Parity Games, Imperfect Information and Structural Complexity[☆]

Bernd Puchala[a], Roman Rabinovich[b]

[a]*Mathematical Foundations of Computer Science, RWTH Aachen University*
[b]*Logic and Semantics, Technical University Berlin*

Abstract

We address the problem of solving parity games with imperfect information on finite graphs of bounded structural complexity. It is a major open problem whether parity games with *perfect* information can be solved in Ptime. Restricting the structural complexity of the game arenas, however, often leads to efficient algorithms for parity games. Such results are known for graph classes of bounded tree-width, DAG-width, directed path-width, and entanglement, which we describe in terms of cops and robber games. Conversely, the introduction of imperfect information makes the problem more difficult, it becomes Exptime-hard. We analyse the interaction of both approaches.

We use a simple method to measure the amount of "unawareness" of a player, the amount of imperfect information. It turns out that if it is unbounded, low structural complexity does not make the problem simpler. It remains Exptime-hard or Pspace-hard even on very simple graphs.

For games with bounded imperfect information we analyse the powerset construction, which is commonly used to convert a game of imperfect information into an equivalent game with perfect information. This construction preserves boundedness of directed path-width and DAG-width, but not of entanglement or of tree-width. Hence, if directed path-width or DAG-width are bounded, parity games with bounded imperfect information can be solved in Ptime. For DAG-width we follow two approaches. One leads to a generalization of the known fact that perfect information parity games are in Ptime if DAG-width is bounded. We prove this theorem for *non-monotone* DAG-width. The other approach introduces a cops and *robbers* game (with multiple robbers) on directed graphs, considered in [RT09] for undirected graphs. We show a tight linear bound for the number of additional cops needed to capture an additional robber.

Keywords: parity games, imperfect information, graph searching games

[☆]This work was supported by the projects *Games for Analysis and Synthesis of Interactive Computational Systems (GASICS)* and *Logic for Interaction (LINT)* of the *European Science Foundation*.

Email addresses: puchala@logic.rwth-aachen.de (Bernd Puchala), roman.rabinovich@tu-berlin.de (Roman Rabinovich)

1. Introduction

Parity games play a key role in the theory of verification and synthesis of state-based systems. They are the model-checking games for the modal μ-calculus, a powerful specification formalism for verification problems. Moreover, parity objectives can express all ω-regular objectives and therefore capture fundamental properties of non-terminating reactive systems, cf. [Tho95]. Such a system can be modeled as a two-player game (the players are called 0 and 1) where changes of the system state correspond to changes of the game position. Situations where the change of the system can be controlled correspond to positions of Player 0, uncontrollable situations correspond to positions of Player 1. A winning strategy for Player 0 yields a controller that guarantees satisfaction of some ω-regular specification.

In a parity game, the players move a token along the edges of a labeled graph by choosing appropriate edge labels, called actions. The vertices of the graph, called positions, are labeled with natural numbers and the winner of an infinite play of the game is determined by the parity of the least color which occurs infinitely often.

The problem to determine, for a given parity game $\mathcal{G}$ and a position v, whether Player 0 has a winning strategy for $\mathcal{G}$ from v, is called the strategy problem. The algorithmic theory of parity games with perfect information has received much attention during the past years, cf. [Jur00].

However, assuming that both players have perfect information about the history of events in a parity game is not always realistic. For example, if the information about the system state is acquired by imprecise sensors or the system encapsulates private states which cannot be read from outside, then a controller for this system must rely on the information about the state and the change of the system to which it has access. A technique to solve the strategy problem in presence of imperfect information is to track the knowledge of the game of Player 0, thus reducing the problem to a strategy problem for a game with perfect information on another graph [Rei84]. This procedure is often referred to as *powerset construction* and we call the constructed graph the *powerset graph.*

Such a knowledge tracking is inherently unavoidable and leads to an exponential lower bound for the time complexity of the strategy problem for reachability games with imperfect information [Rei84] and a super-polynomial lower bound for the memory needed to implement winning strategies in reachability games [BCD$^+$08, Puc].

Our goal is to find interesting special cases of the problem that can be solved in Ptime. A simple, yet effective, approach is to bound the amount of uncertainty of Player 0. This is appropriate in situations where, e.g., the imprecision of the sensors or the amount of private information of the system does not grow when the size of the system grows. Then the game which results from the powerset construction has polynomial size, so solving imperfect information parity games reduces to the strategy problem for parity games with perfect information.

However, it is not known whether the latter problem can be solved efficiently, i.e., in Ptime, and the question whether this is possible remains one of the most intriguing in game theory.

To obtain a class of parity games with imperfect information that we can solve in Ptime, we thus have to bound certain other parameters. A natural approach is to restrict the structural complexity of the game graphs with respect to an appropriate measure. Several such measures have proven to be very useful in algorithmic graph theory. Many problems, including the strategy problem for perfect information parity games which are intractable in general can be solved efficiently on classes of graphs where such measures are bounded. It has been shown that parity games played on graphs of bounded tree-width, path-width, directed path-width, DAG-width or entanglement can be solved in polynomial time [BDH+12, BG05a]. A natural question is whether these results can also be obtained for games with imperfect information. For each of those complexity measures we answer two questions about parity games on graphs of bounded complexity:

1. Are the games with (in general *unbounded*) imperfect information solvable in Ptime?

2. Are the games with *bounded* imperfect information solvable in Ptime?

For two other important measures: directed tree-width [JRST01] and Kelly-width [HK07] the problem remains open, for directed tree-width even in the case of perfect information.

Organization and results.. In Section 2 we introduce the basic notions we use throughout the paper. In Section 3 we consider unbounded imperfect information. For all complexity measures we work with we prove that there are classes of graphs G with complexity at most two such that the size of the powerset graph and its complexity are both exponential in the size of G. We further show that the strategy problem even for simpler reachability games with imperfect information is Exptime-hard on graphs with entanglement and directed path-width at most two. On acyclic graphs, solving reachability games turns out to be Pspace-complete. This shows that bounding the structural complexity of graphs does not substantially decrease the computational complexity of the strategy problem, as long as the amount of imperfect information is unbounded.

In Section 4 we consider parity games with bounded imperfect information. In this case, the graphs which result from the powerset construction have polynomial size. Thus if the construction additionally preserves boundedness of appropriate graph complexity measures, then the corresponding strategy problem is in Ptime. We obtain that the powerset construction, while preserving neither boundedness of entanglement nor of tree-width, does preserve boundedness of directed path-width. The case of DAG-width is much more involved. However, it is also more interesting: DAG-width is bounded in directed path-width, but not the other way around. DAG-width (as well as the other measures) can be defined as a graph searching game where a team of cops tries to capture a robber in the given graph. The player move alternately. The cops occupy some vertices and can change their placement arbitrarily in their move. The robber runs between vertices along cop free paths. The DAG-width of a graph is the minimal number of cops needed to capture the robber in a *monotone* way, i.e., such that the robber can never occupy a vertex that has already been unavailable for him. The problem with DAG-width is now that while capturing the robber is preserved after the applying the powerset construction, the monotonicity is not. For entanglement the monotonicity is not needed, for directed path-width we obtain it for free: if k cops capture the robber, then k cops can also do it in a monotone way [Hun06].

We discuss three approaches to this problem. One of them fails giving us an example (see Theorem 23) which at least partially explains the difficulty with monotonicity for DAG-width. Two other approaches lead to solutions of the problem. The first one is presented in Section 5. We prove that parity games (with perfect information) can be solved efficiently not only if DAG-width is bounded, but also even if *non-monotone* DAG-width is bounded. The idea of the proof is from the solution of the strategy problem for parity games on graphs of bounded DAG-width via *simulated games* by Fearnley and Schewe [FS12]. It turns out that their construction can be used also for the case of bounded non-monotone DAG-width. This relativizes the importance of monotonicity for DAG-width, as the strategy problem for parity games is the only known to become easier if DAG-width is bounded and not known to become easier when the more general directed tree-width is bounded.

The other approach, that we pursue in Section 6 is a generalization of the graph searching game for DAG-width to a game where the cops have to capture *multiple* robbers. The robbers correspond to multiple plays of the parity game with imperfect information that Player 0 considers to be possible in a position. Thus if the amount of imperfect information is at most r, we consider the game with r robbers. The new game also generalizes a similar game on undirected graphs from [RT09] by Richerby and Thilikos. Our setting is, however, different, which makes our main result about the game in a sense more general (we discuss the connection to the game from [RT09] in Section 6). We prove that if k cops can capture r robbers, then kr cops can capture r robbers. This is the technically most involved proof, the main problem is again to preserve monotonicity. However, this result allows us to preserve monotonicity also for DAG-width while translating a cop strategy from the game with imperfect information to a game with perfect information. Thus we establish a connection between imperfect information in parity games and a multiagent graph searching game. Interestingly, if the cops have to capture infinitely many robbers, the game turns out to be equivalent to the game that characterizes directed path-width and is also defined by means of imperfect information. This is the same situation as in [RT09] for the undirected case.

2. Preliminaries

We assume that the reader is familiar with basic notions from the graph theory. All graphs in this work are directed, finite and without multi-edges. (An undirected graph is a graph with a symmetric edge relation.) For sets $X \subseteq V$, $G - X$ denotes the subgraph of G induced by the vertices of G that are not in X. By $\mathrm{Reach}_G(X)$ we denote the set of vertices reachable from X in G. A strongly connected component, or simply a *component*, is a maximal subset of the graph such that, from each vertex to each vertex, there is a path in that subgraph. If U is a set of vertices in the graph G and $v \notin U$ is a vertex, then $\mathrm{C}_U^G(v)$ is the component of $G - U$ containing v. A (directed) rooted tree is an orientation of an undirected tree where all edges are oriented away from a designated vertex, the root. The depth of a rooted tree is the number of vertices on its longest path. For a finite sequence π of elements, $\mathrm{last}(\pi)$ denotes the last element of π. If v is a vertex and E the set of edges, then vE is $\{w \mid (v, w) \in E\}$. If $\sim$ is an equivalence relation, we write $[v]_\sim$ or just $[v]$ for the equivalence class of v. The set of natural numbers is denoted by ω.

2.1. Games

We consider finite two-player zero-sum games with imperfect information and perfect recall, i.e., any play is won by either of the players and both players never forget any information that has already been available for them. The players are called Player 0 and Player 1. Formally, a game *arena* is a tuple $\mathcal{A} = (V, V_0, E)$ where (V, E) is the *game graph*, and $V_0 \subseteq V$ is the set of positions on which Player 0 has to move. Let A be a finite set of *actions*. A *game* is a tuple $\mathcal{G} = (V, V_0, (E_a)_{a\in A}, v_0, \sim, \Omega)$ where $(V, V_0, \bigcup_{a\in A} E_a)$ is an arena with $|vE_a| \leq 1$ for all $v \in V$ and $a \in A$. Thus all edges leaving the same vertex are uniquely labeled and the player who moves at v determines the next position by choosing one of those labels. Furthermore, $v_0 \in V$ is the initial position, and $\Omega \subseteq V^\omega$ is the *winning condition* for Player 0. For convenience, we define $V_1 = V \setminus V_0$ and $E = \bigcup_{a\in A} E_a$. The game graph of $\mathcal{G}$ is $G = (V, E)$. We write $v \xrightarrow{a} w$ if $(v, w) \in E_a$ and $v \xleftrightarrow{a} w$ if $(v, w) \in E_a$ and $(w, v) \in E_a$. For $v \in V$, $\mathrm{act}(v) = \{a \in A \mid vE_a \neq \emptyset\}$. A *play* is a maximal finite or infinite sequence $v_0a_0v_1a_1v_2a_2 \ldots \in (VA)^*V \cup (VA)^\omega$ such that $(v_i, v_{i+1}) \in E_{a_i}$ for all $i \geq 0$. A finite play $\pi = v_0a_0 \ldots v_n$ is won by Player $i \in \{0, 1\}$ if and only if $v_n \in V_{i-1}$ and $v_nE = \emptyset$. An infinite play π is won by Player 0 if and only if $\pi \in \Omega$, otherwise it is won by Player 1.

Common winning conditions are *reachability* (Player 0 wins a play if it reaches a vertex from a given set), *safety* (Player 0 wins if the play never reaches a given set of vertices), or *parity* (the vertices are colored by linearly ordered colors; Player 0 wins if the minimal infinitely often seen color is even).

A *history* is a finite prefix π of a play with $\mathrm{last}(\pi) \in V$. The set of all histories of a game $\mathcal{G}$ is $\mathcal{H}(\mathcal{G})$. Now we can define the last component of a game: $\sim$ is an equivalence relation on $\mathcal{H}(\mathcal{G})$. For $\pi, \pi' \in \mathcal{H}(\mathcal{G})$ we say that Player 0 cannot distinguish between them if $\pi \sim \pi'$.

A *strategy* for Player i is a partial function $g\colon (VA)^*V_i \to A$ and if $i = 0$, then g must be based only on the information available for Player 0: if $\pi \sim \pi'$, then $g(\pi) = g(\pi')$. Let $\pi = v_0a_0v_1a_1v_2\ldots$ be a history or a play. We say that it is *consistent* with g if for all j with $v_j \in V_i$ we have $a_j = g(v_0a_0\ldots a_{j-1}v_j)$. We call a strategy g for Player i *winning* from v_0 if Player i wins every play π in $\mathcal{G}$ from v_0 that is consistent with g. We are interested only in winning strategies for Player 0, so we consider only games where Player 1 has perfect information. If we introduced imperfect information for both players, a non-winning strategy for Player 0 could exist even if there were no winning counter-strategy for Player 1.

In order to speak about decision problems for games of imperfect information we have to represent $\sim$ in a finite way. For that we consider equivalence relations $\sim^V \subseteq V^2$ and $\sim^A \subseteq A^2$ on positions and on actions of the game, respectively, and extend them to $\sim$. In this case we also write $(V, V_0, (E_a)_{a\in A}, \sim^v, \sim^A, \Omega)$ instead of $(V, V_0, (E_a)_{a\in A}, \sim, \Omega)$. Relations $\sim^V$ and $\sim^A$ must satisfy the following conditions. For winning conditions defined by a coloring of the arena vertices we abuse the notation and denote by $\Omega(v)$ the color of vertex v.

1. If $u \sim^V v$, then $u, v \in V_0$ or $u, v \notin V_0$ (Player 0 knows when it is his turn).

2. if for some $v \in V$, $a, b \in \mathrm{act}(v)$ and $a \neq b$, then $a \not\sim^A b$ (Player 0 distinguishes available actions).

3. if $u, v \in V_0$ with $u \sim^V v$, then $\mathrm{act}(u) = \mathrm{act}(v)$ (Player 0 knows which actions are available).

4. if $u \sim^V v$, then $\Omega(u) = \Omega(v)$ (game colors are observable for Player 0).

The equivalence relation $\sim$ on histories is induced by $\sim^V$ and $\sim^A$ as follows. For $\pi = v_0a_0\ldots a_{n-1}v_n$ and $\pi' = w_0b_0\ldots b_{m-1}w_m \in V(AV)^*$, we have $\pi \sim \pi'$ if and only if $n = m$ and $v_j \sim^V w_j$ and $a_j \sim^A b_j$ for all j.

The *winning region* of Player i in $\mathcal{G}$ is the set of all positions $v \in V$ such that Player i has a winning strategy for $\mathcal{G}$ from v.

We say that a class C of games has *bounded imperfect information*, if there is some $r \in \omega$ such that for every game $\mathcal{G} = (V, V_0, (E_a)_{a\in A}, v_0, \sim, \Omega)$ from C and for any position $v \in V$, the equivalence class $[v]_{\sim^V} := \{w \in V \mid v \sim^V w\}$ of v has size at most r. Notice that the equivalence classes $[a]_{\sim^A} := \{b \in A \mid a \sim^A b\}$ of actions $a \in A$ may, however, be arbitrarily large. If $r = 1$, we have a game of perfect information, in which case we omit the component $\sim$ and the actions, so a game with perfect information can be formalized as a tuple (V, V_0, E, v_0, Ω). Being in a positon $v \in V$, a player choses an edge $(v, w) \in E$ and thus determines the next position w. In this case a play is defined in an obvious way analogously to a play in the general case as a sequence of positions.

2.2. Powerset Construction

A usual method to solve games with imperfect information is a powerset construction originally suggested by John H. Reif in [Rei84]. The construction turns a game with imperfect information into a *non-deterministic* game with perfect information such that the existence of winning strategies for Player 0 is preserved.

A *non-deterministic parity game* is defined as a deterministic game, but the condition $|vE_a| \leq 1$ is dropped. Plays, strategies and winning strategies are defined as before. In particular, a strategy is winning for Player 0 if all plays consistent with it are won by Player 0, regardsless which non-deterministic choices are made. In general, even finite non-deterministic games are not determined (i.e., neither of the players may have a winning strategy) and hence not equivalent to deterministic games. However, for each non-deterministic game $\mathcal{G}$ and each player $i \in \{0,1\}$, we can construct a deterministic game $\mathcal{G}^i$ such that the existence of winning strategies for Player i is preserved. The non-determinism can be resolved by giving player $1-i$ control of non-deterministic choices. For any $v \in V$ and any $a \in \text{act}(v)$ we add a unique a-successor of v to the game graph which belongs to player $1-i$ and from which he can choose any a-successor of v in the original game graph. The color of such a new position is the color of its unique predecessor.

Formally, for a parity game $\mathcal{G} = (V, V_0, (E_a)_{a\in A}, v_0, \sim, \Omega)$ where $\sim$ is defined by some $\sim^V$ and $\sim^A$, we construct the *powerset* game $\overline{\mathcal{G}} = (\overline{V}, \overline{V}_0, (\overline{E}_a)_{a\in A}, \overline{v}_0, \overline{\Omega})$ with perfect information. Without loss of generality we always assume that $[v_0] = \{v_0\}$. For $S \subseteq V$ and $B \subseteq A$, let $\text{succ}_B(S) := \{v \in V \mid \text{ there are } s \in S \text{ and } b \in B \text{ such that } b \in \text{act}(s) \text{ and } v \in sE_b\}$. The components of $\overline{\mathcal{G}}$ are defined as follows:

- $\overline{V} = \{\overline{v} \in 2^V \mid \overline{v} \subseteq [u] \text{ for some } u \in V\}$ and $\overline{V}_0 = \overline{V} \cap 2^{V_0}$;
- for all $a \in A$, $\overline{E}_a = \{(\overline{v}, \overline{w}) \mid \overline{w} = \text{succ}_{[a]}(\overline{v}) \cap [u] \text{ for some } u \in \text{succ}_{[a]}(\overline{v})\}$;
- $\overline{v}_0 = \{v_0\}$;
- $\Omega(\overline{v}) = \Omega(v)$ for some $v \in \overline{v}$ (note that colors are observable).

One can see that this construction preserves winning strategies for Player 0. We will always assume that the graph game $\overline{G}$ of $\overline{\mathcal{G}}$, the *powerset graph*, is only the part of the graph reachable from $\{v_0\}$. The following lemma, whose proof is straightforward, states the key property for the correctness of the construction.

Lemma 1. *For each history $\overline{\pi} = \overline{v}_0 a_1 \overline{v}_1 \ldots a_n \overline{v}_n$ in $\overline{\mathcal{G}}$ and all $u_n \in \overline{v}_n$, there is a history $\pi = u_0 a'_1 u_1 \ldots a'_n u_n$ in $\mathcal{G}$ such that $u_i \in \overline{v}_i$ and $a'_i \sim^A a_i$ for all i.*

2.3. Graph searching games

In this section we introduce several measures for structural complexity of graphs, which we define by means of graph searching games. The actions play no role here, so we may assume that the edges are not labeled and the players choose an outgoing edge to determine their move. Hereby Player 0 does not see which edge was chosen by Player 1, he can only distinguish between positions. The games are played by a robber and a team of k cops where k is a parameter of the game. In a position, the robber occupies a vertex and each of the cops either also occupies a vertex or is outside of the graph. In a move, the cops announce their next placement. Then the robber chooses a new vertex that is reachable from his current vertex via paths that do not contain any vertices occupied by cops. In the next position, the robber is on his new vertex and the cops are placed as they have announced. The cops try to capture the robber, i.e., to reach a position where he has no legal move. If they never capture him, the robber wins. Modifications of this basic game define a complexity measure of a graph by the *cop number*: the least number of cops needed to capture the robber.[1]

DAG-width. A *DAG-width game* (or the cops and robber game) $\mathcal{G}_k(G)$ is a game with perfect information [BDH$^+$12]. The game is played on a directed graph $G = (V, E)$, which is different from the game graph, by two players. Cop positions are of the form (U, v) where $U \subseteq V$ is the set of at most k vertices occupied by cops (if $|U| < k$, we say that the rest of the cops is outside of the graph) and $v \in V \setminus U$ is the vertex occupied by the robber. Robber positions are of the form (U, U', v) where U and v are as before and $U' \subseteq V$ is the set of at most k vertices announced by the cops that will be occupied by them in the next position. From a position (U, v), the cops can move to a robber position (U, U', v). From a position (U, U', v), the robber can move to a cop position (U', v') where $v' \in \text{Reach}_{G-(U\cap U')}(v) \setminus U'$. In the first move, the robber is placed on any vertex, i.e., the first move is $\bot \to (\emptyset, v)$ for any $v \in V$. Hereby $\bot$ is an additional dummy first position of any play.

A play of a DAG-width game is *(robber-)monotone* if the robber cannot occupy any vertex that has been already unavailable for him. Formally, the play contains no position (U, U', v) such that some $u \in U \setminus U'$ is reachable from v in $G - (U \cap U')$. A finite play is won by cops if it is monotone. Non-monotone plays and infinite plays are won by the robber.

For a graph G, the least k such that the cops have a winning strategy for the game $\mathcal{G}_k(G)$ is the *DAG-width* $\text{dagw}(G)$ of G, defined in [BDHK06a, Obd06], see also [BDH$^+$12]. The *non-monotone DAG-width* $\text{nm-dagw}(G)$ is the same as DAG-width, but the requirement for the cops to guarantee monotonicity is dropped. We define the *tree-width* $\text{tw}(G)$ as $\text{dagw}(G^{\leftrightarrow}) - 1$, where the game is played on the graph $G^{\leftrightarrow} = (V, E^{\leftrightarrow})$ with $E^{\leftrightarrow} = \{(v, w) \mid (v, w) \in E \text{ or } (w, v) \in E\}$.

Directed path-width. Directed path-width of a graph G is the minimal number of cops minus one that have a monotone winning strategy against an *invisible* robber on G. This is a game with imperfect information for the cop player where cop strategies are functions f that map sequences of cop placements to a next placement: $f : (2^V)^* \to 2^V$. In other words, the *directed path-width game* or the *cops and invisible robber game* is defined as the cops and robber game, but now the equivalence relation contains all pairs of positions. We can also define this game a one-player perfect information game if we assume that the robber

[1] DAG-width, tree-width and directed path-width are usually defined in terms of graph decompositions.

occupies every vertex which is considered by the cops to be possibly occupied. Let $G = (V, E)$ be a graph. Positions of the game have the form (U, U', R) where $|U|, |U'| \leq k$ and $R \subseteq V$. The initial position is $\perp$ and the next one is $(\emptyset, \emptyset, V)$. From a position (U, U', R) the cops can move to any position (U', U'', R') where $R' = \mathrm{Reach}_{G-(U \cap U')}(R) \setminus U''$. A play $(U_0, U_0, R_0)(U_0, U_1, R_1)(U_1, U_2, R_2)\ldots$ is monotone if R_i are monotonically non-increasing. The cops win monotone finite plays, the robber wins (i.e., the cops lose) non-monotone plays and infinite plays. The *directed path-width* of G is the least number k such that $k + 1$ cops have a winning strategy on G.

Obviously, $\mathrm{dagw}(G) \leq \mathrm{dpw}(G) + 1$ for any graph G. Moreover the directed path-width of a graph is not bounded by its DAG-width, that means, there is a class of directed graphs such that the DAG-width is bounded and the directed path-width is unbounded on this class.

Entanglement. In the *entanglement* game [BG05a], in each position, the robber is on a vertex r of the graph. In each round, the cop player may do nothing or place a cop on r, either from outside the graph if there are any cops left or from a vertex v which was previously occupied by a cop and is then freed. No matter what the cops do, the robber must go from his recent vertex r to a new vertex r', which is not occupied by a cop along an edge $(r, r') \in E$. If the robber cannot move, he loses. So formally, the entanglement game on G is a game with perfect information and a position of the entanglement game on G is a tuple (U, r) if it is the cops' turn or a tuple (U, U', r) if it is the robber's turn, with $U' = (U \setminus \{v\}) \cup \{r\}$ for some $v \in U$ (the cop is coming from v to r) or $U' = U \cup \{r\}$ (a new cop from outside is coming to r). From (U, r) the cops can move to a position of the form (U, U', r'). On his turn, the robber can move from (U, U', r) to a position (U', r') where $(r, r') \in E$ and $r' \notin U'$. The entanglement of a graph G, denoted $\mathrm{ent}(G)$ is the minimal number k such that k cops win the entanglement game on G.

It is known that bounded entanglement implies bounded non-monotone DAG-width, but not vice versa [BDHK06b]. It is easy to see that bounded directed path-width implies bounded DAG-width and bounded non-monotone DAG-width, but not vice versa.

Using decompositions to solve parity games. We will measure the complexity of a game by the complexity of its underlying graph, so, e.g., if $\mathcal{G} = (V, V_0, (E_a)_{a \in A}, v_0, \sim, \Omega)$, then $\mathrm{dagw}(\mathcal{G}) = \mathrm{dagw}(V, \bigcup_{a \in A} E_a) = \mathrm{dagw}(G)$.

We defined DAG-width, tree-width and directed path-width in terms of monotone winning strategies. A monotone winning strategy for k cops on G yields a decomposition of G into parts of size at most k which are only sparsely related among each other. (The particular measure determines what "sparsely" precisely means.) Such decompositions often allow for efficient dynamic solutions of hard graph problems.

Entanglement is defined in terms of strategies which are not necessarily monotone and a decomposition in the above sense is known only for $k = 2$, see [GKR09]. Nevertheless, parity games can be solved efficiently on graph classes of bounded entanglement.

Theorem 2 ([Obd03, BDH$^+$12, BG05b])**.** *Parity games can be solved in* Ptime *on classes of graphs of bounded tree-width, DAG-width, (and hence directed path-width), or entanglement.*

Monotonicity costs. In the following, let $\mathcal{M} = \{\mathrm{tw}, \mathrm{dagw}, \mathrm{dpw}, \mathrm{ent}\}$. We say that a measure $X \in \mathcal{M}$ has *monotonicity costs* at most f for a function $f : \omega \to \omega$ if, for any graph G on which k cops have a winning strategy for the X-game on G, $k + f(k)$ cops have a monotone winning strategy for the X-game on G. We say that X has *bounded monotonicity costs* if there is a function $f : \omega \to \omega$ such that X has monotonicity costs at most f. Tree-width has monotonicity costs 0, see [ST93], and the same holds for directed path-width, [Bar06, Hun06]. On the contrary, DAG-width does not have monotonicity costs 0: there is a class of graphs G_n, such that $3n - 1$ cops have a winning strategy on G_n, but $\mathrm{dagw}(G_n) = 4n - 2$, see [KO08a]. Whether DAG-width has bounded monotonicity costs, is an open problem [BDH$^+$12, KO08b].

3. Unbounded imperfect information

If imperfect information is unbounded, then the powerset construction can produce a graph which is super-polynomially larger than the original graph. Moreover, we show that the values of all measures we consider become unbounded and super-polynomial in the size of the given graph.

Let $\mathcal{M} = \{\mathrm{tw}, \mathrm{dagw}, \mathrm{dpw}, \mathrm{ent}\}$ and let G_n be the undirected $n \times n$-grid $G_n = (V_n, E_n)$ with $V_n = \{(i, j) \mid 1 \leq i, j \leq n\}$ and $\big((i_1, j_1), (i_2, j_2)\big) \in E$ if and only if $|i_1 - i_2| + |j_1 - j_2| = 1$. We will need the well-known fact that, for any $n > 1$, we have $X(G_n) \geq n$ for all $X \in \mathcal{M}$.

Proposition 3. *There is a family of games $\mathcal{G}_n$ with imperfect information such that for all $X \in \mathcal{M}$, $X(\mathcal{G}_n) \leq 2$, but $X(\overline{\mathcal{G}}_n)$ is super-polynomial in the size of $\mathcal{G}_n$, where $\overline{\mathcal{G}}_n$ is the powerset graph of $\mathcal{G}_n$.*

Proof. From a very simple graph, we generate a graph containing an undirected square grid of super-polynomial size as a subgraph. This is possible because we can consider large equivalence classes of positions and actions.

Consider a disjoint union of n directed cycles of length 2 with self-loops on each vertex where any two positions are equivalent. Additionally we have an initial position such that, by applying the powerset construction from this position, we obtain a set which contains exactly one element from each cycle. Continuing the construction, we obtain sets that represent binary numbers with n digits and for each digit we have an action which causes exactly this digit to flip. So, using the Gray-code, we can create all binary numbers with n digits by successively flipping each digit. If we do this independently for the first $n/2$ digits and for the last $n/2$ digits, it is easy to see that the resulting positions are connected in such a way, that they form an undirected grid $\overline{G}_n$ of size $2^{n/2} \times 2^{n/2}$, for which we have $X(\overline{G}_n) \geq 2^{n/2}$ for any measure $X \in \mathcal{M}$.

To be more precise, for even $n \geq 2$, let $\mathcal{G}_n = (V_n, V_0 = \emptyset, (E^n_a)_{a\in A_n}, \sim_n, \Omega)$ where $\Omega = \emptyset$, $\sim_n$ is induced by $\sim^V_n$ and $\sim^A_n$ (which we define below) and $G_n = (V_n, E_n = \bigcup_{a\in A_n} E^n_a)$ is the following game graph. The set of vertices is $\{v_0\} \cup \{(i,j) \mid 1 \leq i \leq n, j \in \{0,1\}\}$ where i denotes the number of the cycle and j is the number of a vertex in the cycle. The actions are $A_n = \{a_i \mid 1 \leq i \leq n\} \cup \{\neg_i \mid 1 \leq i \leq n\}$. Here the actions a_i lead from v_0 to the cycles: $v_0 \xrightarrow{a_i} (0,i)$ for $1 \leq i \leq n$. Further actions build the cycles:

- $(i,j) \xrightarrow{\neg_i} (i, 1-j)$ for $1 \leq i \leq n$ and $j \in \{0,1\}$.
- $(i,j) \xrightarrow{\neg_k} (i,j)$ for $1 \leq i \leq n$ with $k \neq i$ and $j \in \{0,1\}$.

Imperfect information is defined by $(i,j) \sim^V_n (k,l)$ and $a_i \sim^A_n a_k$ for any $1 \leq i,j,k,l \leq n$. So each two positions from any two cycles are indistinguishable and each two of the actions a^i_σ are indistinguishable.

In Figure 1, the game $\mathcal{G}_2$ and the powerset game $\overline{\mathcal{G}}_n$ are depicted. The position $\{v_0\}$ of the powerset game is omitted and a position $\{(0,j_1),(1,j_2)\}$ is represented as j_1j_2.

It is clear that $X(\mathcal{G}_n) \leq 2$ for any measure $X \in \mathcal{M}$. Indeed, DAG-width is tree-width plus one and tree-width is one here, because the underlying graphs are undirected trees. The entanglement game is won by two cops: the cops force the robber to v_0 and then one of them occupies v_0. The robber goes into some cycle i and the other cop occupies $(i,0)$. Then the first cop occupies $(i,1)$. In the cops and invisible robber game, one cop is placed on v_0 and then the two other cops visit successively every cycle, so $\mathrm{dpw}(\mathcal{G}_n) = 2$.

Performing the powerset construction on $\mathcal{G}_n$ from v_0 we obtain the graph $\overline{G}_n$. Obviously, $\overline{G}_n$ contains the position $\{(1,0),\dots,(n,0)\}$. From this position, an undirected square grid of super-polynomial size is constructed as follows. The positions of $\overline{G}_n$ (except for $\{v_0\}$) are precisely the sets of vertices of G_n that contain exactly one vertex from every cycle of G_n, i.e., $\overline{V}_n = \big\{\{v_0\}\big\} \cup \big\{\{(1,j_1),\dots,(n,j_n)\} \mid j_i = 0,1\big\}$. Action $\neg_i$ switches the vertex in the ith cycle and lets the other cycles unchanged.

Now we observe how the powerset construction orders the positions of $\overline{G}_n$ in a square grid. We successively apply actions $\neg_i$ for $i \in \{1,\dots,n/2\}$ to create each vertex $\{(1,j_1),\dots,(n/2,j_{n/2}),\ (1,0),\dots,(n,0)\}$ with $j_1,\dots,j_{n/2} \in \{0,1\}$. In each step we can change exactly one j_r to $1-j_r$, so the creation of all these vertices from $\{(1,0),\dots,(n/2,0),(n/2+1,0),\dots,(n,0)\}$ can, for instance, be done using the usual Gray-code for binary numbers: we get the next vertex by applying $\neg_i$ to the previous vertex $\{(1,j_1),\dots,(n/2,j_{n/2}),\ (n/2+1,0),\dots,(n,0)\}$, which changes exactly one position (i,j_i). This undirected path forms the upper horizontal side of the grid. Analogously, by successively applying the actions $\neg_i$ for $i \in \{n/2+1,\dots,n\}$ we can create each vertex $\{(1,0),\dots,(n/2,0),\ (n/2+1,j_{n/2+1}),\dots,(n,j_n)\}$ with $j_{n/2+1},\dots,j_n \in \{0,1\}$ using the Gray-code. This undirected path forms the left vertical side of the grid.

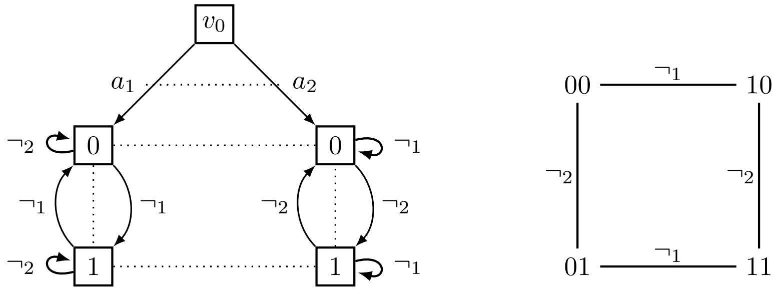

Figure 1: The game $\mathcal{G}_2$ and the powerset graph $\overline{G}^2$

Likewise, given any vertex $\{(1, j_1),\ \ldots,\ (n/2, j_{n/2}),\ (n/2+1, 0),\ \ldots,\ (n, 0)\}$ we can create any vertex $\{(1, j_1), \ldots, (n/2, j_{n/2}), (n/2+1, j_{n/2+1}), \ldots, (n, j_n)\}$ by successively applying the actions $\neg_i$ for $i \in \{n/2+1, \ldots, n\}$ in the same order as before and given any vertex $\{(1, 0),\ \ldots,\ (n/2, 0),\ (n/2+1, j_{n/2+1}),\ \ldots,\ (n, j_n)\}$, by successively applying the actions $\neg_i$ for $i \in \{1, \ldots, n/2\}$, we can create any vertex $\{(1, j_1),\ \ldots,\ (n/2, j_{n/2}),\ (n/2+1, j_{n/2+1}),\ \ldots,\ (n, j_n)\}$. All these paths form a $2^{n/2} \times 2^{n/2}$-grid and therefore, the tree-width of $\overline{G}_n$ is super-polynomial in the size of $\mathcal{G}_n$. Furthermore, using that $\overline{G}_n$ is undirected one easily checks that for all $X \in \mathcal{M} \setminus \{\text{ent}\}$, $X(\overline{G}_n) \geq \text{tw}(\overline{G}_n)$. For entanglement, Berwanger et al. showed in [BDHK06b] that non-monotone DAG-width of a graph (which is at most its tree-width plus one) is at most its entanglement plus one, so $\text{ent}(\overline{G}_n) \geq n + 2$, for $n \geq 3$. □

Remark 4. *The super-polynomial size of the resulting graph is not needed for unbounded growth of graph complexity. By the same technique, replacing n cycles by two undirected n-paths with similar actions and self-loops on all positions leads to an $n \times n$-grid.*

Proposition 3 shows that Reif's construction does not help to solve parity games efficiently even if the game graphs are simple. Before we show that the problem is, in fact, very hard, let us note that on trees, imperfect information does not provide additional computational complexity. The powerset graph of a tree is again a tree (recall that we delete non-reachable positions) where the set of positions on each level partitions the set of positions on the same level of the original tree. Thus the new tree can be computed in polynomial time and is at most as big as the original tree.

For the following proofs we need the notion of an alternating Turing machine. An alternating Turing machine $M = (Q, \Gamma, \Sigma, q_0, \Delta, Q_{\text{acc}}, Q_{\text{rej}})$ is defined as a deterministic Turing machine, but now the set of non-final states is partitioned in Q_{det}, $Q_\exists$ and $Q_\forall$. Whether a word is accepted by M is defined by game semantics. There are two players, both having perfect information: the existential Player $\exists$ and the universal Player $\forall$. If M is in a state from Q_{det}, then there is exactly one next configuration as for deterministic Turing machines. If M is in a state q from $Q_\exists$, the existential player resolves the non-determinism choosing a transition $(q,a) \to (q',a',s) \in \Delta$ and if M is in a state from $Q_\forall$, the universal player moves. The existential player tries to accept the input word, the universal player aims to reject it or to drive M into an infinite computation. A word w is accepted by M if the existential player has a winning strategy from the initial configuration of M on w. The complexity classes APSPACE, ASPACE($S(n)$), APTIME, and ATIME($S(n)$) (for a function $S\colon \omega \to \omega$) are defined using alternating Turing machines as the classes PSPACE, SPACE($S(n)$), PTIME, and TIME($S(n)$) with deterministic Turing machines. Our proofs are based on the following facts, see for example [WW86].

Lemma 5.

(1) APSPACE = EXPTIME.

(2) APTIME = PSPACE.

Theorem 6. *The following problem is* EXPTIME*-hard. Given an imperfect information reachability game $\mathcal{G}$ with* $\mathrm{ent}(\mathcal{G}) \leq 2$ *and* $\mathrm{dpw}(\mathcal{G}) \leq 3$ *and a position v_0, does Player 0 have a winning strategy from v_0 in $\mathcal{G}$?*

Proof. By Lemma 5, for any $L \in$ EXPTIME, there is an alternating Turing machine $M = (Q, \Gamma, \Sigma, q_0, \Delta)$ with only one tape and space bound n^k for some $k \in \omega$, where n is the size of the input, that recognizes L. As usual, Q is the set of states, Γ and Σ are the input and the tape alphabets with $\Gamma \subseteq \Sigma$, q_0 is the initial state, and Δ is the transition relation. First assume that M is deterministic. We describe the necessary changes to prove the general case later.

Let $A = \Sigma \,\dot\cup\, (Q \times \Sigma) \,\dot\cup\, \{\#\}$. Then each configuration C of M is described by a word $C = \#w_1 \dots w_{i-1}(qw_i)w_{i+1} \dots w_t \in A^*$ over A where w_j is the jth symbol on the tape and the reading head is at symbol number i (counting from 0). Since M has space bound n^k and we have $k \geq 1$, without loss of generality we can assume that $|C| = n^k + 1$ for all configurations C of M on inputs of length n. Moreover, for a configuration C of M and $1 \leq i \leq n^k$ the symbol number i of the successor configuration C' only depends on the symbols number $i-1$, i and $i+1$ of C. So there is a function $f : A^3 \to A$ such that for any configuration C of M and any $i \leq n^k$, if the symbols number i, $i+1$ and $i+2$ of C are a_1, a_2 and a_3, then the symbol number i of the successor configuration C' of C is $f(a_1, a_2, a_3)$.

For each input word $u \in \Gamma^*$ we construct a game $\mathcal{G}_u$ with imperfect information such that the player called Constructor has a winning strategy for $\mathcal{G}_u$ if and only if M accepts u. The idea for the game corresponding to u is the following. Player Constructor selects symbols from A such that the sequence constructed in this way forms an accepting run of M on u. In order to check the correctness of the construction, player Verifier may, at any point during the play, but only *once*, memorize some $i \in \{1, \dots, n^k\}$, and a_i, a_{i+1} and a_{i+2} chosen by Constructor within the recent configuration. In the next configuration, Verifier checks the ith symbol chosen by Constructor to be correct according to a_{i-1}, a_i and a_{i+1}, and the function f. If the ith symbol proves incorrect, Constructor loses, otherwise, Verifier loses. If Verifier never checks a transition, Constructor wins if and only if he reaches an accepting configuration. Constructor must not notice when Verifier memorizes the recent position, which defines the imperfect information in the game. Then Constructor has a winning strategy in the game if and only if M accepts u. To justify the bounds on the graph complexity measures that we have claimed, we define the game more formally.

The set of positions is $\{v_0\} \cup \{C, V\} \times A \times \{0, \dots, n^k\} \times Q \times \{-, 1, \dots, n^k\} \times A^3$, so a position has the form $(\sigma, a, i, q, j, a_1, a_2, a_3)$ where σ is the player to move, a is the recent symbol chosen by Constructor and i is the number of a in the recent configuration. Furthermore, q is the last state in Q chosen by Constructor, and j and a_1, a_2, a_3 represent the information memorized by Verifier: j is the number of the symbol to be verified in the next configuration, and a_1, a_2 and a_3 are symbols number $j-1$, j and $j+1$, respectively . All actions are indistinguishable for Constructor and we omit them in the description. The sign $-$ in the four last components of a position means that Verifier did not memorize the corresponding element.

A play begins in position v_0, which belongs to Verifier. He moves to a position $(C, \#, 0, q_0, -, -, -, -)$ or to position $(C, \#, 0, q_0, j, a_1, a_2, a_3)$ where $1 \leq j \leq n^k$ and a_1, a_2, a_3 are symbols number $j-1$, j and $j+1$ of the initial configuration of M on u.

As long as Verifier does not memorize any symbol, Constructor moves from position $(C, a, i, q, -, -, -, -)$ with $0 \leq i < n^k$ to some position $(V, a', i+1, q', -, -, -, -)$ choosing the next symbol and giving Verifier the possibility to memorize it. Hereby either $a' = (q', a'') \in Q \times \Sigma$ (for some $a'' \in \Sigma$), or $a' \in \Sigma$ and $q' = q$. As Verifier does not memorize anything yet, he chooses $(C, a', i+1, q, -, -, -, -)$ as the next position (the other possible move is to memorize a'). If $i = n^k$, then the next position is $(C, \#, 0, \perp, -, -, -, -)$, i.e., Constructor chooses $\#$ and Verifier does not memorize it. Hereby $\perp$ is some fixed state in Q, i.e., the state q is forgotten in this move. We need this to reduce the structural complexity of the game graph. A move of Constructor and an answer of Verifier constitute a round.

Now assume that Verifier decides to memorize the tuple (a_1, a_2, a_3) where a_1 is the current symbol number $i < n^k$, and a_2 and a_3 are the (yet not determined) symbols that will be chosen in the next two rounds. Then from a position $(V, a_1, i, q, -, -, -, -)$ Verifier moves to $(C, a_1, i, q, -, a_1, -, -)$. Then Constructor moves to some $(V, a_2, i+1, q', -, a_1, -, -)$ (where the update of q to q' is as before), then Verifier moves to $(C, a_2, i+1, q', i+1, a_1, a_2, -)$ and Constructor moves to some $(V, a_3, i', q'', i+1, a_1, a_2, -)$ where q' is again updated as before and i' depends on i. If $i+2 \leq n^k$, then $i' = i+2$. Otherwise $i+2 = n^k+1$, then $i' = 0$ (and $a_3 = \#$). Verifier moves to $(C, a_3, i', q'', i+1, a_1, a_2, a_3)$. From this position, the players, first, finish the current configuration and, second, play in the next configuration until the position with index $i+1$ is reached, both in the same way as they played without any memorized information. Formally, we just substitute in the above positions the four last elements $(\dots, -, -, -, -)$ by $(\dots, i+1, a_1, a_2, a_3)$. When a position $(C, b, i+1, s, i+1, a_1, a_2, a_3)$ is reached, the play stops and Verifier wins if and only if $f(a_1, a_2, a_3) = b$. At any *other* position $(\sigma, a, i, q, j, a_1, a_2, a_3)$ (where j and all a_k can be $-$), if q is accepting, Constructor wins and if q is rejecting, Verifier wins. In the remaining case of an infinite play (Verifier never memorizes anything and no final state is reached), Verifier wins.

Imperfect information is defined by making all positions $(\sigma, a, i, q, j, a_1, a_2, a_3)$ and $(\sigma', a', i', q', j', a_1', a_2', a_3')$ indistinguishable for Constructor if $\sigma = \sigma'$, $a = a'$, $i = i'$, and $q = q'$, i.e., Constructor does not know whether Verifier memorized anything.

It is clear that u is accepted by M if and only Constructor has a winning strategy in the game $\mathcal{G}_u$. If u is accepted, then Constructor just constructs the accepting run of M. If not, M rejects (as M recognizes an EXPTIME language, it always stops). In order not to lose by reaching a rejecting state, Constructor has to cheat. However, cheating is not a winning strategy for Constructor because Verifier can memorize the place in the previous configuration that does not match the same place in the current configuration and win.

We now analyze the structural complexity of the game graph, see Figure 2. The *main subgame* $\mathcal{S}$ consists of positions of the form $(\sigma, a, i, q, -, -, -, -)$ without memorization that build a DAG with a unique root $(C, \#, 0, \bot, -, -, -, -)$ and $2{\cdot}n^k{+}1$ layers. A layer number i with an even i has the form $(C, a, i, q, -, -, -, -)$. From every such position there is an edge to every position of the form $(V, a', i + 1, q', -, -, -, -)$ of the next layer. Analogously, from every position of layer number $i + 1$ there is an edge to every position of layer number $i + 2$. Finally, from every position of the last layer, there is an edge back to the root $(C, \#, 0, \bot, -, -, -, -)$. This constitutes the only cyclicity in the graph. Additionally, there are edges from v_0 to every position $(V, a, 1, \bot, -, -, -, -)$.

From every of n^k Verifier positions P in the main subgame and from v_0, Verifier can start memorizing information. Then the play continues in a *checking subgame* $\mathcal{C}_P$ and never returns to the main subgame, so we can consider their complexities independently. Every checking subgame is again a DAG, which consists of two sub-DAGs. The first one is a copy of the remaining part of the main subgame (with changed four last components); the other one is a copy of the part of the main subgame which has been played until Verifier intended to memorize information (again with changed four last components). There are no outgoing edges from the last level of a checking subgame.

It is clear that $\operatorname{ent}(\mathcal{G}_u) \leq 1$ (place the cop on the root $(C, \#, 0, \bot, -, -, -, -)$ and wait until the robber reaches a leaf of the resulting DAG) and $\operatorname{dpw}(\mathcal{G}) \leq 1$ (place one cop on the root and capture the robber with the other cop on the resulting DAG). Notice that we are still considering the special case where M is deterministic. Obviously, $\mathcal{G}_u$ can be constructed from a given input $u \in \Gamma^*$ in polynomial time.

Now consider the general case, where M is not necessarily deterministic. We let Constructor play the role of the existential player and Verifier the role of the universal player. As before, Constructor writes symbols of the current configuration (now including existential choices) and Verifier checks that the current configuration can follow the previous one. However, if we let Constructor check universal choices of Verifier in the same way (by privately remembering a place

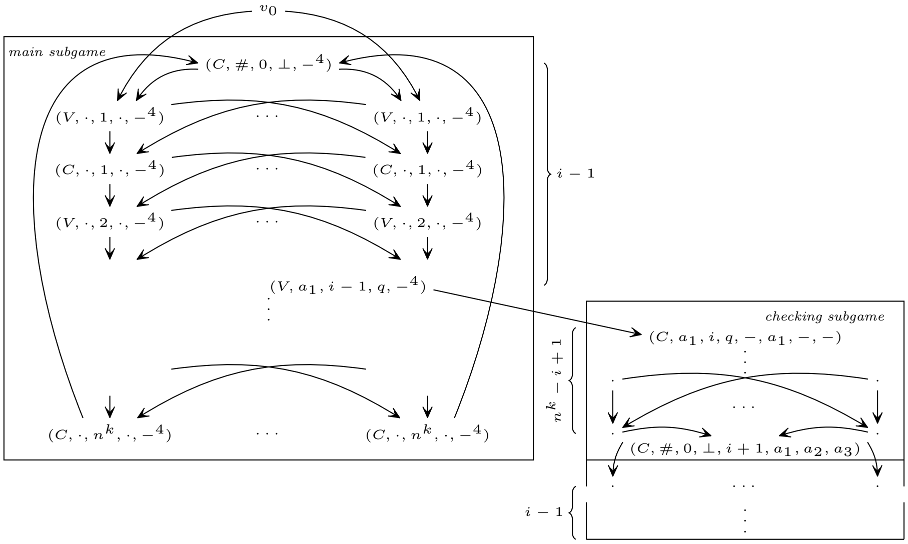
v_0
main subgame
$(C, \#, 0, \bot, -^4)$
$(V, \cdot, 1, \cdot, -^4)$
$\cdots$
$(V, \cdot, 1, \cdot, -^4)$
$(C, \cdot, 1, \cdot, -^4)$
$\cdots$
$(C, \cdot, 1, \cdot, -^4)$
$(V, \cdot, 2, \cdot, -^4)$
$\cdots$
$(V, \cdot, 2, \cdot, -^4)$
$i - 1$
$(V, a_1, i - 1, q, -^4)$
$(C, \cdot, n^k, \cdot, -^4)$
$\cdots$
$(C, \cdot, n^k, \cdot, -^4)$
checking subgame
$(C, a_1, i, q, -, a_1, -, -)$
$n^k - i + 1$
$\cdots$
$(C, \#, 0, \bot, i + 1, a_1, a_2, a_3)$
$\cdots$
$i - 1$

Figure 2: The game graph of $\mathcal{G}_u$ (with only one checking subgame). "$-^4$" is short for "-,-,-,-".

in the previous configuration), the reduction to the games does not work. Indeed, it can happen that M accepts u, but Constructor has no winning strategy: he does not know which place in which configuration he should remember. For this reason, we explicitly remember the last choice of Verifier in the position of the game.

Without loss of generality we can assume that each non-terminal configuration of M has exactly two successor configurations. If there is a configuration C with just a single successor configuration, then we add a default successor to C which leads to acceptance if C is universal and which leads to rejectance if C is existential. If there is a configuration with $b > 2$ successors, then we replace this b-branching by a binary branching configuration tree of depth b by modifying the transition function of M in an appropriate way. Obviously, this construction can be done in such a way that it merely increases the state space of M and the time bound by a constant factor, but not the space bound.

Now, instead of one function f, we have two functions $f_1, f_2 : A^3 \to A$, such that the following holds. If C is a configuration of M, $s \in \{1, 2\}$ and $1 \leq i \leq n^k$, and the symbols number i, $i + 1$ and $i + 2$ of C are a_1, a_2, a_3, then the symbol number $i + 1$ of the successor configuration number s (there are two successor configurations) of C is $f_s(a_1, a_2, a_3)$. Thus, the main subgame $\mathcal{S}$ and every checking subgame $\mathcal{C}_P$ are replaced by two copies $\mathcal{S}^s$ and $\mathcal{C}_P^s$ for $s \in \{0, 1\}$. Thus every position except v_0 has an additional component 0 or 1, which we make the first one, so a position has the form $(s, \sigma, a, i, q, j, a_1, a_2, a_3)$. Intuitively, the previous non-deterministic (existential or universal) choice is memorized in the first component of a position.

Edges from v_0 to $\mathcal{S}$ go now to both copies. Edges from the leaves of $\mathcal{S}$

to its root go now from leaves of both subgames to the roots of both subgames (thus introducing new cycles). If the state of the current configuration is universal, the leaf positions now belong to Verifier, i.e., we have positions $(s, \sigma, a, n^k, q, j, a_1, a_2, a_3)$ where $\sigma = C$ if q is existential and $\sigma = V$ if q is universal. The edges are thus $(s, P, a, n^k, j, a_1, a_2, a_3)$ to $(s', C, \#, 0, j, a_1, a_2, a_3)$ for $s, s' \in \{0, 1\}$. The edges in the checking subgames are changed analogously (without introducing new cycles, because there are no edges from the leaves to the roots).

Imperfect information is defined as before with the additional condition that Constructor observes the copy of the subgame in which the play currently takes place.

Clearly these modifications merely increase the entanglement of the graph from at most 1 to at most 2 (place two cops on both roots of $\mathcal{S}^0$ and of $\mathcal{S}^1$). The directed path-width is now at most 2 (place two cops on the roots and use the third cop to capture the robber on the resulting DAG). □

Remark 7. *The (undirected) path-width and the tree-width of the game graph are also bounded. Both* $\mathcal{S}^s$ *have edges only from one layer to the next one and from the leaves to both roots. Each layer has* $|\{0,1\} \times \{C, V\} \times A \times Q| = 4 \cdot |A| \cdot |Q|$ *elements, so* $8 \cdot |A| \cdot |Q| + 2$ *cops capture the robber in* $\mathcal{S}^s$ *by blocking both roots and occupying one layer after another successively. In* $\mathcal{C}_P$ *the layers are larger and have size at most* $4 \cdot |A| \cdot |Q| \cdot |A|^3$ *(note the last but four component* j *is fixed and depends only on* P*). Hence,* $8 \cdot |A|^4 \cdot |Q|$ *cops capture the robber there. If the robber is visible,* $k_1 = 8 \cdot |A|^4 \cdot |Q|$ *suffice, because if the robber goes to some* $\mathcal{C}_P$*, then a cop occupies* P *and there is no way back for the robber from* $\mathcal{C}_P$*. If the robber is invisible, the cops search every* $\mathcal{C}_P$ *immediately after occupying* P*. In the meanwhile, one layer in* $\mathcal{S}^s$ *must remain blocked, so the cops can get along with* $k_2 = 8 \cdot |A^4| \cdot |Q| + 4 \cdot |A| \cdot |Q|$ *cops. Assuming that* M *recognizes an* Exptime*-complete problem,we obtain that the strategy problem for reachability games with imperfect information on graphs of tree-width at most* k_1 *and path-width at most* k_2 *is* Exptime*-hard.*

The cases of entanglement and directed path-width at most 1 remain open for reachability games, but we can solve them for *sequence-forcing* games. A sequence-forcing condition can be described by a pair (S, Ω) where $\Omega : V \to C \subset \omega$ is a coloring of game positions by natural numbers and $S \subseteq \{1, \ldots, r\}^k$ is a set of sequences of length k for some $k \in \omega$. Player 0 wins an infinite play π of a sequence-forcing game if for some $i \in \omega$ we have $\Omega(\pi(i))\,\Omega(\pi(i+1)) \ldots \Omega(\pi(i+k)) \in S$. Clearly if k is fixed, sequence-forcing games can be polynomially reduced to reachability games by using a memory which stores the last k colors that have occurred. (Notice that this reduction may, however, increase the complexity of the game graph.) In particular, the strategy problem for sequence-forcing games with fixed k is in Ptime. On the other side, the strategy problem for sequence-forcing games with imperfect information is Exptime-hard on graphs of entanglement and directed path-width at most 1, already for $k = 3$.

Theorem 8. *Sequence-forcing games with imperfect information on graphs of entanglement and directed path-width at most* 1 *are* Exptime*-complete.*

Proof. We modify the proof of Theorem 6 as follows. From the nodes on level $2 \cdot n^k + 1$ of $\mathcal{S}^s$ for $s \in \{0, 1\}$ we do not allow moves directly back to the roots, but we redirect all edges to a single (new) position 0, which is common for both $\mathcal{S}^s$, belongs to Verifier and has color 0. From this position, Verifier may move to position 2, which belongs to Constructor and has color 2, or to position 1, which belongs to Verifier and has color 1. From 0 Constructor chooses whether to proceed in $\mathcal{S}^0$ or in $\mathcal{S}^1$ and from 1 Verifier makes this choice. So as Constructor does not notice where the play proceeds in the main subgame or in some checking subgame, the same construction is performed in the checking subgames at places where the configurations of M change and imperfect information is defined accordingly. All old positions obtain color 0 except for positions $(s, \sigma, a, i, q, -, -, -, -)$ on the last levels of $\mathcal{S}^s$ where q is universal: they are colored with 1.

Now, $S = \{(0, 0, 1)\}$, that means, the unique sequence that Constructor wants to enforce is $(0, 0, 1)$. This forces Verifier into giving control back to Constructor if the state in the recent configuration is existential. Then the proof of Theorem 6 carries over. Note that a player still wins if his opponent has move, but is unable to do it, in particular, the players win at their old winning positions.

If a cop occupies position 0 in the modified game, the game graph becomes acyclic, so the entanglement of the whole graph is 1 and its directed path-width is 2. □

Finally, if the we consider acyclic game graphs, the strategy problem for imperfect information reachability games is Pspace-complete. Notice that acyclic graphs are precisely those having DAG-width 1.

Theorem 9. *The strategy problem for reachability games with imperfect information on acyclic graphs is* Pspace*-complete.*

Proof. First we prove the membership in Pspace. Let $\mathcal{G}$ be a game on an acyclic graph with imperfect information and let v_0 be the initial position. The idea is that carrying out the powerset construction on an acyclic graph G we again obtain an acyclic graph $\overline{G}$ where by Lemma 1, the paths in $\overline{G}$ are not longer than the paths in G, so we can solve the reachability game on $\overline{G}$ by an APtime algorithm. Starting from $\{v_0\}$, we proceed as follows. Given a position $\overline{v} \in \overline{V}$ in the corresponding game $\overline{G}$ with perfect information, if $\overline{v} \in \overline{V}_0$, then the existential player guesses a successor of $\overline{v}$ and if $\overline{v} \in \overline{V}_1$, then the existential player chooses a successor position of $\overline{v}$. If the computation reaches a leaf node in $\overline{V}_1$, the algorithm accepts and if the computation reaches a leaf node in $\overline{V}_0$, the algorithm rejects. The construction of a successor position of some position $\overline{v}$ can obviously be done in polynomial time. Moreover, if $\overline{\pi} = \overline{v}_0, \overline{v}_1, \ldots, \overline{v}_k$ is any path in $\overline{G}$, then according to Lemma 1, there is a path $\pi = v_0, v_1, \ldots, v_k$ with $v_i \in \overline{v}_i$ for $i \in \{0, \ldots, k\}$. Since G is acyclic, $k \leq n$. So, the computation stops after at most n steps.

Conversely, let $L \in$ PSPACE be some decision problem. Then, according to Lemma 5, there is an alternating Turing machine M with only one tape and time bound n^k for some $k \in \omega$ that recognizes L. We use the same construction as in the proof of Theorem 6. Since M has time bound n^k and only a single tape, M has also space bound n^k. So we can describe configurations of M in the very same way as in the proof of Theorem 6 and we can construct a game with positions as before. However, the essential difference here is that at a leaf position of $\mathcal{S}^s$, the next move does not lead back to the top of $\mathcal{S}^s$ or $\mathcal{S}^{1-s}$ (for $s \in \{0, 1\}$), but it leads to the roots of new copies of $\mathcal{S}^0$ and $\mathcal{S}^1$. This chain of copies of $\mathcal{S}^s$ stops after n^k steps.

If some input u is accepted by M, then Constructor can prove this by constructing at most $|u|^k$ configurations, so winning strategies carry over between the game constructed in the proof of Theorem 6 and the game constructed here. Moreover, since the graph we have constructed is acyclic by definition, the proof is finished. □

4. Bounded imperfect information

We turn to the case where the size of the equivalence classes of positions is bounded. We show that tree-width and entanglement become unbounded after the application of the powerset construction, but non-monotone DAG-width and directed path-width do not. The more difficult case of DAG-width is treated in Sections 5 and 6.

4.1. Negative results

The first observation is that bounded tree-width may become unbounded when applying the powerset construction. Afterwards we will see, that the same result holds for entanglement.

Proposition 10. *For every $n > 3$, there are games $\mathcal{G}^n$, with bounded imperfect information and tree-width and directed path-width 1, and DAG-width, Kelly-width and entanglement 2 such that the corresponding powerset games $\overline{\mathcal{G}}^n$ have unbounded tree-width.*

Proof. The game graph of $\mathcal{G}^n$ is a disjoint union of n undirected paths of length n together with another vertex v_0^n and directed edges from v_0^n to every other vertex. Imperfect information connects vertices from neighbor paths. The graph G^4 (without v_0^4) is shown in Figure 3 (on the left). Formally for any even natural number $n > 3$, let $\mathcal{G}^n = (V^n, V_0^n, E^n, v_0^n, \sim^{V,n}, \sim^{A,n}, \Omega)$ be the following game:

- $V^n = V_1^n = \{v_0^n\} \cup \{(i,j) \,|\, 1 \le i, j \le n\}$, i.e., $V_0^n = \emptyset$;
- actions play no role and we do not consider them;
- $E^n = \{\big(v_0^n, (i,j)\big) \mid 1 \le i, j \le n\} \cup \{\big((i,j), (i+1,j)\big), \big((i+1,j), (i,j)\big) \mid 1 \le i, j \le n\}$;
- $\sim^{A,n} = V^n \times V^n$ (Player 0 does not distinguish any actions), and for $i < n$,

  - if i is odd and j is even, then $(i,j) \sim^{V,n} (i+1,j)$,
  - if i is even and j is odd, then $(i,j) \sim^{V,n} (i+1,j)$;

- $\Omega = \emptyset$ (the winning condition does not play any role here).

The values of directed measures for $\mathcal{G}^n$ are clear. For entanglement the strategy is to chase the robber with one cop until he goes to the right. Then the play proceeds in rounds. In a round one cop (at the beginning the first cop) is a left bound for the robber movements. The other cop chases the robber until he goes to the right. Continuing in this way two cops capture the robber.

The powerset graph $\overline{G}^n$ has a structure similar to the Gaifman graph of $\mathcal{G}^n$. It has the same paths whose vertices have the form $\{(i,j)\}$, for $1 \leq i,j \leq n$, and are now connected by a gadget consisting of a new vertex $\big\{(i,j),(i\pm 1,j)\big\}$ (hereby, ±1 depends on parities of i and j) and directed edges going from that vertex to the row above and to the row below. A connection is in an odd column if the lower row is odd and in an even column if the lower row is even (starting with the odd row 1), see Figure 3 (the graph on the right).

Formally, $\overline{\mathcal{G}}^n = (\overline{V}^n, \overline{V}_0^n, \overline{E}^n, \overline{v}_0^n)$ (we omit actions, the absent imperfect information and the winning condition that play no role) where

- the positions are defined by

$$\overline{V}^n = \big\{\{v_0^n\}\big\} \cup \big\{\{(i,j)\} \mid (i,j) \in V^n\big\} \cup \big\{\{(i,j),(i+1,j)\} \mid i+j = 1 \mod 2\big\}\,,$$

- no positions belong Player 0: $\overline{V}_0^n = \emptyset$,

- the moves are

$$\begin{aligned}\overline{E}^n = &\Big\{\big(v_0^n, \{(i,j)\}\big) \mid 1 \leq i,j \leq n\}\\ &\cup\big\{\big(v_0^n, \{(i,j),(i+1,j)\}\big) \mid i+j = 0 \mod 2\big\}\\ &\cup\Big\{\big(\{(i,j),(i+1,j)\}, \{(i+b,j+c)\}\big) \mid b \in \{0,1\}, c \in \{1,-1\}, \text{ and}\\ &\qquad\qquad \{(i+b,j+c)\} \in \overline{V}^n, (i,j) \sim^{V,n} (i+1,j)\}\Big\}\,,\end{aligned}$$

  and

- the starting position is $\overline{v}_0^n = \{v_0^n\}$.

We show that $\overline{G}^n$ has an $(m \times m)$-grid as minor where $m = n$ if n is even and $m = n-1$ if n is odd. We cut off $\overline{v}_0^n$, and if n is odd, we cut off the nth column. Further, we delete edges $\big(\{(i,j),(i+1,j)\},\{(i+b,j+1)\}\big)$ if i is odd and $\big(\{(i,j),(i-1,j)\},\{(i+b,j+1)\}\big)$ if i is even. The result is shown in Figure 3 (the graph on the right). Now, the directions of edges are forgotten, i.e., instead of edges $\big\{\{(i,j),(i+1,j)\},\{(i+b,j+c)\}\big\}$ we have edges $\big\{\{(i,j),(i+1,j)\},\{(i+b,j+1)\}\big\}$. We obtain a *wall-graph* defined in [Kre09] where it is shown that such graphs have high tree-width. Indeed, we contract edges $\big((i,j),(i,j+1)\big)$ for all i and add j. The result is an $(m/2 \times n-1)$-grid, from which it is easy to obtain an $(m \times m)$-grid by further edge contractions. It is well known that the tree-width of an $(m \times m)$-grid is m. □

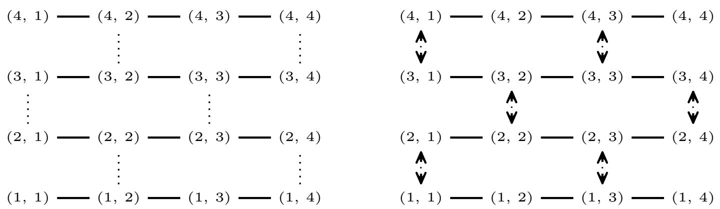

Figure 3: Game graph G^4 (without v_0) and a subgraph of its powerset graph $\overline{G}^4$.

Note that if we consider the whole game structure, i.e., the Gaifman graph of the given game, it is almost of the same shape as the powerset graph and its tree-width is unbounded as well. In fact, we will see in Corollary 24 that the tree-width of a powerset graph is bounded in the tree-width of the Gaifman graph of the given game.

Proposition 11. *For every $n > 3$, there are games $\mathcal{G}^n$, with bounded imperfect information such that* $\operatorname{ent}(\mathcal{G}^n) = 2$ *and the corresponding powerset games $\overline{\mathcal{G}}^n$ have unbounded entanglement.*

Proof. The game graph of $\mathcal{G}^n$ (see Figure 4) consists of two disjoint copies T_1 and T_2 of the full undirected binary tree of depth n. From a vertex in T_1, a path of length two leads to the corresponding vertex in T_2 and there are no paths from T_2 to T_1. The paths from T_1 to T_2 are supplied with imperfect information in such a way that in the powerset graph there appear connections also from T_2 to T_1. Thus, in $\overline{\mathcal{G}}_n$, corresponding vertices are now connected in both directions.

Let $n \in \omega$ be even. We define the game $\mathcal{G}^n = (V^n, V_0^n, E^n, v_0^n, \sim^{V,n}, \sim^{A,n}, \emptyset)$ where $\sim^{A,n}$ plays no role, so we do not define it. Let α be the mapping $\{0,1\} \to \{a,b\}$ with $0 \mapsto a$, $1 \mapsto b$ and let β be the mapping $\{a,b\} \to \{\overline{0},\overline{1}\}$, $a \mapsto \overline{0}$, $b \mapsto \overline{1}$. We generalize α to words: $\alpha(u_1 \dots u_n) = \alpha(u_1) \dots \alpha(u_n)$, and analogously for β. The components of the game can now be defined as follows.

- $V^n = V_1^n = \{v_0^n\} \cup T_1 \cup T_2 \cup a\{a,b\}^{<n}$ where $T_1 = 0\{0,1\}^{<n}$ and $T_2 = \overline{0}\{\overline{0},\overline{1}\}^{<n}$ (so $V_0^n = \emptyset$);
- E^n has edges
  - $v_0^n \to 0$,
  - $u \to u0$ and $u \to u1$ for any $u \in 0\{0,1\}^{<n-1}$,
  - $u \to u$ for any $u \in 0\{0,1\}^{<n}$,
  - $u\overline{0} \to u\overline{0}$ and $u\overline{1} \to u\overline{1}$ for any $u \in \overline{0}\{\overline{0},\overline{1}\}^{<n-1}$,
  - $u \to \alpha(u)$ and $\alpha(u) \to u$ for any $u \in 0\{0,1\}^{<n}$,
  - $u \to \beta(u)$ for any $u \in a\{a,b\}^{<n}$;
- $u \sim^n \beta(\alpha(u))$, for any $u \in 0\{0,1\}^{<n}$.

In the informal description above, T_1 is induced by vertices in $0\{0,1\}^{n-1}$ and T_2 by vertices in $\overline{0}\{\overline{0},\overline{1}\}^{n-1}$. Intermediate vertices are those from $a\{a,b\}^{n-1}$.

Clearly, tree cops can capture the robber on G^n, so $\operatorname{tw}(G^n) = 2$. Let us convince ourselves that $\operatorname{ent}\mathcal{G}^n = 2$. First, two cops are needed already on the subgraph induced by 0 and 00. On the other hand, 2 cops suffice to capture the robber. The strategy is to play on T_1 in a top-down manner. The robber choses a branch of T_1 and the cops play on that branch as on the path in the proof of Proposition 10. Finally the robber is forced to visit an intermediate vertex $\alpha(u)$ for some u. Note that the cops are placed on the robber vertex in every move, hence when the robber is on $\alpha(u)$, u is occupied by a cop, which forces the robber to proceed to T_2 in the next move. On T_2, he is captured in the same way as on T_1.

The powerset graph $\overline{G}^n$ (see Figure 4) has $\{0\}$ as a position and therefore also $\{a\}$ and $\{0,\overline{0}\}$. From $\{0\}$, one possibility is to remain in $\{0\}$, another is to go to $\{a\}$. In $\mathcal{G}^n$, from a, there are edges to 0 and to $\overline{0}$, which are indistinguishable, so, in $\overline{G}^n$, there is an edge $\{a\} \to \{0,\overline{0}\}$. From 0, the pebble can return to a and both from 0 and from $\overline{0}$ it can move to 0, so in $\overline{G}^n$ we have edges $\{0,\overline{0}\} \to \{a\}$ and $\{0,\overline{0}\} \to \{0\}$. The described structure is repeated in the lower levels, because from $\{u\}$, for $u \in 0\{0,1\}^{<n-1}$, there is an edge to $\{u0\}$ and to $\{u1\}$, and analogously for $\{\overline{u}\}$.

Essentially, $\overline{G}^n$ has the same vertices as $\mathcal{G}^n$. We can identify $u \in 0\{0,1\}^{<n}$ with $\{u \in 0\{0,1\}^{<n}\}$, $\alpha(u)$ with $\{\alpha(u)\}$, and $\beta(\alpha(u))$ with $\{\beta(\alpha(u))\}$.

It remains to prove that the entanglement of the powerset graphs is unbounded. We adapt the proof from [BGKR12] for similar graphs and show that $\operatorname{ent}(\overline{\mathcal{G}}^n) \geq n/2 - 1$. In the following, we identify vertices u and $\alpha(u)$ for simplicity of explanation, which, obviously, does not change the entanglement.

We show by induction on n that for every even n, the robber can starting from vertex 0 or from vertex $\overline{0}$

- escape $n/2 - 2$ cops and
- after the $(n/2-1)$th cop enters $\overline{\mathcal{G}}^n$,
  - if started in 0, reach $\overline{0}$, and
  - if started in $\overline{0}$, reach 0.

This suffices to prove unboundedness, as the robber has a winning strategy on $\overline{\mathcal{G}}^{n+1}$ in this case: he switches between the two subtrees of the root.

For $n = 2$, it is trivial. Assume that the statement is true for some even n and consider the situation for $n + 2$. We need two strategies: one for 0 as the starting position and one for $\overline{0}$. By symmetry, it suffices to describe only a strategy for 0. For a word $u \in \{0,1\}^{\leq n+1} \cup \{\overline{0},\overline{1}\}^{\leq n+1}$, let T^u be the subgraph induced by the subtree of T_1 rooted at u and by the corresponding subtree of T_2. The robber can play in a way such that the following invariant is true.

> If the robber is in T^{0xy}, for $x, y \in \{0,1\}$, and starts from $0y$, there are no cops on $\{\overline{0}, \overline{0x}\}$.

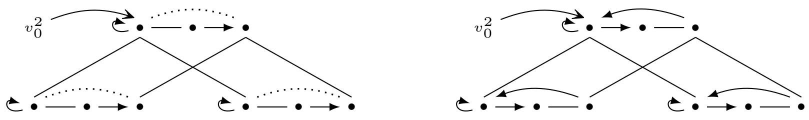

Figure 4: Game graph $\mathcal{G}_2$ and its powerset graph $\overline{G}^2_{v_0}$.

By induction, it follows from the invariant that $\overline{0}$ and $\overline{0y}$ are reachable for the robber.

At the beginning, the robber goes to the (cop-free) subtree T^{000} via the path $(0, 00, 000)$ and plays there from 000 according to the strategy given by the induction hypothesis for T^{000}. Also by induction, $\overline{000}$ remains reachable and thus so is $\overline{0}$ via $\overline{00}$. Either that play lasts for ever (and we are done), or the $(n/2 - 2)$nd cop comes to T^{000} and the robber can reach $\overline{000}$. While he is doing that, no cops can be placed outside of T^{000} as the robber does not leave T^{000}.

Assume that the robber enters a tree T^{0xy}, for $x, y \in \{0, 1\}$ which is free of cops (which is, in particular, the case at the beginning). By symmetry, we can assume that $x = y = 0$. Further, assume without loss of generality that the robber enters T^{000} at 000. Either the play remains in T^{000} infinitely long (and we are done), or the $(n/2 - 1)$-st cop enters T^{000} and the robber reaches $\overline{000}$. Note that while the robber is moving towards $\overline{000}$, no cops can be placed outside of T^{000} as the robber does not leave T^{000}.

If the last cop is already placed, the robber goes to $\overline{00}$ and then to $\overline{0}$, which are not occupied by cops by the invariant, and we are done. If the last cop is not placed yet, all cops are in T^{000}, so the robber runs along the path $\overline{000}, \overline{00}, 00, 0, 01, 010$ to T^{010}. Note that the vertices $\overline{0}$ and $\overline{01}$ are not occupied by cops, so the invariant is still true. The robber plays as in T^{000} and so on. □

4.2. Some positive results

Now we prove that in contrast to tree-width and entanglement, *non-monotone* DAG-width is preserved by the powerset construction.

Proposition 12. *Let $\mathcal{G} = (V, V_0, E, v_0, \sim, \Omega)$ be a parity game with imperfect information such that the size of the $\sim$-classes is bounded by some r. If* $\text{nm-dagw}(G) \le k$, *then* $\text{nm-dagw}(\overline{G}) \le k \cdot r \cdot 2^{r-1}$.

Proof. First, we describe our proof idea informally. We follow a play on G that corresponds to a set of at most r plays on $\overline{G}$ which Player 0 considers possible in the parity game with imperfect information. We translate robber moves from $\overline{G}$ to the plays on G, look for the answers of the cops prescribed by their winning strategy for G and translate them back to $\overline{G}$ combining them into one single move.

A position in the parity game on $\overline{G}$ corresponds to at most r positions in the parity game on G, so if the robber occupies a vertex $[v]$ in $\overline{G}$, we consider, for any $w \in [v]$, the possibility that the robber occupies w in G. Some plays we considered until some position may prove to be impossible when the play evolves, some plays may split in multiple plays. For any robber move to $w \in [v]$, the strategy for the cops in the game on G supplies an answer, moving the cops from C_w to C'_w. All these moves are translated into a move in which the cops occupy precisely the vertices of $\overline{G}$ that include a vertex from some C'_w. These moves of the cop player on $\overline{G}$ can be realized with $k \cdot r \cdot 2^{r-1}$ cops.

The rough argument why robber moves can indeed be translated from $\overline{G}$ to G is that, by Lemma 1, for any path $\overline{u}^0, \overline{u}^1, \ldots, \overline{u}^t$ in $\overline{G}$ and for any $u^t \in \overline{u}^t$, there is a path $u^0, u^1, \ldots, u^t$ in G such that $u^i \in \overline{u}^i$ for any $i \in \{0, \ldots, t\}$. It also follows that if a play is infinite on $\overline{G}$, then at least one corresponding play on G is infinite as well. Hence, if we start from a winning strategy for k cops for the game on G, no strategy for the robber can be winning against $k \cdot r \cdot 2^{r-1}$ cops on $\overline{G}$.

Now we give a more formal proof. Let f be a winning strategy for k cops for the DAG-width game on G (positional strategies suffice) and let $\overline{g}$ be any strategy for the robber for the DAG-width game on $\overline{G}$. We construct a play $\overline{\pi}_{f\overline{g}}$ on $\overline{G}$ that is consistent with $\overline{g}$ (and depends on f), but is won by the cops. The proof is by induction on the length of the finite prefixes (i.e., histories) of $\overline{\pi}_{f\overline{g}}$. While constructing $\overline{\pi}_{f\overline{g}}$ we simultaneously construct, for every history $\overline{\pi}$ of $\overline{\pi}_{f\overline{g}}$ of length i a finite tree $\zeta(\overline{\pi})$ whose branches are histories of length at most i in the DAG-width game on G, such that the following conditions hold. Let

$$\overline{\pi} = \bot \cdot (\overline{C}_1, \overline{v}_1)(\overline{C}_1, \overline{C}_2, \overline{v}_1)(\overline{C}_2, \overline{v}_2) \ldots (\overline{C}_i, \overline{v}_i)$$

(if it ends in a cop position), or

$$\overline{\pi} = \bot \cdot (\overline{C}_1, \overline{v}_1)(\overline{C}_1, \overline{C}_2, \overline{v}_1)(\overline{C}_2, \overline{v}_2) \ldots (\overline{C}_i, \overline{C}_{i+1}, \overline{v}_i)$$

(if it ends in a robber position).

(1) Each history in $\zeta(\overline{\pi})$ is consistent with f.

(2) $\overline{v}_j = \{v \in V \mid$ at level j of $\zeta(\overline{\pi})$, there is (C, v) or $(C, C', v)\}$ for all $j \leq i$. Moreover, for each $v \in V$, on each level there is at most one position of the form (C, v) or (C, C', v).

(3) For all $j \leq i+1$, $\overline{C}_j = \{\overline{w} \in \overline{V} \mid$ at level j, there is (C, C', v) or (C', v) with $\overline{w} \cap C' \neq \emptyset\}$.

(4) Let $\overline{\pi}'$ be a prefix of $\overline{\pi}$. If $\zeta(\overline{\pi}')$ has depth r, then $\zeta(\overline{\pi})$ has depth at least r and up to level r, $\zeta(\overline{\pi}')$ and $\zeta(\overline{\pi})$ coincide.

To begin the induction, consider any play prefix $\overline{\pi}$ of length 1, i.e., any possible initial move $\bot \rightarrow (\emptyset, \overline{w})$ of the robber player. With $\overline{\pi}$ we associate the tree $\zeta(\overline{\pi})$ consisting of the root $\bot$ with successors $(\emptyset, v)$ for $v \in \overline{w}$. Clearly, conditions (1)–(4) hold.

For the translation of the robber moves in the induction step, consider a play prefix

$$\overline{\pi} = \bot \cdot (\overline{C}_1, \overline{v}_1)(\overline{C}_1, \overline{C}_2, \overline{v}_1)(\overline{C}_2, \overline{v}_2) \dots (\overline{C}_i, \overline{C}_{i+1}, \overline{v}_i)$$

with $i \geq 1$ and let, by induction hypothesis, $\zeta(\overline{\pi})$ be constructed up to level $2i$. Consider a robber move from $\overline{v}_i$ to $\overline{v}_{i+1}$, so $\overline{v}_{i+1} \notin \overline{C}_i$ and $\overline{v}_{i+1}$ is reachable from $\overline{v}_i$ in the graph $\overline{G} - (\overline{C}_i \cap \overline{C}_{i-1})$. Let $\overline{v}^0, \overline{v}^1, \dots, \overline{v}^t$ be a path from $\overline{v}_i = \overline{v}^0$ to $\overline{v}_{i+1} = \overline{v}^t$ in $\overline{G} - (\overline{C}_i \cap \overline{C}_{i-1})$. Then by Lemma 1, there are $v \in \overline{v}_{i+1}$ and $u \in \overline{v}_i$, and a path $u^0, u^1, \dots, u^t$ from $u = u^0$ to $v = u^t$ in G with $u^l \in \overline{v}^l$, for $l = 0, \dots, t$. Let W the set of all such v. By Conditions (2) and (4) for $\zeta(\overline{\pi})$, there is some history $\pi \in \zeta(\overline{\pi})$ which ends in a position (C_i^v, C_{i+1}^v, u). So, C_i^v corresponds to $\overline{C}_i$ and C_{i+1}^v corresponds to $\overline{C}_{i+1}$ in the sense of Condition (3). We now extend π to the history $\pi \cdot (C_{i+1}^v, v)$. The set of all such histories extended in this way by (C_{i+1}^w, w) for all $w \in W$ forms the tree $\zeta(\overline{\pi})$.

We have to show that each such move to (C_{i+1}^v, v) is possible, i.e., that $v \notin C_{i+1}^v$ and v is reachable from u in $G - (C_i^v \cap C_{i+1}^v)$. As $\overline{v}_{i+1} \notin \overline{C}_i$, by Condition (3), we have $\overline{v}_{i+1} \cap C_{i+1}^v = \emptyset$, which implies $v \notin C_{i+1}^v$. Now assume towards a contradiction that v is not reachable from u in $G - (C_i^v \cap C_{i+1}^v)$. Then there is some $l \in \{1, \dots, t\}$ such that $u^l \in C_i^v \cap C_{i+1}^v$ (notice that $u^0 = u \notin C_i^v \cap C_{i+1}^v$, otherwise the position with u would not be legal as is given by induction). Then since $u^l \in \overline{v}^l$, we have $\overline{v}^l \in \overline{C}_i^v \cap \overline{C}_{i+1}^v$, by (3), which contradicts the fact that $\overline{v}^0, \overline{v}^1, \dots, \overline{v}^t$ is a path in $\overline{G} - (\overline{C}_i^v \cap \overline{C}_{i+1}^v)$.

We check that Conditions (1)–(4) hold after the construction. For Conditions (2), (3) and (4) this is obvious. For (1), since all play prefixes in $\zeta(\overline{\pi})$ up to level $2i$ are consistent with f by induction and all extensions of the play prefixes are robber moves, all play prefixes in $\zeta(\overline{\pi})$ are still consistent with f.

To translate the answer of the cops, assume that we have already constructed $\zeta(\overline{\pi})$ up to level $2i + 1$, for some $i \geq 0$. Note that there are at most r branches of length $2i + 1$. Let W be the set of robber vertices in the last positions of those branches. For any maximal branch π^v of $\zeta(\overline{\pi})$ ending with a position with robber vertex v where

$$\pi^v = \bot \cdot (C_1^v, v_1)(C_1^v, C_2^v, v_1) \dots (C_i^v, v_i)\,,$$

$i \geq 1$ and $v = v_i$, consider the set $C_{i+1}^v = f((C_i^v, v_i))$ of positions chosen to be occupied by the cops in the next move according to f. We define $\overline{C}_{i+1}$ by

$$\overline{C}_{i+1} = \{[u] \in \overline{V} \mid [u] \cap \bigcup_{v \in W} C_i^v \neq \emptyset\}\,,$$

i.e., the cops occupy those $[w]$ that contain a vertex from some C_i^v.

This yields the play prefix $\overline{\pi}' = \overline{\pi} \cdot (\overline{C}_{i+1}, \overline{C}, \overline{v}_{i+1})$ and we associate an extension $\zeta(\overline{\pi}')$ of $\zeta(\overline{\pi})$ with it. The extension is obtained by appending position (C_i^v, C_{i+1}^v, v) to each branch of length $2i + 1$ ending with a position with robber vertex v. It is trivial that all Conditions (1)–(4) hold.

Assume that $\overline{\pi}_{f\overline{g}}$ is infinite, i.e., won by the robber. Then $\zeta(\overline{\pi}_{f\overline{g}})$ is infinite as well. Since ζ is finitely branching, by König's Lemma, there is some infinite

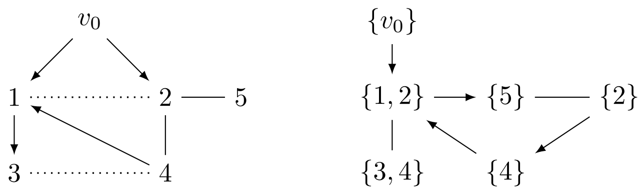

Figure 5: Monotone strategy is translated to a non-monotone one.

path π through ζ. By Condition (1), π is a play in the DAG-width game on G which is consistent with f. Since π is infinite, this contradicts the fact that f is a winning strategy for the cop player.

It remains to count the number of cops used by the cop player in $\overline{\pi}_{f\overline{g}}$. Consider any position $(\overline{C}_i, \overline{C}_{i+1}, \overline{v}_i)$ occurring in $\overline{\pi}_{f\overline{g}}$. By Condition (2), at level $2i$ of $\zeta(\overline{\pi})$, there occur at most $|\overline{v}_i| \leq r$ many play prefixes. Each such play prefix is consistent with f, so at most k vertices are occupied by the cops. Hence, by Condition (3), $|\overline{C}_{i+1}| \leq k \cdot r \cdot 2^{r-1}$ (note that there are 2^{r-1} subsets of set with r elements which contain a fixed vertex). □

We stress that this strategy translation does not necessarily preserve monotonicity, as the following example shows.

Example 13. *We give an example where the strategy translation from Proposition 12 does not preserve monotonicity of the cop strategy. Consider the graph G depicted in Figure 5 and the following monotone (partial) strategy for the cops. First, put a cop on v_0. If the robber goes to* 1*, put a cop on* 1 *and then move the cop from* 1 *to* 3*. If the robber goes to* 2*, put a cop on* 5 *and if the robber goes to* 4*, put a new cop on* 4*. In the game on the powerset graph, consider the following play, which is consistent with the translated cop strategy. First, the cops occupy $\{v_0\}$. Let the robber go to $\{1,2\}$ in which case the cops occupy $\{1,2\}$ and $\{5\}$. Now the robber goes to $\{3,4\}$, so the cop from $\{1,2\}$ is removed. At this moment, the vertex $\{1,2\}$ becomes available for the robber again, so the translated strategy is non-monotone. Notice that, nevertheless,* $\mathrm{dagw}(\overline{G}) = 2$.

Thus our construction does not guarantee that the DAG-width of the powerset graph is bounded in the DAG-width of the original graph and we cannot conclude that a bound on DAG-width allows us to solve parity games in polynomial time. Although not actually our goal, we can consider even stronger conditions on the structural complexity of given graphs. In the remaining of the section we show that *directed path-width* is bounded by a construction similar to that from the proof of Proposition 12 (so the conclusion is also stronger).

For DAG-width we give two solutions, each leading to a result that is also of independent interest. In Section 5 we describe how to use a technique by Fearnley and Schewe from [FS12] for solving parity games on graphs where the DAG-width is not necessarily bounded, but the *non-monotone* DAG-width is. Thus we obtain a stronger result: parity games with imperfect information can be solved in PTIME on classes of graphs of bounded non-monotone DAG-width. In particular, this holds for graphs of bounded DAG-width.

In the remaining sections we go still another way to prove the latter result. Although it is more cumbersome than the solution following Fearnley and Schewe, we also present it because it enlightens the connection between bounded imperfect information and graph searching. It also contains some results on graph searching that are independent of solving parity games.

Proposition 14. *Let $\mathcal{G} = (V, V_0, E, v_0, \sim, \Omega)$ be a parity game with imperfect information in which the size of the $\sim$-classes is bounded by some r. If* $\operatorname{dpw}(G) \leq k$*, then* $\operatorname{dpw}(\overline{G}) \leq k \cdot 2^{r-1}$.

Proof. Let f be a monotone winning strategy for k cops in the directed path-width game on $\mathcal{G}$ and let

$$\pi = \bot \cdot (C_0, C_0, R_0)(C_0, C_1, R_1) \dots (C_{n-1}, C_n, R_n)$$

be the unique play which is consistent with f. It is finite, as f is winning. Recall that the directed path-width game is, essentially, a one player game and there is a bijection between strategies and plays, so it suffices to construct a (not necessarily monotone) play

$$\overline{\pi} = \{\bot\} \cdot (\overline{C}_0, \overline{C}_0, \overline{R}_0)(\overline{C}_0, \overline{C}_1, \overline{R}_1)(\overline{C}_1, \overline{C}_2, \overline{R}_2) \dots$$

of the game on $\overline{G}$ that is won by the cops where, for all i, we have $|\overline{C}_i| \leq k2^{r-1}$. We construct $\overline{\pi}$ inductively by the length of its finite prefixes such that the following invariant holds.

1. $\overline{C}_0 = \emptyset$ and $\overline{R}_0 = \overline{V}$ (at the beginning, there are no cops in the graph and the robber occupies the whole graph),
2. $\overline{R}_n = \emptyset$ (at the end, the robber is captured),
3. $\overline{R}_{i+1} = \mathrm{Reach}_{\overline{\mathcal{G}} - (\overline{C}_i \cap \overline{C}_{i+1})}(\overline{R}_i) \setminus \overline{C}_{i+1}$, for all i (every move is legal),
4. $\overline{R}_{i+1} \subseteq \overline{R}_i$, i.e., the play is monotone,
5. $\bigcup \overline{R}_i \coloneqq \{v \in V \mid \text{ there is some } \overline{w} \in \overline{R}_i \text{ with } v \in \overline{w}\} \subseteq R_i$, i.e., if the robber occupies a vertex $\overline{w}$ in $\overline{\pi}$ and $v \in \overline{w}$, then the robber occupies v. Note that vertices (positions) in $\overline{\pi}$ are sets of vertices in π.

The last two properties of $\overline{\pi}$ imply the statement of the proposition. Indeed, by Property 4, the play is monotone. Furthermore, the robber is finally captured if and only if the play is finite and ends in a position $(\overline{C}_{n-1}, \overline{C}_n, \overline{R}_n)$ where $\overline{R}_n = \emptyset$. Assume that $\overline{\pi}$ is infinite, then all $\overline{R}_i \neq \emptyset$, but then there is some v and $\overline{w}$ with $v \in \overline{w} \in \overline{R}_i$ such that v is not occupied in the ith position of π (by Property 5 of the invariant), so π is not winning, but that contradicts the assumption.

The construction just follows the invariant. Let $\overline{\pi}_0 = \{\bot\}(\emptyset, \emptyset, \overline{V})$ and, for $i > 0$, let $\overline{\pi}_i = \overline{\pi}_{i-1} \cdot (\overline{C}_{i-1}, \overline{C}_i, \overline{R}_i)$ such that, for all $\overline{v} \in \overline{V}$ we have $\overline{v} \in \overline{C}_i$ if and only if $\overline{v} \cap C_i \neq \emptyset$. In other words, we place a cop on a vertex $\overline{v}$ in a position of $\overline{\pi}$ if, in the corresponding position of π, we place a cop on some vertex in $\overline{v}$. As there are 2^{r-1} subsets of C_i that contain a fixed vertex v, the size of all $\overline{C}_i$ is at most $k \cdot 2^{r-1}$.

It remains to show Properties 4 and 5 of the invariant. Assume that the play is not monotone, then there is some i, some $\overline{w} \in \overline{R}_i$ and some $\overline{v} \in \overline{C}_i \setminus \overline{C}_{i+1}$ such that $(\overline{w}, \overline{v}) \in \overline{E}$, i.e., a cop was removed from $\overline{v}$ and the robber occupies $\overline{v}$ following one single edge. By Reif's construction, for all $y \in \overline{v}$, there is some $x \in \overline{w}$ with $(x, y) \in E$. As $\overline{w} \in \overline{R}_i$, by induction, we have $\overline{w} \subseteq R_i$. On the other hand, as $\overline{v} \in \overline{C}_i \setminus \overline{C}_{i+1}$, by the construction of $\overline{\pi}$, we have $\overline{v} \cap C_i \neq \emptyset$ and $\overline{v} \cap C_{i+1} = \emptyset$. In other words, in π some vertex y of $\overline{v}$ is left by a cop and all vertices of $\overline{w}$ are occupied by the robber in the ith move. However, the robber can move from some $x \in \overline{w}$ to y, which causes non-monotonicity in π, but we assumed that π is monotone. Thus $\overline{R}_{i+1} \subseteq \overline{R}_i$.

It remains to prove Property 5 of the invariant. Assume that it does not hold and suppose, i is the least index with $\bigcup \overline{R}_i \not\subseteq R_i$. Then there exist some $\overline{w} \in \overline{R}_i$ and some $v \in \overline{w}$ such that $v \notin R_i$. By induction hypothesis, the move from position $i-1$ to position i is monotone, so $\overline{w} \subseteq \overline{R}_{i-1}$. Then by the choice of i, $v \in R_{i-1}$. So we have $v \in R_{i-1} \setminus R_i$ and thus $v \in C_i \setminus C_{i-1}$. By the construction of $\overline{\pi}$, we have $\overline{w} \in \overline{C}_i$, a contradiction to $\overline{w} \in \overline{R}_i$. □

Corollary 15. *Parity games with bounded imperfect information can be solved in polynomial time on graphs of bounded directed path-width.*

Finally, we remark that our direct translation of the robber moves back to the game on G cannot be immediately applied to the games which define Kelly-width and directed tree-width. In the Kelly-width game, the robber can only move if a cop is about to occupy his vertex. It can happen that the cops occupy a vertex $\{v_1, \ldots, v_l\}$ in $\overline{G}$ but not all vertices $v_1, \ldots, v_l$ in G. In the directed tree-width game, the robber is not permitted to leave the strongly connected component in which he currently is, which again obstructs a direct translation of the robber moves from $\overline{G}$ back to G. Furthermore, it is not known whether parity games with perfect information can be sovled in polynomial time if directed tree-width is bounded.

5. Simulated parity games

Simulated parity games were introduced by Fearnley and Schewe in [FS11, FS12]. The idea of a simulated parity game is to decompose the original game into smaller games such that one can control cycles that appear when the game pebble revisits a vertex more efficiently. The simulated game (in that both players have perfect information) starts on a small subgame S. If a cycle is reached within that subgame, the game stops and the winner is determined as in the usual parity game. Otherwise consider the first visited vertex $v \notin S$. One of the players (it does not matter which one, say, Player 0) gives some promise: he claims, for every vertex in S, that he can guarantee a certain value, the best color, in any play from v to that vertex. More precisely, for every vertex $w \in S$, he announces a color c and asserts that no worse color will be seen in a play from v to w if w will be the first vertex in S visited from now on. Player 1 either accepts for some vertex w in S, then the game continues from w and the minimum color seen since v is set to c, or he rejects. In the latter case the game continues in a next small subgame S' containing v. This continues in the same way, except that now when the play leaves S', the assertions of Player 0 for S' are added to those for S. A play ends either if it reaches a vertex for that Player 0 promised a color (and Player 0 wins if and only if he could keep his promise), or a cycle is closed (then the parity condition applies). This idea of closing cycles is similar to the idea from [BG05b] of alternating cycle detection while playing the entanglement game.

The game is parametrized by two functions: function Next determines the next subgame S' and function Hist forgets some of the promises of Player 0 and is used for optimization. If every play of the simulated game is finite (intuitively, Hist does not forget too much), then Player 0 wins the original game if and only if he wins the simulated game (from the same vertex). Examples of Next and Hist are given in [FS12]: Hist forgets every promise except those from the last subgame and Next follows a tree decomposition or a DAG decomposition.

If we construct Next and Hist such that they fulfill some conditions on certain classes of graphs, then we can solve parity games in PTIME on those classes. The conditions are:

- Every play of the simulated game is finite.
- There is a data structure to store the promises of Player 0 that uses only a logarithmic amount of space in the size of the parity game.

An alternating Turing machine just plays the simulated game and determines the winner. In the rest of the section we describe the simulated game formally and prove that for graph classes where a bounded number of cops can capture a robber (not necessarily in a monotone way) we can indeed find appropriate functions Next and Hist.

Let $\mathcal{G} = (V, V_0, E, v_0, \Omega)$ be a parity game with perfect information. The *significance order* on the set of colors is defined by $a \prec b$ if a is better for Player 0 than b, i.e., a is even and b is odd, or both are even and $a < b$, or both are odd and $a > b$. For a positional strategy f of Player σ, a set F of vertices and two vertices s and t, let $\mathrm{Paths}_f^F(s,t)$ be the set of paths from s to t avoiding F (except of t if $t \in F$) and consistent with f, i.e., if f is a strategy for Player σ, then for all consecutive v and w on a path in $\mathrm{Paths}_f^F(s,t)$, if v is a vetrex of Player σ, then $f(v) = w$. For a sequence P of vertices let $\mathrm{mincol}(P)$ be the minimal with respect to $<$ color appearing in P. We denote the best possible color that a strategy f guarantees on $\mathrm{Paths}_f^F(s,t)$ by $\mathrm{best}_f^F(s,t)$, i.e.,

$$\mathrm{best}_f^F(s,t) = \mathrm{opt}_{P \in \mathrm{Paths}_f^F(s,t)} \mathrm{mincol}(P)$$

where opt = max if f is a Player 0 strategy and opt = min if f is a Player 1 strategy, both with respect to the significance order. Let C be the set of used colors. A *strategy profile* for a set of vertices F is a function $\mathrm{Profile}^F_{f,s}\colon F \to C \cup \{-\}$ defined by:

$$\mathrm{Profile}^F_{f,s}(t) = \begin{cases} - & \text{if } \mathrm{Paths}^F_{f,s}(t) = \emptyset, \\ \mathrm{best}^F_f(s,t) & \text{otherwise.} \end{cases}$$

An *abstract profile* is a function $P^F_s\colon F \to C \cup \{-\}$. A profile is what a strategy f can actually guarantee, an abstract profile is what Player 0 promises when the play leaves the current subgame. In particular $P^F_s(t) = -$ means that t should not be reached at all. Of course, Player 0 is free to promise something that cannot be guaranteed by any of his strategies.

When a play of the simulated game returns to a subgame it left in the past, we must have stored enough information to check whether Player 0 could keep his promise. The data structure for this is a history, which is a set of records.[2] A *record* is a triple (F, c, P) where $F \subseteq V$, P is an abstract profile for F and c is a color in C. Hereby P stores a promise of Player 0 and c is the minimal color seen since the promise was made. For a record (F, c, P) and a color c' define $\mathrm{Update}((F, c, P), c')$ by $(F, \min(c, c'), P)$ where the minimum is with respect to $<$ and $\min(-, c') = c'$, for all $c' \in C$. A *history* is a set $\mathcal{H}$ of records. For a history $\mathcal{H}$ and a color c, we define $\mathrm{Update}(\mathcal{H}, c')$ by $\{\mathrm{Update}(F, c, P), c') \mid (F, c, P) \in \mathcal{H}\}$.

The function Next maps a tuple $(S, v, \mathcal{H})$ where $S \subseteq V$, $v \in V \setminus S$ and $\mathcal{H}$ is a history to a set $S' \subseteq V$. Hist is a history updating function: it deletes some elements from a given history, i.e., $\mathrm{Hist}(\mathcal{H}) \subseteq \mathcal{H}$. The game $\mathrm{Simulate}_{\mathcal{G}}(S, \mathcal{H}, v, \mathrm{Next}, \mathrm{Hist})$ is played on G as follows.

The positions of the simulated game are of the form $(S, v, \mathcal{H}, \Pi)$ where $v \in V$ and Π is a sequence of triples (u, c, w) with $u, w \in V$ and $c \in C$. Hereby v is the current vertex and Π stores the simulated play prefix played so far. By abuse of notation we apply mincol also to sequences of colors with the obvious meaning and extend it to sequences of triples (u, c, w): $\mathrm{mincol}\big((u_1, c_1, w_1), \dots, (u_n, c_n, w_n)\big) = \mathrm{mincol}(c_1, \dots, c_n)$. Furthermore, if Π has the form

$$(u_1, c_1, w_1) \dots (u_m, c_m, w_m)(u_{m+1}, c_{m+1}, w_{m+1}) \dots (u_n, c_n, , w_n)(u_m, c_m, w_m)\,,$$

i.e., it ends in a cycle, then define

$$\mathrm{Winner}(\Pi) = \mathrm{mincol}\big((u_m, c_m, w_m), \dots, (u_n, c_n, w_n)\big) \mod 2\,.$$

The game is played in rounds. Let $(S, v, \mathcal{H}, \Pi)$ be the current position. A round consists of the following steps.

1. If $v \in V_\sigma$, then Player σ chooses some $v' \in vE$.
2. If $v' \in S$ or $v' \in F$ for some $(F, c, P) \in \mathcal{H}$, then $\Pi' = \Pi \cdot (v, \Omega(v'), v')$, the new position is $(S, v', \mathcal{H}, \Pi')$ and the play continues from Step 5.

[2] Note that we redefined the notion of a history for this Section.

3. Player 0 announces an abstract profile P' for v' and S.

4. Player 1 can play `accept` and choose some $w \in S$ with $P'(w) \neq -$, or play `reject`.

    - If Player 1 chooses `accept`, then the next position is $(S, w, \mathcal{H}, \Pi'')$ where $\Pi'' = \Pi' \cdot (v', \min(\Omega(v'), w), w)$.
    - If Player 1 chooses `reject`, then the history and S are updated as follows and the play continues from Step 5:
        - $\mathcal{H}' = \text{Update}(\mathcal{H}, \Omega(v'))$;
        - $\mathcal{H}'' = \mathcal{H}' \cup \{(F, \Omega(v'), P')\}$;
        - $\mathcal{H}''' = \text{Hist}(\mathcal{H}'')$;
        - $S' = \text{Next}(S, v', \mathcal{H}''')$;
        - The winner is the winner of $\text{Simulate}_{\mathcal{G}}(S', \mathcal{H}''', v', \text{Next}, \text{Hist})$.

5. If $v' \in F$ for some $(F, c, P) \in \mathcal{H}$, then the game stops. Let the current position be $(S, v', \mathcal{H}, \Pi^*)$. Player 0 wins the play if $\min(\text{mincol}(\Pi^*), c) \leq P(v')$; Player 1 wins the play if either $\min(\text{mincol}(\Pi^*), c) > P(v')$ or $P(v') = -$.

6. If Π^* ends with a cycle, then the winner of the play is $\text{Winner}(\Pi)$.

The initialization round is played as follows. If $v_0 \in S$, then the play starts in the first regular round. If $v_0 \notin S$, Player 0 announces an abstract strategy profile P for v_0 and S and the history is initialized with $\mathcal{H} = \{(v_0, \Omega(v_0), v_0)\}$.

Theorem 16 ([FS12])**.** *Let $\mathcal{G} = (V, V_0, E, v_0, \Omega)$ be a parity game. Then, for all S, $\mathcal{H}$,* Next*, and* Hist*, if all plays of* $\text{Simulate}_{\mathcal{G}}(S, \mathcal{H}, v_0, \text{Next}, \text{Hist})$ *are finite, then Player* 0 *has a winning strategy for $\mathcal{G}$ if and only if Player* 0 *has a winning strategy for* $\text{Simulate}_{\mathcal{G}}(S, \mathcal{H}, v_0, \text{Next}, \text{Hist})$.

Now we prove the main result of this section. The proof is an adaptation of the proof of Theorem 6.4 from [FS11]. We show that a winning strategy of cops in the cops and robber game induces functions Next and Hist that satisfy the conditions of Theorem 16. In addition, the resulting play can be solved in deterministic polynomial time, which leads to an efficient solution of parity games with perfect information on graphs where non-monotone DAG-width is bounded and thus also of parity games with bounded imperfect information on those graphs.

Theorem 17. *Let $\mathcal{G} = (V, V_0, E, v_0, \Omega)$ be a parity game. Let k cops have a (not necessarily monotone) strategy in the cops and robber game on G that guarantees a capture of the robber. Then there is are functions* Next *and* Hist *ans some $S \subseteq V$ such that for all $\mathcal{H}$ and v_0 the game* $\text{Simulate}_{G}(S, \mathcal{H}, v, \text{Next}, \text{Hist})$ *has no infinite plays. Furthermore, given $\mathcal{G}$, in deterministic polynomial time in the size of $\mathcal{G}$, we can construct a representation of the simulated game and solve it.*

Proof. It is clear that in the cops and robber game, positional strategies for both players suffice. Let f be a positional strategy for k cops that guarantees a capture of the robber. For every position of the simulated game $\mathrm{Simulate}_G(S, \mathcal{H}, v, \mathrm{Next}, \mathrm{Hist})$ with $v \notin S$ we define $\mathrm{Next}(S, v, \mathrm{Hist}) = f(S, v)$. We define S to be the answer of the cops to the first move of the robber to v_0, i.e., $S = f(\emptyset, v_0)$. The function Hist forgets all records from the history up to the last one. If $|\mathcal{H}| = 1$, then $\mathrm{Hist}(\mathcal{H}) = \mathcal{H}$. Otherwise $|\mathcal{H}| = 2$, and, by the definition of Next, there are some F_1, F_2, c_1, c_2, P_1 and P_2 such that $\mathcal{H} = \{(F_1, c_1, P_1), (F_2, c_2, P_2)\}$ and $f(F_1, v) = F_2$, where v i the current vertex. We set $\mathrm{Hist}(\mathcal{H}) = \{(F_2, c_2, P_2)\}$.

We show that all plays of $\mathrm{Simulate}_G(S, \mathcal{H}, v_0, \mathrm{Next}, \mathrm{Hist})$ are finite. Note that since f can be non-monotone, it is possible that the pebble in the simulated game returns to a vertex it left in the past, but the play does not stop because we forgot the promise of Player 0 for that vertex.

Assume that there is an infinite play π of $\mathrm{Simulate}_G(S, \mathcal{H}, v_0, \mathrm{Next}, \mathrm{Hist})$. We describe an infinite play π' of the cops and robber game that is consistent with f. Let $(v_1, w_1), (v_2, w_2) \dots$ be the infinite sequence of all pebble moves in the simulated game with $v_i \in S_i$ and $w_i \notin S_i$ where $(S_i)_{i \geq 1}$ is the sequence of the subgames appearing in π. Then π' is the play in which the robber chooses $v_0, w_1, w_2, w_3, \dots$ and the cops play according to f. Then π' is infinite, which is a contradiction because f is winning, but we still have to show that π' is well-defined, i.e., that all robber moves are possible.

The first robber move to v_0 is trivially possible. The cops answer occupying $S_1 = f(\emptyset, v_0)$. As $w_1 \notin S_1$, there is no cop on w_1. Because the simulated game proceeded from v_1 to w_1 (not necessarily in one move), there is a path from v_1 to w_1, so robber move to w_1 is possible. The same argument applies for all w_i with $i > 1$.

To solve $\mathrm{Simulate}_{\mathcal{G}}(S, \mathcal{H}, v_0, \mathrm{Next}, \mathrm{Hist})$ we construct an alternating Turing machine that just plays the game. This can obviously be done in polynomial time. We have to prove that the Turing machine uses only a logarithmic amount of space in the size of $\mathcal{G}$. As the cops and robber game admits positional winning strategies for both players, we can assume that if a position of the game repeats in a play, then the robber wins the play. There are $|V|^k$ possible cop placements and at most $|V|$ possible robber placements, i.e., at most $|V|^{k+1}$ positions.

The data structures are variables S, $\mathcal{H}$, Π and v.

- By construction S is a cop placement, so we need $k \cdot \log |V|$ bits to store it.
- In any position $\mathcal{H}$ always contains only one record (F, c, P) with $|F| \leq |V|$ and can be stored using $\log |V|$ bits for F, $\log |C|$ bits for c and $k \cdot \log |C|$ bits for P because P contains at most k pairs (w, c_w) with $w \in F$ and $c_w \in C \cup \{-\}$. One can represent P as a list of length at most k of colors from $C \cup \{-\}$. Thus $\mathcal{H}$ can be stored using $\log |V| + k \cdot (\log |C| + 1)$ bits.
- We need $\log |V|$ bits to store v.

- The variable Π is a sequence of at most $|S|$ tuples (v, c, w). If (v_1, c_1, w_1) and $(v_2.c_2, w_2)$ are two consecutive tuples in Π, then $w_1 = v_2$, so we only have to remember pairs (c, w) where c is a color and w is one of at most $|S|$ vertices. As $|S| \leq k$, we need at most $k \cdot (\log |C| + \log k)$ bits for Π. Note that although in the initialization round $v \notin S$ is possible, we do not need to memorize v because in this case there can be no closed cycle in S containing v.

Summing up, the alternating Turing machine needs at most $(k+2) \cdot \log |V| + 2k \cdot \log |C| + k(\log k + 1)$ bits. This leads to a deterministic algorithm running in time $\mathcal{O}(|V|^{k+2} \cdot |C^{2k}| \cdot k^k)$. □

The powerset construction produces a graph $\overline{G}$ that is only polynomially larger than the original graph G. By Proposition 12, the non-monotone DAG-width of $\overline{G}$ is bounded in the DAG-width of G, so as a corollary from Theorem 17 we obtain the following result.

Corollary 18. *Parity games with bounded imperfect information can be solved in polynomial time on graphs of bounded DAG-width.*

6. Bounded imperfect information and multiple robbers

In this section we follow another approach to prove Corollary 18. We translate imperfect information bounded by some constant r into a new graph searching game by introducing r robbers instead of one. This game is a generalization of a similar helicopter cops and robber game with multiple robbers (that we refer to as a helicopter game, for short) defined by Richerby and Thilikos in [RT09]. The helicopter game is played on an undirected graph by a team of k cops and a gang of r robbers where k and r are parameters of the game. The cops move as in the cops and (single) robber game (up to a non-essential new kind of *sliding* cop moves), and each robber moves independently of the others also as in the game with a single robber. If a robber is captured, he is taken away from the graph. When all robbers are captured, the cops win. Infinite plays, in which at least one robber survives for ever, are won by the robbers. The cops also lose if they perform a non-monotone move, i.e., if a robber can reach a vertex that was previously unavailable for the robbers. Richerby and Thilikos show that the number of additional cops needed to capture a team of robbers with one additional robber grows at most logarithmically in r.

Our game differs from the helicopter game in three aspects. First, we do not allow sliding moves, but this can introduce a difference in the cop number by at most one. Second, we play on directed graphs, and we will see that this permits the robbers to coordinate their efforts in a new way to escape from the cops. Third, in our game the robbers can jump to each other, i.e., a robber can leave his vertex and play from the vertex occupied by another robber. This rule may seem somewhat unnatural, but we introduce it for several reasons. First, we will see that this rule supplies the robbers with more power. In particular, the logarithmic upper bound from [RT09] does not hold any more. We however show that the number of additional cops is bounded in r and grows at most linearly in r, which is our main result about graph searching games. Hence the additional power of the robber gang makes the boundedness result stronger (for the cost of a worse bound). The second reason to allow the robber jumping is that this fits our purpose to solve parity games with bounded imperfect information in polynomial time. Finally, our graph searching game may be used to model parallel processes that must be served in some way. Some processes may terminate or may be ultimately served and thus finished, some can produce new processes if the total number does not exceed some bound. Every process corresponds to a robber and resources used to serve them correspond to the cops. Captured robbers describe terminated processes and creating new processes is modeled by multiple robbers running from the vertex of one robber in different directions. The cop number describes the minimal amount of resources needed to serve all processes. In our case, processes are possible plays of a parity game.

The rest of the paper is structured as follows. In Section 7 we prove Theorem 25, which states that if k cops capture one robber on a graph, then $k \cdot r$ cops capture r robbers on that graph. In particular the number of new cops needed to capture r robbers is bounded only in r and k by a linear function. We show in Theorem 40, Section 7.3, that a linear bound is unavoidable in our setting. As our example graphs are undirected, this is not due to directed edges in the graphs, but is caused by the ability of the robbers to jump.

Before we turn to the analysis of games with multiple robbers, let us show how we can use Theorem 25 to obtain Corollary 18. Given a parity game with imperfect information bounded by r on a graph G of DAG-width k, we find a winning strategy for $k \cdot r$ cops against r robbers on G. This strategy can be used to construct a winning strategy for $k \cdot r \cdot 2^{r-1}$ cops against one robber on the powerset graph $\overline{G}$ (so $\operatorname{dagw}(\overline{G}) \leq k \cdot r \cdot 2^{r-1}$). We show how to do this in Lemma 19. As the size of $\overline{G}$ is polynomially bounded in the size of G, we can solve the parity game with perfect information on $\overline{G}$ in polynomial time in the size of G.

6.1. Boundedness of DAG-width and parity games

Going to the powerset graph, we associate every play we consider to be possible on the original graph (there are at most r such plays) with one robber. Tracking at most r plays corresponds to playing against at most r robbers simultaneously.

Lemma 19. *If* $\operatorname{dagw}_r(G) \leq k$*, then* $\operatorname{dagw} \overline{G} \leq k \cdot 2^{r-1}$.

Proof. Let f be a winning strategy for the cops in the game against r robbers on G. We follow a play π consistent with f and a play $\overline{\pi}$ of the game against one robber on $\overline{G}$ simultaneously. Cop moves are translated from π to $\overline{\pi}$ and robber moves are translated in the opposite direction. We maintain two invariants.

> **(Robbers)** If the robber occupies a vertex $\overline{v} = \{v_1 \cdots, v_s\} \in \overline{V}$ in a position of $\overline{\pi}$, then in the corresponding position of π (after the same number of moves), the robbers occupy the set $\overline{v} \subseteq V$.

(Cops) If the cops occupy a set U in a postion of π, then, for every $u \in U$, the cops occupy every $\overline{u}$ in the corresponding position of $\overline{\pi}$.

Consider any strategy $\overline{g}$ for the robber player for the game with one robber on $\overline{G}$. We construct a play $\overline{\pi}_{fg}$ of this game that is consistent with $\overline{g}$ (and depends on f), but is winning for the cops. As $\overline{g}$ is arbitrary, it follows that the cops have a winning strategy.

We construct $\overline{\pi}_{fg}$ by induction in the length of its finite prefixes. For every finite prefix $\overline{\pi}_i$ of $\overline{\pi}_{fg}$ of length i we define a history π_i of a play on G that is consistent with f and has length i. Hereby, for all even $i \geq 2$, if $(\overline{U}_j, \overline{U}_{j+1}, \overline{v}_j)$ is the jth position of π_i, then $(U_j, U_{j+1}, \overline{v}_j)$ is the jth position of π_i such that $\overline{U}_j = \{\overline{u} \in \overline{V} \mid \overline{u} \cap U_j \neq \emptyset\}$ and $\overline{U}_{j+1} = \{\overline{u} \in \overline{V} \mid \overline{u} \cap U_{j+1} \neq \emptyset\}$, for all $j \leq i$.

For $i = 0$, let $\overline{\pi}_i = \pi_i = \perp$. For the translation of a robber move, let $\overline{\pi}_i$ and π_i be constructed and let the robber move from $(\overline{U}_i, \overline{U}_{i+1}, \overline{v}_i)$ to $(\overline{U}_{i+1}, \overline{v}_{i+1})$ in the game on $\overline{G}$. We define $\overline{\pi}_{i+1} = \overline{\pi}_i \cdot (\overline{U}_{i+1}, \overline{v}_{i+1})$ and $\pi_{i+1} = \pi_i \cdot (U_{i+1}, \overline{v}_{i+1})$ and show that going from $\overline{v}_i$ to $\overline{v}_{i+1}$ is a legal robber move in the game on G.

As the move from $\overline{v}_i$ to $\overline{v}_{i+1}$ is legal on $\overline{G}$, we have $\overline{v}_{i+1} \notin \overline{U}_{i+1}$ and $\overline{v}_{i+1} \in \text{Reach}_{\overline{G} - (\overline{U}_i \cap \overline{U}_{i+1})}(\overline{v}_i)$. Let $P = \overline{v}_i, \overline{v}^1, \ldots, \overline{v}^t, \overline{v}_{i+1}$ be a path from $\overline{v}_i$ to $\overline{v}_{i+1}$ in $\overline{G} - (\overline{U}_i \cap \overline{U}_{i+1})$. Let $v \in \overline{v}_{i+1}$. Then by Lemma 1, there is some $u \in \overline{v}_i$ and a path $u = u^0, u^1, \ldots, u^t, v$ in G with $u^l \in \overline{v}^l$, for $l = 0, \ldots, t$. We have to show that $v \notin U_{i+1}$ and that v is reachable from u in $G - (U_i \cap U_{i+1})$.

First, $\overline{v}_{i+1} \notin \overline{U}_{i+1}$ and therefore $\overline{v}_{i+1} \cap U_{i+1} = \emptyset$, which implies $v \notin U_{i+1}$. Now assume towards a contradiction that v is not reachable from u in $G - (U_i \cap U_{i+1})$. Then there is some $l \in \{1, \ldots, t\}$ such that $u^l \in U_i \cap U_{i+1}$. However, since $u^l \in \overline{v}^l$, by the induction hypothesis, we have $\overline{v}^l \in \overline{U}_i \cap \overline{U}_{i+1}$, but $\overline{v}^1, \ldots, \overline{v}^t$ is a path in $\overline{G} - (\overline{U}_i \cap \overline{U}_{i+1})$.

To translate the answer of the cops, consider the set $U = f(U_i, \overline{v}_i)$, which f prescribes to occupy in the next move, so $\pi_{i+1} = \overline{\pi}_i \cdot (U_i, U_{i+1}, \overline{v}_i)$. Let the next move in $\overline{\pi}$ be defined by $\overline{U}_{i+1} = \{\overline{u} \in \overline{V} \mid \overline{u} \cap C \neq \emptyset\}$, and hence, $\overline{\pi}_{i+1} = \overline{\pi}_i \cdot (\overline{U}_i, \overline{U}_{i+1}, \overline{v}_i)$.

Finally, play $\overline{\pi}_{fg}$ is the limit of all $\overline{\pi}_i$, i.e., the 0th position of $\overline{\pi}_{fg}$ is $\perp$, and the ith position is $(\overline{U}_i, \overline{U}_{i+1}, \overline{v})$, if i is a positive even number, and $(\overline{U}_i, \overline{v}_i)$ if i is odd.

We have to show that $\overline{\pi}_{fg}$ is won by the cops, i.e., that it is monotone and the robber is captured. To prove the monotonicity, assume towards a contradiction that the play $\overline{\pi}_{fg}$ is not monotone, i.e., there is some position $(\overline{U}_i, \overline{U}_{i+1}, \overline{v}_i)$ of $\overline{\pi}_{fg}$ such that there is some $\overline{u} \in \overline{U}_i \setminus \overline{U}_{i+1}$ reachable from $\overline{v}_i$ in $\overline{G} - (\overline{U}_i \cap \overline{U}_{i+1})$. Let $\overline{v}_i, \overline{v}^1, \ldots, \overline{v}^t, \overline{u}$ be a path from $\overline{v}_i$ to $\overline{u}$ in $\overline{G}$ with $\overline{v}^l \notin \overline{U}_i$, for all $l \in \{1, \ldots, t\}$. Since $\overline{u} \in \overline{U}_i \setminus \overline{U}_{i+1}$, by the construction of π, there is some $u \in \overline{u}$ with $u \in U_i \setminus U_{i+1}$. Moreover, by Lemma 1, there is some $v_i \in \overline{v}_i$ and a path $v_i, v^1, \ldots, v^t, u$ in G with $v^l \in \overline{v}^l$, for all $l \in \{1, \ldots, t\}$. By the construction of π_i all $v^l \notin U_i$, thus u is reachable from v_i in $G - U_i$, which contradicts the monotonicity of f. Hence, $\overline{\pi}_{fg}$ is monotone.

Consider the play π_{fg} obtained as a limit of all π_i. If $\overline{\pi}_{fg}$ is infinite, then π_{fg} is infinite as well, which is impossible, as π_{fg} is consistent with f.

Finally, we count the number of cops used by the cop player in $\overline{\pi}_{fg}$. Consider any position $(\overline{U}_i, \overline{U}_{i+1}, \overline{v}_i)$ occurring in $\overline{\pi}_{fg}$. Since π_{fg} is consistent with f, for the corresponding position $(U_i, U_{i+1}, \overline{v}_i)$ in π_{fg}, we have $|U_{i+1}| \leq k$. From the construction of $\overline{\pi}_{fg}$, it follows that $|\overline{U}_{i+1}| \leq k \cdot 2^{r-1}$. Therefore, the robber does not have a winning strategy against $k \cdot 2^{r-1}$ cops in the game on $\overline{G}$. By determinacy, $k \cdot 2^{r-1}$ cops have a winning strategy. □

6.2. The multiple robbers game

Let $G = (V, E)$ be a graph and $k, r \in \omega$. The *k cops and r robbers game* $\mathcal{G}_k^r(G)$ is defined as follows. A position has the form (U, R) or (U, U', R) where $U, U', R \subseteq V$ with $|U|, |U'| \leq k$ and $|R| \leq r$. Hereby U and U' are as in the game with one robber and R are the vertices occupied by the robbers. From a cop position (U, R), the cops can move to any position (U, U', R) as in the game with one robber. From a robber position (U, U', R), the robbers can move to any position (U', R') such that $R' \cap U' = \emptyset$ and each $r' \in R'$ is reachable from some $r \in R$ in $G - (U \cap U')$. In the first move, the robbers can go from the initial position $\perp$ to any position $(\emptyset, R)$ with $|R| \leq r$.

Notice that this definition blurs the role of single robbers: first, a robber can leave the graph and, second, one robber can induce multiple robbers in the next position. Indeed, there may be distinct $v_1, v_2 \in R'$ reachable (in $G - (U \cap U')$) only from one vertex $v \in R$. In this case, we say informally that robber v_1 *runs* and robber v_2 *jumps* if we assume that the robber on v_1 was on v before the move and the robber on v_2 was on a vertex w with $v_2 \notin \mathrm{Reach}_{G-(U \cap U')}(w)$. However, this distinction is not formalized (we could also swap the roles of v_1 and v_2) and used only to develop better intuition.

A play of a cops and multiple robbers game is *(robber-)monotone* if the play contains no position (U, U', R) such that some $u \in U \setminus U'$ is reachable from some $r \in R$ in $G - (U \cap U')$. Monotone finite plays are won by the cops, non-monotone plays and infinite plays are won by the robbers.

A *memory strategy* for the cop player in a cops and multiple robbers game is a memory structure $\mathcal{M} = (M, \mathrm{init}, \mathrm{upd})$ together with a strategy function $f : M \times 2^V \times V \to 2^V$ (for the cop strategy), respectively $f : M \times 2^V \times 2^V \to 2^V$ (for the robber strategy). Hereby M is a set of memory states, $\mathrm{init} : V \to M$, respectively $\mathrm{init} : 2^V \to M$ is the memory initialization function mapping the robbers placement after the first move of the robbers to a memory state, and $\mathrm{upd} : M \times 2^V \times 2^V \times V \to M$, respectively $\mathrm{upd} : M \times 2^V \times 2^V \times 2^V \to M$ is the memory update function, which maps a memory state and a cop respectively a robber position to a new state. A memory strategy is positional if $|M| = 1$, in which case M can be omitted. Winning strategies, plays, histories and consistency are defined in the usual way, analogously to the case of a single robber. A cop strategy is monotone, if every play consistent with it is monotone. As the cops have a reachability winning condition, the cops and multiple robbers games are positionally determined. We will use memory strategies because they allow us more intuitive descriptions.

The least k such that the cops have a winning strategy for the k cops and r robbers game on G is denoted by $\mathrm{dagw}_r(G)$. Note that the DAG-width of a graph G is $\mathrm{dagw}_1(G)$. We define $\mathrm{tw}_r(G)$ analogously to the case of one robber, i.e., $\mathrm{tw}_r(G) = \mathrm{dagw}_r(G^{\leftrightarrow}) - 1$ where $G^{\leftrightarrow}$ is as G, but with the edge relation replaced by its symmetrical closure. Recall from Section 2.2 that $\overline{G}$ is the graph obtained from a graph G by applying the powerset construction.

As discussed above, to complete our second proof of Corollary 18 it remains to show that Theorem 25 holds. We do this in the next section.

7. From one robber to r robbers

As the first step we show that we can assume without loss of generality two restrictions on robbers strategies. A robber strategy f is *isolating* if no two robbers can reach one another, i.e., if in any cop position (U, R) of any play consistent with f, for all $v, w \in R$, we have $v \notin \mathrm{Reach}_{G-U}(w)$. An important special case of this rule is that there can never be two robbers in the same component. Intuitively, if $v \in \mathrm{Reach}_{G-U}(w)$, then the robber on v is redundant: the robbers can place him on v also in the next move. He can go to v by first jumping to the robber on w and then running from w to v.

The second restriction on the robber moves is that each of them leaves his vertex either if he jumps to another robber (a reason for a jump can be that he is needed somewhere else) or if otherwise (if he does not jump, but runs) the destination of his run would become unreachable for him in the next move. Formally, we say that a robber strategy f is *prudent* if, for each robber move $(U, U', R) \to (U', R')$ consistent with f, we have $r' \notin \mathrm{Reach}_{G-U'}(R)$, for any $r' \in R' \setminus R$. This is not a proper restriction to the robber moves either. Indeed, running within the same component makes no sense, as the set of vertices reachable for the robber does not change. Running outside of the current component makes even less sense, as that set becomes smaller.

Lemma 20. *If r robbers have a winning strategy against k cops, then r robbers have an isolating prudent winning strategy against k cops.*

Proof. Given a set of vertices U, we say that R and $\hat{R}$ are equivalent, $R \equiv_U \hat{R}$, if for all $r \in R$ there is some $\hat{r} \in \hat{R}$ and vice versa, for all $\hat{r} \in \hat{R}$ there is some $r \in R$ such that r and $\hat{r}$ are in the same component of $G - U$.

Let f be a positional winning strategy for r robbers in the monotone multiple robbers game on G against k cops. We construct an isolating prudent strategy $\hat{f}$ for r robbers against k cops by induction on the play length playing simultaneously a play π consistent with f and a play π' consistent with $\hat{f}$. We translate a cop move from $\hat{\pi}$ to π and a robber move from π to $\hat{\pi}$ and show the following. If $(U, U', R) \to (U', R')$ is the ith robber move in π and $(\hat{U}, \hat{U}', \hat{R}) \to (\hat{U}', \hat{R}')$ is the ith robber move in $\hat{\pi}$, then

- $U = \hat{U}$ and $U' = \hat{U}'$, and
- $\mathrm{Reach}_{G-U'}(R') \subseteq \mathrm{Reach}_{G-U'}(\hat{R}')$.

Clearly, this implies that $\hat{f}$ is a winning strategy. In the beginning of a play, the first robber move is translated as in the general case. Cop moves are translated without any change, so the invariant is not broken.

For the translation of a robber move, consider the topological order $\underline{\rightsquigarrow}$ on vertices of $G - U'$ where $v \underline{\rightsquigarrow} w$ if $w \in \mathrm{Reach}_{G-U'}(v)$. Let $\alpha\colon G\big/_{\equiv_{U'}} \to G$ be a choice function on $G\big/_{\equiv_{U'}}$. For every set of vertices R, let $\mathrm{topomin}\colon 2^G \to 2^G$ be defined by $\mathrm{topomin}(R) = \{\alpha(r) \mid r \text{ is } \underline{\rightsquigarrow}\text{-minimal in } R\}$.

Let $A = \{v' \in R' \setminus R \mid v' \in \mathrm{Reach}_{G-U'}(R)\}$ and let $\beta\colon A \to R$ be some function with $v' \in \mathrm{Reach}_{G-U'}(\beta(v'))$. It is exists by the definition of robber moves. Let $\gamma\colon R' \to G$ with $\gamma(v') = \beta(v')$ if $v' \in A$ and $\gamma(v') = v'$ otherwise. If f prescribes to move from (U, U', R) to (U', R'), then $\hat{f}$ prescribes to move from $(U, U', \hat{R})$ to $(U', \hat{R}')$ where $\hat{R}' = \mathrm{topomin}(\{\gamma(v') \mid v' \in R'\})$. Then $\hat{f}$ is isolating and prudent. Note that, by the first part of the invariant, strategy $\hat{f}$ is well defined. The invariant follows directly from the construction.

□

7.1. Tree-width and componentwise hunting

Before we prove our main result of this section, let us first consider the same problem for the game characterizing tree-width.

Lemma 21. *For all G and $k, r > 0$, if $\mathrm{tw}(G)+1 \leq k$, then $\mathrm{tw}_r(G)+1 \leq r\cdot(k+1)$.*

It follows from this lemma that if tree-width is fixed, then parity games with bounded imperfect information are solvable in polynomial time because $\mathrm{tw}_r(G) \leq r \cdot k$ implies $\mathrm{dagw}_r(G) \leq r \cdot k$. As the path-width of a graph is always at least its tree-width, we obtain the same result for (undirected) path-width.

Proof. Without loss of generality let G be undirected. Let f be a monotone winning strategy for k cops in the game on G against one robber. As f is monotone, we can assume that cops are not placed on vertices that are already unavailable for the robber, i.e., for a move $(U, v) \to (U, U', v)$ we always have $U' \setminus U \subseteq \mathrm{Reach}_{G-(U\cap U')}(v)$. (Otherwise, instead of f, consider a strategy that is as f, but never places cops on such vertices. This strategy will be still monotone and winning and will use at most k cops.) We construct a monotone strategy $\otimes_r f$ for $k \cdot r$ cops in the game on G with r robbers that is winning against each isolating robber strategy.

Intuitively, the cop player uses r teams of cops with k cops in each team. Every team plays independently of each other chasing its own robber according to f. We maintain the invariant that in each cop position (U, R) that is consistent with $\otimes_r f$, there is a partition $(U_1, \cdots, U_r)$ of U and an enumeration of $v_1, \cdots, v_r$ of R such that for each v_i, $(U \setminus U_i) \cap \mathrm{Reach}_{G-U_i}(v_i) = \emptyset$, i.e., cops on U_i block v_i from other cops, and that (U_i, v_i) is consistent with f in the game with one robber. The next move of the cops is $\otimes_r f(U, R) = \bigcup_{i=1}^{r} f(U_i, v_i)$. By a simple induction on the length of a play it is easy to see that the invariant holds, which implies that the cops monotonically catch all r robbers. □

The reason why the proof is so simple is that in an *undirected* graph the set of vertices reachable from a given position is precisely the connected component which contains these positions. Thus the strategy f does not need to place cops on vertices outside the robber component. For directed graphs, this is not true and the simple translation of strategies is not possible without certain refinement any more. Consider the following possible situation. The cops play simultaneously against all robbers according to a winning strategy f in the game against one robber as before. A slightly different variant of this approach (that will be used in the proof of Theorem 25) is that they choose one of them (say, occupying some vertex v_1) to play against him further while the cops of other teams wait for this robber to be caught. The robbers stay in two distinct components on v_1 and v_2. The problem is that v_2, may prevent playing against v_1. If f says to place a cop on a vertex v that is reachable from v_2, it may become impossible to reuse the cop from v later playing against v_1, although f prescribes to do so: v_2 would induce non-monotonicity on v.

One approach to solve this problem is to change f such that it does not prescribe to place cops outside of the robber component. It would suffice to prove that there is a function $F : \omega \to \omega$ such that every cop winning strategy f for k cops against one robber can be transformed into a winning strategy f' for $F(k)$ cops against one robber that never prescribes to place cops outside of the robber component. In other words, strategy f' should fulfill the following property: in a position (U, v), if C is the component of $G - U$ with $v \in C$, then $f'(U, v) \subseteq C$. However, such a function F does not exist, as we will show in Theorem 23. For this proof we need a statement about cop strategies. In the next lemma we show that any cop positional winning strategy for the game with one robber can be modified without using additional cops to obtain a new positional strategy that obeys the following rules. It does not place a cop on a vertex that is already unavailable for the robber and always prescribes to place new cops. In a graph G, for a set A and a vertex $v \notin A$, let $\mathrm{front}_G(v, A)$ be the inclusion minimal subset B of A such that $\mathrm{Reach}_{G-A}(v) = \mathrm{Reach}_{G-B}(v)$. It is easy to see that B is unique and thus well-defined.

Lemma 22. *On a graph G, if f is a positional monotone winning strategy for k cops against one robber, then there is a positional monotone winning strategy f^* for k cops against one robber, such that, for every position (U, v) that appears in a play consistent with f^*, we have that $f^*(U, v) \setminus U \neq \emptyset$ and that any $u \in f^*(U, v) \setminus U$ is reachable from v in $G - U$.*

Proof. We construct f^* by induction on the length of the finite prefixes π of plays consistent with f together with finite prefixes π^* of plays consistent with f^* such that the following invariant holds:

- $|\pi^*| \leq |\pi|$;
- if $\mathrm{last}(\pi) = (U, v)$ and $\mathrm{last}(\pi^*) = (U^*, v^*)$, then
    - $v = v^*$,
    - $U^* \subseteq U$ and
    - $\mathrm{Reach}_{G-U}(v) = \mathrm{Reach}_{G-U^*}(v)$;
- if $\mathrm{last}(\pi) = (U, U', v)$ and $\mathrm{last}(\pi^*) = (U^*, U^{*\prime}, v^*)$, then

- $v = v^*$,
- $U^* \subseteq U$, $U^{*\prime} \subseteq U'$ and
- $\mathrm{Reach}_{G-(U\cap U')}(v) = \mathrm{Reach}_{G-(U^*\cap U^{*\prime})}(v)$.

Notice that the invariant immediately implies that f^* is winning for the cops.

For a cop position (U, v), let $(U = U_0, v), (U_1, v), \dots, (U_m, v)$ be defined by the following rule:

- $U_0 = U$, $U_1 = f(U_0, v)$ (so $m \geq 1$), and
- if $\mathrm{front}_G(v, U_{i-1}) \neq \mathrm{front}_G(v, U_i)$, then $U_{i+1} = f(U_i, v)$, otherwise, $m = i$ and U_{i+1} does not exist.

Then $f^*(U, v) = U_m$. Intuitively, we skip all cop moves according to f in which new cops are only placed on or removed from vertices behind the front, i.e., on vertices that are not reachable from the robber vertex because of other cops. The next cop move according to f^* is the first move according to f where the cops are placed between the robber and the front (when the front changes) under the assumption that the robber does not move while the cops move behind the front.

At the beginning, we have $\pi = \pi^* = \perp$ and the invariant trivially holds. In general, let f^* be defined for all positions in plays up to a certain length. Consider finite histories π and π^* as above.

Let $\mathrm{last}(\pi) = (U, U', v)$ and $\mathrm{last}(\pi^*) = (U^*, U^{*\prime}, v)$ and let the robber move from $\mathrm{last}(\pi^*)$ to a position $(U^{*\prime}, v')$. Then we extend π by position (U', v') and the invariant holds again.

Let $\mathrm{last}(\pi) = (U, v)$ and $\mathrm{last}(\pi^*) = (U^*, v^*)$. Then the next move of the cops is to $f^*(U^*, v)$ and the invariant still holds. As $U^* \subseteq U$ and $U^{*\prime} \subseteq U'$, f^* uses at most k cops. Note that f^* is positional. □

Now we prove that the cops have to place themselves outside of the robber component.

Theorem 23. *For $n > 0$ there are graphs G_n such that* $\mathrm{dagw}(G_n) \leq 3$ *for all n, but any winning cop strategy which is restricted to place cops only inside the robber component, uses at least $n + 1$ cops.*

Proof. The graph G_n is the disjoint union of an undirected and a directed tree, both of the same shape: full trees of branching degree and depth $n + 1$, with some additional edges connecting the trees, see Figure 6.

Let, for $i \in \{0, 1\}$ and $m, n > 0$, $A(i, m, n) = (\{1, \dots, n\} \times \{i\})^{\leq m}$ be the set of all sequences of length at most m over the alphabet $\{1, \dots, n\}$ labeled with i (the labeling is used to distinguish the trees). For $v = (v_0, i), \dots, (v_l, i) \in A(i, m, n)$, let v' be the word $(v_0, 1 - i), \dots, (v_l, 1 - i) \in A(1 - i, m, n)$.

The vertex set of G_n is defined by $V_n = V_n^0 \cup V_n^1$ where $V_n^0 = A(0, n, n + 1)$ and $V_n^1 = A(1, n, n + 1)$.

The edges are defined by $E_n = E_n^0 \cup E_n^1 \cup E_n'$. Hereby

$$E_n^0 = \{(v, vj), (vj, v) \mid v \in A(0, n-1, n+1), j \in A(0, 1, n+1)\},$$

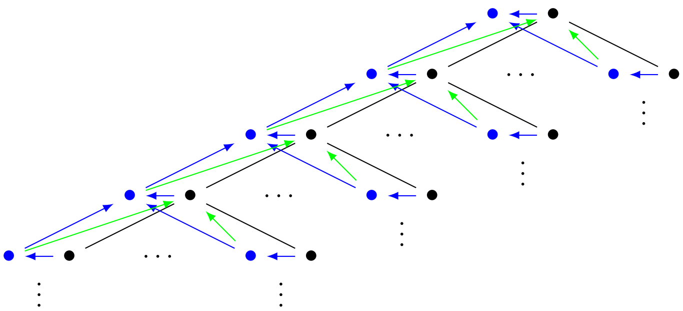

Figure 6: $\mathrm{dagw}(G_n) = 4$, but the robber wins against n cops if they move only into his component.

$$E_n^1 = \{(vj, v) \mid v \in A(1, n-1, n+1), j \in A(1, 1, n)\}\,, \text{ and}$$

$$E_n' = \{(v, v') \mid v \in A(0, n, n{+}1)\} \cup \{(vj, v') \mid v \in A(1, n{-}1, n{+}1), j \in A(1, 1, n)\}\,.$$

It is easy to see that cops three capture the robber. They occupy both roots $(0, 1)$ and $(1, 1)$ in the first move. By symmetry we can assume that the robber goes to the left-most subtree. Then the third cop is placed on the successor $(1, 2)$ of $(1, 1)$ and then the cop from $(1, 1)$ is moved to $(0, 2)$. In this manner, the cops work through both trees top-down and the robber is captured in some leaf.

For the second statement, define $T_n^0 = (V_n^i, E_n^i)$ for $i = \{0, 1\}$ and note that it makes no sense for the cops to leave out holes, i.e., to place cops on subtrees of T_n^0 or T_n^1 rooted at a vertex $v \in V_n^0$, respectively, at v', if v is reachable from the robber vertex via some cop free path. Indeed, due to the high branching degree, the robber can switch between subtrees of v going into those having no cop in them until v is occupied by a cop. In that position the cops from other components than that of the robber can be removed by Lemma 22. So we can assume that the cops play top-down, i.e., they never leave out holes. Then the robber strategy is just to stay in the left-most branch. Note that after a vertex $v \in V_n^0$ is occupied by a cop, vertex $v' \in V_n^1$ is not in the robber component any more. Thus the cops occupy successively $(\varepsilon, 0)$, $(1, 0)$, $(2, 0)$, and so on. In that way, more and more cops become tied, i.e., for every cop on a vertex $(j, 0)$, there is a cop-free path from the robber vertex to $(i, 0)$. □

Before we proceed with the case of directed graphs, let us mention that bounded tree-width of the *Gaifman graphs* of given games already implies that parity games with imperfect information are solvable in PTIME. The Gaifman graph of a relational structure $S = (A, R_1, \ldots, R_m)$ is the undirected graph with vertices A and an edge between vertices v and w if v and w appear in the same tuple in some relation R_i for $i \in \{1, \ldots, m\}$. Thus a $\sim$-equivalence class in a game with imperfect information induces a clique in the Gaifman graph consisting of all equivalent vertices. Thus if the tree-width of the Gaifman graphs of some games is bounded, then so is the imperfect information. This implies the following corollary.

Corollary 24. *Parity games whose Gaifman graphs have bounded tree-width can be solved in deterministic polynomial time.*

7.2. Generalization to the directed case

We are ready to prove our main result of Section 6.

Theorem 25. *For* $k, r > 0$*, if* $\mathrm{dagw}(G) \leq k$*, then* $\mathrm{dagw}_r(G) \leq k \cdot r$.

The rest of the section is devoted to the proof of this theorem. Let f be a positional monotone winning strategy for k cops against one robber on a directed graph G. According to Lemma 22 we can assume without loss of generality that for any history π' consistent with f such that $\mathrm{last}(\pi') = (U, v)$ we have $f(\pi') \setminus U \neq \emptyset$ and any $u \in f(\pi') \setminus U$ is reachable from v in $G - U$. Moreover, due to Lemma 20 it suffices to construct a strategy $\otimes_r f$ for $r \cdot k$ cops against r robbers which is winning against all isolating prudent robber strategies. First, we sketch a description of a memory strategy $\otimes_r f : \mathcal{M} \times (2^V \times 2^V) \to 2^V$ and the corresponding memory structure.

Without loss of generality we can assume that G is strongly connected. Indeed, given a winning strategy for $k \cdot r$ cops against r robbers on every strongly connected component of G, we can traverse the graph by applying the strategy to the topologically minimal components, then eliminate them and continue in that way until all robbers are captured.

Informal description, some invariants and some elements of the memory structure

The cops play in r teams of k cops. Consider a position (U, R) in a play with r robbers. With every vertex $v \in R$ occupied by a robber, we associate a team of cops $U_i \subseteq V$ with $|U_i| \leq k$. With each U_i we associate a history ρ_i of the game against one robber that is consistent with f such that (U_i, v) is the last position of ρ_i. We formulate this as an invariant in the game with r robbers:

(Cons) Any history ρ_i is consistent with f.

For any position that appears in a play against r robbers, we keep $s \leq r$ histories ρ_i in memory and write $\rho = \rho_1 \cdot \ldots \cdot \rho_s$. This sequence of histories is the main part of the memory. The following invariant says that, up to the last robber moves, all ρ_i are linearly ordered by $\sqsubset$.

(Lin) $\rho_1 \sqsubset \rho_2 \sqsubset \ldots \sqsubset \rho_s$.

Sequence ρ is constructed and maintained in the memory in the following way. At the beginning of a play, we set $\rho = \rho_1 = \perp$, i.e., ρ consists of one play prefix containing only the initial position. When the play with r robbers goes on, but only one robber is in the graph, ρ_1 grows together with the play with r

robbers and the latter gets the form $\bot \cdot (U^1, R^1) \cdots (U^m, R^m)(U^m, U^{m+1}, R^m)$ where all R_i are singletons. While playing this part of the play, all teams make the same moves according to f. We store the sequence in the memory as $\rho = \rho_1 = \bot \cdot (U^1, v^1) \cdots (U^m, v^m)(U^m, U^{m+1}, v^m)$ where $\{v^i\} = R^i$ (see Figure 7). When more robbers come into the graph, they go into different components (because they play according to an isolating strategy) and the cops choose one of them, say on a vertex b_1. We associate $\rho_2 = \rho_1 \cdot (U^m, U^{m+1}, b_1)$ with that robber and store $\rho = \rho_1 \cdot \rho_2$ in the memory. Note that ρ_1 ends with a robber position. Assume for a moment that only the robber in $\mathrm{C}_U(b_1)$ moves where U is the placement of the cops in the position when new robbers entered the graph. Then only this robber is pursued by its team of cops according to f, but cops are not placed on vertices v if $v \in \mathrm{Reach}_{G-U^m}(R_1)$ where R_1 is the set of robbers distinct from b_1. The cops belonging to other teams remain idle. Cop moves are appended to ρ_2, however, without respecting the omitted placements. To put it differently, let W_i be the last cop placement in ρ_i and let b_2 be the last robber vertex in ρ_2. Then in a position (U, R) of the game with r robbers, we have

$$\otimes_r f(U, R) = (U \setminus W_2) \cup \big(f(W_2, b_2) \setminus \mathrm{Reach}_{G-W_1}(b_2)\big)\,.$$

Hereby, $U \setminus W_2$ are cops from the team associated with ρ_1. Note that $\otimes_r f$ depends also on the memory state, but we will not write this explicitly. For the memory state update, in ρ_2, not the actual move $f(W_2, b_2) \setminus \mathrm{Reach}_{G-W_1}(b_2)$ is stored, but the intended one, i.e., $f(W_2, b_2)$. If later new robbers come and occupy different components of $\mathrm{C}_U(b_2)$, we again choose one of them (say, on b_3), create ρ_3 and set ρ_3, W_3 and b_3 analogously to ρ_2, W_2 and b_2, and store $\rho = \rho_1 \cdot \rho_2, \rho_3$. Analogously, the cops play according to

$$\otimes_r f(U, R) = (U \setminus W_2) \cup \Big(f(W_3, b_3) \setminus \big(\mathrm{Reach}_{G-W_2}(b_2) \cup \mathrm{Reach}_{G-W_1}(b_1)\big)\Big)\,.$$

Histories in ρ are subject to change, so at different points of time, ρ and ρ_i are different objects, but we will not reflect that in our notation to avoid unnecessary indexes. It will be always clear from the context what ρ is. Note that cops from teams $\sqsubset$-smaller than 3 (in general, s) cannot be removed from their vertices, as, according to f, omitted placements must be performed first. Hence, taking the cops may infer non-monotonicity. For example, both cops from ρ_1 in Figure 7 cannot be removed before the omitted placement in ρ_1 is performed. Note also that there may be more than one robber in R_i associated to a play ρ_i if $i < s$ and at most one robber is associated with ρ_s.

Now we describe the remaining elements of the memory. A complete element of the memory structure has the form

$$\zeta = (\rho_1, R_1, O_1) \cdot \ldots \cdot (\rho_{s-1}, R_{s-1}, O_{s-1}) \cdot \rho_s\,.$$

Hereby ρ_i are as before and, for $i < s$, ρ_i ends with a robber position. The last robber moves associated with ρ_i (other robbers may join the robber from ρ_i) are stored in R_i. Whether ρ_s ends with a robber or a cop position depends on the

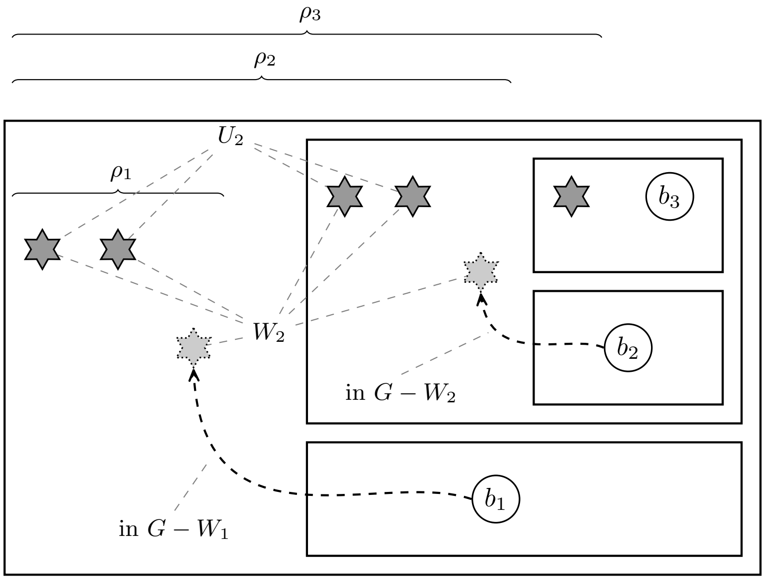

Figure 7: Memory used by strategy $\otimes_r f$ on the graph G. Squares are robber components. Stars denote cop vertices, dotted light gray stars denote vertices where cop placements were omitted.

current position in the game with r robbers: either both end with a cop position, or both end with a robber position. Set R_i represents the vertices occupied by robbers that are associated with ρ_i. Elements O_i are sets of vertices where cops of longer histories are not placed because, roughly, those vertices are reachable from R_i in $G - W_i$. However, we will see later that, in fact, sets O_i are more dynamic.

The strategy we described so far is the strategy from as constructed in the case of undirected graphs, just this time with omitted placements of cops. Now we drop the assumption that robbers from R_i stay idle. They may prevent the cops to play against the robber from the longest history ρ_s. One possibility is that one of them, say the robber from $b_i \in R_i$, for some $i < s$, jumps to the robber on b_s, in a position (U, U', R) of the game with r robbers. Then both robbers (the one from b_s and the one who jumped to b_s) run to vertices b'_i and b'_s in different components of $\mathrm{C}^{U}_{U'}(b_s)$.[3] Now some cops from U' may be reachable from b'_i and cannot be removed as f may prescribe to play against b'_s later. Previously, we used cops from team corresponding to robber b'_i who remained on U_i (which is now U') and cops from team b'_s pursued b'_s. Thus we have to reuse cops from U_i, but they cannot be just removed before cop placements are made up that were omitted because of the robber on b_i. Instead, we let the

[3] Recall the definition of a $\mathrm{C}^{U}_{U'}(b_s)$ on Page 5.

cops from U_i play according to f from U_i until they occupy the same vertices as cops from U_{i+1} of the next longer history. While this is done the cop vertices are stored in ρ_i. Then ρ_i and ρ_{i+1} are merged.

Note that it does not suffice to catch up all moves between the ends of ρ_i and ρ_{i+1} in one move placing cops as in the last position of ρ_{i+1}. The robber may use the absence of the cops in the intermediate positions and run to a vertex such that the resulting placement of that robber and the cops is not consistent with f.

There is an other case when the cops have to play in a different way: the robber corresponding the longest history is captured or jumps away. In this case, his component is not reachable for any robber any more, as the robbers play according to an isolating strategy. We remove the cops from the graph placed since the last position in ρ_{s-1}, i.e., since the last time the robbers from ρ_{s-1} and ρ_s ran into different components. Then we choose another robber from R_{s-1} to chase and append a new history to ρ.

Formal description, the rest invariants and the full memory structure

Now we present the strategy $\otimes_r f$ and the memory updates formally. Given a position (U, R) or (U, U', R) of the game with r robbers and a memory state

$$\zeta = ((\rho_1, R_1, O_1), \ldots, (\rho_{s-1}, R_{s-1}, O_{s-1}), \rho_s)\,,$$

we define the new set $U' = \otimes_r f((U, R), \zeta)$ of vertices occupied by cops (if the current position belongs to the cops) and the new memory state

$$\zeta' = ((\rho'_1, R'_1, O'_1), \ldots, (\rho'_{s'-1}, R'_{s'-1}, O'_{s-1}), \rho'_{s'})\,.$$

We also maintain some additional invariants. To describe them, we define W_i^{-1}, W_i, W^i, b_i, U_i, R_i and O_i such that

- $\mathrm{last}(\rho_i) = (W_i^{-1}, W_i, b_i)$, for $i \in \{1, \ldots, s-1\}$,
- $\mathrm{last}(\rho_s) \in \{(W_s, b_s), (W_s^{-1}, W_s, b_s)\}$,
- $U_i = W_i \setminus O^{i-1}$, $U^i = \bigcup_{j=1}^{i} U_j$ and $W^i = \bigcup_{j=1}^{i} W_j$, for $i \in \{1, \ldots, s\}$,
- $R^i = \bigcup_{j=1}^{i} R_j$ and $O^i = \bigcup_{j=1}^{i} O_j$, for $i \in \{1, \ldots, s-1\}$,
- $R_s = \{b_s\}$, if $b_s \in R$ and $R_s = \emptyset$ otherwise.

In other words, W_i is the placement of the cops in the last position of the play ρ_i as it is stored (without respecting that some moves were omitted), b_i is the stored position of the robber in that play (but the robber may be somewhere else in the play with r robbers). Furthermore, $U_i \subseteq U \cap W_i$ is the set of cops who are indeed placed and belong to ρ_i, and O_i is the set of vertices on which we do not place cops from $\sqsubseteq$-greater plays even if f prescribes to do so.

Invariants.

(Robs) The sets R_i are pairwise disjoint and $R = \bigcup_{i=1}^{s} R_i$.

(Cops) $U = \bigcup_{i=1}^{s} U_i$.

(Omit) For all $i \in \{1, \ldots, s-1\}$, $R_i \subseteq O_i = \mathrm{Reach}_{G-W_i}(O_i)$.

(Ext) For all $i \in \{1, \ldots, s-1\}$, $O_i \subseteq \mathrm{Reach}_{G-W_i^{-1}}(b_i)$.

Conditions (Omit) and (Ext) describe what sets O_i actually are. We assume that a robber may occupy or reach b_i. From here, he threatens all vertices that are reachable from b_i in $\mathrm{Reach}_{G-W_i^{-1}}(b_i)$, i.e., if he is bounded in his moves only by his own cops W_i^{-1}. Note that W_i^{-1} are the cops from the previous position of ρ_i, but the cops W_i are not placed yet: the robber can run in $G-(W_i^{-1} \cap W_i)$, but, as f is monotone, we can consider $G - W_i^{-1}$ instead of $G - (W_i^{-1} \cap W_i)$. In particular, the placement R_i of the robbers is reachable from b_i in $G - W_i^{-1}$. Furthermore, O_i are closed under reachability *after* the cops are placed on W_i.

In addition to (Cops), we also assume that, if (U, R) is a cop position and $b_s \in R$ (the stored vertex of the robber in the longest play is indeed occupied by a robber), then $\mathrm{last}(\rho_s) = (W_s, b_s)$.

The first part of (Omit) together with (Ext) guarantees that the last move of each robber who is associated with ρ_i is consistent with it.

Lemma 26. *For all $i \leq s$ and for all $b \in R_i$, $\rho_i \cdot (W_i, b)$ is consistent with f.*

Proof. By (Omit) we have $b \in O_i$ and therefore, using (Ext), we obtain that b is reachable from b_i in $G - W_i^{-1}$. Moreover, as $\mathrm{last}(\rho_i) = (W_i^{-1}, W_i, b_i)$ and ρ_i is consistent with f according to (Cons), $\rho_i \cdot (W_i, b)$ is consistent with f as well. □

The next lemma, which follows from the monotonicity of f, states that every (stored) robber is bounded by his cops on their last vertices and is not affected by previous placements.

Lemma 27.

(1) For any $i \in \{1, \ldots, s-1\}$ and any $b \in R_i$, $\mathrm{Reach}_{G-W_i}(b) = \mathrm{Reach}_{G-W^i}(b)$.

(2) $\mathrm{Reach}_{G-W_s}(b_s) = \mathrm{Reach}_{G-W^s}(b_s)$.

Proof. Consider some $i \in \{1, \ldots, s-1\}$ and some $b \in R_i$. As $W_i \subseteq W^i$, we have $\mathrm{Reach}_{G-W_i}(b) \supseteq \mathrm{Reach}_{G-W^i}(b)$, so assume that the converse inclusion $\mathrm{Reach}_{G-W_i}(b) \subseteq \mathrm{Reach}_{G-W^i}(b)$ does not hold. Then there is some $u \in W^{i-1} \setminus W_i$ such that $u \in \mathrm{Reach}_{G-W_i}(b)$. Now if $j \in \{1, \ldots, i-1\}$ such that $u \in W_j$, then due to (Lin), $\rho_j \sqsubset \rho_i$. Moreover, $\mathrm{last}(\rho_j) = (W_j^{-1}, W_j, b_j)$ and, by Lemma 26, $\rho_i \cdot (W_i, b_i)$ is consistent with f, but as ρ_j is consistent with f as well due to (Cons), $\mathrm{Reach}_{G-W_i}(b) \cap W_j \neq \emptyset$ contradicts the monotonicity of f (which is violated in position (W_i^{-1}, W_i, b_i)).

For b_s, the argument is the same. □

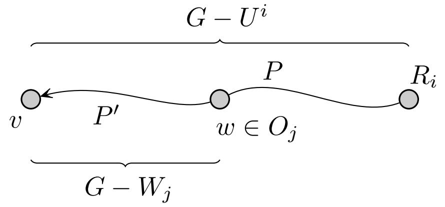

Figure 8: $v \in \text{Reach}_{G-W_j}(O_j)$ implies $v \in O_j$ by (Omit)

The following lemma is one of the key arguments for monotonicity of $\otimes_r f$. It states that the robbers (who are indeed on the graph in the play with r robbers) associated with play ρ_i are bounded by the cops (who are indeed on the graph in the play with r robbers) in a way that they can reach only vertices in O_i, which are not occupied by cops from longer plays. The lemma can be directly derived from (Omit) without using other invariants.

Lemma 28. *For $i \leq s-1$, $\text{Reach}_{G-U^i}(R_i) \subseteq O^i$.*

Proof. Let $v \in \text{Reach}_{G-U^i}(R_i)$ and let P be a path from R_i to v in $G - U^i$ as show in Figure 8. If $v \in \text{Reach}_{G-W_i}(R_i)$, then by (Omit) we have $v \in \text{Reach}_{G-W_i}(O_i) = O_i \subseteq O^i$. Let therefore $v \notin \text{Reach}_{G-W_i}(R_i)$. Then $P \cap W_i \neq \emptyset$ and we consider the minimal $l \leq i$ such that $P \cap W_l \neq \emptyset$ and some $w \in P \cap W_l$. As $P \cap U^i = \emptyset$ we have $w \notin U^i$ and thus $w \notin U_l$, as $U_l \subseteq U^i$ by the definition of U^i. As $U_l = W_l \setminus O_{l-1}$, this yields $w \in O^{l-1}$, that means, $w \in O_j$ for some $j < l$. Now v is reachable from w in G via some path $P' \subseteq P$ and, due to the minimal choice of l, $P \cap W_j = \emptyset$. Hence, $P' \cap W_j = \emptyset$, see Figure 8. This yields $v \in \text{Reach}_{G-W_j}(w) \subseteq \text{Reach}_{G-W_j}(O_j)$ and as, by (Omit), $\text{Reach}_{G-W_j}(O_j) = O_j$ it follows that $v \in O_j \subseteq O^i$. □

Finally, we formulate the fact that the reachability area of a robber is not restricted by cops of longer histories as a direct corollary of Lemma 28.

Corollary 29. *For all $i \in \{1, \ldots, s-1\}$ and all $b \in R_i$ we have* $\text{Reach}_{G-U}(b) = \text{Reach}_{G-U^i}(b)$.

We proceed with a description of $\otimes_r f$ and the memory update.

Initial Move. As we assumed that G is strongly connected, by Lemma 20, the robbers do not split in the first move. So let the initial move be $\perp \to (\emptyset, \{b\})$. After the move, the memory state is set to $\rho = \rho_1 = \big((\emptyset, b)\big)$. All the invariants hold obviously for $(\emptyset, \{b\})$ and $\big((\emptyset, b)\big)$.

Now we consider some cop position (U, R) and some memory state ζ such that all invariants are fulfilled.

Move of the Cops. In the following, we define the new set $U' = \otimes_r f((U,R),\zeta)$ of vertices occupied by cops and the new memory state

$$\zeta' = ((\rho'_1, R'_1, O'_1), \ldots, (\rho'_{s'-1}, R'_{s'-1}, O'_{s-1}), \rho'_{s'}) \,.$$

Case I: $b_s \notin R$
That means, the robber b_s which is stored in the longest history is not on the graph any more. Hence, if $s = 1$ (the memory contains only one history), then that robber has been captured and, as there are no other robbers, all the robbers are captured and the cops have won. Otherwise, we set $U' := U^{s-1} = \bigcup_{i=1}^{s-1} U_i$, i.e., we remove the cops corresponding to the longest history from the graph. For the memory update, consider ρ_{s-1} and distinguish two cases:

- $R_{s-1} = \emptyset$
  That means, there are no robbers on the graph that are associated with the next longest history. The new memory state ζ' is obtained from ζ by deleting ρ_s and replacing $(\rho_{s-1}, R_{s-1}, O_{s-1})$ by the history $\rho_{s-1} \cdot (W_{s-1}, b_s)$. Note that we could delete all last plays from ρ that have no associated robbers on the graph at once, but, for the ease of proving our invariants, we do it step by step.

- $R_{s-1} \neq \emptyset$
  In this case, we have to select one of the robbers from R_{s-1} that we want to pursue next. Choose some robber $b \in R_{s-1}$ and define $\tilde{O}_{s-1} := \mathrm{Reach}_{G-W_{s-1}}(R_{s-1} \setminus \{b\})$. Then the new memory state ζ' is obtained from ζ by replacing $(\rho_{s-1}, R_{s-1}, O_{s-1})$ by $(\rho_{s-1}, R_{s-1} \setminus \{b\}, \tilde{O}_{s-1})$ and replacing ρ_s by $\rho_{s-1} \cdot (W_{s-1}, b)$.

Case II: $b_s \in R$.
Case II.1: There is some $i \in \{1, \ldots, s-1\}$ such that $R_i = \emptyset$.
That means, there is no robber associated with history ρ_i. First, consider the next robber move in ρ_i according to ρ_{i+1} (note that $i < s$, so ρ_{i+1} exists). Consider a vertex $\tilde{b}_i \in V$ and the suffix η of ρ_{i+1} such that $\rho_{i+1} = \rho_i \cdot (W_i, \tilde{b}_i) \cdot \eta$. We distinguish three more cases.

(a) $\rho_{i+1} = \rho_i \cdot (W_i, \tilde{b}_i) = \rho_s$, i.e., η is empty.
In this case, ρ_i already reached the end of ρ_s, but is not deleted yet. Indeed, all histories ρ_i, for $i \leq s-1$, end with a robber position. If η is empty, then $i = s-1$. Set $U' := U$, i.e., the cops stay idle, and update the memory by deleting (ρ_i, R_i, O_i) from ζ.

For the other cases, we set

- $\tilde{W}_i := f(W_i, \tilde{b}_i)$ and
- $U' := \bigcup_{j \neq i} U_j \cup (\tilde{W}_i \setminus O^{i-1})$

to define the next cop move and

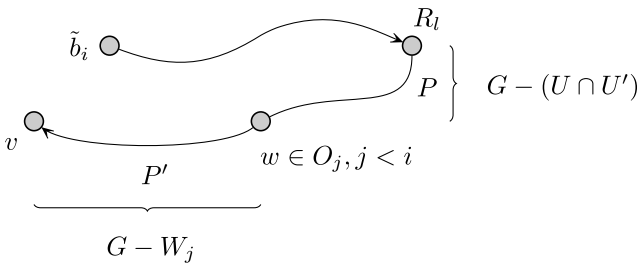

Figure 9: Robbers from longer histories than ρ_i cannot cause non-monotonicity.

- $\tilde{O}_i = (O_i \cap \mathrm{Reach}_{G-W_i}(\tilde{b}_i)) \setminus \tilde{W}_i$ and
- $\tilde{\rho}_i = \rho_i \cdot (W_i, \tilde{b}_i) \cdot (W_i, \tilde{W}_i, \tilde{b}_i)$

for the definition of the memory update.

(b) $\tilde{\rho}_i \neq \rho_{i+1}$.
That means, we have not reached the end of the next history. In this case, we replace (ρ_i, R_i, O_i) by $(\tilde{\rho}_i, R_i, \tilde{O}_i)$.

(c) $\tilde{\rho}_i = \rho_{i+1}$.
The memory update is to replace $(\rho_{i+1}, R_{i+1}, O_{i+1})$ by $(\rho_{i+1}, R_{i+1}, O_{i+1} \cup \tilde{O}_i)$ and to remove (ρ_i, R_i, O_i). Note how we conservatively updated O_i.

Case II.2: For all $i \in \{1, \dots, s-1\}$ we have $R_i \neq \emptyset$.
In this case, the cops play against the robber from ρ_s. We define

- $\tilde{W}_s = f(W_s, b_s)$ and
- $U' \coloneqq \bigcup_{j<s} U_j \cup (\tilde{W}_s \setminus O^{s-1})$

and, for the memory update, we replace ρ_s by $\rho'_s = \rho_s \cdot (W_s, \tilde{W}_s, b_s)$.

As a next step, we prove that the cop moves from U to U' is monotone, i.e., that no robber can reach any vertex from $U \setminus U'$ in $G - (U \cap U')$.

Lemma 30. $(U \setminus U') \cap \mathrm{Reach}_{G-(U\cap U')}(R) = \emptyset$.

Proof. We go through the cases defined in the description of the cop move.

Case I. If $b_s \notin R$ we have $U' = U^{s-1}$ so, by (Cops), $U^{s-1} \subseteq U \cap U'$. Moreover, (Robs) yields $R = \bigcup_{i=1}^{s-1} R_i$ and hence, using Lemma 28, we obtain $\mathrm{Reach}_{G-(U\cap U')}(R) \subseteq O^{s-1}$. Due to the definition of U_s we have $O^{s-1} \cap U_s = \emptyset$, which implies $\mathrm{Reach}_{G-(U\cap U')}(R) \cap U_s = \emptyset$ and thus, by (Cops), the move of $\otimes_r f$ is monotone in this case.

Case II. Here we have $b_s \in R$ and two further cases.

Case II.1: there is some $i \in \{1, \dots, s-1\}$ such that $R_i = \emptyset$. In Subcase (a), the cops stay idle, so the move is monotone. Otherwise we have $U' = \bigcup_{j \neq i} U_j \cup \tilde{U}_i$ with $\tilde{U}_i = \tilde{W}_i \setminus O^{i-1}$ where $\tilde{W}_i = f(W_i, \tilde{b}_i)$ and $\rho_{i+1} = \rho_i \cdot (W_i, \tilde{b}_i) \cdot \eta$ are as above. Assume that this move is not monotone, i.e., there is some $v \in U \setminus U'$ with $v \in \mathrm{Reach}_{G-(U \cap U')}(R)$. Then $v \in U_i \setminus \tilde{U}_i$, by the definition of U' and (Cops).

We distinguish, which robbers can reach v. First, consider robbers from smaller histories than ρ_i, that means, from the set R^{i-1}. As $U^{i-1} \subseteq U \cap U'$, by Lemma 28, we obtain $\mathrm{Reach}_{G-(U \cap U')}(R^{i-1}) \subseteq O^{i-1}$. Due to the definition of U_i, we have $O^{i-1} \cap U_i = \emptyset$ and hence $v \notin \mathrm{Reach}_{G-(U \cap U')}(R^{i-1})$, i.e., no robber from R^{i-1} can cause non-monotonicity.

As $R_i = \emptyset$, we have $v \in \mathrm{Reach}_{G-(U \cap U')}(R^{>i})$ where $R^{>i} = \bigcup_{l=i+1}^{s-1} R_l \cup \{b_s\}$ is the set of robbers from longer histories than ρ_i. Consider some path P from $R^{>i}$ to v in $G - (U \cap U')$ as shown in Figure 9.

First, we show that $v \notin \mathrm{Reach}_{G-(W_i \cap \tilde{W}_i)}(R^{>i})$, i.e., that the robber has to visit omitted vertices. For $l \in \{i+1, \dots, s-1\}$ and any $b \in R_l$, by (Lin), $\rho_i \cdot (W_i, \tilde{b}_i)$ is a strict prefix of $\rho_l \cdot (W_l, b)$ and, by Lemma 26, both of these histories are consistent with f. So, by monotonicity of f, any robber $b \in R_l$ is reachable from $\tilde{b}_i$ in $G - W_i$ and hence in $G - (W_i \cap \tilde{W}_i)$. Moreover, as we are in Case II.1 (b) or (c), the same arguments show that b_s is also reachable from $\tilde{b}_i$ in $G - W_i$ and hence in $G - (W_i \cap \tilde{W}_i)$. Therefore, if $v \in \mathrm{Reach}_{G-(W_i \cap \tilde{W}_i)}(R^{>i})$, then $v \in \mathrm{Reach}_{G-(W_i \cap \tilde{W}_i)}(\tilde{b}_i)$. But as $v \in U_i \subseteq W_i$ this contradicts monotonicity of f since $\rho_i \cdot (W_i, \tilde{b}_i) \cdot (W_i, \tilde{W}_i, \tilde{b}_i)$ is consistent with f. Hence, $v \notin \mathrm{Reach}_{G-(W_i \cap \tilde{W}_i)}(R^{>i})$.

As the robber visits omitted vertices, $P \cap (W_i \cap \tilde{W}_i) \neq \emptyset$. We consider the minimal $l \leq i$ such that $P \cap \widehat{W}_l \neq \emptyset$ where $\widehat{W}_j = W_j$ for $j < i$ and $\widehat{W}_i = W_i \cap \tilde{W}_i$. We define $\widehat{U}_j$ analogously. The meaning of $\widehat{W}_j$ is that it contains precisely the vertices occupied by cops according to ρ_j which remained idle in the last move. Let w be some vertex in $P \cap \widehat{W}_l$. First, as $w \in P$, $w \notin U \cap U'$, so (Cops) and the definition of U' yield $w \notin \widehat{U}_l$. Therefore, $w \in \widehat{W}_l \setminus \widehat{U}_l$ and hence, using the definitions of U_l and $\tilde{U}_i$, if $l = i$, we obtain $w \in O^{l-1}$, i.e., $w \in O_j$ for some $j < l$. Moreover, v is reachable from w in G via some path $P' \subseteq P$ and, due to the minimal choice of l, $P' \cap \widehat{W}_j = P' \cap W_j = \emptyset$, so $v \in \mathrm{Reach}_{G-W_j}(w) \subseteq \mathrm{Reach}_{G-W_j}(O_j) = O_j \subseteq O^{i-1}$. The last equality is due to (Omit). But as $O^{i-1} \cap U_i = \emptyset$, $v \in O^{i-1}$ is a contradiction to $v \in U_i$.

Finally, consider Case II.2, i.e., for all $i \in \{1, \dots, s-1\}$ we have $R_i \neq \emptyset$. First notice that, due to definition of U' and (Cops), $U \setminus U' \subseteq U_s$. For robbers other than b_s the same arguments as in Case I and Case II.1, using (Robs) and Lemma 28, show that they cannot cause non-monotonicity. The argument for b_s is the same as in Case II.1: assume that b_s causes non-monotonicity at some vertex v. As ρ_s is consistent with f due to (Cons) and f is monotone, b_s can reach v only via O^{s-1} (using (Cops)). However, O^{s-1} is closed under reachability in $G - U$ and v cannot be in O^{s-1}, so this is impossible. □

For the cop move, it remains to prove that all invariants still hold after the move. We first give a separate lemma for (Robs), (Lin), (Cons) and (Ext) and prove them quite briefly as they can be obtained easily from the induction hypothesis, using the definition of the cop move.

Lemma 31. *(Robs), (Lin), (Cons) and (Ext) are preserved by the cop move.*

Proof. (Robs) follows immediately from the induction hypothesis. Linearity of $\sqsubset$ is obviously preserved in Case I, Case II.1 (a) and (b) and in Case II.2. In Case II.1 (b), we have to show that $\tilde{\rho}_i \sqsubset \rho_{i+1}$. First notice that $\rho_i \cdot (W_i, \tilde{b}_i) \sqsubset \rho_{i+1}$ as $\rho_{i+1} = \rho_i \cdot (W_i, \tilde{b}_i)\eta$ and $\eta \neq \emptyset$. Furthermore, the first position in η is $(W_i, \tilde{W}_i, \tilde{b}_i)$ as ρ_{i+1} is consistent with f by (Cons) and $\tilde{W}_i = f(W_i, \tilde{b}_i)$. As $\tilde{\rho}_i \neq \rho_{i+1}$ it follows that $\tilde{\rho}_i \sqsubset \rho_{i+1}$.

For (Cons), consider first Case I. If $R_{s-1} = \emptyset$, then $\rho'_{s'} = \rho_{s-1} \cdot (W_{s-1}, b_s)$. As $\text{last}(\rho_s) \in \{(W_s^{-1}, W_s, b_s), (W_s, b_s)\}$ and $\rho_{s-1} \sqsubset \rho_s$ and due to (Cons) both of these histories are consistent with f, which is monotone, b_s is reachable from b_{s-1} in $G-(W_{s-1}^{-1} \cap W_{s-1})$, so $\rho_{s-1} \cdot (W_{s-1}, b_s)$ is consistent with f. If $R_{s-1} \neq \emptyset$, then $\rho_{s-1} \cdot (W_{s-1}, b)$ is consistent with f for any $b \in R_{s-1}$ due to Lemma 26. In Case II.1 (a) and (b), (Cons) follows immediately from the induction hypothesis. In Case II.1 (b), (Cons) follows from (Lin) as $\rho'_{s'} = \rho_s$ is consistent with f and $\tilde{\rho}_i \sqsubset \rho_s$. Finally, in Case II.2, ρ_s is consistent with f due to (Cons) and $W'_s = f(W_s, b_s)$, so $\rho'_{s'} = \rho'_s$ is consistent with f as well.

To prove (Ext) first notice that in Case I, if $R_{s-1} = \emptyset$, then (Ext) follows immediately from the induction hypothesis. Moreover, if $R_{s-1} \neq \emptyset$, then $s' = s$ and we have to show that $O'_{s-1} = \tilde{O}_{s-1} \subseteq \text{Reach}_{G-W_{s-1}^{-1}}(b_{s-1})$. As, by Lemma 26, for any $b' \in R_{s-1}$ the history $\rho_{s-1}(W_{s-1}, b')$ is consistent with f, which is monotone, the reachability area of any $b' \in R_{s-1}$ in $G-W_{s-1}$ is a subset of the reachability area of b_{s-1} in $G - W_{s-1}^{-1}$. Hence, by definition of $\tilde{O}_{s-1}$, the statement follows. In Case II, (Ext) follows easily from the induction hypothesis, using the definition of $\tilde{O}_i$ in Case II.1 (b) and (c). □

For the remaining two invariants (Omit) and (Cops), we have two separate lemmas which we prove in greater detail. The most interesting cases in the proofs of these two invariants are Cases II.1 (b) and (c). The crucial point here is the new set $\tilde{O}_i$. See Figure 10 for an illustration.

Lemma 32. *(Omit) is preserved by the cop move.*

Proof. In Case I, if $R_{s-1} = \emptyset$, (Omit) follows immediately from the induction hypothesis, so consider Case II where $R_{s-1} \neq \emptyset$. We have $s' = s$, $W'_{s-1} = W_{s-1}$ and $O'_{s-1} = \tilde{O}_{s-1} = \text{Reach}_{G-W_{s-1}}(R_{s-1} \setminus \{b\})$. Clearly, this yields that O'_{s-1} is closed under reachability in $G-W_{s-1}$. Moreover, by (Omit), $R_{s-1} \subseteq \text{Reach}_{G-W_{s-1}}(O_{s-1})$, so we have $R_{s-1} \cap W_{s-1} = \emptyset$ an hence $R_{s-1} \setminus \{b\} \subseteq \text{Reach}_{G-W_{s-1}}(R_{s-1} \setminus \{b\}) = O'_{s-1}$.

Consider Case II.1. In Case (a), (Omit) follows immediately from the induction hypothesis. In Case (b), $R'_i \subseteq O'_i$ is trivial as $R'_i = R_i = \emptyset$, so we have to show that $O'_i = \text{Reach}_{G-W'_i}(O'_i)$. We have $O'_i = \tilde{O}_i = (O_i \cap \text{Reach}_{G-W_i}(\tilde{b}_i)) \setminus \tilde{W}_i$

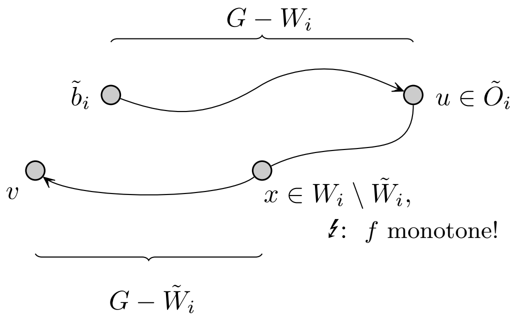

Figure 10: $\tilde{O}_i$ is closed under reachability in $G - \tilde{W}_i$

and $W'_i = \tilde{W}_i = f(W_i, \tilde{b}_i)$. Moreover, by the definition of $\tilde{O}_i$ in this case, we have $\tilde{O}_i \cap \tilde{W}_i = \emptyset$, so $\tilde{O}_i \subseteq \mathrm{Reach}_{G-\tilde{W}_i}(\tilde{O}_i)$.

As a next step, we show that $\tilde{O}_i$ is closed under reachability in $G - \tilde{W}_i$. Let $v \in \mathrm{Reach}_{G-\tilde{W}_i}(\tilde{O}_i)$. Clearly, $v \notin \tilde{W}_i$. Let $u \in \tilde{O}_i$ such that v is reachable from u in $G - \tilde{W}_i$. As $\tilde{O}_i = (O_i \cap \mathrm{Reach}_{G-W_i}(\tilde{b}_i)) \setminus W_i$, we have $u \in \mathrm{Reach}_{G-W_i}(\tilde{b}_i)$ and $v \in \mathrm{Reach}_{G-\tilde{W}_i}(u)$. Therefore, there is a cop-free path from $\tilde{b}_i$ to v via u in $G - (W_i \cap \tilde{W}_i)$. By (Cons), ρ_i is consistent with f and $\tilde{W}_i = f(W_i, \tilde{b}_i)$, so, as f is monotone, this path must be cop-free in $G - W_i$, see Figure 10. Thus, $v \in \mathrm{Reach}_{G-W_i}(u)$ and as $u \in O_i$ (by the definition of $\tilde{O}_i$) and $u \in \mathrm{Reach}_{G-W_i}(\tilde{b}_i)$, we have $v \in \mathrm{Reach}_{G-W_i}(O_i)$ and $v \in \mathrm{Reach}_{G-W_i}(\tilde{b}_i)$. By (Omit), we have $\mathrm{Reach}_{G-W_i}(O_i) = O_i$, so $v \in O_i \cap \mathrm{Reach}_{G-W_i}(\tilde{b}_i)$ and as $v \notin \tilde{W}_i$ this yields $v \in \tilde{O}_i$.

In Case (c), we have to show that $R'_i \subseteq O'_i$ and that O'_i is closed under reachability in $G - W'_i$. We have $R'_i = R_{i+1}$, $O'_i = O_{i+1} \cup \tilde{O}_i$ and $W'_i = W_{i+1}$. By (Omit), $R_{i+1} \subseteq O_{i+1} \subseteq O_{i+1} \cup \tilde{O}_i$. Moreover, as in Case (b), $\tilde{O}_i$ is closed under reachability in $G - \tilde{W}_i$ and as $\tilde{\rho}_i = \rho_{i+1}$ we have $\tilde{W}_i = W_{i+1}$. By (Omit), O_{i+1} is closed under reachability in $G - W_{i+1}$, so the union $O_{i+1} \cup \tilde{O}_i$ is closed under reachability in $G - W_{i+1}$ as well. Finally, in Case II.2, (Omit) follows again from the induction hypothesis. □

Lemma 33. *(Cops) is preserved by the cop move.*

Proof. We have to show that $U' = \bigcup_{j=1}^{s'} U'_j$ where $s' \in \{s-1, s\}$ is the length of ζ'. Note that, by the definition, $U'_j = W'_j \setminus (O^{j-1})'$ for $j = 1, \dots, s'$.

In Case I, Case II.1 (a) and Case II.2, this can easily be obtained using the induction hypothesis and the definition of U'. Consider Case II (b). We have $s' = s$ and $O'_j = O_j$, for $j \neq i$, and $O'_i = \tilde{O}_i \subseteq O_i$. As, moreover, $W'_j = W_j$ for $j < i$, we have $U'_j = U_j$, for $j < i$. Furthermore, $U'_i = W'_i \setminus (O^{i-1})' = \tilde{W}_i \setminus O^{i-1}$ and, as $(O^{j-1})' \subseteq O^{j-1}$, for $j = 1, \dots, s$, we have $U_j \subseteq U'_j$, for $j > i$. Hence, $U' \subseteq \bigcup_{j=1}^{s} U'_j$ and it remains to show $\bigcup_{j=1}^{s} U'_j \subseteq U'$.

Towards a contradiction, assume that there is some $v \in (\bigcup_{j=1}^{s} U'_j) \setminus U'$. Then $v \in U'_j$, for some $j > i$, and, as $v \notin U' \supseteq U_j$, we have $v \in W_j \setminus (O^{j-1})'$, but $v \notin O^{j-1}$. Since $O'_l = O_l$ for $l \neq i$, we have $v \in O_i \setminus O'_i = O_i \setminus \tilde{O}_i$. So, by the definition of $\tilde{O}_i$, we have $v \in \tilde{W}_i$ or $v \notin \text{Reach}_{G-W_i}(\tilde{b}_i)$. As $v \notin U'$ we have $v \notin \tilde{W}_i \setminus O^{i-1}$ and, as $v \notin O^{j-1} \supseteq O^{i-1}$, it follows that $v \notin \tilde{W}_i$, so $v \notin \text{Reach}_{G-W_i}(\tilde{b}_i)$. Let $\rho^* = \widehat{\rho}(W^{-1}, W, b)$ be the shortest prefix of ρ_j such that $v \in W$. Note that such a prefix exists as $v \in W_j$. Due to (Cons), $\tilde{\rho}_i$ and ρ^* are consistent with f and f is monotone, so since $v \notin \tilde{W}_i$ we have $\tilde{\rho}_i \sqsubset \rho^*$ and as $v \notin \text{Reach}_{G-W_i}(\tilde{b}_i)$, we also have $v \notin \text{Reach}_{G-W^{-1}}(b)$. However, this is a contradiction to the fact that f is active.

Finally, in Case (c), we have $s' = s - 1$, as we delete the ith element of ζ. Hence, we have a shift of indexes. Accounting for this fact, (Cops) can be proven analogously to the Case (b). □

Move of the Robbers. Let R' be the set of vertices occupied by robbers after their move. If $R' = R$, we do not update the memory. This happens in particular after the cop move in Case I and in Case II.1 (a) of the cop move: in those cases, we do not place new cops on the graph, so the robbers stay idle because they stick to a prudent strategy. We will not consider these cases.

Let $R' \neq R$ and consider the memory state

$$\zeta = \big((\rho_1, R_1, O_1), \ldots, (\rho_{s-1}, R_{s-1}, O_{s-1}), \rho_s\big)$$

before the robber move from R to R'. Note that $b_s \in R$.

We will also need the memory state

$$\overline{\zeta} = ((\overline{g}_1, \overline{R}_1, \overline{O}_1), \ldots, (\overline{g}_{\overline{s}-1}, \overline{R}_{\overline{s}-1}, \overline{O}_{s-1}), \overline{g}_{\overline{s}})$$

and the set U^{-1} of vertices occupied by cops before the last cop moves.

Assignment of the robbers to histories. We assign every robber $b \in R'$ to the shortest history ρ_i with $b \in O_i$, which yields the new set $\tilde{R}_i$ replacing R_i:

> If $b \in O^{s-1}$, then let $i = \min\{j \in \{1, \ldots, s-1\} \mid b \in O_j\}$ and assign b to ρ_i. Otherwise assign b to ρ_s.

The crucial point we have to prove about the memory update after a robber move is that a robber assigned to a certain history is consistent with it according to f. For the robbers assigned to histories ρ_i with $i < s$ this follows easily from the fact that $\tilde{R}_i \subseteq O_i$, similar as in Lemma 26. For the robbers in $\tilde{R}_s$ this is, however, much more involved. We have to show that each such robber can be reached from $\overline{b}_{\overline{s}} = b_s$ in the graph $G - \overline{W}_{\overline{s}}$ which then shows that prolonging the longest history by a move from b_s to some robber from $\tilde{R}_s$ yields again an f-history. This property is proved in the following lemma.

Lemma 34. $\tilde{R}_s \subseteq \text{Reach}_{G-\overline{W}_{\overline{s}}}(\overline{b}_{\overline{s}})$.

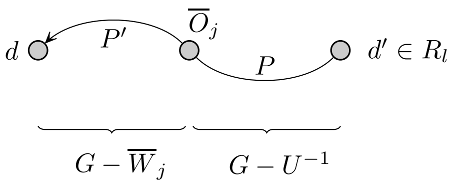

Figure 11: Any $d \in \tilde{R}_s \setminus \mathrm{Reach}_{G-\overline{W}_{\overline{s}}}(\overline{b_{\overline{s}}})$ is in $\overline{O}^{s-1}$.

Proof. Let $d \in \tilde{R}_s$. As the robbers have moved from R to R' in their move, there is some $d' \in R$ such that d is reachable from d' in $G - (U^{-1} \cap U)$. As we have already shown in Lemma 30, the move from U^{-1} to U was monotone, so d is reachable from d' in $G - U^{-1}$. Let P be a path from d' to d in $G - U^{-1}$ and assume that $d \notin \mathrm{Reach}_{G-\overline{W}_{\overline{s}}}(\overline{b_{\overline{s}}})$. We show that then $d \in O^{s-1}$ in contradiction to $d \in \tilde{R}_s$ as by the definition of $\tilde{R}_s$, $\tilde{R}_s \cap O^{s-1} = \emptyset$. By (Robs) for $\overline{\zeta}$, $R = \bigcup_{i=1}^{\overline{s}}(\overline{R}_i)$, so there is some (unique) $l \leq \overline{s}$ with $d' \in \overline{R}_l$.

First we show $d \in \overline{O}^{\overline{s}-1}$, see Figure 11. If $d' \neq \overline{b_{\overline{s}}}$, then according to (Omit) for $\overline{\zeta}$ we have $d' \in \overline{R}_l \subseteq \overline{O}_l$ and as $d' \in P$, we have $P \cap \overline{O}^l \neq \emptyset$. In the other case we have $d' = \overline{b_{\overline{s}}}$ so $d \in \mathrm{Reach}_{G-U^{-1}}(\overline{b_{\overline{s}}})$ and as, by (Cops) for $\overline{\zeta}$, $\overline{U_{\overline{s}}} \subseteq U^{-1}$, we have $d \in \mathrm{Reach}_{G-\overline{U_{\overline{s}}}}(\overline{b_{\overline{s}}})$. However, by our assumption, $d \notin \mathrm{Reach}_{G-\overline{W_{\overline{s}}}}(\overline{b_{\overline{s}}})$, so by the definition of $\overline{U_{\overline{s}}}$, $P \cap \overline{O}^{\overline{s}-1} \neq \emptyset$. Hence, in any case we have $P \cap \overline{O}_j \neq \emptyset$ for some $j \leq \min\{\overline{s}-1, l\}$ and we consider the minimal such j. Then by (Cops) for $\overline{\zeta}$, $\overline{U}_j \subseteq U^{-1}$, so d is reachable from $\overline{O}_j$ in $G - \overline{U}_j$ via a path $P' \subseteq P$, see Figure 11. So if $d \notin \mathrm{Reach}_{G-\overline{W}_j}(\overline{O}_j)$, then, by the definition of $\overline{U}_j$, we have $P \cap \overline{O}^{j-1} \neq \emptyset$, which contradicts the minimality of j. Hence, $d \in \mathrm{Reach}_{G-\overline{W}_j}(\overline{O}_j) = \overline{O}_j$ by (Omit) for $\overline{\zeta}$.

Now we show that d is also in O^{s-1}. We distinguish the moves that the cops may have made. Case I and Case II.1 (a) of the cop move do not have to be considered here as discussed above. If $\overline{O}_j = O_j$, which in particular holds in Case II.2, then $d \in O_j \subseteq O^{s-1}$. Now assume that $\overline{O}_j \neq O_j$, so we are in Case II.1 (b) or (c). Let i be as in these cases. Then for all $m < i$, we have $\overline{O}_m = O_m$, so $j \geq i$. Moreover, for all $m > i$, $\overline{O}_m = O_m$ (in Case II.(b)) or $\overline{O}_m \subseteq O_{m-1}$ (in Case II.(c)), so either $d \in O^{s-1}$ or $j \leq i$. The remaining case is $j = i$. Note that in this case, $j < l$ as either $l = \overline{s}$ and $j \leq \overline{s} - 1$ or $l < \overline{s}$. In the latter case, the reason is that $j \leq l$ and $\overline{R}_j = \overline{R}_i = \emptyset$ and $d' \in \overline{R}_l \neq \emptyset$. We show that $d \in \tilde{O}_j$, then by the definition of the memory update $d \in O_j$ and hence $d \in O^{s-1}$.

By definition, $\tilde{O}_j = (\overline{O}_j \cap \mathrm{Reach}_{G-\overline{W}_j}(\tilde{b}_j)) \setminus \tilde{W}_j$ where $\tilde{W}_j = W'_j = W_j$ and $\tilde{b}_j = b_j$. We have already shown that $d \in \overline{O}_j$. In order to see that $d \notin W_j$ notice that $d \in R'$, and $U_j \subseteq U$ according to (Cops), so $d \notin U_j$. Hence, if $d \in W_j$, we have $d \in \overline{O}^{j-1} = O^{j-1}$ by the definition of U_j, contradicting $d \in \tilde{R}_s$. Thus, $d \notin W_j$ and it remains to show that $d \in \mathrm{Reach}_{G-\overline{W}_j}(b_j)$. First notice

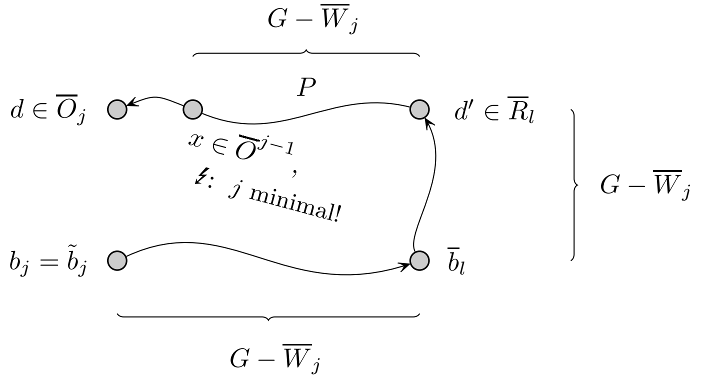

Figure 12: If $i = j < l$, the robber $\tilde{b}_j$ can still reach d in the graph $G - \overline{W}_j$ via d'.

that since $j < l$, we have $\tilde{\rho}_j \preccurlyeq \overline{g}_{j+1} \preccurlyeq \overline{g}_l$. So as, according to (Cons), all these histories are consistent with f, which is monotone, $\overline{b}_l$ is reachable from b_j in $G - \overline{W}_j$, see Figure 12. Now if $l < \overline{s}$, $d' \in \overline{R}_l$, so by (Ext), d' is reachable from $\overline{b}_l$ in $G - \overline{W}_l^{-1}$. Moreover, using again that $\tilde{\rho}_j \preccurlyeq \overline{g}_l$ are both consistent with f and that f is monotone, this yields that d' is reachable from $\overline{b}_l$ in $G - \overline{W}_j$. If, on the other hand, $l = \overline{s}$, then $d' = \overline{b}_{\overline{s}} = \overline{b}_l$, so clearly, d' is reachable from $\overline{b}_l$ in the graph $G - \overline{W}_j$. Therefore, d' is reachable from b_j in the graph $G - \overline{W}_j$ and as, by (Cops), $\overline{U}_j \subseteq U^{-1}$, d is reachable from d' in $G - \overline{U}_j$ via P. Hence, if d is not reachable from b_j in $G - \overline{W}_j$, then due to the definition of $\overline{U}_j$ there is some vertex from $\overline{O}^{j-1}$ on the path P which contradicts the minimality of j. Hence, $d \in \mathrm{Reach}_{G-\overline{W}_j}(b_j)$. □

Memory update. For the memory update, we distinguish three cases according to the number of robbers that have been assigned to $\tilde{R}_s$, and according to whether the last position of ρ_s belongs to the cops or to the robber. We simplify the case distinction by proving that if we did not play against the robber in the longest history in the last cop move, then at most the robber $b_s = \overline{b}_{\overline{s}}$ can be consistently associated with ρ_s.

Lemma 35. *If ρ_s ends with a position of the cop player, then $\tilde{R}_s \subseteq \{b_s\}$.*

Proof. Assume that ρ_s ends with a cop position, i.e., $\rho_s = \widehat{\rho}_s(W_s, b_s)$. Then the last cop moves was not as in Case II.2. As Case I and Case II.1 (a) do not need to be considered as discussed above, we have $W_s = \overline{W}_{\overline{s}}$ (and $b_s = \overline{b}_{\overline{s}}$). So Lemma 34 yields $\tilde{R}_s \subseteq \mathrm{Reach}_{G-W_s}(b_s)$. By Lemma 27 we have $\mathrm{Reach}_{G-W_s}(b_s) = \mathrm{Reach}_{G-W^s}(b_s)$, so $\tilde{R}_s \subseteq \mathrm{Reach}_{G-W^s}(b_s) \subseteq \mathrm{Reach}_{G-U}(b_s)$. Since $b_s \in R$, it follows that $\tilde{R}_s \not\subseteq \{b_s\}$ contradicts the assumption that the robbers use a prudent strategy. □

There remain two other cases.

Case 1: ρ_s ends with a position of the robber player and $|\tilde{R}_s| \geq 1$.

Intuitively, this case means that the last cop move was according to ρ_s and $|\tilde{R}_s| \geq 1$. In other words, at least one of the robbers from R' can be consistently associated with ρ_s. (As we will see in Lemma 36, it follows from (Cons) that all robbers from $\tilde{R}_s$ can be associated with ρ_s.)

We choose one of the robbers $b \in \tilde{R}_s$ which we pursue further (that means, b will be the new robber from the longest history), and add a new history $\rho_{s'} = \rho_{s+1}$ extending ρ_s by the robber move from b_s to b. The remaining robbers $\tilde{R}_s \setminus \{b\}$ are still associated with ρ_s. The new set $O'_{s'-1} = O'_s$ contains exactly the vertices reachable from $\tilde{R}_s \setminus \{b\}$ in $G - W_s$.

Formally, we choose some $b \in \tilde{R}_s$, define $\tilde{O}_s = \mathrm{Reach}_{G-W_s}(\tilde{R}_s \setminus \{b\})$ and set

$$\zeta' = \big((\rho_1, \tilde{R}_1, O_1), \dots, (\rho_{s-1}, \tilde{R}_{s-1}, O_{s-1}), (\rho_s, \tilde{R}_s \setminus \{b\}, \tilde{O}_s), \rho_s \cdot (W_s, b)\big)\,.$$

Case 2: ρ_s ends with a position of the cop player or $|\tilde{R}_s| = 0$.

This case means that either we did not play according to ρ_s, or we did, but b_s was captured or returned to an shorter ρ_i.

We define

$$\zeta' = \big((\rho_1, \tilde{R}_1, O_1), \dots, (\rho_{s-1}, \tilde{R}_{s-1}, O_{s-1}), \rho_s\big)\,.$$

Invariants after the robber move. Now we prove that all invariants still hold after the robber move.

Lemma 36. *All invariants are preserved by the robber move.*

Proof. (Robs) holds by the definition of the sets $\tilde{R}_i = R'_i$ and the construction of the memory update. (Lin) and (Cops) are obvious.

To prove (Omit), first notice that by (Omit) for ζ, each set O_i for $i = 1, \dots, s-1$ is closed under reachability in $G - W_i$ and as, for $i = 1, \dots, s-1$, we have $O'_i = O_i$ and $\rho'_i = \rho_i$, the invariant holds for all $i = 1, \dots, s-1 \geq s'-2$. Moreover, $R'_i = \tilde{R}_i \subseteq O_i = O'_i$ holds by the definition of the sets $\tilde{R}_i$ for $i = 1, \dots, s-1$. In particular, in Case 2, there is nothing to show, so consider Case 1. We have $s' = s+1$ and $O'_{s'-1} = O'_s = \tilde{O}_s = \mathrm{Reach}_{G-W_s}(\tilde{R}_s \setminus \{b\})$, so O'_s is obviously closed under reachability in $G - W_s$ and as $W'_{s'-1} = W_s$, O'_s is closed under reachability in $G - W'_{s'-1}$. It remains to show that $R'_{s'-1} \subseteq O'_{s'-1}$. First, we have $W_s \cap (\tilde{R}_s \setminus \{b\}) = \emptyset$. Assume that, to the contrary, there is some $v \in W_s \cap (\tilde{R}_s \setminus \{b\})$. Then $v \notin U$ (as a cop and a robber cannot be on the same vertex) and according to (Cops) we have $U = \bigcup_{i=1}^{s} U_i$. So $v \notin U_s$ and hence, according to the definition of U_s, $v \in O^{s-1}$, which contradicts $v \in \tilde{R}_s$. So, indeed, $W_s \cap (\tilde{R}_s \setminus \{b\}) = \emptyset$. Hence, by the definition of $O'_{s'-1}$ in Case 1, we have $R'_{s'-1} = \tilde{R}_s \setminus \{b\} \subseteq \tilde{O}_s = O'_{s'-1}$ and thus, (Omit) follows.

Notice that, by (Ext) for ζ, $O_i \subseteq \mathrm{Reach}_{G-W_i^{-1}}(b_i)$ for $i = 1, \dots, s-1$ and as $O'_i = O_i$ and $\rho'_i = \rho_i$ for $i = 1, \dots, s-1$, the invariant holds for all $i = 1, \dots, s-1 \geq s'-2$. In particular, in Case 2, there is nothing to show and we consider Case 1. First, notice that $(W'_{s'-1})^{-1} = W_s^{-1} = \overline{W}_s$ and $b'_{s'-1} = b_s = \overline{b}_s$, so according to Lemma 34, we have $R'_{s'-1} \subseteq \tilde{R}_s \subseteq \mathrm{Reach}_{G-W_s^{-1}}(b_s)$. Moreover, by the definition, $O'_{s'-1} = \tilde{O}_s = \mathrm{Reach}_{G-W_s}(\tilde{R}_s \setminus \{b\})$. So if $v \in \tilde{O}_s$, then v is reachable from some $\widehat{b} \in \tilde{R}_s \setminus \{b\}$ in $G - W_s$ and as $\tilde{R}_s \subseteq \mathrm{Reach}_{G-W_s^{-1}}(b_s)$, $\widehat{b}$ is reachable from b_s in $G - W_s^{-1}$. Thus, v is reachable from b_s in $G - (W_s^{-1} \cap W_s)$ and, as $\rho_s = \widehat{\rho}(W_s^{-1}, W_s, b_s)$ is consistent with f by (Cons) for ζ and f is monotone, we have $v \in \mathrm{Reach}_{G-W_s^{-1}}(b_s)$.

Finally, for (Cons), Case 2 is trivial. For Case 1, as ρ_s is consistent with f by (Cons), it suffices to show that $b \in \mathrm{Reach}_{G-W_s^{-1}}(b_s)$. However, we have shown in Lemma 34 that $\tilde{R}_s \subseteq \mathrm{Reach}_{G-\overline{W}_s}(\overline{b}_s)$ and as in Case 1 we have $\overline{b}_s = b_s$ and $\overline{W}_s = W_s^{-1}$, this follows from $b \in \tilde{R}_s$. □

It remains to show that, first, $\otimes_r f$ uses at most $r \cdot k$ cops and, second, playing according to $\otimes_r f$ the cops capture all robbers.

Using at most $k \cdot r$ cops. By (Cops), the number of cops is bounded by $|\bigcup_{i=1}^{s} U_i|$. By definition of U_i, we have $|\bigcup_{i=1}^{s} U_i| \leq |\bigcup_{i=1}^{s} W_i|$. Due to (Cons), all W_i have size at most k. Thus we have to show that there are at most r distinct sets W_i.

Lemma 37. *For any memory state ζ consistent with $\otimes_r f$ we have $|\zeta| \leq r + 1$ and, if $|\zeta| = r + 1$, then $W_s = W_{s-1}$.*

Proof. In the following, we denote by ζ the memory state before and by ζ' the memory state after the cop move (and before the robber move) and by ζ'' the memory state after the robber move.

If $|\zeta| \leq r$, then, by inspecting all cases, we can see that $|\zeta''| \leq r$, or, in Case 1 of the robber move, $|\zeta''| \leq r + 1$ and $W_s = W_{s-1}$. Consider the case $|\zeta| = r + 1$ and $W_s = W_{s-1}$. As $|R| \leq r$, it follows from (Robs) that $R_i = \emptyset$, for some $i \in \{1, \ldots, s-1\}$ or $b_s \notin R$.

If $b_s \notin R$, then, after the cop move, we either have $|\zeta'| = r$ (if $R_{s-1} = \emptyset$), or $|\zeta'| = r+1$ and $W'_{s'} = W'_s = W_{s-1} = W'_{s-1} = W'_{s'-1}$ (if $R_{s-1} \neq \emptyset$). Moreover, in that case the memory state after the robber moves (which is empty) is the same as after the cop moves.

Now assume that $b_s \in R$ and let $i \in \{1, \ldots, s-1\}$ be such that $R_i = \emptyset$. Then in the cop move, we are in Case II.1. If we are in Case II.1 (a) or in Case II.1 (c), then we have $|\zeta'| = r$ after the cop move, so after the robber move, $|\zeta''| \leq r + 1$ holds. If we are in Case II.1 (b), then after the cop move, we have $s' = s$, $\rho'_{s-1} = \rho_{s-1}$ and $\rho'_s = \rho_s$. Hence, $W'_{s-1} = W'_s$ and, as ρ_s ends with a cop position (because after the robber move, ρ_s always ends in a cop position and Case II. (b) does not change ρ_s), ζ'' is constructed according to Case 3 of the memory update after the robber move. Hence, $|\zeta''| = |\zeta'| = r + 1$ and $W''_{s''} = W'_{s'} = W'_{s-1} = W''_{s''-1}$. □

Capturing all robbers. To prove that $\otimes_r f$ is winning, we, first, prove that an additional invariant holds.

(Progress) For $i \in \{2, \ldots, s-1\}$, $R_i \cap O^{i-1} = \emptyset$ and $b_s \notin O^{s-1}$.

The invariant expresses that an O_i can only be a reason not to place *any* cops when playing against robbers from smaller histories. Indeed, any winning strategy finally places a cop into the robber component, so after some omitted placements, some cop is really placed. This is true, in particular, for $i = s$, which guarantees that the set of vertices available to the robbers shrinks.

The reason why we have to maintain that property also for the shorter play prefixes is that when the robber leaves b_s one of shorter ρ_i becomes the longest one.

Basically, (Progress) follows from the assumption that the robbers use an isolating strategy. However, as the sets O_i are defined with respect to reachability in $G-W_i$ and not in $G-U$, we have to transfer that topological incomparability from $G-U$ to $G-W_i$.

Lemma 38. *(Progress) is preserved by both cop and robber moves.*

Proof. First, consider the situation after the cop move. In Case I, we have $R'_j = R_j$ and $O_j = O'_j$ for $j = 1, \ldots, s-2$ and hence, $R'_j \cap (O^{j-1})' = \emptyset$ by (Progress) for ζ. Moreover, if $R_{s-1} = \emptyset$, then $s' = s-1$, so $s'-1 = s-2$ and it remains to show that $b_{s'} \notin O^{s'-1}$. However, as $b_{s'} = b_s$ and $O^{s'-1} = O^{s-2}$, this follows immediately from (Progress) for ζ.

If $R_{s-1} \neq \emptyset$, then $s' = s$ and $R'_{s-1} \subseteq R_s$ so $R'_{s-1} \cap (O^{s-2})' = \emptyset$ and $b'_s = b \notin (O^{s-2})'$ follows again immediately from $(O^{s-2})' = O^{s-2}$ and (Progress) for ζ. So it remains to show that $b \notin O'_{s-1} = \tilde{O}_{s-1} = \mathrm{Reach}_{G-W_{s-1}}(R_{s-1} \setminus \{b\})$. As the robbers play according to an isolating strategy, $b \notin \mathrm{Reach}_{G-U}(R_{s-1} \setminus \{b\})$. Assume that $b \in \mathrm{Reach}_{G-W_{s-1}}(R_{s-1} \setminus \{b\})$. Then due to Lemma 27, $b \in \mathrm{Reach}_{G-W^{s-1}}(R_{s-1} \setminus \{b\}) \subseteq \mathrm{Reach}_{G-U^{s-1}}(R_{s-1} \setminus \{b\})$. Moreover, by Corollary 29, $\mathrm{Reach}_{G-U^{s-1}}(R_{s-1} \setminus \{b\}) = \mathrm{Reach}_{G-U}(R_{s-1} \setminus \{b\})$, which is a contradiction. In Case II, (Progress) for ζ' follows easily from (Progress) for ζ using the definition of the memory update.

Now consider the situation after the robber move. In Case 2, (Progress) holds by the construction of the sets $\tilde{R}_i = R'_i$ for $i = 1, \ldots, s$. Moreover, in Case 1, $R'_i \cap (O^{i-1})' = \emptyset$ holds for $i = 1, \ldots, s'-1$ by the construction of the sets R'_i as well and $b \notin (O^{s'-2})' = O^{s-1}$ holds by the construction of $\tilde{R}_s$.

It remains to show that $b \notin O'_{s'-1} = \tilde{O}_s = \mathrm{Reach}_{G-W_s}(\tilde{R}_s \setminus \{b\})$. As the robber plays according to an isolating strategy, we have $b \notin \mathrm{Reach}_{G-U}(\tilde{R}_s \setminus \{b\})$. Assume that $b \in \mathrm{Reach}_{G-W_s}(\tilde{R}_s \setminus \{b\})$. Then as $W_s = W'_{s'-1}$ and $\tilde{R}_s \setminus \{b\} = R'_{s'-1}$, Lemma 27 for the memory state ζ' after the robber move yields $b \in \mathrm{Reach}_{G-(W^{s'-1})'}(R'_{s'-1}) = \mathrm{Reach}_{G-W^s}(\tilde{R}_s \setminus \{b\}) \subseteq \mathrm{Reach}_{G-U^s}(\tilde{R}_s \setminus \{b\})$. Moreover, $U^s = U$, so $b \in \mathrm{Reach}_{G-U}(\tilde{R}_s \setminus \{b\})$, which is a contradiction. □

We conclude the proof of Theorem 25 with the following lemma, whose proof uses (Progress) to show that all robbers are finally captured in any play consistent with $\otimes_r f$.

Lemma 39. *$\otimes_r f$ is winning.*

Proof. First observe that every cop that is placed on the graph according to the longest history restricts the set of vertices reachable for the robber on b_s because f is active.

Assume that there is a play

$$\pi = \bot \cdot (U_0, R_0) \cdot (U_0, U_1, R_0) \cdot (U_1, R_1) \dots$$

consistent with $\otimes_r f$ and a position (U_j, R_j) of π after which the set of vertices reachable for the robber in the longest history ρ_s remains constant. (Due to the monotonicity of $\otimes_r f$, it never becomes smaller.) As the robbers play according to a prudent strategy, R_i also remains constant. Let $b(i)$ be vertex b_s stored in the memory after move number i. Then $\mathrm{Reach}_{G-U_l}(b(j)) = \mathrm{Reach}_{G-U_{l+1}}(b(l+1))$, for $l \geq j$. As the robber strategy is prudent, it follows that $b(j) = b(l)$, i.e., the robber does not change his vertex after move number j.

It suffices to prove that Case II.2 appears infinitely often. If it does, we place new cops on $f(W_s, b_s) \setminus O^{s-1}$ again and again. As $\mathrm{Reach}_{G-U_l}(b(l)) = \mathrm{Reach}_{G-U_{l+1}}(b(l+1))$, for all $l \geq j$, it follows that $\otimes_r f$ never places cops into $\mathrm{Reach}_{G-U(l)}(b(l))$ and thus $U_l = U_{l+1}$, by the definition of U_l. Since $\otimes_r f$ places cops according to f, it prescribes to place cops only in O^{s-1}. Therefore, b_s is never occupied by any cop according to f due to the invariant (Progress). Hence, f is not winning, which contradicts our assumption.

Assume that after some position, Case II.2 does not appear. Then Case I or Case II.1 appear infinitely often. In both cases, s does not increase.

In Case I, if $R_{s-1} = \emptyset$, then the number s of histories in ζ decreases. If $R_{s-1} \neq \emptyset$, then $|R_{s-1}|$ decreases.

In Case II.1, histories that are shorter than ρ_s are extended or deleted (which decreases s), if they reach the next play prefix. The length of the longest history in ζ is an upper bound for the growth of their lengths. As the robbers do not change their placement, $|R_{s-1}|$ will never increase again. Together, either s or $|R_{s-1}|$ decrease, so Cases I and II.1 can appear only finitely many times. It follows that we have Case II.2 infinitely many times, but that contradicts our assumption. □

This finishes the proof of Theorem 25.

7.3. Robbers hierarchy, imperfect information and directed path-width

In this section we extend the results from [RT09] about the dependence of cop number on the number of robbers to our setting. For the same graph G, increasing the number of robbers induces a hierarchy of cop numbers that are needed to capture the robbers. It is clear that less robbers do not demand more cops. Furthermore, one robber corresponds to the DAG-width game and $|G|$ robbers to the directed path-width game, hence we have the following scheme:

$$\mathrm{dagw}(G) = \mathrm{dagw}_1(G) \leq \mathrm{dagw}_2(G) \leq \ldots \leq \mathrm{dagw}_{|G|}(G) = \mathrm{dpw}(G)$$

where n is the number of vertices of G. In general, i.e., on some graphs, this hierarchy does not collapse, because path-width is not bounded in tree-width.

We give explicit lower bounds for the stages. In a sense, DAG-width can be approximated by a refinement of directed path-width, but there are infinitely many stages of approximation. This result is analogous to similar results in [RT09] and in [FFN09].

Theorem 40. *For every $k > 0$, there is a class G^k of graphs such that, for all $G \in \mathsf{G}^k$, we have* $\mathrm{dagw}_1(G) = 2 \cdot k$ *and, for all $r > 0$, there exists $G^k_r \in \mathsf{G}^k$ with*

(1) $\mathrm{dpw}(G^k_r) = k \cdot (r+1)$*, and*

(2) for all $i \in \{1, \dots, r\}$, $\mathrm{dagw}_i(G^k_r) \geq \frac{i \cdot (k-1)}{2}$.

Proof. Class G^k consists of graphs G^k_r, for each $r > 0$. Every G^k_r is the lexicographic product $T_r \oplus K_k$ of the full undirected tree T_r with branching degree $\lceil \frac{r}{2} \rceil + 2$ and of height $r+1$, with the k-clique K_k. In other words, G^k_r is T_r where every vertex v is replaced by a k-clique $K(v)$ and if (v, w) is an edge of T_r, then all pairs (v', w') with $v' \in K(v)$ and $w' \in K(w)$ are edges of G^k_r.

It is clear that $\mathrm{dagw}_1(G^k_r)$ is $2 \cdot k$: the cops play as on T_r occupying $K(v)$ instead of single tree vertex v and leaving $K(v)$ if v is left.[4] We have to show that $\mathrm{dpw}(G^k_r) = k(r+1)$ and that $\mathrm{dagw}_i(G^k_r) \geq \frac{i \cdot (k-1)}{2}$.

We start with directed path-width. A similar proof can be found, for example, in [Bod98]. Note that the branching degree of all T_r is at least 3. Let us see that the statement follows from $\mathrm{dpw}(T_r) = r + 1$. First, as for DAG-width above, we have $\mathrm{dpw}(G^k_r) \leq k \cdot (r+1)$. The statement of the other direction follows from the fact that it makes no sense for the cops to occupy only a part of a k-clique. We formulate that statement as a small lemma.

Lemma 41. *Every winning strategy f for $k(r+1)$ cops can be turned into a winning strategy f' for $k(r+1)$ cops that always prescribes to occupy whole k-cliques.*

Proof. Strategy f' is as follows. If f prescribes to occupy only a part of a clique, then f' does not place any cops in the clique, otherwise f and f' are the same. Assume that f' is not winning. Then there is a cop move $(U, R) \to (U, U', R)$ such that a path P from R to $U \setminus U'$ exists in $G - (U \cap U')$. Consider a path P' that is as P, but for vertices v occupied by cops, it contains a vertex $w \in K(v)$ that is cop-free. It is clear that such a vertex w always exists. Then P' is an evidence that f is not monotone, which is a contradiction to our assumption. □

We prove $\mathrm{dpw}(T_r) = r + 1$ by induction on r. The case $r = 1$ is trivial. If $r + 1$ cops win on T_r, then $r + 2$ cops win on T_{r+1} by placing a cop on the root and applying the strategy for $r + 1$ cops from the induction hypothesis for every subtree.

[4] The idea to use the lexicographic product and of the proof is due to [Hun07].

The other direction (that $\mathrm{dpw}(T_r) \geq r+1$) is also proven by induction on r. The induction base is clear. Assume that $\mathrm{dpw}(T_r) \geq r+1$. In T_{r+1}, let the direct successors of the root be $v_1, \ldots, v_m$ (recall that $m \geq 3$). All subtrees T^i rooted at v_i, for $i \in \{1, \ldots, m\}$, must be decontaminated (i.e., the robber must be expelled from T^i) and $r+1$ cops are needed for that. Assume without loss of generality that T^1 is the first and T^2 is the second decontaminated subtree. In some position all $r+1$ cops are in T^2. However, there is a path from T^m via the root of the whole tree to T^1. Thus T^1 becomes recontaminated, which contradicts the monotonicity of directed path-width [Hun06].

It remains to show that $k \cdot i$ robbers win against $\frac{i\cdot(k-1)}{2}$ cops on G_r^k. We show only that i robbers win against $\lfloor \frac{i}{2} \rfloor$ cops on T_r, the result with factor k follows as above. As in the proof of Theorem 23, we can assume that the cops play top-down because the tree has a high branching degree.

The winning strategy for robbers is to tie every cop. A cop is *tied* if there is a cop-free path from a robber to the cop. When a cop is placed on a vertex v, the robbers occupy two subtrees of v. As there are at least two robbers for each cop, this is always possible. A cop is untied only if two other cops in both subtrees chosen by the robbers become tied, so at every tree level at least one more cop becomes tied. At the latest when a cop reaches level $\lfloor \frac{i\cdot(k-1)}{2} \rfloor$, all cops are tied. □

8. Discussion and future work

We analyzed the connection between imperfect information in parity games and structural complexity of game graphs. If the amount of imperfect information is unbounded, restricting structural complexity of game graphs does not lead to lower computational complexity of the strategy problem. For the case of bounded imperfect information we showed that some graph complexity measures have unbounded values when performing the powerset construction, and some are still bounded. As side effects of our proofs we showed that, first, monotonicity of DAG-width is not necessary for an efficient solution of the strategy problem for perfect information parity games, and, second, that introducing new robbers demands only linearly more cops to capture them. We believe that those results are also of independent relevance.

To complete the picture, it would be interesting to prove that Kelly-width and directed tree-width also remain bounded after performing the powerset construction. For directed tree-width it is not known whether perfect information parity games can be solved in PTIME, so a bound would not immediately imply an efficient solution of parity games with imperfect information. It would be also worth attention to analyze which other variants of the graph searching game with multiple robbers make sense and what are the differences between them, our version and the games from [RT09].

8.1. Acknowledgments

We thank Łukasz Kaiser for many inspiring discussions, Tsvetelina Yonova-Karbe and Sebastian Siebertz for the proof reading.

References

[Bar06] J. Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.

[BCD$^+$08] D. Berwanger, K. Chatterjee, L. Doyen, T. A. Henzinger, and S. Raje. Strategy Construction for Parity Games with Imperfect Information. In *CONCUR 2008 - Concurrency Theory, 19th International Conference*, pages 325–339, 2008.

[BDH$^+$12] Dietmar Berwanger, Anuj Dawar, Paul Hunter, Stephan Kreutzer, and Jan Obdržálek. The DAG-width of directed graphs. *J. Comb. Theory, Ser. B*, 102(4):900–923, 2012.

[BDHK06a] D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. DAG-Width and Parity Games. In *STACS 06*, volume 3884 of *LNCS*, pages 524–536, 2006.

[BDHK06b] D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. DAG-Width and Parity Games. In *STACS 06*, volume 3884 of *LNCS*, pages 524–536. Springer-Verlag, 2006.

[BG05a] D. Berwanger and E. Grädel. Entanglement – a measure for the complexity of directed graphs with applications to logic and games. In *LPAR 04*, volume 3452 of *LNCS*, pages 209–223. Springer, 2005.

[BG05b] D. Berwanger and E. Grädel. Entanglement – a measure for the complexity of directed graphs with applications to logic and games. In *LPAR 2004*, volume 3452 of *LNCS*, pages 209–223. Springer, 2005.

[BGKR12] Dietmar Berwanger, Erich Grädel, Łukasz Kaiser, and Roman Rabinovich. Entanglement and the Complexity of Directed Graphs. *Theoretical Computer Science*, 463(0):2–25, 2012. Special Issue on Theory and Applications of Graph Searching Problems.

[Bod98] H.L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[FFN09] Fedor V. Fomin, Pierre Fraigniaud, and Nicolas Nisse. Nondeterministic graph searching: From pathwidth to treewidth. *Algorithmica*, 53(3):358–373, 2009.

[FS11] John Fearnley and Sven Schewe. Time and space results for parity games with bounded treewidth. *CoRR*, abs/1112.0221, 2011.

[FS12] John Fearnley and Sven Schewe. Time and parallelizability results for parity games with bounded treewidth. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP (2)*, volume 7392 of *Lecture Notes in Computer Science*, pages 189–200. Springer, 2012.

[GKR09] Erich Grädel, Łukasz Kaiser, and Roman Rabinovich. Directed Graphs of Entanglement Two. In *Proceedings of the 17th International Symposium on Fundamentals of Computation Theory*, volume 5699 of *LNCS*, pages 169–181. Springer, 2009.

[HK07] P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. In *SODA*, pages 637–644, 2007.

[Hun06] Paul Hunter. Losing the +1. or directed path-width games are monotone. Online available at http://www.comlab.ox.ac.uk/people/paul.hunter/papers/losing.pdf, 2006.

[Hun07] P. Hunter. *Complexity and Infinite Games on Finite Graphs.* PhD thesis, Computer Laboratory, University of Cambridge, 2007.

[JRST01] T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *J. Comb. Theory, Ser. B*, 82(1):138–154, 2001.

[Jur00] Marcin Jurdziński. *Games for Verification: Algorithmic Issues.* PhD thesis, University of Aarhus, 2000.

[KO08a] S. Kreutzer and S. Ordyniak. Digraph Decompositions and Monotonicity in Digraph Searching. In *GTCCS, WG 08, Durham*, pages 336–347, 2008.

[KO08b] S. Kreutzer and S. Ordyniak. Digraph Decompositions and Monotonicity in Digraph Searching. In *Graph-Theoretic Concepts in Computer Science: 34th International Workshop, WG 2008, Durham, UK, June 30 - July 2, 2008. Revised Papers*, pages 336–347. Springer-Verlag, 2008.

[Kre09] Stephan Kreutzer. On the parameterised intractability of monadic second-order logic. In Erich Grädel and Reinhard Kahle, editors, *CSL*, volume 5771 of *Lecture Notes in Computer Science*, pages 348–363. Springer, 2009.

[Obd03] Jan Obdrzálek. Fast mu-calculus model checking when tree-width is bounded. In Warren A. Hunt Jr. and Fabio Somenzi, editors, *CAV*, volume 2725 of *Lecture Notes in Computer Science*, pages 80–92. Springer, 2003.

[Obd06] J. Obdržálek. Dag-width: connectivity measure for directed graphs. In *SODA*, pages 814–821, 2006.

[Puc] B. Puchala. Infinite Two Player Games with Partial Information: Logic and Algorithms. Diploma Thesis, RWTH Aachen, 2008.

[Rei84] J. H. Reif. The Complexity of Two-player Games of Incomplete Information. *Journal of Computer and System Sciences*, 29:274–301, 1984.

[RT09] David Richerby and Dimitrios M. Thilikos. Graph Searching in a Crime Wave. *SIAM J. Discrete Math.*, 23(1):349–368, 2009.

[ST93] P. D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *J. Comb. Theory Ser. B*, 58(1):22–33, 1993.

[Tho95] W. Thomas. On the Synthesis of Strategies in Infinite Games. In *Proceedings of the 12th Annual Symposium on Theoretical Aspects of Computer Science, STACS '95*, pages 1–13, 1995.

[WW86] K. Wagner and G. Wechsung. *Computational Complexity.* D. Reidel Publishing Company, 1986.