shorter than the distance from v to the closest cop in P_y.

  Invariants 2 and 3 are true for the same reasons as explained for state 1. The robber x_1 moves in such a way that invariant 4 is true. There is still at least one free robber in X and one free robber in Y as before. If invariant 1 is violated, we are in state 1, if the closest cop has a distance to v smaller than $10n$. Otherwise, if the distance is larger than $60n$, we are in state 3. In that case x_1 is at the same vertex as z_2 and frees z_2.

- **State 3**: In this state, the cops have committed so far to freeze Y, that z_2 can free Z and return back to his vertex without being disturbed by a cop. The following invariants hold:
  1. All cops are in $P_y \cup T_y$ and the cop closest to v has a distance larger than $10n$ to v.

2 Not all cops are in T_y.

3 x_1 is in P_z and his distance to v is by $10n$ shorter than the distance from v to the closest cop in P_y.

4 If the cop closest to v is at a distance smaller than $60n$, then z_2 stays at distance $50n$ to v. Otherwise, z_2 is $10n$ closer to v than the closest cop.

5 The robbers x_2, y_2, z_1 and all robbers in Z are frozen on the vertices they were placed on initially.

6 y_1 is at distance $10n$ to the closest cop, or frozen at the vertex with distance $100n$ to v.

The robbers x_1, z_2 and y_1 move in such a way, that the invariants 3, 4 and 6 are maintained. This is possible because the robbers are in P_z, while the cops are all in P_y. If the 1 invariant is violated we are back in state 1. Invariant 5 is true for the same reasons as before. From invariant 2 follows that there is at least one free robber in Y and additionally there is a free robber in X as before. If invariant 2 is violated, we can assume that all robbers in Y are frozen. It takes the first cop at least $100n$ moves to get back to v and by that time we can assume that we are in a state that is equivalent to state 1. When the last cop enters T_y, z_2 is at a distance of $90n$ to v. The robber can free z_1 and Z, and can get back to the vertex with distance $50n$ in less than $70n$ moves ($10n$ to get to T_z, $2n$ to unfreeze Z and $50n$ to get back). After $10n$ moves z_1 is freed by z_2 and after another $90n$ moves, he is at distance $10n$ to v. This means that z_1 can stay at a distance of at most $10n$ to the closest cop, after that cop reached v after the at least $100n$ moves. That state is now equivalent with Y being frozen instead of Z.

The game will always be in one of the three states and therefore never end. The robbers win. ∎

Let us assume that for all trees $c_r^{\mathrm{FT}}(T) \leq c + 1$ and that this bound is tight, i.e. there is a tree for which the freeze tag cop number is equal to $c + 1$. Then we know that $c_{3r+2}^{\mathrm{FT}}(T) \leq c + 2$ for all trees according to Theorem 3.3. Using Theorem 3.5 we can construct a tree T' such that $c_{3r+6}^{\mathrm{FT}}(T') > c + 1$. If for the values of r between $3r + 2$ and $3r + 5$, there exists a tree on which $c + 1$ cops win remains to be solved.
Next we will look at graphs with a bounded treewidth. To define the treewidth, we first need to introduce the concept of a *tree decomposition*. Let $G = (V, E)$ be a graph. Then a tree decomposition is a tree with vertices $X_1, \ldots, X_n$ where each X_i is a subset of V and the following conditions are met:

- Every vertex in V is contained in at least one tree vertex.
- For every vertex v in V, the tree vertices containing v form a component.
- For every edge in V, there exists a tree vertex that contains these two vertices.

For a tree decomposition we define the width as the size of the largest X_i minus one. The treewidth of a graph is then the minimal width of a tree decomposition of that graph.
Our goal is to show that for graphs with a fixed treewidth, the freeze tag cop number is still logarithmic in the number of robbers.

Theorem 3.6: *For a graph G of treewidth k, we have $c_r^{FT}(G) \in \mathcal{O}(k \log r)$.*

Proof. We prove the theorem by showing that $c_r^{\mathrm{FT}}(G) \leq c_{\frac{r}{2}}^{\mathrm{FT}}(G) + 2k + 2$. Consider a tree decomposition T of width k with vertices $V_1 \dots V_n$ where $V_i \subseteq V(G)$, $|V_i| = k + 1$ and $V_i \neq V_j$ for every i, j.
We describe a winning strategy for the cops. Let V_x be some vertex of the tree T. We place a cop on each of the vertices in V_x. We placed $k + 1$ cops and the remaining $k + 1 + c_{\frac{r}{2}}^{\mathrm{FT}}(G)$ cops are placed arbitrarily. Now we look at the components of $G - V_x$. If each component contains at most $\frac{r}{2}$ robbers, then the $c_{\frac{r}{2}}^{\mathrm{FT}}(G)$ cops can go from component to component and freeze every robber. In this case the cops win.
Otherwise, there can be at most one component with more than $\frac{r}{2}$ robbers. Let v be a vertex in this component. Consider the component C of $T - V_x$ with a vertex who contains v(note that there is only one such component). Let V_y be the neighbor of V_x in C. Now $k + 1$ of the remaining $k + 1 + c_{\frac{r}{2}}^{\mathrm{FT}}(G)$ cops move each to a different vertex in V_y. Then we repeat the process with V_y instead of V_x until all components contain at most $\frac{r}{2}$ robbers.
Next we show that this happens after finitely many steps. We look at the components of $G - V_y$. If all components contain at most $\frac{r}{2}$ robbers we are finished. Otherwise, the component containing more than $\frac{r}{2}$ is a proper subset of the previous component. That is because the vertices in V_x where guarded, and no robber could leave the component. There is at least one vertex in V_y that is not in V_x and is thereby not part of the component anymore. In each iteration the size of the component containing more than $\frac{r}{2}$ robbers decreases and therefore after finitely many iterations all components contain at most $\frac{r}{2}$ robbers. ∎

4 Graphs with Large Freeze Tag Cop Numbers

In Chapter 2 we looked at several very simple families of graphs and showed that their freeze tag cop number is constant. For trees, we were able to show in Chapter 3, that the freeze tag cop number is logarithmic in the number of robbers. In this chapter we will investigate graphs with a larger freeze tag cop number and will try to come as close to the upper bound from Lemma 2.1 as possible.
First we will look at grid graphs and show that the number of cops necessary is asymptotically equivalent to the square root of the number of robbers.

Theorem 4.1: *On a grid r^2 robbers can win against $\frac{r}{20}$ cops, i.e. $c_{r^2}^{FT} > \frac{r}{20}$.*

Proof. Let G be a grid of size $r \times r$. We describe a winning strategy for the robbers. We place one robber at every vertex, no matter where the cops were placed. Assume that half of the robbers are frozen (if fewer robbers are frozen we pretend that more robbers are frozen). Next we prove the claim that we can find a matching with size at least $\frac{r}{2}$ between vertices with free and frozen robbers.
Case 1: Every row contains a free and a frozen robber.
Then we find a match in every row for a total of r matches.
Case 2: There is a row that contains only free or only frozen robbers.
We can assume, that there is a row that contains only frozen robbers. From this it follows that every column contains a frozen robber. Because only $\frac{r^2}{2}$ robbers are frozen there can be at most $\frac{r}{2}$ columns containing r frozen robbers. This means there are at least $\frac{r}{2}$ columns containing a free robber. In each of these columns we find a match.
Now we look at the subset of these $\frac{r}{2}$ matches where the frozen robber cannot be reached by a cop in his next move. We want to prevent the cops from freezing two robbers on one vertex, which could happen if the frozen robber is near a cop. Every cop can reach at most 5 different vertices in one move. Therefore, our subset has size at least $\frac{r}{2} - 5c = \frac{r}{4}$. In the next 2 turns for every of these pairs the free robber frees the frozen robber and moves back. After two moves there is one robber on every vertex again and the cops can freeze at most $\frac{r}{20} \cdot 2$ robbers in the meantime. This is because a cop can never freeze more than one robber on a single vertex. The total number of frozen robbers is then at most $\frac{r^2}{2} - \frac{r}{4} + \frac{r}{10} < \frac{r^2}{2}$. Because less than half of the robbers are frozen we can repeat this strategy indefinitely and the robbers win. ∎

Next we will generalize the strategy we just used. We can place one robber on every vertex and assume half of them are frozen. Then we need to find a matching between free and frozen robbers. One tool that will help us calculate a lower bound for the size of the matching is the *minimal bisection.*

Definition 4.2: *The* minimum bisection problem *is to partition the set of vertices into two sets in such a way, that the number of edges between the sets is minimal. Additionally, the sizes of the two sets may differ by at most one. The minimal number of edges in this case is called* minimum bisection.

The minimum bisection of a graph together with the maximum degree is enough to calculate a lower bound on the size of the matching. With that lower bound and the maximum degree of a graph we can then determine a lower bound on the cop number for graphs.

Theorem 4.3: *Let G be a graph on r vertices with a maximum degree of d and with a minimal bisection of b. Then $c_r^{FT}(G) \geq \frac{b}{5d+1}$.*

Proof. We describe a strategy for r robbers, to win against $c = \frac{b}{5d+1}$ cops. We place one robber on every vertex and assume that half of the robbers are frozen. It follows, that there are at least b edges, where the robber on one vertex is free and the robber on the other vertex is frozen. Every cop can reach at most $d + 1$ vertices in one move. In total the cops can cover at most $c(d + 1)$ of these edges. Therefore, there are $b - c(d + 1)$ of these edges where the vertex with the frozen robber cannot be reached by a cop in one move. We can greedily find a matching of size $\frac{b-c(d+1)}{2d}$ between vertices with free and frozen robbers. We do this by iteratively choosing an edge and removing all other edges adjacent to the two vertices. We remove less than $2d$ edges in every iteration. Therefore, every two moves at least $\frac{b-c(d+1)}{2d}$ free robbers can free a neighboring frozen robber and not be caught there. Then they return to their vertex where they potentially can be caught. Because the cops can never freeze two robbers on the same vertex, the maximum number of robbers the cops can catch in two moves is $2c$. After two moves, there are at most $\frac{r}{2} - \frac{b-c(d+1)}{2d} + 2c = \frac{r}{2} - \frac{b-c(d+1)+4cd}{2d} = \frac{r}{2} - \frac{b-c(5d+1)}{2d} = \frac{r}{2}$ robbers frozen. Consequentially, the cops will never be able to freeze all robbers and the robbers win. ∎

Our goal is to find graphs for which we get a high lower bound on the cop number, using the formula we just proved in Theorem 4.3. So we are looking for graphs with a large minimal bisection and a small maximum degree. Next, we will show that by applying our strategy for grids to cubes with more dimensions we can construct graphs whose freeze tag cop number is arbitrarily close to being linear in the number of robbers. First we define what we mean by more dimensional cubes.

Definition 4.4: *Let $H = H(k, n)$ be a graph where $V(H) = \{1, \ldots, k\}^n$. Two vertices u, v are adjacent if and only if there exists an $i \in \{1, \ldots, n\}$ such that: $\forall j \in \{1, \ldots, n\}, j \neq i : u_j = v_j$ and $|u_i - v_i| = 1$. We say that H is a k-ary n-dimensional cube.*

Grid graphs, for example, are k-ary 2-dimensional cubes, because they are 2-dimensional and k is their side length. The number of vertices in $H(k, n)$ are k^n and the maximum degree is $2n$. Their minimal bisection is easy to determine, as it is k^{n-1}(see [LC16] without proof). Now we only need to choose the right values for k and n.

Theorem 4.5: *For every $\varepsilon \in \mathbb{R}, \varepsilon > 0$ and $r \in \mathbb{N}_+$ there exist values for r' ($r' \geq r$), k and n, such that $c_{r'}^{FT}(H(k, n)) > r'^{1-\varepsilon}$.*

Proof. We show that the inequality holds for $n = \lceil \frac{2}{\varepsilon} \rceil$, $k = 10n + 1 + r$ and $r' = k^d$ (note that r' are the number of vertices in H). It obviously holds that $r' = k^d \geq r$. The minimum bisection of H is k^{n-1}. Every vertex in H has degree at most $2n$. Together with Theorem 4.3, this yields that $c_{r'}^{\mathrm{FT}}(H) \geq \frac{k^{n-1}}{10n+1}$. Next we show that: $\frac{k^{n-1}}{10n+1} > r'^{1-\varepsilon}$.

$$
\begin{aligned}
&\frac{k^{n-1}}{10n+1} > r'^{1-\varepsilon} \\
\iff &\frac{k^{n-1}}{10n+1} > k^{n-n\varepsilon} \\
\iff &\frac{k^{n-1}}{k^{n-n\varepsilon}} > 10n+1 \\
\iff &k^{n\varepsilon-1} > 10n+1 \\
\iff &k^{\lceil \frac{2}{\varepsilon} \rceil \varepsilon-1} > 10n+1 \\
\Longleftarrow &k^{\frac{2}{\varepsilon}\varepsilon-1} > 10n+1 \\
\iff &k > 10n+1 \\
\iff &10n+1+r > 10n+1 \\
\iff &r > 0
\end{aligned} \tag{4.1}
$$

We have thereby proven that $c_{r'}^{\mathrm{FT}}(H) \geq \frac{k^{n-1}}{10n+1} > r'^{1-\varepsilon}$. ∎

In different words, this means that for every function $f(r) \in o(r)$ we can construct a graph G such that $c_r^{\mathrm{FT}}(G) > f(r) + c(G)$ for large enough values of r. Last but not least, we show that graphs exist for which the freeze tag cop number is exactly linear in the number of robbers. We will look at 3-regular graphs. Their maximum degree is 3 and the minimal bisection for large random 3-regular graphs is about $\frac{1}{10}$ of the number of vertices([LM20]). Therefore, we can show that there are graphs where the freeze tag cop number is linear in the number of robbers.

Theorem 4.6: *For sufficiently large $r \in \mathbb{N}_+$ there exist 3-regular graphs G, such that $c_r^{FT}(G) \geq \frac{r}{160}$.*

Proof. Let G be a 3-regular graph with r vertices and a minimal bisection of size $> \frac{r}{10}$(such a graph exists according to [LM20]). Theorem 4.3 yields that $c_r^{\mathrm{FT}}(G) \geq \frac{\frac{r}{10}}{5 \cdot 3+1} = \frac{r}{160}$. ∎

5 Conclusion

In this thesis, we explored the freeze tag variant of the game cops and robbers. We first related the freeze tag cop number to the cop number. We showed trivial lower and upper bounds that hold on every graph for the freeze tag cop number, $c_r^{\mathrm{FT}}(G)$, with regard to the number of robbers r. We then proved that a constant number of cops suffices to catch any number of robbers on paths, cycles, complete graphs and graphs containing a universal vertex.

The same holds for a subclass of the chordal graphs known as interval graphs. However, it remains an open question whether there are tighter bounds than the trivial ones from Chapter 2 on the freeze tag cop number for chordal graphs in general.

Following that in Chapter 3, we showed for trees, that there is a function $f \in \mathcal{O}(\log r)$ such that $f(r) > c_r^{\mathrm{FT}}(T)$ for every tree T. We also proved that this bound is tight, i.e. that there is no such function in $o(\log r)$. We then extended this result by proving a logarithmic upper bound for graphs with bounded treewidth.

In Chapter 4 we first examined grid graphs and showed that the freeze tag cop number is at least proportional to the square root of the number of robbers. This represents asymptotically faster growth than for any other family of graphs studied up to that point. We then generalized the approach to k-ary n-cubes and showed that we can construct graphs whose freeze tag cop number can get arbitrarily close to being linear in the number of robbers.

Unfortunately we were not able to present a graph whose freeze tag cop number is strictly linear in the number of robbers. But we could show that graphs exist for which this is the case, although we do not know how they look like. Identifying or constructing a group of graphs whose freeze tag cop number is linear in the number of robbers remains an open problem.

The most important question that came up while working on this thesis was whether there exists a constant k such that $k \log(r) c(G) > c_r^{\mathrm{FT}}(G)$ for every graph G. But in the last chapter we could show that this is not the case.

Bibliography

[AF84] M. Aigner and M. Fromme. “A game of cops and robbers”. In: *Discrete Applied Mathematics* Volume 8 (1984), pp. 1–12. ISSN: 0166-218X. DOI: *https://doi.org/10.1016/0166-218X(84)90073-8*.

[FKL12] Alan Frieze, Michael Krivelevich, and Po-Shen Loh. “Variations on cops and robbers”. In: *Journal of Graph Theory* Volume 69 (2012), pp. 383–402.

[Hah07] Ge na Hahn. “Cops, robbers and graphs”. In: *Tatra Mt. Math. Publ* Volume 36 (2007), pp. 163–176.

[HIL24] Pamela E. Harris, Erik Insko, and Florian Lehner. “Tipsy cop and tipsy robber: Collisions of biased random walks on graphs”. In: *Theoretical Computer Science* Volume 997 (2024), p. 114512. ISSN: 0304-3975. DOI: *https://doi.org/10.1016/j.tcs.2024.114512*.

[LC16] Yamin Li and Wanming Chu. “Adjusting Parameters of k-Ary n-Cube to Achieve Better Cost Performance”. In: *2016 IEEE Trustcom/BigDataSE/ISPA*. 2016, pp. 1218–1225. DOI: *10.1109/TrustCom.2016.0197*.

[LM20] Lyuben Lichev and Dieter Mitsche. “On the minimum bisection of random 3-regular graphs”. In: *arXiv preprint arXiv:2009.00598* (2020).