Monotonicity of the cops and robber game for bounded depth treewidth

Isolde Adler
University of Bamberg, Germany

Eva Fluck
RWTH Aachen University, Germany

Abstract

We study a variation of the cops and robber game characterising treewidth, where in each play at most q cops can be placed in order to catch the robber, where q is a parameter of the game. We prove that if k cops have a winning strategy in this game, then k cops have a monotone winning strategy. As a corollary we obtain a new characterisation of bounded depth treewidth, and we give a positive answer to an open question by Fluck, Seppelt and Spitzer (2024), thus showing that graph classes of bounded depth treewidth are homomorphism distinguishing closed.

Our proof of monotonicity substantially reorganises a winning strategy by first transforming it into a pre-decomposition, which is inspired by decompositions of matroids, and then applying an intricate breadth-first 'cleaning up' procedure along the pre-decomposition (which may temporarily lose the property of representing a strategy), in order to achieve monotonicity while controlling the number of cop placements simultaneously across all branches of the decomposition via a vertex exchange argument. We believe this can be useful in future research.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases tree decompositions, treewidth, treedepth, cops-and-robber game, monotonicity, homomorphism distinguishing closure

1 Introduction

Search games were introduced by Parsons and Petrov in [34, 35, 36] and since then gained much interest in many (applied and theoretical) areas of computer science and in discrete mathematics [5, 9, 8, 26, 17, 33, 24, 15, 23, 21]. In search games on graphs, a fugitive and a set of searchers move on a graph, according to given rules. The searchers' goal is to capture the fugitive, and the fugitive tries to escape. Here the interest lies in minimising the resources needed to guarantee capture. Typically this means minimising the number of searchers, but we also seek to bound the number of new placements of searchers. Search games have proven very useful for providing a deep understanding of structural and algorithmic properties of width parameters of graphs, such as treewidth [7, 41], pathwidth [8], cutwidth [28], and directed treewidth [25], treedepth [31], and b-branched treewidth [14, 30].

The crux in relating a given variant of a search game to a width parameter often lies in the question of whether the game is *monotone*, i. e. whether the searchers always have a winning strategy in which a previously cleared area never needs to be searched again – without needing additional resources. Furthermore, monotonicity of a search game provides a polynomial space certificate for proving that determining the winner is in NP.

In their classic paper [41], Seymour and Thomas proved monotonicity of the cops and robber game characterising treewidth. They use a very elegant inductive argument via the dual concept of *brambles*. In this paper we study a variation of this game, where k cops try to capture a robber, but they are limited to making at most q cop placements, for a fixed number $q \in \mathbb{N}$. It is an open question from [13], whether this game is monotone. We give a positive answer to this question.

The notion of *treedepth* was first introduced by Nešetřil and Ossona de Mendes [31]. They exhibit a number of equivalent parameters, and a characterisation by a monotone game is implicitely given. This was subsequently made more explicit in [19]. In [18], a characterisation by a different game called *lifo game* is given for which monotonicity is proven. The game we study can be seen as generalising the monotone game implicit in [31]. However, it is strictly more general and it is not monotone by definition.

Recently, width parameters received a renewed interest in the context of counting homomorphisms and the expressive power of logics [12, 20, 10, 39, 13]. In this context a non-monotone search game characterisation of the width parameter is useful to ensure that there are no graphs of higher width that can be added to the graph class without changing the expressive power [32, 13]. The main obstacle then is to find such a non-monotone characterisation, as the natural characterisation as a search-game of many graph parameters is inherently monotone. *Bounded depth treewidth* and the game studied in this paper were first defined in [13]. An equivalent characterisation of these graph classes by so-called k-pebble forest covers of depth q, which is bounded width tree depth, was already given in [1].

Homomorphism counts. Homomorphism counts are an emerging tool to study equivalence relations between graphs. Many equivalence relations between graphs can be characterized as homomorphism indistinguishability relations, these include graph isomorphism [27], graph isomorphism relaxations [29, 22, 38], cospectrality [11] and equivalence with respect to first-order logic with counting quantifiers [12, 20, 10, 13]. In order to study the expressiveness of such equivalence relations, it is crucial to know under which circumstances distinct graph classes yield distinct equivalence relations. Towards this question one considers the closure of a graph class under homomorphism indistinguishability. Let $\mathcal{F}$ be a graph class. Two graphs G, H are *homomorphism indistinguishable over* $\mathcal{F}$, if for all $F \in \mathcal{F}$ the number of homomorphisms from F to H equals the number of homomorphisms from F to G. The graph class $\mathcal{F}$ is *homomorphism distinguishing closed*, if for every graph $F \notin \mathcal{F}$ there exists two graphs G, H, that are homomorphism indistinguishable over $\mathcal{F}$ but that do not have the same number of homomorphisms from F. It has been conjectured by Roberson [37], that all graph classes that are closed under taking minors and disjoint unions are homomorphism distinguishing closed. So far the list of graph classes for which the conjecture is confirmed is short: the class of all planar graphs [37], graph classes that are essentially finite [40], the classes of all graphs of tree width at most $k-1$ [32] and the classes of all graphs of tree depth at most q [13]. The latter two results rely on characterisations of the graph classes in terms of non-monotone cops-and-robber games. We study *bounded depth treewidth*, which bounds both the width and the depth simultaneously. We give a game characterisation that does not rely on monotonicity, and as a consequence we obtain that graph classes of bounded depth treewidth are also homomorphism distinguishing closed.

Our contribution. We show the following (cf. Theorem 27).

Fix integers $k, q \geq 1$. For every graph G the following are equivalent.

- *G has a tree decomposition of width at most $k-1$ and depth at most q.*
- *k cops have a monotone winning strategy in the cops and robber game on G with at most q placements.*
- *k cops have a winning strategy in the cops and robber game on G with at most q placements.*

The equivalence between the last two statements gives a positive answer to an open question from [13]. Our proof of monotonicity gives both a proof of monotonicity for the classical cops and robber game characterising treewidth as well as for the game characterising treedepth as special cases. As a corollary, we obtain the following (cf. Theorem 19).

Let $k, q \geq 0$ be integers. The class of graphs having a tree decomposition of width at most $k-1$ and depth at most q is homomorphism distinguishing closed.

Proof techniques. In contrast to the proof of monotonicity of the classic cops and robber game [41], our proof does not use a dual concept such as brambles. Instead, we modify a (possibly non-monotone) winning strategy, turning it first into what we call a *pre-decomposition*, and then cleaning it up while keeping track of width and depth, thus finally transforming the pre-decomposition into a monotone winning strategy. Our concept of pre-decomposition is inspired by decompositions of matroids and it is based on ideas from [3, 6]. Our cleaning-up technique is similar to the proof of monotonicity of the the game for b-branching treewidth [30]. However, the cleaning-up technique in [30] loses track of the number of cop placements, as local modifications may have non-local effects that are not controlled. We need to keep track in order to control the depth.

This poses a major challenge which we resolve in our proof by a fine grain cleaning-up technique in our pre-decompositions based on a careful decision of which vertices to 'push up and through the tree' and which to 'push down'. The vertices 'pushed up' may have an effect on the part of the pre-decomposition that was processed in previous steps, which we manage to control by a vertex exchange argument. Additionally we keep track of how the first modification at some node in the pre-decomposition relates back to the original strategy. We believe that our techniques will also help in future research.

Our proof provides an independent proof of monotonicity of the classic game characterising tree-width as a special case, namely when q is greater than or equal to the number of vertices of the graph. Our proof strategy is entirely different, as it does not use an equivalence via a dual object such as brambles. Instead, we provide a more direct transformation of a (possibly non-monotone) winning strategy.

Further related research. Search games are used to model a variety of real-world problems such as searching a lost person in a system of caves [34], clearing contaminated tunnels [26], searching environments in robotics [23], and modelling bugs in distributed environments [17], cf. [16] for a survey.

There is a fine line between games that are monotone and those that are not. For example, the marshalls and robber game played on a hypergraph is a natural generalisation of the cops and robber game, it is related to hypertree-width, but it is not monotone [2]. However, the monotone and the non-monotone variants are strongly related [4] to eachother.

Structure of the paper. In Section 2 we fix our notation and we define tree decompositions of bounded depth and width. Section 3 introduces pre-tree decompositions, relevant properties, and establishes a relation to tree decompositions. The game is introduced in Section 4, and in Section 5 we give the main proofs, showing how to make a strategy tree exact while maintaining the bounds on width and depth. The insights given by our answer to the open question in the area of homomorphism counts are briefly discussed in Section 6.

2 Preliminaries

Sets and partitions Let A be a finite set. We write 2^A to denote the power-set of A and, for $k \in \mathbb{N}$, $\binom{A}{\leq k}$ to denote all subsets of A of size $\leq k$. $\mathrm{Part}(A)$ is the set of all *partitions* of

A, where we allow partitions to contain multiple (but finite) copies of the empty set. Let $P = \{X_1, \ldots, X_d\} \in \mathrm{Part}(A)$ and $F \subseteq A$. For $i \in [d]$, the partition

$$P_{X_i \to F} \coloneqq \{X_1 \setminus F, \ldots, X_{i-1} \setminus F, X_i \cup F, X_{i+1} \setminus F, \ldots, X_d \setminus F^i\},$$

is called the *F-extension in X_i of P*. A function $w\colon \mathrm{Part}(A) \to \mathbb{N}$ is *submodular* if, for all $P, Q \in \mathrm{Part}(A)$, for all sets $X \in P$ and $Y \in Q$ with $X \cup Y = A$, it holds that

$$w(P) + w(Q) \geq w(P_{X \to \overline{Y}}) + w(Q_{Y \to \overline{X}}).$$

Let $f\colon A \to B$ be a function and $C \subseteq A$. By $f|_C$ we denote the restriction of f to C, i. e. $f|_C\colon C \to B$ and $f|_C(c) = f(c)$, for all $c \in C$.

Graphs A graph G is a tuple $(V(G), E(G))$, where $V(G)$ is a finite set of vertices and $E(G) \subseteq {V(G) \atop \leq 2}$ is the set of edges. We usually write uv or vu to denote the edge $\{u, v\} \in E(G)$. If G is clear from the context we write V, E instead of $V(G), E(G)$. We write $I(G)$ to denote the set of isolated vertices in G, that is for every $v \in I(G)$, there is no $u \in V(G)$ with $u = v$ and $uv \in E(G)$. By G° we denote the graph obtained from G by adding all self-loops that are not present in G, that is $V(G^\circ) \coloneqq V(G)$ and $E(G^\circ) \coloneqq E(G) \cup \{vv \mid v \in V(G)\}$. For $v \in V$ we write $E_G(v) \coloneqq \{uv \mid uv \in E(G)\}$ for the edges incident to v.

A *tree* is a graph where any two vertices are connected by exactly one path. A *rooted tree* (T, r) is a tree T together with some designated vertex $r \in V(T)$, the *root* of T. By $L(T)$ we denote the set of all *leaves* of T, that is $L(T) \coloneqq \{v \in V(T) \mid |N(v)| = 1\}$. All vertices that are not leafs are called *inner vertices*.

At times, the following alternative definition is more convenient. We can view a rooted tree (T, r) as a pair $(V(T), \ )$, where is a partial order on $V(T)$ and for every $v \in V(T)$ the elements of the set $\{u \in V(T) \mid u \quad v\}$ are pairwise comparable: The minimal element of is precisely the root of T, and we let $v \quad w$ if v is on the unique path from r to w. Let $t, t \ \in V(T)$, we call $t^* \in V(T)$ the *greatest common ancestor* if $t^* \quad t, t$ but for all $t \ \in V(T)$ with $t^* \prec t$ either $t \quad t$ or $t \quad t$.

▶ **Definition 1.** *Let G be a graph, let (T, r) be a rooted tree and let $\beta\colon V(T) \to 2^{V(G)}$ be a function from the nodes of T to sets of vertices of G. We call (T, r, β) a* tree decomposition *of G, if*
(T1) $_{t \in V(T)} G[\beta(t)] = G$*, and*
(T2) *for every vertex $v \in G$, the graph $T_v \coloneqq T[\{t \in V(T) \mid v \in \beta(t)\}]$ is connected.*
The sets $\beta(t)$ are called the bags *of this tree decomposition.*

The *width* of a tree decomposition (T, r, β) is $\mathrm{wd}(T, r, \beta) \coloneqq \max_{t \in V(T)} |\beta(t)| - 1$, the *depth* is $\mathrm{dp}(T, r, \beta) \coloneqq \max_{\ \in L(T)} | \ _{t \in P} \ \beta(t)|$, where P is the path from to the root. The *tree width* of a graph G is the minimum width of any tree decomposition of G, the *tree depth* of a graph G is the minimum depth of any tree decomposition (see cf [13]). For $k, q \geq 1$ we define the class $\mathcal{T}_q^k$ to be all graphs that have a tree decomposition (T, r, β) with $\mathrm{wd}(T, r, \beta) \leq k - 1$ and $\mathrm{dp}(T, r, \beta) \leq q$. The following lemma is a well known consequence from (T2).

▶ **Lemma 2.** *Let G be a graph and $U \subseteq V(G)$ connected in G. Let (T, r, β) be a tree decomposition of G, then $T_U \coloneqq T[\{t \in V(T) \mid U \cap \beta(t) = \emptyset\}]$ is connected.*

3 Pre-tree decomposition, exactness and submodularity

Here we consider a definition of tree decompositions that is inspired by matroid tree decompositions. We relax this definition into what we call a pre-tree decomposition.

▶ **Definition 3.** *Let* $G = (V(G), E(G))$ *be a graph. Let* $X \subseteq E(G)$. *We define* $\delta(X) := \{v \in V(G) \mid \exists e \in X, e \ \in E(G) \setminus X, v \in e \cap e \ \}$. *Let* π *be a partition of* $E(G)$. *We define*

$$\delta(\pi) := \{v \in V(G) \mid \exists X \in \pi, v \in \delta(X)\}.$$

A tuple (T, r, β, γ), *where* (T, r) *is a (rooted) tree,* $\beta\colon V(T) \to 2^{V(G)}$ *and* $\gamma\colon \overrightarrow{E(T)} \to 2^{E(G)}$, *is a* (rooted) pre-tree decomposition *if:*

(PT1) $\beta(r) = \emptyset$ *and for every connected component* C *of* G, *there is a child* c *of the root with* $\gamma(r, c) = E(C)$.

(PT2) *For every leaf* $\ \in L(T)$ *with neighbour* t, *it holds that* $|\gamma(t, \ )| \leq 1$.

(PT3) *For every internal node* $t \in V(T) \setminus L(T)$, *we define* $\pi_t := (\gamma(t, t_1), \ldots, \gamma(t, t_d))$, *where* $N(t) = \{t_1, \ldots, t_d\}$ *an arbitrary enumeration of the neighbours of* t, *and for a leaf* $\ \in L(T)$ *with parent* p *we define* $\pi \ := (\gamma(\ell, p), \overline{\gamma(\ell, p)})$. *For every* $t \in V(T)$, *the tuple* π_t *is a partition of* $E(G)$ *and* $\beta(t) \supseteq \delta(\pi_t)$.

(PT4) *For every edge* $st \in E(T)$, *it holds that* $\gamma(s, t) \cap \gamma(t, s) = \emptyset$.

We call an edge $st \in E(T)$ exact *if* $\gamma(s, t) \cup \gamma(t, s) = E(G)$, *we call* (T, r, γ, β) exact, *if every edge is exact and* $\beta(t) = \delta(\pi_t)$, *for all* $t \in V(T)$. *We call* $\beta(t)$ *the* bag *at node* t *and* $\gamma(s, t)$ *the* cone *at edge* st.

▶ **Observation 4.** *Let* $(T \ , r)$ *be a subtree of* (T, r) *with the same root. If all edges in* T *are exact then* $\{\gamma(t, \ ) \mid \ \in L(T \ ), t \text{ parent of } \ \}$ *is a partition of* $E(G)$.

Similar to the definition of width and depth for tree decompositions we define the width and depth of a pre-tree decomposition.

▶ **Definition 5.** *The* width *of a partition* π *of the edges of a graph is*

$$\mathrm{wd}(\pi) := |\delta(\pi)|.$$

The width *of a pre-tree decomposition is*

$$\mathrm{wd}(T, r, \beta, \gamma) := \max_{t \in V(T)} |\beta(t)| - 1.$$

The depth *of a rooted pre-tree decomposition is*

$$\mathrm{dp}(T, r, \beta, \gamma) := \max_{t \in V(T)} \sum_{s \in P_t \setminus \{r\}} |\beta(s) \setminus \beta(p_s)|,$$

where P_t *is the unique path from the root* r *to* t *and* p_s *is the parent of* s.

The reader may note that the width of a Pre-tree decomposition only gets smaller if one sets $\beta(t) := \delta(\pi_t)$, for all nodes $t \in V(T)$, but the depth can get larger. We show that the width of a partition of the edges as defined above is submodular. We need this property to show that our main construction does not enlarge the width of the pre-tree decomposition.

▶ **Lemma 6.** *For every graph* G, wd *is submodular.*

Proof. Let $P = \{X_1, \ldots, X_d\}, Q = \{Y_1, \ldots, Y_d\} \in \mathrm{Part}(E(G))$. We prove that

$$\mathrm{wd}(P) + \mathrm{wd}(Q) \geq \mathrm{wd}(P_{X_1 \to \overline{Y_1}}) + \mathrm{wd}(Q_{Y_1 \to \overline{X_1}}),$$

which is enough to prove the lemma by symmetry.

If $X_1 = E(G)$, then $P = P_{X_1 \to \overline{Y_1}}$ and $Q = Q_{Y_1 \to \overline{X_1}}$, thus the lemma holds.

If $X_1 = \emptyset$, then $Q_{Y_1 \to \overline{X_1}} = (E(G), \emptyset, \dots, \emptyset)$ and thus $\mathrm{wd}(Q_{Y_1 \to \overline{X_1}}) = 0$. Furthermore $\delta(P_{X_1 \to \overline{Y_1}}) \subseteq \delta(P) \cup \delta(Y_1) \subseteq \delta(P) \cup \delta(Q)$ and thus $\mathrm{wd}(P_{X_1 \to \overline{Y_1}}) \leq \mathrm{wd}(P) + \mathrm{wd}(Q)$, thus the lemma holds.

If $Y_1 = E(G)$ and $Y_1 = \emptyset$ the lemma holds analogously.

Thus let $\emptyset = X_1, Y_1 = E(G)$. Trivially we get that $\delta(P_{X_1 \to \overline{Y_1}}) \subseteq \delta(P) \cup \delta(Y_1)$ and $\delta(Q_{Y_1 \to \overline{X_1}}) \subseteq \delta(Q) \cup \delta(X_1)$. Assume there exists some $v \in \delta(P_{X_1 \to \overline{Y_1}}) \setminus \delta(P)$, then $v \in \delta(Y_1)$ and thus $v \in \delta(Q)$. Furthermore we get that $E(v) \cap X_1 = \emptyset$ and thus $E(v) \subseteq Y_1 \cup \overline{X_1}$. But then $v \notin \delta(Q_{Y_1 \to \overline{X_1}})$. Analogously we can show that $\delta(Q_{Y_1 \to \overline{X_1}}) \setminus \delta(Q) \quad \cap \delta(P_{X_1 \to \overline{Y_1}}) = \emptyset$. Thus all in all every vertex that is newly introduced to one of $\delta(P_{X_1 \to \overline{Y_1}}), \delta(Q_{Y_1 \to \overline{X_1}})$ is removed from the other and therefore the lemma holds. ◀

The next lemma shows that a pre-tree decomposition of a graph G is indeed a relaxation of a tree decomposition of G. If every edge is exact and all bags are exactly the boundary of the partition then we can construct a tree decomposition. We need to start with a pre-tree decomposition of the graph G° with all self-loops added to ensure that every non-isolated vertex does appear in some bag and that the components correspondent to isolated vertices are covered by the pre-tree decomposition. On the other hand we can transform a tree decomposition into a pre-tree decomposition, by copying the tree decomposition of each connected component of G and adding leafs that correspond to the edges of G°.

▶ **Lemma 7.** *Let $k, q \geq 1$. Let $G = (V, E)$ be a graph. Any tree decomposition of G of width $\leq k - 1$ and depth $\leq q$ gives rise to an exact pre-tree decomposition of G° of width $\leq k - 1$ and depth $\leq q$ and vice versa.*

Proof. Let (T, r, β, γ) be an exact pre-tree decomposition of G° of width $\leq k$ and depth $\leq q$. We define $\beta : V(T) \to 2^{V(G)}$ as follows

$$\beta\ (t) := \begin{array}{ll} \{v\} & \text{if } t \in L(T), r \text{ parent of } t \text{ and } \gamma(r, t) = \{vv\}, \\ \beta(t) & \text{otherwise.} \end{array}$$

▷ Claim 8. $(T, \beta\ )$ is a tree decomposition of width $\leq k$ and depth $\leq q$.

Proof. From (PT1), (PT2) and Observation 4 we get that for every edge $uv \in E(G^\circ)$ there is some leaf with parent p and $\gamma(p, \ ) = \{uv\}$. Thus if $u = v$, then $\beta\ (t) = \{v\}$ and thus $vv \in E(G[\beta\ (t)])$. Otherwise $uu, vv \in E(G^\circ) \setminus \{uv\}$ and thus $u, v \in \beta\ (t)$ and $uv \in E(G[\beta\ (t)])$. All in all we get that (T1) holds.

Assume there exists a $v \in V(G)$ such that T_v is not connected. Let T_1, T_2 be two disjoint connected components of T_v and let $P = t_1, \dots, t_a$ be the shortest T_1-T_2-path in T. Then $v \notin \delta(\gamma(t_1, t_2)) \subseteq \delta(\pi_{t_2})$ and thus $E(v) \cap \gamma(t_1, t_2) = \emptyset$. As all edges in P are exact it holds that $\gamma(t_1, t_2) \supseteq \gamma(t_a, s)$, for all $s \in N(t_a) \setminus \{t_{a-1}\}$. And thus it holds that $E(v) \cap \gamma(t_a, s) = \emptyset$ and $E(v) \subseteq \gamma(t_a, t_{a-1})$. This contradicts $v \in \ _{s \in N(t_a)} \delta(\gamma(t_a, s))$. Therefore (T2) also holds and $(T, \beta\ )$ is a tree decomposition.

The width and depth are obvious as $k, q \geq 1$. ◁

Now let (T, r, β) be a tree decomposition of G of width $\leq k$ and depth $\leq q$. W.l.o.g. β is *tight*, that is for all $t \in V(T)$ and $v \in \beta(t)$, that $(T, r, \beta\ )$, where $\beta\ (t) := \beta(t) \setminus \{v\}$ and $\beta\ (s) = \beta(s)$, for all $s \in V(T) \setminus \{t\}$, is not a tree decomposition of G. We construct a new tree T with root r and functions $\beta : V(T\ ) \to 2^{V(G)}$, $\gamma \colon \overrightarrow{E(T\ )} \to 2^{E(G^\circ)}$, $f \colon V(T\ ) \setminus (L(T\ ) \cup \{r\ \}) \to V(T)$ as follows. Let C be a connected component of G and let $V_C := \{t \in V(T) \mid V(C) \cap \beta(t) = \emptyset\}$. By Lemma 2 V_C is connected. If C is an isolated vertex v, then

$V_C = \{t\}$, for some $t \in V(T)$. We add a new node t_v to T and connect it to the root. We set $\beta\ (t_v) = \emptyset$, $\gamma(r\ , t_v) = \{vv\}$ and $\gamma(t_v, r\ ) = E(G^\circ) \setminus \{vv\}$. Otherwise let T_C be a copy of the subtree induced by V_C with root r_C and vertices V_C^* and $f|_{V_C^*} \colon V_C^* \to V_C$ the natural bijection between the copies and their originals. We attach r_C to the root r . For every $v \in V(C)$, there is some $t_v \in V_C$ such that $v \in \beta(t_v)$, as C is not an isolated vertex. We add a new leaf t_v that we attach to $f|^{-1}_{V(T_C)}(t_v)$ and set $\beta\ (t_v) = \{v\}$, $\gamma(f|^{-1}_{V_C^*}(t_v), t_v) = \{vv\}$ and $\gamma(t_v, f|^{-1}_{V_C^*}(t_v)) = E(G^\circ) \setminus \{vv\}$. For every $e \in E_G(C)$ there is some $t_e \in V_C$ such that $e \subseteq \beta(t_e)$. We add a new leaf t_e that we attach to $f|^{-1}_{V_C^*}(t_e)$ and set $\beta\ (t_e) = e$, $\gamma(f|^{-1}_{V_C^*}(t_e), t_e) = \{e\}$ and $\gamma(t_e, f|^{-1}_{V_C^*}(t_e)) = E(G^\circ) \setminus \{e\}$. For every node $t \in V_C^*$ with parent p we add all edges $e \in E_G(C)$, where t_e is a descendant of t, and all self-loops $vv \in E_{G^\circ}(C)$, where t_v is a descendant of t, to $\gamma(p, t)$. Furthermore we set $\gamma(t, p) := E(G^\circ) \setminus \gamma(p, t)$ and $\beta\ (t) := \delta(\pi_t) \subseteq \beta(f(t)) \cap V(C)$. By tightness of β there is some $v \in \beta(f(\ ))$ such that $T_v = \{f(\ )\}$, for every $\ \in L(T_C)$, thus no leaf of T_C is a leaf in T , thus $(T\ , r\ , \beta\ , \gamma)$ satisfies (PT2). (PT1), (PT3) and (PT4) hold by construction. Furthermore every edge is exact by construction. Thus $(T\ , r\ , \beta\ , \gamma)$ is an exact pre-tree decomposition of G°.

The width is obvious as every bag in β is a subset of some bag in β. To see that the depth bound also holds we observe two things. For every leaf $\ \in L(T\ )$ with parent p we get that $\beta\ (\ ) \setminus \beta\ (p) = \emptyset$. For every inner node $t \in V(T\ ) \setminus L(T\ )$ with parent p we get that $\beta\ (t) \setminus \beta\ (p) \subseteq \beta(f(t))$ and, if $p = r$, $\beta\ (t) \setminus \beta\ (p) \subseteq \beta(f(t)) \setminus \beta(f(p))$, by the tightness of β. ◀

The next observation can be seen by following the same arguments as in the proof of Lemma 7.

▶ **Observation 9.** *Let (T, r, β, γ) be a pre-decomposition of some graph G. Let $(T\ , r)$ be a subtree of (T, r) with the same root. If all edges in T are exact then, for every $v \in V(G)$, the set $\{t \in V(T\ ) \mid v \in \delta(\pi_s)\}$ is connected in T . In particular every $t \in V(T\ )$ satisfies*

$$\sum_{s \in P_t \setminus \{r\}} |\delta(\pi_s) \setminus \delta(\pi_{p_s})| = |\bigcup_{s \in P_t} \delta(\pi_s)|.$$

We conclude this section with an observation about the cones along a path of exact edges. It is a direct consequence of exactness and the fact that the cones incident to a vertex form a partition of the edges.

▶ **Observation 10.** *Let (T, r, β, γ) be a pre-decomposition of some graph G. Let $P = t_1, \ldots, t$ a path in T, such that every edge $t_i t_{i+1}$, for $i \in [\ -1]$, is exact. Then it holds that $\gamma(t_1, t_2) \supseteq \gamma(t_2, t_3) \supseteq \ldots \supseteq \gamma(t_{\ -1}, t\ )$.*

4 The game

In the cops-and-robber game on a graph G, the cops occupy sets X of at most k vertices of G, and the robber moves on edges of G. In order to make the rules precise, we need *edge components* of G that arise when the cops are blocking a set X.

▶ **Definition 11.** *Let $G = (V, E)$ be a graph and $X \subseteq V$. We let* the edge component graph *of G with respect to X be the graph G^X obtained as the disjoint union of the following graphs. (In order to make all graphs disjoint we introduce copies of vertices where needed.)*

- *For every $uv \in E(G[X])$, the graph $G_{uv} := (\{u, v\}, \{uv\})$, and*

- *for every connected component C of $G\backslash X$, the graph G_C, with $V(G_C) := V(C) \cup N_G(V(C))$ and $E(G_C) := E(C) \cup E(V(C), X)$, where $E(V(C), X)$ is the set of edges of G incident to both a vertex of C and a vertex in X.*

The reader may note that G^X may contain multiple copies of the vertices in X, but exactly one copy of each edge in G.

▶ **Observation 12.** *There is a natural bijection $\Psi \colon E(G^X) \to E(G)$ between the edges of G^X and the edges of G.*

▶ **Definition 13** (q-placement k-cops-and-robber game)**.** *Let G be a graph and let $k, q \geq 1$. The* q-placement k-cops-and-robber game $\mathrm{CR}_q^k(G)$ *is defined as follows:*

We have a counter j that indicates how many times cops are placed on the graph. If G does not contain any edges the cop player wins immediately.

- *We initialize the counter $j = 0$.*
- *The cop positions are sets $X \in V(G)^{\leq k}$.*
- *The robber position is an edge $uv \in E(G)$.*
- *The initial position (X_0, u_0v_0) of the game is $X_0 = \emptyset$ and $u_0v_0 \in E(G)$, thus the game starts with no cops positioned on G and the robber on an arbitrary edge in a connected component of G of his choice.*
- *For $X \subseteq V(G)$ and $uv \in E(G)$, we write $\gamma_{uv}^X := \Psi(E(C))$ for the component C of the graph G^X, such that $uv \in E(\gamma_{uv}^X)$. Thus if the cops are at positions X and robber at an edge uv we write (X, γ_{uv}^X) for the position of the game.*
- *In round i the cop-player can move from the set X_{i-1} to a set X_i, if $X_i \subseteq X_{i-1}$, or if $j < q$, the cop-player can pick a vertex $v \in V(G)$ and play to $X_i := X_{i-1} \cup \{v\}$ and increase j by one.*
- *In round i the robber-player can move along a path with no internal vertex in $X_{i-1} \cap X_i$. Thus the robber-player can move to some edge u_iv_i, such that the edge $\Psi^{-1}(u_iv_i)$ is in a connected component of G^{X_i} that is contained in $\Psi^{-1}(\gamma_{u_{i-1}v_{i-1}}^{X_{i-1} \cap X_i})$ via a path $p = w_1, \dots, w$ where $\{w_1, w_2\} = \{u_{i-1}, v_{i-1}\}$ and $\{w_{-1}, w\} = \{u_i, v_i\}$ and $\{w_2, \dots, w_{-1}\} \cap X_i \cap X_{i-1} = \emptyset$.*
- *The cop-player wins in round i, if $\{u_i, v_i\} \subseteq X_i$, and we say the cop-player* captures *the robber in round i. The robber-player wins if the cop-player has not won and $j = q$.*

If we further restrict the movement of the cops such that always $\gamma_{u_{i-1}v_{i-1}}^{X_{i-1}} \supseteq \gamma_{u_{i-1}v_{i-1}}^{X_{i-1} \cap X_i}$ holds, we write $\text{mon-CR}_q^k(G)$ *and call the game* monotone q-placement k-cops-and-robber game.

The game played on the graph G° corresponds to the game on G, where the robber can hide both inside a vertex or an edge. It is easy to see that this does not benefit the robber player, that is he wins the game $\mathrm{CR}_q^k(G)$ if and only if he wins the game $\mathrm{CR}_q^k(G^\circ)$, as the components that are reachable by the robber are essentially the same. In [13], the authors introduce a cops-and-robber game, where the robber can only hide in the vertices. Again this does not pose a restriction for the robber with the same argument as above. There is a tight connection between the cops-and-robber game defined above and tree decompositions of graphs.

▶ **Lemma 14** ([13])**.** *Let G be a graph and $k, q \in \mathbb{N}$. The cop player wins* $\text{mon-CR}_q^k(G)$ *if and only if $G \in \mathcal{T}_q^k$.*

Towards strengthening the above connection to also include the non-monotone game we first introduce how to construct a pre-tree decomposition from a winning strategy of the cop player.

▶ **Definition 15** (strategy tree)**.** *Let G be a graph without isolated vertices and let $k, q \in \mathbb{N}$. Let $\sigma\colon V(G)^{\leq k} \times E(G) \to V(G)^{\leq k}$ a cop strategy such that that for all $X \in V(G)^{\leq k}$, for all $uv \in E(G)$ and for all $u\ v\ \in \gamma^X_{uv}$ we have that $\sigma(X, uv) = \sigma(X, u\ v\ )$. We write $\sigma(X, \gamma^X_{uv})$ instead of $\sigma(X, uv)$.*

The strategy tree *of σ is a pre-tree decomposition (T, r, β, γ), inductively defined as follows:*

- *$\beta(r) = \emptyset$,*
- *for every connected component C of G, there is a child c of the root r and $\gamma(r, c) = E(C)$,*
- *for every node $t \in V(T) \setminus \{r\}$ with parent $s \in V(T)$,*
  - *if the robber player is caught, we set $\beta(t) = e$, where $\gamma(s,t) = \{e\}$,*
  - *else $\beta(t) = \sigma(\beta(s), \gamma(s,t))$ and*
  - *for every connected component C of $G^{\beta(t)}$, that has a non-empty intersection with $\Psi^{-1}(\gamma(s,t))$, there is a child c of t and $\gamma(t,c) = \Psi(E(C))$,*
  - *$\gamma(t,s) \coloneqq E(G) \setminus {}_{c \text{ child of } t}\, \gamma(t,c)$, if $t \notin L(T)$, and*
  - *$\gamma(t,s) \coloneqq E(G) \setminus \gamma(s,t)$, if $t \in L(T)$.*

We call $t \in V(T)$ a branching node *if the cop player placed a new cop incident to the robber escape space.*

Observe that if $t \in V(T)$ is a leaf, then the robber is captured and the depth of (T, r, β, γ) is $\leq q$ if and only if σ is a winning strategy in $\mathrm{CR}^k_q(G)$.

Note that w.l.o.g. every child of the root is a branching node, as the cop player w.l.o.g. only plays positions that are inside the component the robber chose in the first round. If the game is played on G°, then every branching node that does not correspond to the placement of a cop onto an isolated vertex has more than one child. We observe that the monotone moves of the cop player correspond to the exact edges in the strategy tree.

▶ **Observation 16.** *For edge $st \in E(T)$, where s is the parent of t it holds that the move $\sigma(\beta(s), \gamma(s,t))$ is monotone if and only if st is exact. Moreover, if st is not exact, then $\beta(t) \quad \beta(s)$.*

The following two observations about the self-loops in the graph G° are key to prove the construction in the next section does not enlarge the depth of the pre-tree decomposition.

▶ **Observation 17.** *When considering the game on G°, all self-loops vv incident to $\beta(s)$ are either contained in $\gamma(s,p)$ or there is a child c of s such that $\gamma(s,c) = \{vv\}$.*

▶ **Observation 18.** *Let $v \in V(G)$ be a non-isolated vertex and the game played on G°. A node s has a child c with $\gamma(s,c) = \{vv\}$, for some self-loop vv if and only if s is a branching node and v is the vertex the cops picked.*

5 Making a strategy tree exact

Our goal is to prove the following theorem.

▶ **Theorem 19.** *Let $G = (V(G), E(G))$ be a graph, let $k, q \geq 1$ and let (T, r, β, γ) be a strategy tree for some cop strategy $\sigma\colon V(G^\circ)^k \times E(G^\circ) \to V(G^\circ)^k$. If σ is a winning strategy in $\mathrm{CR}^k_q(G^\circ)$, then there is a tree decomposition of G with width $\leq k$ and depth $\leq q$.*

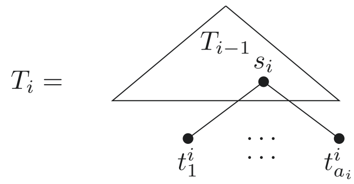

Figure 1 The subtree T_i appearing in the construction.

To prove this we construct an exact pre-tree decomposition of G° from the strategy tree, starting at the root r and traversing the tree nodes in a breadth-first-search. We then use Lemma 7 to get the desired tree decomposition. When we consider a node we change the pre-tree decomposition so that all incident edges are exact afterwards. Note that by the choice of the traversal we only need to consider outgoing edges.

The construction. Let (T, r, β, γ) be the pre-tree decomposition of G° from a winning strategy as in Theorem 19. Let $s_1, \dots, s_{n_T}$ be an order of the nodes of T in bfs where $s_1 = r$. Let $\beta_0 \coloneqq \beta$ and $\gamma_0 \coloneqq \gamma$. We construct a sequence $(T, r, \beta_0, \gamma_0), \dots, (T, r, \beta_{n_T}, \gamma_{n_T})$ of pre-tree decompositions, such that $(T, r, \beta_{n_T}, \gamma_{n_T})$ is exact. We say s_i is *considered in step i*. Let

$$T_i \coloneqq T[\{s_1, \dots, s_i\} \cup N_T(\{s_1, \dots, s_i\})].$$

See Figure 1 for an illustration of T_i. (It will become clear that this is the subtree of all nodes where the pre-tree decomposition is modified in or before step i. We also point out that edges from T_i to $T \setminus T_i$ may become non-exact during our modification process.)

If s_i is a leaf, there are no outgoing edges that are not exact, and we set $\beta_i \coloneqq \beta_{i-1}$ and $\gamma_i \coloneqq \gamma_{i-1}$. Otherwise let $t_1^i, \dots, t_{a_i}^i \in N_T(s_i)$ be all children of s_i.

- We pick pairwise disjoint $F_1^i, \dots, F_{a_i}^i \subseteq E(G^\circ)$, with

  $$F_j^i \subseteq \overline{\gamma_{i-1}(t_j^i, s_i)} \cap \overline{\gamma_{i-1}(s_i, t_j^i)},$$

  such that the partition π^* that results from taking the F_j^i-extensions in $\gamma_{i-1}(s_i, t_j^i)$ (in arbitrary order) has the minimum size boundary. If there are multiple optimal choices for $F_1^i, \dots, F_{a_i}^i$ we select the one that minimizes the size of $\quad_{j \in [a_i]} F_j^i$, if there are still several options we break ties arbitrarily.
- Let $F^i \coloneqq \quad_{j \in [a_i]} F_j^i$ and $F_j^{*i} \coloneqq \quad \overline{\gamma_{i-1}(t_j^i, s_i)} \cap \overline{\gamma_{i-1}(s_i, t_j^i)} \quad \cup F^i \quad \setminus F_j^i$.

  For every $p \in V(T_i)$ with child c we set

$$\gamma_i(p, c) \coloneqq \begin{cases} \quad \gamma_{i-1}(s_i, t_j^i) \setminus F^i \quad \cup F_j^i & \text{if } (p, c) = (s_i, t_j^i), \text{ for some } j \in [a_i], \\ \gamma_{i-1}(p, c) \setminus F_j^{*i} & \text{if } t_j^i = p, \text{ for some } j \in [a_i], \\ \gamma_{i-1}(p, c) \cup F^i & \text{if } c \quad s_i, \\ \gamma_{i-1}(p, c) \setminus F^i & \text{otherwise,} \end{cases}$$

and

$$\gamma_i(c, p) \coloneqq \begin{cases} \gamma_{i-1}(c, p) \cup F_j^{*i} & \text{if } (p, c) = (s_i, t_j^i), \text{ for some } j \in [a_i], \\ \gamma_{i-1}(c, p) & \text{if } t_j^i = p, \text{ for some } j \in [a_i], \\ \gamma_{i-1}(c, p) \setminus F^i & \text{if } c \quad s_i, \\ \gamma_{i-1}(c, p) \cup F^i & \text{otherwise,} \end{cases}$$

and all other $uv \in \overrightarrow{E(T)}$ we set $\gamma_i(u, v) \coloneqq \gamma_{i-1}(u, v)$. Furthermore we set

$$\beta_i(t) \coloneqq \begin{cases} \delta(\pi_t^i) & \text{if } t \in V(T_i), \\ \beta_{i-1}(t) & \text{otherwise.} \end{cases}$$

Intuitively in the construction above we push the change at s_i through T_{i-1}, that is for all edges in T_{i-1} we add F^i to the directed edge that points away from s_i and remove F^i from the edges in the other direction. We obtain the following observation.

▶ **Observation 20.** *Let $i, j \in [n_T]$ such that s_i is the parent of s_j. Then $\gamma_\alpha(s_i, s_j) \subseteq \gamma(s_i, s_j)$, for all $\alpha < i$.*

The proof. We prove Theorem 19 in three steps. First we prove that the construction indeed yields an exact pre-tree decomposition. Next we show that the width can be bounded as desired and lastly we prove that the construction yields the desired depth.

▶ **Lemma 21.** *For all $i \in [n_T]$, $(T, r, \beta_i, \gamma_i)$ is a pre-tree decomposition. Furthermore all edges in $E(T_i)$ are exact.*

Proof. (PT1) holds as all edges leaving the root are already exact in γ, thus we change nothing in step 1 where the root is considered and every F^i, with $i > 1$, only contains edges from a single component of G by construction.

We observe that the changes from γ_{i-1} to γ_i at some node $t \in V(T_i) \setminus \{s_i\}$ corresponds to an F^i- or F_j^{*i}-extension of π_t^{i-1} at the set that corresponds to the edge, that points towards s_i. Furthermore π_s^i is a partition of the edges by construction and for all $t \in V(T_i)$ we set $\beta_i(t) = \delta(\pi_t^i)$. As γ_i and β_i are equal to γ and β at all vertices that are not part of $V(T_i)$, this shows by induction that (PT3) still holds.

Next we observe that at every edge that is not incident to some t_j^i we add to one direction exactly what we remove from the other direction. Furthermore by construction the edges $s_i t_j^i$ are exact after the construction. Lastly, for all children c of t_j^i, we only remove edges from $\gamma_i(t_j^i, c)$. Thus again by induction we get that (PT4) holds and that all edges of T_i are exact.

It remains to show that (PT2) holds. Let $i \in [n_t]$ and $j \in [a_i]$ such that $t_j^i \in L(T)$. From Observation 20 we know that $\gamma_{i-1}(s_i, t_j^i) \subseteq \gamma(s_i, t_j^i)$ and thus $|\gamma_{i-1}(s_i, t_j^i)| \leq 1$. Furthermore we know that $\gamma_{i-1}(t_j^i, s_i) = \gamma(t_j^i, s_i) = E(G^\circ) \setminus \gamma(s_i, t_j^i)$. Therefore we get that $F_j^i \subseteq \gamma(s_i, t_j^i)$ and thus $\gamma_i(s_i, t_j^i) \subseteq \gamma(s_i, t_j^i)$. By construction, in step $i \in [n_T]$, such that $s_i \;= t_j^i$, we do nothing. And in all other steps $\alpha > i$, we have that $t_j^i \quad s_\alpha$ and thus we only remove edges from $\gamma_\alpha(s_i, t_j^i)$. This shows that for all $\alpha \in [n_T]$ we have $|\gamma_\alpha(s_i, t_j^i)| \leq 1$. ◀

Hence, for $i = n_T$, we get that $(T, r, \beta_{n_T}, \gamma_{n_T})$ is an exact pre-tree decomposition. Note that it is possible that $\gamma_{n_T}(s, t)$ is empty for an edge $st \in \overrightarrow{E(T)}$. By Lemma 7 we obtain a tree decomposition, from this pre-tree decomposition. We show below that the width and depth are as stated in the theorem.

Our construction does not change the width of the decomposition. To prove this we observe that in step i the bound in s_i is minimal. We then push the change through the subtree T_i and find that if a change would increase the width, we could push this change back to the node s_i and find an even smaller bound there, which contradicts the minimality of our choice.

▶ **Lemma 22.** $\mathrm{wd}(T, r, \beta_i, \gamma_i) \leq \mathrm{wd}(T, r, \beta, \gamma)$*, for all $i \in [n_T]$.*

Proof. We prove the statement for all $0 \leq i \leq n_T$ by induction. As $(T, r, \beta_0, \gamma_0) = (T, r, \beta, \gamma)$, the statement clearly holds for $i = 0$. Next we show that $\mathrm{wd}(T, r, \beta_i, \gamma_i) \leq \mathrm{wd}(T, r, \beta_{i-1}, \gamma_{i-1})$, for all $i \in [n_T]$. Obviously $|\beta_i(t)| = \beta_{i-1}(t)$, for all $t \notin T_i$. Furthermore by construction $|\beta_i(s_i)| \leq |\beta_{i-1}(s_i)|$. Let $j \in [a_i]$, let $X := \gamma_i(s_i, t^i_j)$ and let $Y := \gamma_{i-1}(t^i_j, s_i)$. We observe that

$$\pi^i_{t^i_j} = \pi^{i-1}_{t^i_j, Y \to \overline{X}}.$$

Thus it holds that

$$|\beta_i(t^i_j)| = |\,\mathrm{wd}(\pi^{i-1}_{t^i_j, Y \to \overline{X}})| \leq |\,\mathrm{wd}(\pi^{i-1}_{t^i_j})| \leq |\beta_{i-1}(t^i_j)|$$

as otherwise by submodularity for the partitions $\pi^{i-1}_{t^i_j}$ and $\pi^i_{s_i}$, we get that

$$|\,\mathrm{wd}(\pi^i_{s_i, X \to \overline{Y}})| < |\,\mathrm{wd}(\pi^i_{s_i})|,$$

which contradicts the minimality of the bound for $F^i_1, \dots, F^i_{a_i}$.

Lastly assume there is a node t in $V(T_i) \setminus \{s_i, t^i_1, \dots, t^i_{a_i}\}$ such that $|\beta_i(t)| > |\beta_{i-1}(t)|$. We assume t is of minimal distance to s_i with this property. Let $x_0 = t, x_1, \dots, x_b = s_i$ be the path from t to s_i. By minimality of the distance we know that $|\beta_i(x_1)| \leq |\beta_{i-1}(x_1)|$. Additionally we know that all edges on the path from s_i to x_1 are exact in γ_i, as well as the edge $x_1 t$ in γ_{i-1}. Now let $Y := \gamma_{i-1}(t, x_1)$ and, for all $0 \leq \alpha < b$, let $X_\alpha := \gamma_i(x_{\alpha+1}, x_\alpha)$ and $Z_\alpha := \gamma_i(x_\alpha, x_{\alpha+1})$. The transition from $i-1$ to i at t corresponds to $\pi^{i-1}_{t,Y \to F} = \pi^{i-1}_{t,Y \to \overline{X_0}}$. Thus if $\mathrm{wd}(\pi^{i-1}_{t,Y \to F}) = |\beta_i(t)| > |\beta_{i-1}(t)| = \mathrm{wd}(\pi^{i-1}_t)$ we get by submodularity that $\mathrm{wd}(\pi^i_{x_1}) > \mathrm{wd}(\pi^i_{x_1, X_0 \to \overline{Y}})$. As the edge $x_1 t$ was exact at step $i - 1$, we know that

$$F\ := \overline{Y} \setminus X_0 = F^i \setminus Y \subseteq F^i.$$

We now push this change back to s_i along the path $x_1, \dots, x_b$ and we again find a contradiction to the minimality of the bound of $F^i_1, \dots, F^i_{a_i}$. For this, let us assume we have pushed the change to x_α, that is we changed $\pi^i_{x_\alpha}$ to $\pi^*_{x_\alpha} = \pi^i_{x_\alpha, X_{\alpha-1} \to F}$ and we know that $\mathrm{wd}(\pi^*_{x_\alpha}) < \mathrm{wd}(\pi^i_{x_\alpha})$. As the edge $x_\alpha x_{\alpha+1}$ is exact in γ_i, we get that $\pi^*_{x_\alpha, (Z_\alpha \setminus F\ ) \to \overline{X_\alpha}} = \pi^i_{x_\alpha}$. Let

$$\pi^*_{x_{\alpha+1}} := \pi^i_{x_{\alpha+1}, X_\alpha \to \overline{(Z_\alpha \setminus F\ )}} = \pi^i_{x_{\alpha+1}, X_\alpha \to F}\ ,$$

then by submodularity $\mathrm{wd}(\pi^*_{x_{\alpha+1}}) < \mathrm{wd}(\pi^i_{x_{\alpha+1}})$. When we have pushed the change to $\alpha = b$, we find the desired contradiction. ◀

To prove that our construction does not increase the depth we show that in every step i the depth up to the nodes in T_i is bounded by the depth up to these nodes in the original tree. We prove this by induction on the number of steps. In step i every change in any bag at some node in $V(T_{i-1})$ is closely related to the change at the considered node s_i. Additionally we find that the vertices in a bag at some child of s_i that is not present in the bag at s_i is exactly the vertex the cop player newly placed in the corresponding move of the game.

▶ **Lemma 23.** *For all $i \in [n_T]$ and all $t \in V(T_i)$, it holds that*

$$\sum_{s \in P_t \setminus \{r\}} |\beta_i(s) \setminus \beta_i(p_s)| \leq \sum_{s \in P_t \setminus \{r\}} |\beta(s) \setminus \beta(p_s)|.$$

Proof. Let $\ \in [n_T]$. As by construction $\beta\ (t) = \delta(\pi_t)$, for all $t \in V(T\ )$, we get from Observation 9 and Lemma 21 that $|\ _{s \in P_t} \beta\ (s)| =\ _{s \in P_t \setminus \{r\}} |\beta\ (s) \setminus \beta\ (p_s)|$. Thus it suffices to show that $|\ _{s \in P_t} \beta_i(s)| \leq\ _{s \in P_t \setminus \{r\}} |\beta(s) \setminus \beta(p_s)|$.

We prove the statement by induction on the steps i. Recall that $(T, r, \beta_0, \gamma_0) = (T, r, \beta, \gamma)$, thus the statement holds for $i = 0$. Now assume the statement holds for $i - 1$, thus for all $t \in V(T_{i-1})$ it hold that $| \ _{s\in P_t} \beta_{i-1}(s)| \le \ _{s\in P_t\setminus\{r\}} |\beta(s) \setminus \beta(p_s)|$.

We recall that $V(T_i) = V(T_{i-1}) \cup \{t^i_1, \ldots, t^i_{a_i}\}$ and that $s_i \in L(T_{i-1})$. For the nodes $t \in V(T_{i-1})$ we can directly build upon the induction hypothesis. But the nodes t^i_j, with $j \in [a_i]$, are added into the subtree. Here we need to compare directly to the original bags, as we can no longer use that in step $i-1$ the depth at these nodes is bounded by the depth in the original strategy tree. We can prove for these nodes that every vertex newly placed at one of these nodes in step i is also newly placed in the original strategy. Then we can show that the difference between depth at these nodes and their parent in step i can be bounded by the difference in the original strategy tree.

▷ Claim 24. Every $j \in [a_i]$ satisfies $\beta_i(t^i_j) \setminus \beta_i(s_i) \subseteq \beta(t^i_j) \setminus \beta(s_i)$.

Proof. Let $v \in \beta_i(t^i_j) \setminus \beta_i(s_i)$. As $v \notin \beta_i(s_i)$ we get that $v \notin \delta(\gamma_i(t^i_j, s_i))$ and thus $E_{G^\circ}(v) \cap \gamma_i(t^i_j, s_i) = \emptyset$. By construction we have that $\gamma_i(t^i_j, s_i) \supseteq \ i-1(t^i_j, s_i) = \gamma(t^i_j, s_i)$, and thus $vv \notin \gamma(t^i_j, s_i)$. As $v \in \beta_i(t^i_j) = \delta(\pi^i_{t^i_j})$, there are two distinct children c_1, c_2 of t^i_j such that $v \in \delta(\gamma_i(t^i_j, c\ ))$ and thus $E_{G^\circ}(v) \cap \gamma_i(t^i_j, c\ ) = \emptyset$, for $\ = 1, 2$. By construction we have $\gamma_i(t^i_j, c\ ) \subseteq \gamma_{i-1}(t^i_j, c\ ) = \gamma(t^i_j, c\ )$, for $\ = 1, 2$. And thus $v \in \delta(\pi_{t^i_j}) \subseteq \beta(t^i_j)$. By Observation 17 there thus is a child c of t^i_j such that $\gamma(t^i_j, c) = \{vv\}$ and, by Observation 18, $v \in \beta(t^i_j) \setminus \beta(s_i)$. ◁

The following claim tracks vertices that are added to any bag in $V(T_{i-1})$ at step i.

▷ Claim 25. Let $i \in [n_T]$ and let $t \in V(T_{i-1})$. If $v \in \beta_i(t) \setminus \beta_{i-1}(t)$, then $v \in \beta_i(t^*)$, for all t^* on the path from t to s_i.

Proof. Let $t^* = t$. Let $t\ $ be the next node on the path from t to s_i. Then $\gamma_i(t, t\ ) = \gamma_{i-1}(t, t\ ) \cup F^i$. As $\gamma_i(t, t\ )$ is the only set incident to t where edges are added in step i, we get that $v \in \delta(\gamma_i(t, t\ ))$. And from $v \notin \delta(\gamma_{i-1}(t, t\ ))$ we get that $v \in \delta(F^i)$. Now suppose that $v \notin \beta_i(t^*)$, and thus also $v \notin \delta(\gamma_i(t^*, p))$, where p is the next node on the path from t^* to t. As v is incident to edges in F^i we get that $E_{G^\circ}(v) \cap \gamma_i(t^*, p) = \emptyset$. We know from Lemma 21 that all edges in T_i are exact and thus that $\gamma_i(t\ , t) \subseteq \gamma_i(t^*, p)$ by Observation 10. This is a contradiction to $v \in \delta(\gamma_i(t, t\ )) = \delta(\gamma_i(t\ , t))$ and thus $v \in \beta_i(t^*)$. ◁

The next claim is used to show that a vertex that disappears from a bag in $V(T_{i-1})$ at step i also disappears from the union of bags that determine the depth at that bag, especially if a vertex disappears from the bag at s_i, then it disappears from every bag in $V(T_i)$.

▷ Claim 26. Let $i \in [n_T]$ and let $t \in V(T_{i-1})$. If $v \in \beta_{i-1}(t) \setminus \beta_i(t)$, then $v \notin \beta_i(t^*)$, for all $t^* \in V(T_{i-1})$ such that t is contained in the path from t^* to s_i.

Proof. We have $E_{G^\circ}(v) \cap F^i = \emptyset$.

Let $t = s_i$. As $v \notin \beta_i(s_i)$ we get that $v \notin \delta(\gamma_i(s_i, p_{s_i}))$ and thus $E_{G^\circ}(v) \cap \gamma_i(s_i, p_{s_i}) = E_{G^\circ}(v) \cap \gamma_{i-1}(s_i, p_{s_i}) \cap \overline{F^i} = \emptyset$. Now let $t^* \in V(T_{i-1})$ and $t\ $ be the next node on the path from t^* to s_i. Then by Lemma 21 we get that $\gamma_i(t^*, t\ ) \supseteq \gamma_i(p_{s_i}, s_i) \supseteq E_{G^\circ}(v)$ and thus $v \notin \beta_i(t^*)$.

Otherwise let $t = s_i$ Let $t\ $ be the next node on the path from t to s_i. As $v \notin \delta(\gamma_i(t, t\ ))$ it follows that $E_{G^\circ}(v) \subseteq \gamma_i(t, t\ ) = \gamma_{i-1}(t, t\ ) \cup F^i$, that $v \in \delta(\gamma_{i-1}(t, t\ ))$, and that $E_{G^\circ}(v) \cap \gamma_{i-1}(t\ , t) \subseteq E_{G^\circ}(v) \cap F^i$. Assume there is some $t^* \in V(T_{i-1})$ such that $v \in \beta_i(t^*)$. We observe that due to Lemma 21 and because all edges incident to v are contained in $\gamma_i(t, t\ )$, we get that t is not contained in the path from t^* to s_i. ◁

We are now ready to prove the lemma. Towards this, let $i \geq 1$ and assume the statement holds for $i-1$. We consider all vertices that appear at a bag at any node in T_i due to the changes in step i. Observe that if $\beta_i(s_i) = \beta_{i-1}(s_i)$, then there are no changes to the bags at other nodes than the t^i_j by minimality of $|F^i|$, and if $\beta_i(s_i) = \beta_{i-1}(s_i)$, we have $|\beta_i(s_i)| < |\beta_{i-1}(s_i)|$ again by the minimality of the choice.

Let $t \in V(T_{i-1})$. Let $U := \;_{s\in P_t} \beta_i(s) \setminus \;_{s\in P_t} \beta_{i-1}(s)$. Let t^* be the greatest common ancestor of t and s_i. As t^* is on every path from some node in P_t to s_i, from Claim 25 we know that $u \in \beta_i(t^*) \setminus \beta_{i-1}(t^*)$, for all $u \in U$. Let $W = \beta_{i-1}(t^*) \setminus \beta_i(t^*)$ As by Lemma 22 $|\beta_i(t^*)| \leq |\beta_{i-1}(t^*)|$, we know that $|U| \leq |W|$. By Claim 26 we get that $W \subseteq \;_{s\in P_t} \beta_{i-1}(s) \setminus \;_{s\in P_t} \beta_i(s)$. By this *vertex exchange* we conclude that $\;_{s\in P_t} \beta_i(s) \leq \;_{s\in P_t} \beta_{i-1}(s)$.

Otherwise $t = t^i_j$ for some $j \in [a_i]$. By construction we get $\;_{s\in P_t} \beta_i(s) = \;_{s\in P_{s_i}} \beta_i(s) \cup \beta_i(t) \setminus \beta_i(s_i)$. We have shown above that $| \;_{s\in P_{s_i}} \beta_i(s)| \leq \;_{s\in P_{s_i}\setminus\{r\}} |\beta(s) \setminus \beta(p_s)|$ and by Claim 24 we have $\beta_i(t) \setminus \beta_i(s_i) \subseteq \beta(t) \setminus \beta(s_i)$. Thus we can bound the union $| \;_{s\in P_t} \beta_i(s)| \leq \;_{s\in P_t\setminus\{r\}} |\beta(s) \setminus \beta(p_s)|$. ◀

Proof of Theorem 19. Combining Lemmas 21–23 we get that there exists an exact pre-tree decomposition of G° of width $\leq k$ and depth $\leq q$, if the cop player wins $\mathrm{CR}^k_q(G^\circ)$. The theorem then follows from Lemma 7. ◀

Summarising all results we get the following equivalences.

▶ **Theorem 27.** *Let $k, q \geq 1$ and G be a graph. The following are equivalent:*
(1) *G admits a tree decomposition of width at most $k-1$ and depth at most q.*
(2) *G° admits a tree decomposition of width at most $k-1$ and depth at most q.*
(3) *G° admits an exact pre-tree decomposition of width at most $k-1$ and depth at most q.*
(4) *The cop player wins* mon-$\mathrm{CR}^k_q(G^\circ)$.
(5) *The cop player wins* $\mathrm{CR}^k_q(G^\circ)$.
(6) *The cop player wins* mon-$\mathrm{CR}^k_q(G)$.
(7) *The cop player wins* $\mathrm{CR}^k_q(G)$.

Proof. Every tree decomposition of G is also a tree decomposition of G° and vice versa, thus (1) and (2) are equivalent. Lemma 7 shows the equivalence of (1) and (3). Theorem 19 shows that (5) implies (1). Let G be a graph without edges. The cop player wins on G before the first round and on G° after one placement with one cop. On a connected component with at least one edge the robber escape spaces are essentially the same. Thus for $q, k \geq 1$, by construction, (4) and (6) as well as (5) and (7) are equivalent. Lemma 14 shows that (1) is equivalent to (6) and thus also (2) to (4). Furthermore (6) implies (7) and (4) implies (5) by construction. Thus the theorem follows. ◀

6 Excursion on counting homomorphisms

In this section we give an overview over the field of counting homomorphisms and the equivalence relations on graphs, that can be derived from these counts. We focus ourselves to the results and open questions regarding the homomorphism counts from graphs in the class $\mathcal{T}^k_q$, for fixed $k, q \geq 0$.

We start by recalling the important definitions. Let G, F be two graphs. A *homomorphism* from F into G is a function $\varphi \colon V(F) \to V(G)$, such that for every $uv \in E(F)$, it holds that $\varphi(u)\varphi(v) \in E(G)$. By $\hom(F, G)$ we denote the number of homomorphisms from F into G. Let $\mathcal{F}$ be a graph class. We say two graphs G and H are *homomorphism indistinguishable over* $\mathcal{F}$ if, for every $F \in \mathcal{F}$ it holds that $\hom(F, G) = \hom(F, H)$, we write $G \equiv_{\mathcal{F}} H$. A graph class $\mathcal{F}$ is *homomorphism distinguishing closed* if, for every $F \notin \mathcal{F}$, there exist two graphs G, H such that $G \equiv_{\mathcal{F}} H$, but $\hom(F,G) = \hom(F,H)$.

In [13] the authors have reduced the question whether the class $\mathcal{T}_q^k$ is homomorphism distinguishing closed down to the question if monotonicity is a restriction for the cop player. The definition of the cops-and-robber game the authors use is slightly different. In their game the robber hides in vertices and is caught if a cops occupies the same vertex. With the same arguments as in the proof of Theorem 27 we observe that these games are equivalent. The following lemma is thus implied in [13].

▶ **Lemma 28** ([13]). *Let $k, q \geq 1$. The graph class $\mathcal{C} := \{G \mid$ cop player wins $\mathrm{CR}_q^k(G)\}$ is homomorphism distinguishing closed.*

In this paper we show that the cop player wins $\mathrm{CR}_q^k(G)$if and only if $G \in \mathcal{T}_q^k$, thus the we get the following.

▶ **Theorem 29.** *Let $k, q \geq 0$ be integers. The class $\mathcal{T}_q^k$ is homomorphism distinguishing closed.*

Proof. Assume $k, q \geq 1$. Then theorem follows directly from Lemma 28 and Theorem 27. Thus assume $k = 0$ or $q = 0$. The only graph that has a tree decomposition of depth 0 or width -1 is the empty graph. The only homomorphism from the empty graph into any graph is the empty function, which is a homomorphism independent of the right-hand-side graph. Thus there are no graphs that can be distinguished by the number of homomorphisms from the empty graph. Thus the theorem holds. ◀

7 Conclusion

We gave a new characterisation of bounded depth treewidth by the cops and robber game with both a bound on the number of cops and on the number of placements,where the cops are allowed to make non-monotone moves. As a corollary we gave a positive answer to an open question on homomorphism counts. The core of our contribution is a proof of monotonicity of this game. For this proof we substantially reorganise a winning strategy. First we transform it into a pre-decomposition. Then we apply a breadth-first 'cleaning up' procedure along the pre-decomposition (which may temporarily lose the property of representing a strategy), in order to achieve monotonicity while controlling the number of cop placements simultaneously across all branches of the decomposition via a vertex exchange argument(cf. the proof of Lemma 23). As an interesting observation we obtain that cop moves into the back country, i.e. to positions that are not part of the boundary, can be ignored and the depth of the exact pre-tree decomposition is the number of cops placed into the robber escape space: We observe that in the proof of Claim 24 where we compute how much larger the depth at some node t_j^i at step i is than at the considered node s_i the depth increases only if the node t_j^i is branching by Observation 18 as t_j^i has a child where the cone contains only a self-loop and hence this is a move into the robber space.

▶ **Corollary 30.** $\mathrm{dp}(T, r, \beta_{n_T}, \gamma_{n_T}) \leq \max_{\in L(T)} |\{t \in P \mid t \text{ is branching}\}|$.

In the future, it would be interesting to know if it is possible to give a proof that entirely argues with game strategies (not requiring pre-decopmositions), and we leave this open. We also leave open whether a dual object similar to brambles can be defined for bounded depth treewidth. Finally, given a winning strategy for k cops with q placements, it would be interesting to know if it is possible to bound the number of cops necessary for winning with only $q - 1$ placements in terms of k and q, given that the cop player still can win.

References

1 Samson Abramsky, Anuj Dawar, and Pengming Wang. The pebbling comonad in finite model theory. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005129.

2 Isolde Adler. Marshals, monotone marshals, and hypertree-width. *J. Graph Theory*, 47(4):275–296, 2004. URL: https://doi.org/10.1002/jgt.20025, doi:10.1002/JGT.20025.

3 Isolde Adler. Games for width parameters and monotonicity. *CoRR*, abs/0906.3857, 2009. URL: http://arxiv.org/abs/0906.3857, arXiv:0906.3857.

4 Isolde Adler, Georg Gottlob, and Martin Grohe. Hypertree width and related hypergraph invariants. *Eur. J. Comb.*, 28(8):2167–2181, 2007. URL: https://doi.org/10.1016/j.ejc.2007.04.013, doi:10.1016/J.EJC.2007.04.013.

5 Martin Aigner and M. Fromme. A game of cops and robbers. *Discret. Appl. Math.*, 8(1):1–12, 1984. doi:10.1016/0166-218X(84)90073-8.

6 Omid Amini, Frédéric Mazoit, Nicolas Nisse, and Stéphan Thomassé. Submodular partition functions. *Discret. Math.*, 309(20):6000–6008, 2009. URL: https://doi.org/10.1016/j.disc.2009.04.033, doi:10.1016/J.DISC.2009.04.033.

7 Daniel Bienstock. Graph searching, path-width, tree-width and related problems (A survey). In Fred Roberts, Frank Hwang, and Clyde L. Monma, editors, *Reliability Of Computer And Communication Networks, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, December 2-4, 1989*, volume 5 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 33–50. DIMACS/AMS, 1989. URL: https://doi.org/10.1090/dimacs/005/02, doi:10.1090/DIMACS/005/02.

8 Daniel Bienstock and Paul D. Seymour. Monotonicity in graph searching. *J. Algorithms*, 12(2):239–245, 1991. doi:10.1016/0196-6774(91)90003-H.

9 Hans L. Bodlaender and Dimitrios M. Thilikos. Computing small search numbers in linear time. In Rodney G. Downey, Michael R. Fellows, and Frank K. H. A. Dehne, editors, *Parameterized and Exact Computation, First International Workshop, IWPEC 2004, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3162 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 2004. doi:10.1007/978-3-540-28639-4_4.

10 Anuj Dawar, Tomáš Jakl, and Luca Reggio. Lovász-Type Theorems and Game Comonads. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, June 2021. doi:10.1109/LICS52264.2021.9470609.

11 Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász Meets Weisfeiler and Leman. *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, pages 40:1–40:14, 2018. doi:10.4230/LIPICS.ICALP.2018.40.

12 Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, August 2010. doi:10.1002/jgt.20461.

13 Eva Fluck, Tim Seppelt, and Gian Luca Spitzer. Going Deep and Going Wide: Counting Logic and Homomorphism Indistinguishability over Graphs of Bounded Treedepth and Treewidth. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024)*, volume 288 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:17, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CSL.2024.27, doi:10.4230/LIPIcs.CSL.2024.27.

14 Fedor V. Fomin, Pierre Fraigniaud, and Nicolas Nisse. Nondeterministic graph searching: From pathwidth to treewidth. *Algorithmica*, 53(3):358–373, 2009. URL: https://doi.org/10.1007/s00453-007-9041-6, doi:10.1007/S00453-007-9041-6.

15 Fedor V. Fomin, Petr A. Golovach, and Jan Kratochvíl. On tractability of cops and robbers game. In Giorgio Ausiello, Juhani Karhumäki, Giancarlo Mauri, and C.-H. Luke Ong, editors, *Fifth IFIP International Conference On Theoretical Computer Science - TCS 2008, IFIP 20th World Computer Congress, TC 1, Foundations of Computer Science, September 7-10,*

2008, Milano, Italy, volume 273 of *IFIP*, pages 171–185. Springer, 2008. doi:10.1007/978-0-387-09680-3_12.

16 Fedor V. Fomin and Dimitrios M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008. URL: https://doi.org/10.1016/j.tcs.2008.02.040, doi:10.1016/J.TCS.2008.02.040.

17 Matthew K. Franklin, Zvi Galil, and Moti Yung. Eavesdropping games: a graph-theoretic approach to privacy in distributed systems. *J. ACM*, 47(2):225–243, 2000. doi:10.1145/333979.333980.

18 Archontia C. Giannopoulou, Paul Hunter, and Dimitrios M. Thilikos. Lifo-search: A min-max theorem and a searching game for cycle-rank and tree-depth. *Discret. Appl. Math.*, 160(15):2089–2097, 2012. URL: https://doi.org/10.1016/j.dam.2012.03.015, doi:10.1016/J.DAM.2012.03.015.

19 Archontia C. Giannopoulou and Dimitrios M. Thilikos. A min-max theorem for lifo-search. *Electron. Notes Discret. Math.*, 38:395–400, 2011. URL: https://doi.org/10.1016/j.endm.2011.09.064, doi:10.1016/J.ENDM.2011.09.064.

20 Martin Grohe. Counting Bounded Tree Depth Homomorphisms. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 507–520, New York, NY, USA, 2020. Association for Computing Machinery. event-place: Saarbrücken, Germany. doi:10.1145/3373718.3394739.

21 Martin Grohe and Dániel Marx. Constraint solving via fractional edge covers. *ACM Trans. Algorithms*, 11(1):4:1–4:20, 2014. doi:10.1145/2636918.

22 Martin Grohe, Gaurav Rattan, and Tim Seppelt. Homomorphism Tensors and Linear Equations. In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 70:1–70:20, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2022.70.

23 Geoffrey A. Hollinger, Athanasios Kehagias, and Sanjiv Singh. GSST: anytime guaranteed search. *Auton. Robots*, 29(1):99–118, 2010. URL: https://doi.org/10.1007/s10514-010-9189-9, doi:10.1007/S10514-010-9189-9.

24 Paul Hunter and Stephan Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theor. Comput. Sci.*, 399(3):206–219, 2008. URL: https://doi.org/10.1016/j.tcs.2008.02.038, doi:10.1016/J.TCS.2008.02.038.

25 Thor Johnson, Neil Robertson, Paul D. Seymour, and Robin Thomas. Directed tree-width. *J. Comb. Theory, Ser. B*, 82(1):138–154, 2001. URL: https://doi.org/10.1006/jctb.2000.2031, doi:10.1006/JCTB.2000.2031.

26 Andrea S. LaPaugh. Recontamination does not help to search a graph. *J. ACM*, 40(2):224–245, 1993. doi:10.1145/151261.151263.

27 László Lovász. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(3):321–328, September 1967. doi:10.1007/BF02280291.

28 Fillia Makedon and Ivan Hal Sudborough. On minimizing width in linear layouts. *Discret. Appl. Math.*, 23(3):243–265, 1989. doi:10.1016/0166-218X(89)90016-4.

29 Laura Mančinska and David E. Roberson. Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 661–672, 2020. doi:10.1109/FOCS46700.2020.00067.

30 Frédéric Mazoit and Nicolas Nisse. Monotonicity of non-deterministic graph searching. *Theor. Comput. Sci.*, 399(3):169–178, 2008. URL: https://doi.org/10.1016/j.tcs.2008.02.036, doi:10.1016/J.TCS.2008.02.036.

31 Jaroslav Nesetril and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *Eur. J. Comb.*, 27(6):1022–1041, 2006. URL: https://doi.org/10.1016/j.ejc.2005.01.010, doi:10.1016/J.EJC.2005.01.010.

32 Daniel Neuen. Homomorphism-Distinguishing Closedness for Graphs of Bounded Tree-Width, April 2023. doi:10.48550/arXiv.2304.07011.

33 Jan Obdrzálek. Dag-width: connectivity measure for directed graphs. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 814–821. ACM Press, 2006. URL: http://dl.acm.org/citation.cfm?id=1109557.1109647.

34 T. D. Parsons. Pursuit-evasion in a graph. In Yousef Alavi and Don R. Lick, editors, *Theory and Applications of Graphs*, pages 426–441, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

35 Torrence D Parsons. The search number of a connected graph. In *Proc. 9th South-Eastern Conf. on Combinatorics, Graph Theory, and Computing*, pages 549–554, 1978.

36 Nikolai N. Petrov. A problem of pursuit in the absence of information on the pursued. *Differentsial'nye Uravneniya*, 18(1):345—-1352, 1982.

37 David E. Roberson. Oddomorphisms and homomorphism indistinguishability over graphs of bounded degree, June 2022. doi:10.48550/arXiv.2206.10321.

38 David E. Roberson and Tim Seppelt. Lasserre Hierarchy for Graph Isomorphism and Homomorphism Indistinguishability. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 101:1–101:18, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2023.101.

39 Benjamin Scheidt and Nicole Schweikardt. Counting homomorphisms from hypergraphs of bounded generalised hypertree width: A logical characterisation. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France*, volume 272 of *LIPIcs*, pages 79:1–79:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: https://doi.org/10.4230/LIPIcs.MFCS.2023.79, doi:10.4230/LIPICS.MFCS.2023.79.

40 Tim Seppelt. Logical Equivalences, Homomorphism Indistinguishability, and Forbidden Minors. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2023.82.

41 Paul D. Seymour and Robin Thomas. Graph searching and a min-max theorem for tree-width. *J. Comb. Theory, Ser. B*, 58(1):22–33, 1993. URL: https://doi.org/10.1006/jctb.1993.1027, doi:10.1006/JCTB.1993.1027.