Entanglement and the Complexity of Directed Graphs

Dietmar Berwanger[a], Erich Grädel[b], Łukasz Kaiser[c], Roman Rabinovich[b]

[a] *LSV, CNRS & Ecole Normale Supérieure de Cachan, France*
[b] *Mathematische Grundlagen der Informatik, RWTH Aachen University, Germany*
[c] *LIAFA, CNRS & Université Paris Diderot, France*

Abstract

Entanglement is a parameter for the complexity of finite directed graphs that measures to which extent the cycles of the graph are intertwined. It is defined by way of a game similar in spirit to the cops and robber games used to describe tree width, directed tree width, and hypertree width. Nevertheless, on many classes of graphs, there are significant differences between entanglement and the various incarnations of tree width.

Entanglement is intimately related with the computational and descriptive complexity of the modal μ-calculus. The number of fixed-point variables needed to describe a finite graph up to bisimulation is captured by its entanglement. This plays a crucial role in the proof that the variable hierarchy of the μ-calculus is strict.

We study complexity issues for entanglement and compare it to other structural parameters of directed graphs. One of our main results is that parity games of bounded entanglement can be solved in polynomial time. Specifically, we establish that the complexity of solving a parity game can be parametrised in terms of the minimal entanglement of subgames induced by a winning strategy.

Furthermore, we discuss the case of graphs of entanglement two. While graphs of entanglement zero and one are very simple, graphs of entanglement two allow arbitrary nesting of cycles, and they form a sufficiently rich class for modelling relevant classes of structured systems. We provide characterisations of this class, and propose decomposition notions similar to the ones for tree width, DAG-width, and Kelly-width.

Keywords: Structural graph theory, Graph searching games, Parity games, Digraph algorithms

Email addresses: `dwb@lsv.ens-cachan.fr` (Dietmar Berwanger), `graedel@logic.rwth-aachen.de` (Erich Grädel), `kaiser@liafa.univ-paris-diderot.fr` (Łukasz Kaiser), `rabinovich@logic.rwth-aachen.de` (Roman Rabinovich)

1. Introduction

In recent years, several parameters have been proposed for measuring the structural complexity of directed graphs in a similar way to which *tree width* [27] measures the complexity of undirected graphs. The intuition behind such parameters is that acyclic graphs are simple, and that the complexity of a graph is reflected by the degree in which its cycles are intertwined, or entangled. Two main approaches to making the idea precise rely on *graph decompositions* similar to tree decompositions, and *graph searching games*, also called cops and robber games: here, a number of cops seek to capture a fugitive that can move along the edges of the graph, and the number of cops needed to capture the fugitive determines the complexity of the graph.

Tree width of a directed graph $\mathcal{G}$ can be defined as the tree width of the undirected graph that underlies $\mathcal{G}$. However, discarding the direction of edges may lead to the loss of relevant information. For instance, an acyclic orientation of a complete graph has maximal (undirected) tree width, in spite of the fact that the directed graph is acyclic, and thus simple.

Directed tree width, the first generalisation of tree width to directed graphs, is defined by means of an arboreal decomposition similar to the tree decomposition for the undirected case [17]. A variant of the graph searching game for the undirected case, where the robber is restricted to stay in her strongly connected component, characterises directed tree width only up to a constant additive factor.

DAG-width, introduced in [3, 24, 4], is defined by DAG-decompositions. A DAG-decomposition of width k for a graph $\mathcal{G}$ is described by a directed acyclic graph (DAG) $\mathcal{D}$ and a map that associates, with every node of the DAG, a set of at most k nodes of $\mathcal{G}$, covering the entire graph $\mathcal{G}$ in such a way that, for every $d \in \mathcal{D}$, the edges of $\mathcal{G}$ leaving a node strictly below d are guarded by nodes in d. DAG-width can also be characterised by a variant of a graph searching game (the directed cops and visible robber game), but with the somewhat unsatisfactory restriction that the cops are only allowed to use robber-monotone strategies, i.e., a move of the cops must never enlarge the portion of the graph in which the robber can move. It has been proved [22] that this restriction is necessary: there exist families of graphs where the difference between the DAG-width and the number of cops that can capture the robber with a non-monotone strategy is unbounded.

Kelly-width, introduced in [16], is a similar measure that can be characterised either by a refined notion of decomposition, called Kelly-decomposition, or by a graph searching game in which the robber is invisible to the cops and inert, in the sense that she can move only when a cop is about to land on her current position. Again, the correspondence between decompositions and games only holds with the restriction to monotone strategies [22].

Entanglement, introduced in [5], has been motivated by applications concerning the modal μ-calculus and parity games. It is defined by a game where the moves of both the cops and the robber are more restricted than in other graph searching games: In each move the cops either stay where they are or place one of them on the current position of the robber; here, strategies need not be monotone.

Entanglement is, in a sense, more delicate than (directed) tree width, DAG-width, or Kelly-width [15]. There exist graphs of DAG-width, Kelly-width and directed tree width three and arbitrarily large entanglement. For a survey of further complexity measures for directed graphs, such as pathwidth [26], cycle rank [11], D-width [28], we refer to [14, 25].

The strengths of entanglement are the close connection with modal logics and bisimulation invariant properties, and the natural game-theoretic characterisation. Thus, entanglement has been instrumental in the proof that the variable hierarchy of the modal μ-calculus is strict [7]. Furthermore, parity games can be solved efficiently on game graphs of bounded entanglement — analogous results hold for bounded DAG-width and bounded Kelly-width. Finally, entanglement does not increase when we take bisimulation quotients, and it has been proved that, as a consequence, winning regions of parity games are definable in least fixed point logic on graphs of bounded entanglement [10].

The main weakness of entanglement as a measure (at the current state of the art) is that it does not come with a natural notion of decomposition, such as the ones for tree width, DAG-width, or Kelly-width. Decompositions are crucial for algorithmic applications, since they allow to break structures into smaller parts that can be processed independently. In Section 3 we present a structural characterisation of entanglement in terms of the minimal feedback of finite unravellings of the graph as a tree with back-edges. However, while this produces a game-free definition of entanglement, it does not yield a notion of a decomposition.

For the particular case of graphs with entanglement two, studied in [2, 13], we provide structural characterisations via decompositions similar to the ones for tree width, DAG-width, and Kelly-width. While graphs of entanglement zero and one are very simple, graphs of entanglement two allow complex nesting of cycles, and they are rich enough to model interesting classes of structured systems. We show that all graphs of entanglement two have both DAG-width and Kelly-width three.

2. Entanglement via graph searching games

Let $\mathcal{G} = (V, E)$ be a finite directed graph. The entanglement of $\mathcal{G}$, denoted by $\mathrm{ent}(\mathcal{G})$, is defined through a game $\mathrm{EG}_k(\mathcal{G})$ played in rounds by a robber against k cops according to the following rules. Initially the robber selects an arbitrary position v_0 of $\mathcal{G}$ and the cops are outside the graph. In every round, the cops may either stay where they are, or place one of them on the current position v of the robber. The robber may try to escape by moving to a successor

$w \in vE$ that is not occupied by a cop. If no such position exists, the robber is captured and the cops win. Note that, in every round, the robber is informed about the move of the cops before she decides on her own move, and that she has to leave her current position no matter whether the cops move or not. The entanglement of $\mathcal{G}$ is the minimal number $k \in \mathbb{N}$ such that k cops have a strategy to win the game $\mathrm{EG}_k(\mathcal{G})$.

For a formal definition of strategies in the entanglement game $\mathrm{EG}_k(\mathcal{G})$ on a graph $\mathcal{G} = (V, E)$, we describe a play by a sequence $\pi \in S^{\leq\omega}$, where $S = V \times \mathcal{P}_{\leq k}(V)$. Hereby $\mathcal{P}_{\leq k}(V)$ is the set of subsets of V of size at most k, and $(v, P) \in S$ denotes a position where the robber is on v and the cops occupy the nodes in P. As the turns of the players alternate, we do not represent the turn information explicitly.

Then, a strategy of the robber in $\mathrm{EG}_k(\mathcal{G})$ is a partial function $\rho : S \cup \{\varepsilon\} \to V$ with the property that $\rho(v, P) \in vE \setminus P$. Here, $\rho(\varepsilon)$ describes the choice of the initial node by the robber.

Similarly, a strategy of the cops is a partial function $\sigma : S \to V \cup \{\Box, \bot\}$ describing which cop, if any, moves to the current node occupied by the robber:

- if $\sigma(v, P) = \bot$ then the cops remain idle, and the next position is (v, P) (but now it is the robber's turn);
- if $\sigma(v, P) = \Box$ then it must be the case that $|P| < k$ and the next position is $(v, P \cup \{v\})$ (a cop from outside moves to node v);
- otherwise $\sigma(v, P) = u \in P$ (the cop from node u goes to v), and the next position is $(v, (P \setminus \{u\}) \cup \{v\})$.

Note that we distinguish between cops only according to their position in the graph; in particular, we do not distinguish cops that stay outside of it.

A strategy ρ of the robber together with a strategy σ of the cops define a unique play $\pi = (v_0, P_0)(v_1, P_1)(v_2, P_2)\ldots$ that *follows* ρ and σ. The starting position is $(v_0, P_0) = (\varepsilon, \emptyset)$ meaning that the cops and the robber are outside of the graph. After the initial move of the robber the position is $(v_1, P_1) = (\rho(\varepsilon), \emptyset)$. For every $n > 0$ the node v_{2n+1} occupied by the robber after her $(n+1)$-st move is determined by $\rho(v_{2n}, P_{2n})$, and the set P_{2n} occupied by the cops after their n-th move is determined by $\sigma(v_{2n-1}, P_{2n-1})$. Finally, we have $P_{2n+1} = P_{2n}$ and $v_{2n} = v_{2n-1}$. A play ends with a win for the cops, if, for some n, there is no position $w \in v_{2n}E \setminus P_{2n}$. Infinite plays are winning for the robber. A robber (or cop) strategy is *winning*, if the robber (cop) wins every play that follows it, regardless of the strategy of the opponent.

The entanglement game is a reachability game: the cops try to reach a state of the game at which the robber is captured. It is well known that such games are determined via memoryless strategies, i.e., one of the two players has a winning strategy that depends only on the current position, and not on the history of the play (see, e.g., [31]).

Lemma 2.1. *For every graph $\mathcal{G}$ and every k, the game $\mathrm{EG}_k(\mathcal{G})$ is determined with memoryless winning strategies, that is, either the cops or the robber have a memoryless winning strategy.*

Entanglement is an interesting measure on *directed* graphs. To deal with undirected graphs, we view undirected edges $\{u, v\}$ as pairs (u, v) and (v, u) of directed edges. In the following a graph is always understood to be directed and finite.

To get a feeling for this measure we quote a few observations concerning the entanglement of certain familiar graphs. The proofs are simple exercises.

Proposition 2.2. *Let $\mathcal{G}$ be a directed graph.*

(1) $\text{ent}(\mathcal{G}) = 0$ *if, and only if, $\mathcal{G}$ is acyclic.*
(2) *If $\mathcal{G}$ is the graph of a unary function, then* $\text{ent}(\mathcal{G}) = 1$.
(3) *If $\mathcal{G}$ is an undirected tree, then* $\text{ent}(\mathcal{G}) \leq 2$.
(4) *If $\mathcal{G}$ is the complete directed graph with n nodes, then* $\text{ent}(\mathcal{G}) = n - 1$.

Let C_n denote the directed cycle with n nodes. Given two graphs $\mathcal{G} = (V, E)$ and $\mathcal{G}' = (V', E')$ their asynchronous product is the graph $\mathcal{G} \times \mathcal{G}' = (V \times V', F)$ where

$$F = \{(uu', vv') : [(u, v) \in E \wedge u' = v'] \vee [u = v \wedge (u', v') \in E']\}.$$

Note, that $T_{mn} := C_m \times C_n$ is the $(m \times n)$-torus or, to put it differently, the graph obtained from the directed $(m + 1) \times (n + 1)$-grid by identifying the left and right border and the upper and lower border.

Proposition 2.3.

(1) *For every n,* $\text{ent}(T_{nn}) = n$.
(2) *For every $m \neq n$,* $\text{ent}(T_{mn}) = \min(m, n) + 1$.

Proof. On T_{nn}, a team of n cops can capture the robber by placing themselves on a diagonal, thus blocking every row and every column of the torus. If there are less than n cops, the robber can guarantee the *free-lane* property to hold again and again: there is a cop free column in the torus and a cop free path to this column from her node. At the beginning of a play this is clear. In general, assume that the property holds and let the robber move on a cop free column until a cop announces to land on her node. In that moment, there is another cop free column, say number c, as we have n columns and at most $n - 1$ cops, but one cop is on his way to the robber's node and thus outside of the graph. For the same reason, there is a cop free row r. The robber runs to the crossing of row r and the column she is on and then along row r to column c. When she arrives at column c, the free-lane property holds again. It follows that the robber wins the game.

On T_{mn} with $m < n$, m cops are needed to block every row, and an additional cop forces the robber to leave any row after at most n moves, so that she finally must run into a cop. The same proof as above shows that the robber escapes if there are less than $m + 1$ cops. $\square$

The following proposition characterises the graphs with entanglement one. It is more difficult to describe graphs with entanglement two, and we defer the characterisation to Section 8. The problem of characterising graphs with entanglement 3 and above is open.

Proposition 2.4. *The entanglement of a directed graph is one, if, and only if, the graph is not acyclic, and in every strongly connected component, there is a node whose removal makes the component acyclic.*

Proof. On any graph with this property, one cop captures the robber by placing himself on the critical node in the current strongly connected component when the robber passes there. The robber will have to return to this node or leave the current component. Eventually she will be captured in a terminal component.

Conversely if there is a strongly connected component without such a critical node, then the robber may always proceed from her current position towards an unguarded cycle and thus escape forever. □

As acyclicity in directed graphs is NLOGSPACE-complete [19], we immediately obtain the following corollary.

Corollary 2.5. *For $k = 0$ and $k = 1$, the problem whether a given graph has entanglement k is* NLOGSPACE*-complete.*

3. Entanglement via trees with back-edges and partial unravellings

Let $\mathcal{T} = (V, E)$ be a directed tree. We write $\preceq_E$ for the associated partial order on $\mathcal{T}$, that is, the reflexive, transitive closure of E. A directed graph $\mathcal{T} = (V, F)$ is a *tree with back-edges* if there is a partition $F = E \cup B$ of the edges into tree-edges and back-edges such that (V, E) is indeed a directed tree with edges oriented away from the root, and whenever $(u, v) \in B$, then $v \preceq_E u$.

The following observation shows that, up to the choice of the root, the decomposition into tree-edges and back-edges is unique.

Lemma 3.1. *Let $\mathcal{T} = (V, F)$ be a tree with back-edges and $v \in V$. Then there exists at most one decomposition $F = E \cup B$ into tree-edges and back-edges such that (V, E) is a tree with root v.*

Let $\mathcal{T} = (V, E, B)$ be a tree with back-edges. The *feedback* of a node v of $\mathcal{T}$ is the number of ancestors of v that are reachable by a back-edge from a descendant of v. The feedback of $\mathcal{T}$, denoted $\mathrm{fb}(\mathcal{T})$ is the maximal feedback of nodes on $\mathcal{G}$. More formally,

$$\mathrm{fb}(\mathcal{T}) = \max_{v \in V} |\{u \in V : \exists w (u \preceq_E v \preceq_E w \wedge (w, u) \in B)\}|.$$

We call a back edge (w, u), and likewise its target u, *active* at a node v in $\mathcal{T}$, if $u \preceq_E v \preceq_E w$.

Note that the feedback of $\mathcal{T}$ may depend on how the edges are decomposed into tree-edges and back-edges, i.e., on the choice of the root. Consider, for instance, the following graph C_3^+ (the cycle C_3 with an additional self-loop on one of its nodes). Clearly, for every choice of the root, C_3^+ is a tree with two back-edges. If the node with the self-loop is taken as the root, then the feedback is 1, otherwise it is 2.

Lemma 3.2. *Let $\mathcal{T} = (V, E, B)$ be a tree with back-edges of feedback k. Then there exists a partial labelling $i : V \mapsto \{0, \dots, k-1\}$ assigning to every target u of a back edge an index $i(u)$ in such a way that no two nodes u, u' that are active at the same node v have the same index.*

Proof. The values of this labelling are set while traversing the tree in breadth-first order. Notice that every node u with an incoming back-edge is active at itself. As $\mathcal{T}$ has feedback k, there can be at most $k-1$ other nodes active at u. All of these are ancestors of u, hence their index is already defined. There is at least one index which we can assign to u so that no conflict with the other currently active nodes arises. □

Lemma 3.3. *The entanglement of a tree with back-edges is at most its feedback:* $\mathrm{ent}(\mathcal{T}) \leq \mathrm{fb}(\mathcal{T})$.

Proof. Suppose that $\mathrm{fb}(\mathcal{T}) = k$. By Lemma 3.2 there is a labelling i of the targets of the back-edges in $\mathcal{T}$ by numbers $0, \dots, k-1$ assigning different values to any two nodes u, u' that are active at the same node v. This labelling induces the following strategy for the k cops: at every node v reached by the robber, send cop number $i(v)$ to that position or, if the value is undefined, do nothing. By induction over the stages of the play, we can now show that this strategy maintains the following invariant: in every round of the play on $\mathcal{T}$, when the robber reaches a node v, then all active nodes $u \neq v$ are occupied and, if the current node is itself active, a cop is on the way. To see this, let us trace the evolution of the set $Z \subseteq T$ of nodes occupied by a cop. In the beginning of the play, Z is empty. A node v can be included into Z if it is visited by the robber and active with regard to itself. At this point, our strategy appoints cop $i(v)$ to move to v. Since, by construction of the labelling, the designated cop $i(v)$ must come from a currently inactive position and, hence, all currently active positions except v remain in Z. But if every node which becomes active is added to Z and no active node is ever given up, the robber can never move along a back edge, so that after a finite number of steps she reaches a leaf of the tree and loses. But this means that we have a winning strategy for k cops, hence $\mathrm{ent}(\mathcal{T}) \leq k$. □

It is well-known that every graph $\mathcal{G}$ can be unravelled from any node v to a tree $\mathcal{T}_{G,v}$ whose nodes are the paths in $\mathcal{G}$ from v. Clearly $\mathcal{T}_{G,v}$ is infinite unless $\mathcal{G}$ is finite and no cycle in $\mathcal{G}$ is reachable from v. A *finite unravelling* of a (finite) graph $\mathcal{G}$ is defined in a similar way, but rather than an infinite tree, it produces a finite tree with back-edges. To construct a finite unravelling we proceed as in the usual unravelling process with the following modification: whenever we have a path $v_0 v_1 \dots v_n$ in $\mathcal{G}$ with corresponding node $\overline{v} = v_0 v_1 \dots v_n$ in the unravelling, and a successor w of v_n that coincides with v_i (for any $i \leq n$), then we may, instead of creating the new node $\overline{v}w$ (with a tree-edge from $\overline{v}$ to $\overline{v}w$) put a back-edge from $\overline{v}$ to its ancestor $v_0 \dots v_i$. Clearly this process is nondeterministic. Accordingly, any finite graph can be unravelled, in several different ways, to a finite tree with back-edges.

Definition 3.4. Let $\mathcal{G} = (V, E)$ be a graph and let v_0 be a node in $\mathcal{G}$. A tree $\mathcal{T} = (T, E_T)$ with back-edges is a *finite unravelling* of $\mathcal{G}$, if it is finite and there is a labelling $h : T \to V$ with the following property:

> for all paths $v_0 \dots v_n$ in $\mathcal{G}$, there is a unique path $w_0 \dots w_n$ in $\mathcal{T}$ such that, for all $i \in \{0, \dots, n\}$, we have $h(w_i) = v_i$.

Note that different finite unravellings of a graph may have different feedback and different entanglement.

Clearly the entanglement of a graph is bounded by the entanglement of its finite unravellings. Indeed a winning strategy for k cops on a finite unravelling of $\mathcal{G}$ immediately translates to a winning strategy on $\mathcal{G}$.

Proposition 3.5. *The entanglement of a graph is the minimal feedback (and the minimal entanglement) of its finite unravellings:*

$$\begin{aligned}\operatorname{ent}(\mathcal{G}) &= \min\{\operatorname{fb}(\mathcal{T}) : \mathcal{T} \textit{ is a finite unravelling of } \mathcal{G}\}\\ &= \min\{\operatorname{ent}(\mathcal{T}) : \mathcal{T} \textit{ is a finite unravelling of } \mathcal{G}\}.\end{aligned}$$

Proof. For any finite unravelling $\mathcal{T}$ of a graph $\mathcal{G}$, we have $\operatorname{ent}(\mathcal{G}) \leq \operatorname{ent}(\mathcal{T}) \leq \operatorname{fb}(\mathcal{T})$. It remains to show that for any graph $\mathcal{G}$ there exists some finite unravelling $\mathcal{T}$ with $\operatorname{fb}(\mathcal{T}) \leq \operatorname{ent}(\mathcal{G})$.

To prove this, we view winning strategies for the cops as descriptions of finite unravellings. A strategy for k cops tells us, for any finite path πv of the robber whether a cop should be posted at the current node v, and if so, which one. Such a strategy can be represented by a partial function g mapping finite paths in $\mathcal{G}$ to $\{0, \dots, k-1\}$. On the other hand, during the process of unravelling a graph to a (finite) tree with back edges, we need to decide, for every successor v of the current node, whether to create a new copy of v or to return to a previously visited one, if any is available. To put this notion on a formal ground, we define an *unravelling function* for a rooted graph $\mathcal{G}, v_0$ as a partial function ρ between finite paths from v_0 through $\mathcal{G}$, mapping any path $v_0, \dots, v_{r-1}, v_r$ in its domain to a strict prefix $v_0, v_1, \cdots, v_{j-1}$ such that $v_{j-1} = v_r$. Such a function gives rise to an unravelling of $\mathcal{G}$ in the following way: we start at the root and follow finite paths through $\mathcal{G}$. Whenever the current path π can be prolonged by a position v and the value of ρ at πv is undefined, a fresh copy of v corresponding to πv is created as a successor of π. In particular, this always happens if v was not yet visited. Otherwise, if $\rho(\pi\, v)$ is defined, then the current path π is bent back to its prefix $\rho(\pi)$ which also corresponds to a copy of v. Formally, the unravelling of $\mathcal{G}$ *driven* by ρ is the tree with back edges $\mathcal{T}$ defined as follows:

- the domain of $\mathcal{T}$ is the smallest set T which contains v_0 and for each path $\pi \in T$, it also contains all prolongations πv in $\mathcal{G}$ at which ρ is undefined;
- the tree-edge partition is

$$E^{\mathcal{T}} := \{\, (v_0, \dots, v_{r-1},\ v_0, \dots, v_{r-1}, v_r) \in T \times T \mid (v_{r-1}, v_r) \in E^{\mathcal{G}} \,\};$$

- for all paths $\pi := v_0, \ldots, v_{r-1} \in T$ where $\rho(\pi v)$ is defined, the back-relation $B^{\mathcal{T}}$ contains the pair $(\pi,\ \rho(\pi v))$ if $(v_{r-1}, v) \in E^{\mathcal{G}}$.

We are now ready to prove that every winning strategy g for the k cops on $\mathcal{G}, v_0$ corresponds to an unravelling function ρ for $\mathcal{G}, v_0$ that controls a finite unravelling with feedback k.

Note that the strategy g gives rise to a k-tuple $(g_0, \ldots, g_{k-1})$ of functions mapping every initial segment π of a possible play according to g to a k-tuple $(\, g_0(\pi), \ldots, g_{k-1}(\pi)\,)$ where each $g_i(\pi)$ is a prefix of π recording the state of the play (i.e. the current path of the robber) at the last move of cop i.

Now, for every path π and possible prolongation by v, we check whether, after playing π, there is any cop posted at v. If this is the case, i.e, when, for some i, the end node of $g_i(\pi)$ is v, we set $\rho(\pi\, v) := \pi_i$. Otherwise we leave the value of ρ undefined at π, v. It is not hard to check that, if g is a winning strategy for the cops, the associated unravelling is finite and has feedback k. $\square$

4. Computational complexity

Many algorithmic issues in graph theory are related to the problem of cycle detection, typically, to determine whether a given graph contains a cycle satisfying certain properties. When alternation comes into play, that is, when we consider paths formed interactively, the questions become particularly interesting but often rather complex, too. In this framework, we will study the entanglement of a graph as a measure of how much memory is needed to determine on the fly whether a path formed interactively enters a cycle.

As a basis for later development, let us first describe a procedure for deciding whether k cops are sufficient to capture the robber on a given graph. The following algorithm represents a straightforward implementation of the game as an alternating algorithm, where the role of the robber is played by the universal player while the cops are controlled by the existential player.

procedure Entanglement$(\mathcal{G}, v_0, k)$
input graph $\mathcal{G} = (V, E)$, initial position v_0, candidate $k \leq |V|$
// accept iff $\mathrm{ent}(\mathcal{G}, v_0) \leq k$
$v := v_0, (d_i)_{i \in [k]} := \perp$; // current position of robber and cops
do
  existentially guess $i \in [k] \cup \{\text{pass}\}$ // appoint cop i or pass
  if $i \neq$ pass **then** $d_i := v$ // guard current node
  if $vE \setminus \{d\} = \emptyset$ **then accept**
  else universally choose $v \in vE \setminus \{d\}$;
repeat

Since this algorithm requires space only to store the current positions of the robber and the k cops, it runs in alternating space $O((k+1)\log|V|)$ which corresponds to deterministic polynomial time [9].

Lemma 4.1. *The problem of deciding, for a fixed parameter k, whether a given graph $\mathcal{G}$ with n nodes has $\mathrm{ent}(\mathcal{G}) \leq k$ can be solved in time $O(n^{k+1})$.*

Notice that if we regard the parameter k as part of the input, the algorithm yields an Exptime upper bound for complexity of deciding the entanglement of a graph. We know no non-trivial lower bounds.

5. Parity games

As usual for measures of graph complexity, we are not only interested in computing the entanglement of a graph, but also in identifying complex problems that become tractable when restricted to graphs of small entanglement. In this section, we discuss one prominent example: parity games. These games are subject to an intriguing open problem related to the μ-calculus, the computational complexity of its evaluation problem: Given a formula ψ and a finite transition structure $\mathcal{K}, v$, decide whether ψ holds in $\mathcal{K}, v$. The natural evaluation games for L_μ are parity games [30].

Parity games are path-forming games played between two players on labelled graphs $\mathcal{G} = (V, V_0, E, \Omega)$ equipped with a *priority* labelling $\Omega : V \to \mathbb{N}$. All plays start from a given initial node v_0. At every node $v \in V_0$, the first player, called Player 0, can move to a successor $w \in vE$; at positions $v \in V_1 := V \setminus V_0$, his opponent Player 1 moves. Once a player gets stuck, he loses. If the play goes on infinitely the winner is determined by looking at the sequence $\Omega(v_0), \Omega(v_1), \ldots$ of priorities seen during the play. In case the least priority appearing infinitely often in this sequence is even, Player 0 wins the play, otherwise Player 1 wins.

A *memoryless strategy* for Player i in a parity game $\mathcal{G}$ is a function σ that indicates a successor $\sigma(v) \in vE$ for every position $v \in V_i$. A strategy for a player is *winning*, if he wins every play that is consistent with the strategy. The Memoryless Determinacy Theorem of Emerson and Jutla states that parity games are always determined with memoryless strategies.

Theorem 5.1 (Memoryless Determinacy, [12])**.** *In any parity game, one of the players has a memoryless winning strategy.*

Any memoryless strategy σ induces a subgraph $\mathcal{G}_\sigma$ of the game graph $\mathcal{G}$, obtained by removing every edge $(v, w) \in E$ where $v \in V_i$ and $w \neq \sigma(v)$. Then, σ is a winning strategy for Player i if, and only if, he wins every play on $\mathcal{G}_\sigma$. As these subgames are small objects and it can be checked efficiently whether a player wins every play on a given graph, the winner of a finite parity game can be determined in NP$\cap$co-NP. In general, the best known deterministic algorithms to decide the winner of a parity game have running times that are polynomial with respect to the size of the game graph, but exponential with respect to the number of different priorities occurring in the game [20]. However, for game graphs of bounded tree width, DAG-width or Kelly-width, it is known that the problem can be solved in polynomial time with respect to the the size of the graph, independently of the number of priorities [23, 4, 16].

In the remainder of this section we will show that the entanglement of a parity game graph is a pivotal parameter for its computational complexity. To maintain a close relationship between games and algorithms, we base our analysis on alternating machines (for a comprehensive introduction, see e.g. [1]).

Similar to the robber and cop game, the dynamics of a parity game consists in forming a path through a graph. However, while in the former game the cops can influence the forming process only indirectly, by obstructing ways of return, in a parity game both players determine directly how the path is prolonged. Besides this dynamic aspect, also the objectives of players are quite different at a first sight. While the cops aim at making the play return to a guarded position, each player of a parity game tries to achieve that the least priority seen infinitely often on the path is of a certain parity.

The key insight which brings the two games to a common ground is the Memoryless Determinacy Theorem for parity games: whichever player has a winning strategy in a given game $\mathcal{G} = (V, V_0, E, \Omega)$, also has a memoryless one. This means, that either player may commit, for each reachable position $v \in V$ which he controls, to precisely one successor $\sigma(v) \in vE$ and henceforth follow this commitment in every play of $\mathcal{G}$ without risking his chance to win. It follows that, whenever a play returns to a previously visited position v, the winner can be established by looking at the least priority seen since the first occurrence of v. Therefore we can view parity games on finite game graphs as path forming games of finite duration where the objective is to reach a cycle with minimal priority of a certain parity.

With this insight, we obtain a method for determining the winner of a parity game by simulating the moves of players while maintaining the history of visited positions in order to detect whether a cycle was reached and to keep track of the occurring priorities. To store the full history, an implementation of this method requires space $O(|V| \log |V|)$ in the worst case; since the procedure uses alternation to simulate the single game moves, this situates us in $\textsc{Aspace}(O(|V| \log |V|))$, or $\textsc{Dtime}(|V|^{O(|V|)})$.

What makes this approach highly impractical is the extensive representation of the play history. In fact, the power of alternation is limited to the formation of the path, while the history is monitored in a deterministic way. We can improve this significantly, by interleaving robber and cop games with parity games in such a way that the formation of cycles is monitored using the power of alternation.

Intuitively, we may think of a parity game as an affair involving three agents, Player 0 and 1, and a referee who seeks to establish which of the two has a winning strategy. In the approach presented above, the referee memorises the entire history of the game. But as we could see, the occurrence of a cycle in a path-forming game on a graph $\mathcal{G}$ can already be detected by storing at most $\mathrm{ent}(\mathcal{G})$ many positions. Hence, if we could provide the referee with the power of sufficiently many cops, this would reduce the space requirement. The crux of the matter is how to fit such a three-player setting into the two-player model of alternating computation.

Our proposal is to let one of the players act as a referee who challenges the opponent in the parity game, and, at the same time, controls the cops in an auxiliary cops-and-robber game played on the side, where the path formed in the parity game is regarded as if it would be formed by the robber alone.

Formally, this leads to a new game. For a game graph $\mathcal{G} = (V, V_0, E, \Omega)$, a player $i \in \{0,1\}$, and a number k, the *supercop* game $\mathcal{G}[i,k]$ is played between two players: the *Supercop*, which controls k cops and the positions of V_i, and the *Challenger* which controls the positions in V_{1-i}. Starting from an initial position v_0, in any move, the Supercop may place one of the k cops on the current position v, or leave them in place. If the current position v belongs to V_{1-i}, Challenger has to move to some position $w \in vE$, otherwise the Supercop moves. If a player gets stuck, he loses immediately. The play ends, if it reaches a position w occupied by a cop, and the Supercop wins if the least priority seen since the cop was placed at w has the same parity as i. All infinite plays are winning for the Challenger.

The following lemma states that parity games can be reduced to Supercop games with an appropriate number of cops.

Lemma 5.2.

(1) *If Player i has a winning strategy for the parity game $\mathcal{G}$, then the Supercop wins the supercop game $\mathcal{G}[i,k]$ with $k = \mathrm{ent}(\mathcal{G})$.*

(2) *If, for some $k \in \mathbb{N}$, the Supercop wins the game $\mathcal{G}[i,k]$, then Player i has a winning strategy for the parity game $\mathcal{G}$.*

Proof. Let σ be a memoryless winning strategy of Player i for the game $\mathcal{G}$, and let $\mathcal{G}_\sigma$ be the subgame of $\mathcal{G}$ induced by this strategy. Then, the least priority seen on any cycle of $\mathcal{G}_\sigma$ is favourable to Player i. This remains true for any cycle formed in $\mathcal{G}[i,k]$ where Player i acting as a Supercop follows the same strategy σ. On the other hand, obviously $\mathrm{ent}(\mathcal{G}_\sigma) \leq \mathrm{ent}(\mathcal{G}) = k$, which means that the Supercop also has a strategy to place the k cops so that every path through $\mathcal{G}_\sigma$ will finally meet a guarded position v and hence form a cycle, witnessing that he wins.

For the converse, assume that the Supercop wins the game $\mathcal{G}[i,k]$ whereas Player $1-i$ had a memoryless winning strategy τ in the parity game $\mathcal{G}$. Then Player $1-i$ could follow this strategy when acting as a Challenger in $\mathcal{G}[i,k]$, so that the play remains in $\mathcal{G}_\tau[i,k]$. However, in this game no cycle is favourable to Player i and, hence, the Supercop i cannot win, in contradiction to our assumption. □

Note that computing the winner of a supercop game $\mathcal{G}[i,k]$ requires alternating space $(2k+1)\log|V|$. Indeed, one just plays the game recording the current position of the robber and the current position of each cop, along with the minimal priority that has been seen since he was last posted.

procedure Supercop($\mathcal{G}, v_0, j, k$)
input parity game $\mathcal{G} = (V, V_0, E, \Omega)$, initial position $v_0 \in V$, player j, k cops
// accept iff Supercop has a winning strategy in $\mathcal{G}[j,k]$ with k cops
$v := v_0$ // current position
$(d_i)_{i\in[k]} := \bot$ // positions guarded by cops
$(h_i)_{i\in[k]} := \bot$ // most significant priorities
repeat
  if $j = 0$ **then**

```
      existentially guess  i ∈ [k] ∪ {pass}   // appoint cop i or pass
    else
      universally choose  i ∈ [k] ∪ {pass}   // other player's cop
    if i ≠ pass then
      d_i := v; h_i := Ω(v)                   // guard current node
    v := Move(G, v)                           // simulate a game step
    forall i ∈ [k] do                         // update history
      h_i := min(h_i, Ω(v))
    repeat
  until ( v = d_i for some i )                // cycle detected
  if (j = 0 and h_i is even) or (j = 1 and h_i is odd) then accept
  else reject
```

We are now ready to prove that parity games of bounded entanglement can be solved in polynomial time. In fact we establish a more specific result, taking into account the minimal entanglement of subgames induced by a winning strategy.

Theorem 5.3. *The winner of a parity game $\mathcal{G} = (V, V_0, E, \Omega)$ can be determined in* $\textsc{Aspace}(\mathcal{O}(k \log |V|))$*, where k is the minimum entanglement of a subgame $\mathcal{G}_\sigma$ induced by a memoryless winning strategy σ in $\mathcal{G}$.*

Proof. We first describe the procedure informally, in the form of a game. Given a parity game $\mathcal{G} = (V, V_0, E, \Omega)$ and an initial position v_0, each player i selects a number k_i and claims that he has a winning strategy from v_0 such that $\mathrm{ent}(\mathcal{G}_\sigma) \leq k_i$. The smaller of the two numbers k_0, k_1 is then chosen to verify that Supercop wins the game $\mathcal{G}[i, k_i]$. If this is the case the procedure accepts the claim of Player i, otherwise Player $(1 - i)$ is declared the winner.

Here is a more formal description of the procedure:

```
procedure SolveParity(G, v)
input parity game G = (V, V_0, E, Ω), initial position v ∈ V
// accept iff Player 0 wins the game
existentially guess  k_0 ≤ |V|
universally choose  k_1 ≤ |V|
if k_0 ≤ k_1 then
  if Supercop(G, v, 0, k_0) then accept
  else reject
else
  if Supercop(G, v, 1, k_1) then reject
  else accept
fi
```

We claim that Player 0 has a winning strategy in a parity game $\mathcal{G}, v$ if, and only if, the alternating procedure ParitySolve$(\mathcal{G}, v)$ accepts.

To see this, assume that Player 0 has a memoryless winning strategy σ from v. Then, the guess $k_0 := \mathrm{ent}(\mathcal{G}_\sigma)$ leads to acceptance. Indeed, for $k_1 \geq k_0$, Player 0 wins the supercop game $\mathcal{G}[0, k_0]$ by using the strategy σ as a parity player together with the cop strategy for $\mathcal{G}_\sigma$. On the other hand, for $k_1 < k_0$, the procedure accepts as well, since Player 1 cannot win the supercop game $\mathcal{G}[1, k_1]$ without having a winning strategy for the parity game.

The converse follows by symmetric arguments exchanging the roles of the two players. $\square$

Corollary 5.4. *Parity games of bounded entanglement can be solved in polynomial time.*

6. Descriptive complexity

The modal μ-calculus L_μ, introduced by Kozen [21], is a highly expressive formalism which extends basic modal logic with monadic variables and binds them to extremal fixed points of definable operators.

Syntax. For a set $\textsc{act}$ of actions, a set $\textsc{prop}$ of atomic propositions, and a set $\textsc{var}$ of monadic variables, the formulae of L_μ are defined by the grammar

$$\varphi ::= \text{false} \mid \text{true} \mid p \mid \neg p \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

where $p \in \textsc{prop}$, $a \in \textsc{act}$, and $X \in \textsc{var}$. An L_μ-formula in which no universal modality $[a]\varphi$ occurs is called *existential*.

The number of variables occurring in a formula provides a relevant measure of its conceptual complexity. For any $k \in \mathbb{N}$, the *k-variable fragment* $\mathrm{L}_\mu[k]$ of the μ-calculus is the set of formulae $\psi \in \mathrm{L}_\mu$ that contain at most k distinct variables.

Semantics. Formulae of L_μ are interpreted on transition systems, or Kripke structures. Formally, a transition system $\mathcal{K} = \big(V, (E_a)_{a \in \textsc{act}}, (V_p)_{p \in \textsc{prop}}\big)$ is a coloured graph with edges labelled by action and nodes labelled by atomic propositions. Given a sentence ψ and a structure $\mathcal{K}$ with state v, we write $\mathcal{K}, v \models \psi$ to denote that ψ holds in $\mathcal{K}$ at state v. The set of states $v \in K$ such that $\mathcal{K}, v \models \psi$ is denoted by $[\![\psi]\!]^{\mathcal{K}}$.

Here, we only define $[\![\psi]\!]^{\mathcal{K}}$ for fixed-point formulae ψ. Towards this, note that a formula $\psi(X)$ with a monadic variable X defines on every transition structure $\mathcal{K}$ (providing interpretations for all free variables other than X occurring in ψ) an operator $\psi^{\mathcal{K}} : \mathcal{P}(K) \to \mathcal{P}(K)$ assigning to every set $X \subseteq K$ the set $\psi^{\mathcal{K}}(X) := [\![\psi]\!]^{\mathcal{K},X} = \{v \in K : (\mathcal{K}, X), v \models \psi\}$. As X occurs only positively in ψ, the operator $\psi^{\mathcal{K}}$ is *monotone* for every $\mathcal{K}$, and therefore, by a well-known theorem due to Knaster and Tarski, has a least fixed point $\mathrm{lfp}(\psi^{\mathcal{K}})$ and a greatest fixed point $\mathrm{gfp}(\psi^{\mathcal{K}})$. Now we put

$$[\![\mu X.\psi]\!]^{\mathcal{K}} := \mathrm{lfp}(\psi^{\mathcal{K}}) \text{ and } [\![\nu X.\psi]\!]^{\mathcal{K}} := \mathrm{gfp}(\psi^{\mathcal{K}}).$$

As a modal logic, the μ-calculus distinguishes between transitions structures only up to behavioural equivalence, captured by the notion of bisimulation.

Definition 6.1. A *bisimulation* between two transition structures $\mathcal{K} = \big(V, (E_a)_{a \in A}, (P_i)_{i \in I}\big)$ and $\mathcal{K}' = \big(V', (E'_a)_{a \in A}, (P'_i)_{i \in I}\big)$ is a relation $Z \subseteq V \times V'$ such that for all $(v, v') \in Z$,

- for all $i \in I$, $v \in P_i$ if and only if $v' \in P'$,
- for all $a \in A$ and $w \in V$, if $(v,w) \in E_a$ then there is some $w' \in V'$ with $(v',w') \in E'_a$ and $(w,w') \in Z$, and
- for all $a \in A$ and $w' \in V'$, if $(v',w') \in E'_a$ then there is some $w \in V$ with $(v,w) \in E_a$ and $(w,w') \in Z$.

Two transition structures $\mathcal{K}, u$ and $\mathcal{K}', u'$ are *bisimilar*, if there is a bisimulation Z between them, with $(u,u') \in Z$.

An important model-theoretic feature of modal logics is the *tree model property* meaning that every satisfiable formula is satisfiable in a tree. This is a straightforward consequence of bisimulation invariance, since $\mathcal{K}, u$ is bisimilar to its *infinite unravelling*, i.e., a tree where the nodes correspond to the finite paths in $\mathcal{K}, u$. Every such path π inherits the atomic propositions of its last node v; for every node w reachable from v in $\mathcal{K}$ via an a transition, π is connected to its prolongation by w via an a-transition. Notice that in terms of our notion of unravelling defined in the proof of Proposition 3.5, the infinite unravelling of a system is just the unravelling driven by a function defined nowhere.

The entanglement of a transition system $\mathcal{K} = \big(V, (E_a)_{a \in \text{ACT}}, (V_p)_{p \in \text{PROP}}\big)$ is the entanglement of the underlying graph (V,E) where $E = \bigcup_{a \in \text{ACT}} E_a$. We now show that every transition structure of entanglement k can be described, up to bisimulation, in the μ-calculus using only k fixed-point variables.

Proposition 6.2. *Let $\mathcal{K}$ be a finite transition system with* $\text{ent}(\mathcal{K}) = k$. *For any node v of $\mathcal{K}$, there is a formula $\psi_v \in L_\mu[k]$ such that*

$$\mathcal{K}', v' \models \psi_v \;\Leftrightarrow\; \mathcal{K}', v' \sim \mathcal{K}, v.$$

Proof. According to Proposition 3.5, the system $\mathcal{K}$ can be unravelled from any node v_0 to a finite tree $\mathcal{T}$ with back-edges, with root v_0 and feedback k. Clearly $\mathcal{T}, v_0 \sim \mathcal{K}, v_0$. Hence, it is sufficient to prove the proposition for $\mathcal{T}, v_0$. For every action $a \in \text{ACT}$, the transitions in $\mathcal{T}$ are partitioned into tree-edges and back edges $E_a \cup B_a$.

Let $i : \mathcal{T} \mapsto \{0, \dots, k-1\}$ be the partial labelling of $\mathcal{T}$ defined in Lemma 3.2. At hand with this labelling, we construct a sequence of formulae $(\psi_v)_{v \in T}$ over fixed-point variables $X_0, \dots, X_{k-1}$ while traversing the nodes of $\mathcal{T}$ in reverse breadth-first order.

The atomic type of any node v is described by the formula

$$\alpha_v := \bigwedge_{\substack{p \in \text{PROP} \\ v \in V_p}} p \;\wedge\; \bigwedge_{\substack{p \in \text{PROP} \\ v \notin V_p}} \neg p.$$

To describe the relationship of v with its successors, let

$$\varphi_v := \alpha_v \wedge \bigwedge_{a\in\mathrm{ACT}} \Bigg(\bigwedge_{(v,w)\in E_a} \langle a\rangle\psi_w \wedge \bigwedge_{(v,w)\in B_a} \langle a\rangle X_{i(w)} \\ \wedge\ [a]\Big(\bigvee_{(v,w)\in E_a} \psi_w \vee \bigvee_{(v,w)\in B_a} X_{i(w)}\Big)\Bigg).$$

If v has an incoming back-edge, we set $\psi_v := \nu X_{i(v)}\,.\,\varphi_v$, if this is not the case we set $\psi_v := \varphi_v$. Note that since we proceed from the leaves of $\mathcal{T}$ to the root, this process is well-defined, and that in ψ_v the variables $X_{i(u)}$ occur free, for any node $u \neq v$ that is active at v. In particular the formula ψ_{v_0}, corresponding to the root of $\mathcal{T}$, is closed.

It remains to prove that $\mathcal{K}', v' \models \psi_{v_0} \;\Leftrightarrow\; \mathcal{K}', v' \sim \mathcal{T}, v_0$. We first show that $\mathcal{T}, v_0 \models \psi_{v_0}$, and hence $\mathcal{K}', v' \models \psi_{v_0}$ for any $\mathcal{K}', v' \sim \mathcal{T}, v_0$. To see this we prove that Verifier has a winning strategy for the associated model checking game.

Note that, since ψ_{v_0} has only greatest fixed points, any infinite play of the model checking game is winning for Verifier. It thus suffices to show that from any position of form (v, φ_v), Verifier has a strategy to make sure that the play proceeds to a next position of form (w, φ_w), unless Falsifier moves to position (v, α_v) and then loses in the next move. But by the construction of the formula, it is obvious that Verifier can play so that any position at which she has to move has one of the following three types:

(1) $(v, \langle a\rangle\psi_w)$, where $(v, w) \in E_a$. In this case, Verifier moves to position (w, ψ_w).
(2) $(v, \langle a\rangle X_{i(w)})$, where $(v, w) \in B_a$. In this case Verifier moves to $(w, X_{i(w)})$.
(3) $(w, \bigvee_{(v,w)\in E_a} \psi_w \vee \bigvee_{(v,w)\in B_a} X_{i(w)})$ where $w \in vE_a \cup vB_a$. In this case, Verifier selects the appropriate disjunct and moves to either (w, ψ_w) or $(w, X_{i(w)})$.

In all cases the play will proceed to (w, φ_w). Hence, Falsifier can force a play to be finite only by moving to a position (v, α_v). Otherwise the resulting play is infinite and thus also winning for Verifier.

For the converse, suppose that $\mathcal{K}', v' \not\sim \mathcal{T}, v_0$. Since $\mathcal{T}$ is finite, the non-bisimilarity it witnessed by a finite stage. That is, there is a basic modal formula separating $\mathcal{K}', v'$ from $\mathcal{T}, v_0$, and Falsifier can force the model checking game for ψ_{v_0} on $\mathcal{K}', v'$ in finitely many moves to a position of form (w', α_w) such that w and w' have distinct atomic types. This proves that $\mathcal{K}', v' \not\models \psi_{v_0}$. □

As the entanglement of a transition system regards only the underlying graph, one can easily find examples of high entanglement that can be described with very few variables. For instance, in a transition structure over a strongly connected finite graph with no atomic propositions and only a single action a, all states are bisimilar, and can be described by $\nu X.(\langle a\rangle X \wedge [a]X)$, regardless of the entanglement of the underlying graph. Nevertheless, the following theorem establishes a strong relationship between the notion of entanglement and the descriptive complexity of L_μ.

Theorem 6.3 ([7]). *Every strongly connected graph of entanglement k can be labelled in such a way that no μ-calculus formula with less than k variables can describe the resulting transition structure, up to bisimulation.*

This theorem, which generalises a result of [6], provides the witnesses for the expressive strictness of the μ-calculus variable hierarchy proved in [7].

7. Entanglement and other complexity measures

The definition of entanglement is reminiscent of the characterisation of other complexity measures defined via cops and robber games [29, 18, 4, 16]. However, we will see that entanglement is a quite different, and for some purposes more accurate, measure.

The precise relationship between entanglement and other measures is not well understood yet. The following sufficient condition for the existence of a winning strategy for k cops will be helpful to see that entanglement of the undirected $(n \times n)$-grid is at most $3n$. It is well known that the tree width of the $(n \times n)$-grid is precisely n.

Lemma 7.1. *Let $\mathcal{G} = (V, E)$ be a directed graph. If for some $k \in \mathbb{N}$, there exists a partial labelling $i : V \to [k]$ under which every strongly connected subgraph $\mathcal{C} \subseteq \mathcal{G}$ contains a uniquely labelled node v, that is, $i(v) \neq i(w)$ for all $w \in \mathcal{C}$ with $w \neq v$, then we have $\mathrm{ent}(\mathcal{G}) \leq k$.*

Proof. We may interpret the proposed labelling as a memoryless strategy for the cops, indicating at every position $v \in \mathrm{dom}(i)$ occurring in a play of $\mathcal{G}$, that cop $i(v)$ shall be posted there, or that no cop shall move, if $i(v)$ is undefined.

Towards a contradiction, suppose that, although the cops move according to strategy i, the robber has a strategy to escape. That is, he can form an infinite path without meeting any cop. Let $\mathcal{C}$ be the set of positions visited infinitely often by this path. Clearly, $\mathcal{C}$ induces in $\mathcal{G}$ a strongly connected subgraph. Let $v \in C$ be a node whose label $i(v)$ is unique in $\mathcal{C}$. According to the cop strategy, $i(v)$ never moved since the play stabilised in $\mathcal{C}$. But this contradicts our assumption that the robber has visited every position $v \in \mathcal{C}$ infinitely often. □

Proposition 7.2. *For every n, the undirected $(n \times n)$-grid has entanglement at most $3n$.*

Proof. Consider the labelling $i : [n] \times [n] \to [3n]$ obtained by first assigning the values $0, \ldots, n-1$ to the horizontal median of the grid, i.e. $i(\lceil \frac{n}{2} \rceil, j) := j$ for all $j \in [n]$. For the two $\lfloor \frac{n}{2} \rfloor \times n$ grids obtained when removing the positions already labelled, we proceed independently and assign the values $n, \ldots, n + \lfloor \frac{n}{2} \rfloor$ to their respective medians, and so on, in step k applying the procedure to the still unlabelled domain consisting of 2^k many $\lfloor \frac{n}{2^k} \rfloor \times \lfloor \frac{n}{2^k} \rfloor$ disconnected grids. Summing up, we get at most

$$(n + \frac{n}{2}) + (\frac{n}{2} + \frac{n}{4}) + \cdots = \sum_{k=1}^{n} \frac{3n}{2^k} < 3n$$

many different labels. It is easy to verify that the labelling obtained in this way satisfies the condition of Lemma 7.1, implying that the entanglement of the initial grid is at most $3n$. $\square$

Most often, it turns out that the entanglement of a graph is greater than the other measures. For the next result we need game theoretic characterisations of DAG-width, Kelly-width and directed tree width. We will use the notation $\mathrm{Reach}\,\mathcal{G}(v)$ to denote the set of nodes from which a node v is reachable in a graph $\mathcal{G}$, and write $\mathcal{G} \setminus U$ for the graph obtained by removing a set of nodes U from $\mathcal{G}$; when U consists of a single element u we simply write $\mathcal{G} \setminus u$.

The (directed) k cops and (*visible*) robber game used to characterise DAG width [4] is played on a graph $\mathcal{G} = (V, E)$ similar to the entanglement game. A position where the cops are in turn to move is described by (r, C) where $C \subseteq V^{\leq k}$ is the set of nodes occupied by cops and $r \in V$ is the node occupied by the robber. From (r, C), the cops can move to positions (r, C, C') of the robber, where $C' \subseteq V^{\leq k}$ is the set of nodes the cops announce to occupy. From a position (r, C, C'), the robber can move to the positions (r', C') where $r' \in \mathrm{Reach}_{\mathcal{G}\setminus(C\cap C')}(r)$, i.e., the robber can run along cop-free paths. A play is called *monotone*, if the robber cannot occupy a node that has been previously been unreachable to her. The cops win a monotone play if it reaches a position (r, C, C') with $\mathrm{Reach}_{\mathcal{G}\setminus(C\cap C')}(r) = \emptyset$. The robber wins every play that is non-monotone or infinite. In [4] it is shown that the DAG-width of $\mathcal{G}$ is the minimal number k such that the cops win the k cops and robber game on $\mathcal{G}$.

The rules of the k cops and *invisible inert* robber game for defining Kelly width [15] are similar, except that the cops are not informed about the current position of the robber and (he is invisible), in turn, the robber moves only if the cops threaten to occupy his current position (he is inert). It is convenient to describe game positions by giving the set of possible R of possible locations of the robber, together with the current or announced locations of the cops. Thus, we have a one-player game: the position following (R, C, C') is (R', C') with

$$R' = (R \cup \mathrm{Reach}_{\mathcal{G}\setminus(C\cap C')}(R \cap C')) \setminus C'.$$

The term $\mathrm{Reach}_{\dots}(R \cap C')$ describes the inertness of the robber, whereas the term $R \cup \dots$ means that the robber may stay on the a previous node if no cop is about to occupy it. Kelly-width is the minimal k such that k cops have a strategy to capture the robber in any monotone play.

Finally, the game for directed tree width [18] differs from the k cops and robber game in that, firstly, monotonicity is not required, and secondly, the robber cannot leave her strongly connected component: a move from (r, C, C') to (r', C') is only possible if additionally holds that $r \in \mathrm{Reach}_{\mathcal{G}\setminus(C\cap C')}(r')$.

Proposition 7.3. *There is a family of finite undirected graphs with unbounded entanglement and tree width two, and DAG-width, Kelly-width and directed tree width three.*

Proof. Let $\mathcal{G}_k$ be the graph consisting of two full binary undirected trees whose corresponding nodes are connected to each other: $\mathcal{G}_k = (V_k \uplus \overline{V}_k, E_k \uplus \overline{E}_k \uplus E'_k)$

where $V_k = \{0,1\}^{\leq k-1}$ is the set of words over $\{0,1\}$ of length at most $k-1$, $\overline{V}_k = \{\overline{0},\overline{1}\}^{\leq k-1}$, $(V_k, E_k) = \mathcal{T}$ is an undirected full binary tree, $(\overline{V}_k, \overline{E}_k) = \overline{\mathcal{T}}$ is its copy and $E'_k = \big\{\{u,\overline{u}\} \mid u \in \{0,1\}^{\leq k-1}\big\}$.

It is easy to see that $\mathcal{G}_k$ has tree width 2, and DAG-width, Kelly-width and directed tree width three. For entanglement, we show that, for every even k, the robber starting from node ε or from node $\overline{\varepsilon}$ can ensure to

- escape $k/2-2$ cops and
- after the $(k/2-1)$-th cop enters $\mathcal{G}_k$,
  - if started on ε, reach $\overline{\varepsilon}$, and
  - if started in $\overline{\varepsilon}$, reach ε .

This suffices to describe a winning strategy for the robber on $\mathcal{G}^{k+1}$: by switching between the two subtrees of the root.

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

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

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

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

Assume that the robber enters a tree $\mathcal{T}^{xy}$, for $x, y \in \{0,1\}$ which is free of cops (this 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 $\mathcal{T}^{00}$ at 00. Either the play remains in $\mathcal{T}^{00}$ forever (and we are done), or the $(k/2-1)$-st cop enters $\mathcal{T}^{00}$ and the robber reaches $\overline{00}$. Note that while the robber is moving towards $\overline{00}$, no cops can be placed outside of $\mathcal{T}^{00}$ as the robber does not leave $\mathcal{T}^{00}$.

If the last cop is already placed, the robber goes to $\overline{0}$ and then to $\overline{\varepsilon}$, which are not occupied by cops, according to the invariant, and we are done. If the last cop is not placed yet, all cops are in $\mathcal{T}^{00}$, so the robber runs along the path $\overline{00}, \overline{0}, 0, \varepsilon, 1, 10$ to $\mathcal{T}^{10}$. Note that the nodes $\overline{\varepsilon}$ and $\overline{1}$ are not occupied by cops, so the invariant still holds. The robber plays as in $\mathcal{T}^{00}$ and so on. $\square$

While entanglement is not bounded in the other measures, we can prove that it grows only logarithmically in the size of the graph if the tree width is fixed. For the proof we recall the definition of a tree decomposition [27]. Let $\mathcal{G} = (V, E)$ be an undirected graph. A *tree decomposition* of $\mathcal{G}$ is an undirected tree $\mathcal{T} = (T, F)$ together with a collection $\mathcal{X} = \{X_t \mid t \in T\}$ of subsets of V, called bags, indexed by elements of T that satisfy the following properties.

(1) $\bigcup \mathcal{X} = V$.
(2) For all $\{v, w\} \in E$ there is some some $t \in T$ with $\{v, w\} \subseteq X_t$.
(3) For every $v \in V$, the set $\{t \in \mathcal{T} \mid v \in X_t\}$ induces a (connected) subtree of $\mathcal{T}$.

The width of a tree decomposition $(\mathcal{T}, \mathcal{X})$ is the size of the largest $X_t \in \mathcal{X}$. The tree width of $\mathcal{G}$ is the minimal width over all tree decompositions of $\mathcal{G}$ minus one.

Proposition 7.4. *For any finite undirected graph $\mathcal{G}$ of tree width k, we have that* $\mathrm{ent}(\mathcal{G}) \leq (k+1) \cdot \log |G|$.

Proof. Let $\mathcal{T}$ be a decomposition tree of minimal width of $\mathcal{G}$. Without loss of generality, we can assume that $\mathcal{T}$ is a binary tree. Our argument uses the separator properties of tree decompositions (see, e.g., [8]). In every subtree $\mathcal{S}$ of $\mathcal{T}$, there exists a node s, we may call it the *centre* of $\mathcal{S}$, which balances $\mathcal{S}$ in the sense that the subtrees in $\mathcal{S} \setminus \{s\}$ carry almost the same number of nodes in their bags (differences up to k are admissible). Consider now the following memoryless cop strategy. First, all nodes in the centre s of the decomposition tree receive indices $0, \ldots, k$. Then, we repeat the process independently for the two subtrees disconnected by the removal of s and assign to the nodes in their respective centres indices $k+1, \ldots, 2k+2$. The process ends when all nodes of $\mathcal{G}$ are labelled. In this way, at most $(k+1) \log |V|$ cop indices are assigned. Since the bags of a tree decomposition separate the graph, every strongly connected subgraph of $\mathcal{G}$ will contain at least one unique label. By Lemma 7.1, the constructed labelling indeed represents a memoryless strategy for at most $(k+1) \log |V|$ cops. $\square$

Very little is known about the opposite direction, whether other measures are bounded in entanglement. We have no general characterisation of entanglement in terms of a decomposition and it is not clear how to construct decompositions for other measures from a winning strategy for the cops in the entanglement game. On the other side, it is difficult to translate winning strategies from the entanglement game other graph searching games where monotonicity is required.

In contrast, the relationship between entanglement and path width and directed path width is better understood. The latter measures are defined in the same way as tree width, only that the word "tree" is replaced with "path" and "directed path", respectively. Let $\mathcal{G}$ be an undirected graph. We denote by $\overleftrightarrow{\mathcal{G}}$ the directed graph obtained from $\mathcal{G}$ by replacing every undirected edge $\{v, w\}$ by two directed edges (v, w) and (w, v).

Proposition 7.5.

(1) *Let $\mathcal{G}$ be a directed graph. Then* $\mathrm{ent}(\mathcal{G}) \leq \mathrm{dpw}(\mathcal{G}) + 1$.

(2) *There exists a class of undirected graphs $\mathcal{G}_n$ such that, for every $n > 1$,* $\mathrm{ent}(\overleftrightarrow{\mathcal{G}_n}) = 2$ *and* $\mathrm{pw}(\mathcal{G}_n) = \mathrm{dpw}(\overleftrightarrow{\mathcal{G}_n})$ *are unbounded.*

Proof.

(1) Let $(\mathcal{P}, \mathcal{X}, f)$ be a path decomposition of $\mathcal{G}$ of minimal width k with $\mathcal{P} = (P, E_P)$ and $P = (p_0, \ldots, p_m)$. Then, the largest bag has size $k+1$. The strategy of the Cop player is to expel the robber from $f(p_0)$, then from $f(p_1)$ and so on, until the robber is captured in the last bag. For every such step, at most k cops are needed (remember that the robber has to move when it is her turn).

(2) Let $\mathcal{G}_n$ be the undirected full binary tree of height n. Then $\mathrm{pw}(\mathcal{G}_n) = \mathrm{dpw}(\overleftrightarrow{\mathcal{G}_n})$ are unbounded while n grows, see [8]. In the entanglement game, two cops have the following winning strategy. A play is divided in rounds that are separated by a downward move of the robber. In each round the same cop follows the robber in every move, the other cop remains idle. The cops alternate in each round. It is easy to see that the robber will be captured.

□

8. Graphs of entanglement two

To motivate and present intuitions for the class of graphs of entanglement two, we first introduce a class $\mathcal{F}$ of graphs (V, E, F) where $F \subseteq V$ is a set of marked nodes. The class $\mathcal{F}$ is defined inductively, as follows:

(1) The graph consisting of one marked node and without edges is in $\mathcal{F}$.

(2) $\mathcal{F}$ is closed under removal of edges, i.e., if $(V, E, F) \in \mathcal{F}$ and $E' \subseteq E$ then $(V, E', F) \in \mathcal{F}$.

(3) For $\mathcal{G}_1, \mathcal{G}_2 \in \mathcal{F}$ with marked nodes F_1 and F_2, the disjoint union of $\mathcal{G}_1$ and $\mathcal{G}_2$ with marked $F_1 \cup F_2$ is in $\mathcal{F}$.

(4) For $\mathcal{G}_1 = (V_1, E_1, F_1), \mathcal{G}_2 = (V_2, E_2, F_2) \in \mathcal{F}$, their marked sequential composition $\mathcal{G}$ is in $\mathcal{F}$, where

$$\mathcal{G} = (V_1 \cup V_2, E_1 \cup E_2 \cup F_1 \times V_2, F_1 \cup F_2).$$

(5) For $\mathcal{G} = (V, E, F) \in \mathcal{F}$, the graph $\mathcal{G}'$ with added marked loop is in $\mathcal{F}$, where for a new node v,

$$\mathcal{G}' = (V \cup \{v\}, E \cup (F \times \{v\}) \cup (\{v\} \times V), \{v\}).$$

Notice that the rules (2)–(4) add no cycles and do not increase the entanglement. New cycles are created in (5), but only between the marked nodes and a new node, which afterwards becomes the only marked node.

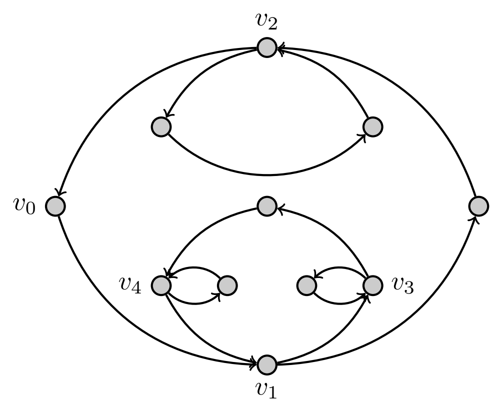

Figure 1: Example graph of entanglement two.

All graphs in the class $\mathcal{F}$ have entanglement two. Before we explain the meaning of the marked nodes F (in Section 8.1) and present the strategy for the cops in $\mathrm{EG}_2(\mathcal{G})$ for graphs $\mathcal{G} \in \mathcal{F}$ (in the proof of Theorem 8.15), let us describe a few sub-classes of $\mathcal{F}$ and possible uses for graphs of entanglement two.

One sub-class of $\mathcal{F}$ consists trees with edges directed to the root and, additionally, any set of back-edges going downwards. More formally, such trees can be described as structures $\mathcal{T} = (T, E_T \cup E_{back})$ where (T, E_T) is a tree with edges directed to the root and for any back-edge $(w, v) \in E_{back}$ it must be the case that w is reachable from v in (T, E_T). Such graphs have entanglement at most two. A winning strategy for the cops is to chase the robber with one cop until she goes along a back-edge (w, v). Then she is blocked by this cop in the subtree rooted at w. Now the second cop chases the robber until she takes another back-edge, and so on, until she is captured at a leaf.

Another class of graphs included in $\mathcal{F}$ are control-flow graphs for well-structured programs (which do not use `goto`). Such graphs are constructed using sequential and parallel composition (corresponding to items (3) and (4) in the definition of $\mathcal{F}$), and loops with a single entry and a single exit point, which are a special case of item (5) in the definition of $\mathcal{F}$.

Consider, for example, the graph depicted in Figure 1. Removing v_0 from this graph leaves only two non-trivial strongly connected components, namely the v_1-loop and the v_2-loop, and one trivial component consisting of a single node.[1] The loops can be decomposed as well by removing v_1 and v_2; finally, the v_3-loop and the v_4-loop can be decomposed as well. This decomposition induces a strategy for the cops: first place one of them on v_0 and then chase the

[1] We consider only non-trivial strongly connected components, i.e., *not* single nodes without self-loops.

robber on v_1 with the other cop. If the robber enters the v_1-loop, the cop from v_0 chases her on v_3 and v_4 and so she is captured. If the robber does not enter the v_1-loop, the cop from v_1 chases her on v_2 and so she is captured.

One of the main results in this section is Theorem 8.14 where we show that a decomposition, generalising the above example, can be found for each graph of entanglement two. As a consequence, we prove in Theorem 8.15 that graphs of entanglement two can be characterised in a way similar to the above definition of the class $\mathcal{F}$. More precisely, a graph has entanglement at most two if, and only if, each of its strongly connected components belongs to a class $\mathcal{F}'$, which is defined similarly to the class $\mathcal{F}$, but with rule (5) changed as follows.

(5′) For $\mathcal{G} = (V, E, F) \in \mathcal{F}'$, the graph $\mathcal{G}'$ with added loop is in $\mathcal{F}'$, where

$$\mathcal{G}' = (V \cup \{v\}, E \cup (F \times \{v\}) \cup (\{v\} \times V), \{v\} \cup F'),$$

and F' is any subset of the previously marked nodes F such that $\mathcal{G}[F']$ is acyclic and no nodes in F' are reachable from $V \setminus F'$.

A consequence of our proofs, stated in Proposition 8.20, is that graphs of entanglement two have both DAG-width and Kelly-width at most 3. This confirms that graphs of entanglement two are simple according to all known graph measures, and strengthens our motivation to study them as the most basic class of graphs where cycles are already nested in interesting ways.

8.1. Entanglement of graphs with exit nodes

In this section, we introduce a technical notion which is crucial for subsequent proofs: the entanglement of a graph with exit nodes. To provide an intuition for this notion, consider the graph in Figure 1 with the node v_0 removed. This graph contains two non-trivial strongly connected components: the v_1-loop and the v_2-loop. The v_2-loop has entanglement one, so it is clearly simpler than the entire graph. On the other hand, the v_1-loop has entanglement two. Nevertheless, we claim that also the v_1-loop is in a sense simpler than the entire graph, despite having the same entanglement. Indeed, observe that not only can two cops capture the robber on the v_1-loop, but they can do it in such a way that the only node through which the robber can exit this loop, v_1, remains blocked during the whole play after the robber visits it. This observation leads to the notion we study here.

8.1.1. Simple and complex components

In the rest of this section, we focus on strongly connected subgraphs of a graph. Let $\mathcal{G}$ be a graph and $\mathcal{G}'$ a strongly connected subgraph of $\mathcal{G}$. The set $\mathrm{Ex}(\mathcal{G}, \mathcal{G}')$ of *exit nodes* of $\mathcal{G}'$ in $\mathcal{G}$ is the set of all $v \in \mathcal{G}'$ for which there is a node $u \in \mathcal{G} \setminus \mathcal{G}'$ with $(v, u) \in E$ (note that we sometimes write $v \in \mathcal{G}$ if $\mathcal{G} = (V, E)$ and $v \in V$).

To study subgraphs that contain exit nodes in a way that is independent of the bigger graph in the context, we say that $\mathcal{G}^*$ is a *graph with exit nodes* when $\mathcal{G}^* = (V, E, F)$, where (V, E) is a graph and F is any subset of V representing the exits. The following notion is used while decomposing a graph $\mathcal{G}$.

Definition 8.1. Let $\mathcal{G}$ be a graph and let $v \in \mathcal{G}$. A *v-component* of $\mathcal{G}$ is a graph $\mathcal{C} = (C, E, F)$ with exit nodes such that (C, E) is a strongly connected component of $\mathcal{G} \setminus v$ and $F = \mathrm{Ex}(\mathcal{G}, \mathcal{C})$.

In a strongly connected graph $\mathcal{G}$, for a node v, let $\leq_v$ be the topological order on the set of strongly connected components of $\mathcal{G} \setminus v$, i.e.,

$$\mathcal{C} \leq_v \mathcal{C}' \Leftrightarrow \text{ there is a path from } \mathcal{C} \text{ to } \mathcal{C}' \text{ in } \mathcal{G} \setminus v.$$

The *entanglement game with exit nodes* $\mathrm{EG}^*_k(\mathcal{G})$ is played on a graph $\mathcal{G} = (V, E, F)$ with exit nodes in the same way as the entanglement game, but with an additional winning condition for the robber: she wins a play if she succeeds in reaching an exit node after the last cop has entered $\mathcal{G}$ from outside. More formally, the robber wins a play if it reaches a position (v, P) such that $v \in F$ and $|P| = k$. (This includes the case when the robber already sits on an exit node at the time when the last cop moves to that node.) In the context of subgraphs inside a larger graph this new winning condition means that the robber can leave the subgraph and get back to the bigger graph.

We define a further variant of the entanglement game to mark the node from that a play starts. Let v be a node of $\mathcal{G}$. The game $\mathrm{EG}^*_k(\mathcal{G}, v)$ is played in the same way as $\mathrm{EG}^*_k(\mathcal{G})$, except that the robber does not choose a node to start on, but starts on v.

Definition 8.2. A graph with exit nodes $\mathcal{G}$ is *k-complex* if the robber has a winning strategy (which we call a *robber $\mathcal{G}$-strategy*) in the entanglement game with exit nodes $\mathrm{EG}^*_{k+1}(\mathcal{G})$. If the cops have a winning strategy in $EG^*_{k+1}(\mathcal{G})$ (called a *cops $\mathcal{G}$-strategy*), then $\mathcal{G}$ is *k-simple.*

To start with, let us show that existence of a node with only k-simple components gives a bound on entanglement.

Proposition 8.3. *If there is a node v in a graph $\mathcal{G}$ such that all v-components are k-simple, then* $\mathrm{ent}(\mathcal{G}) \leq k + 1$.

Proof. Let v be a node such that all v-components of $\mathcal{G}$ are k-simple. Let σ be any strategy for the cops in $\mathrm{EG}_{k+1}(\mathcal{G})$ with the following properties:

- if the robber is on v then chase her there with any cop, i.e., $\sigma(v, P) \neq \perp$,
- if the robber is on a node u that is not in a v-component, then wait: $\sigma(u, P) = \perp$,
- if the robber is on a node u in a k-simple v-component $\mathcal{C}$, then use a $\mathcal{C}$-strategy $\sigma^{\mathcal{C}}$ moving the cop from v only as the last resort, i.e.,

$$\sigma(u, P) = \begin{cases} \sigma^C(u, P \cap \mathcal{C}) & \text{if } \sigma^C(u, P \cap \mathcal{C}) \in \mathcal{C} \text{ or } \sigma^C(u, P \cap \mathcal{C}) = \perp, \\ \Box & \text{if } \sigma^C(u, P \cap \mathcal{C}) = \Box,\ P \setminus \mathcal{C} = \{v\} \text{ and } |P| \leq k, \\ w & \text{if } \sigma^C(u, P \cap \mathcal{C}) = \Box \text{ and } w \in P \setminus \mathcal{C} \text{ with } w \neq v, \\ v & \text{if } \sigma^C(u, P \cap \mathcal{C}) = \Box \text{ and } |P| = k + 1. \end{cases}$$

We show that σ is winning for the cops in $\mathrm{EG}_{k+1}(\mathcal{G})$. Assume that the robber has a counter-strategy ρ to win the play that is consistent with both ρ and σ. First we show that this play visits v. Indeed, if it starts in a node $v_0 \neq v$ then the robber will either be captured in the v-component $\mathcal{C}$ containing v_0 (we can assume that v_0 is in a v-component, otherwise the cops stay idle until the robber enters such a component or visits v), or she will be expelled from $\mathcal{C}$, because the cops use a $\mathcal{C}$-strategy. Since we assumed that the robber wins, she is expelled from $\mathcal{C}$. This will continue until v is reached. In this moment an arbitrary cop goes to v. Afterwards the cop from v is moved only as the $(k+1)$-st one to enter a component $\mathcal{C}$. Therefore the robber will always either be captured in $\mathcal{C}$ or expelled again without using the cop from v — and thus finally captured. $\square$

In the remainder of this section, we prove that the converse holds for the case $k = 1$. This will lead to Theorem 8.11 and provide the basis of a structural characterisation of graphs of entanglement two in Section 8.2.

8.1.2. Independence from the starting node

Lemma 8.4. *Let $\mathcal{G}$ be a strongly connected k-complex graph with exit nodes. Then the robber wins $\mathrm{EG}^*_{k+1}(\mathcal{G}, v)$ for all $v \in \mathcal{G}$.*

Proof. Let us divide the nodes of $\mathcal{G}$ into two subsets: the set V_R of nodes v from which the robber wins $\mathrm{EG}^*_{k+1}(\mathcal{G}, v)$ and the set V_C of nodes v from which the cops win $\mathrm{EG}^*_{k+1}(\mathcal{G}, v)$. These sets are disjoint and as $\mathcal{G}$ is k-complex, V_R is not empty.

Let us assume that V_C is not empty. As $\mathcal{G}$ is strongly connected, there exists an edge from V_C to V_R. Pick such an edge $(w, v) \in E$ and let

- ρ^v be a winning strategy for the robber in $\mathrm{EG}^*_{k+1}(\mathcal{G}, v)$,
- σ^w be a winning strategy for the cops in $\mathrm{EG}^*_{k+1}(\mathcal{G}, w)$.

First, observe that in no play consistent with ρ^v, the robber enters w before the last $(k+1)$-st cop moves into $\mathcal{G}$. Indeed, if this was the case, the cops could just continue playing σ^w from w as if all cops placed already were outside. As σ^w is winning, this continued play has to end in a position where the robber can neither move nor reach an exit node. But this contradicts the fact that the play was consistent with ρ^v, which is winning for the robber.

We show that the following robber strategy ρ^w is winning: first move from w to v and then continue playing ρ^v, ignoring any cop that may be placed on w in the first move of the cops. Indeed, if the cops are idle in the first move, then the play proceeds according to ρ^v and is thus winning for the robber. Otherwise, the play proceeds according to ρ^v as if there was no cop on w. But, as observed above, this infinite play never visits w and thus the cop standing there makes no difference – the play is winning for the robber.

Since ρ^w is winning for the robber in $\mathrm{EG}^*_{k+1}(\mathcal{G}, w)$ and σ^w is winning for the cops in the same game, we get a contradiction. Thus V_C is empty, so all nodes of $\mathcal{G}$ belong to V_R. $\square$

The following result follows from the above lemma by taking $F = \emptyset$.

Corollary 8.5 ([7]). *Let $\mathcal{G}$ be a strongly connected graph of entanglement k. Then the robber wins the game $\mathrm{EG}_k(\mathcal{G})$ with a changed starting rule, stating that, at the beginning of a play it is not the robber, but the cops who choose the node from which the robber has to start.*

To prove a converse of Proposition 8.3 we need to consider various configurations of complex components. We will show that the existence of certain combinations of 1-complex components implies that the graph has entanglement greater than two. This will be used in the Section 8.1.6 to show that every graph of entanglement two contains a node so that after its removal all components are 1-simple. We will later prove that the corresponding property fails for graphs of entanglement $k \geq 3$.

8.1.3. Topologically incomparable components

Lemma 8.6. *Let $\mathcal{G}$ be a strongly connected graph and let $v \in \mathcal{G}$. Further, let $\mathcal{C}_0$ and $\mathcal{C}_1$ be two k-complex v-components. If $\mathcal{C}_0$ and $\mathcal{C}_1$ are incomparable with respect to $\leq_v$, then* $\mathrm{ent}(\mathcal{G}) > k + 1$.

Proof. Assume that $\mathcal{C}_0$ and $\mathcal{C}_1$ have entanglement at most $k+1$. Otherwise the robber wins playing in the component of entanglement greater than $k+1$. Thus, the robber's $\mathcal{C}_0$-strategy ρ^0 and the robber's $\mathcal{C}_1$-strategy ρ^1 prescribe the robber to go to an exit node when all $k+1$ cops arrive in the component. Note that these strategies are not defined for positions (w, P) where, for some $i \in \{0, 1\}$, $w \in \mathrm{Ex}(\mathcal{G}, \mathcal{C}_i)$ and $|P \cap \mathcal{C}_i| = k + 1$.

The following strategy ρ is winning for the robber in $\mathrm{EG}_{k+1}(\mathcal{G})$:

- the robber starts on any node $w \in \mathcal{C}_0$;
- $\rho(w, P) = \rho^i(w, P)$ if
  - $w \in \mathcal{C}_i$, for $i \in \{0, 1\}$, and
  - $|P \cap \mathcal{C}_i| \leq k + 1$, or both $|P \cap \mathcal{C}_i| < k + 1$ and $w \notin \mathrm{Ex}(\mathcal{G}, \mathcal{C}_i)$;
- $\rho(w, P)$ prescribes to run to $\mathcal{C}_{1-i}$ in any possible way if $w \in \mathrm{Ex}(\mathcal{G}, \mathcal{C}_i)$ and $|P \cap \mathcal{C}_i| = k + 1$, for $i \in \{0, 1\}$;
- $\rho(w, P)$ prescribes to run to $\mathcal{C}_{1-i}$ in any possible way if $w \notin \mathcal{C}_0 \cup \mathcal{C}_1$ and $P \cap \mathcal{C}_i \neq \emptyset$, for $i \in \{0, 1\}$;
- $\rho(w, P)$ prescribes to run to $\mathcal{C}_0$ in any possible way if $w \notin \mathcal{C}_0 \cup \mathcal{C}_1$ and $P \cap \mathcal{C}_i \neq \emptyset$, for each $i \in \{0, 1\}$.

To see that ρ is indeed winning for the robber, and that there always is a possible path from w to $\mathcal{C}_i$ in the second and the third cases of the definition above, let us consider a play consistent with ρ.

The robber starts on some node in $\mathcal{C}_0$ and plays ρ^0 until all $k + 1$ cops are in $\mathcal{C}_0$. When the last cop moves to $\mathcal{C}_0$, she reaches an exit node u, because $\mathcal{C}_0$ is

k-complex and ρ^0 was a $\mathcal{C}_0$-strategy. From u, she can run to v and then to $\mathcal{C}_1$ (without entering $\mathcal{C}_0$ again), because the components are incomparable and all paths between them lead through v (note that $v \notin \mathcal{C}_0$ and the graph is strongly connected). Now she plays according to ρ^1 until all $k+1$ cops come to $\mathcal{C}_1$, and analogously proceeds to $\mathcal{C}_0$ via v. This goes on indefinitely, so $k+1$ cops never capture her. $\square$

8.1.4. Disjoint components

We first consider the case of disjoint components that contain each others basis node, and then a more general case.

Lemma 8.7. *Let $\mathcal{G}$ be a strongly connected graph, and let $a_0, a_1 \in \mathcal{G}$ such that, for $i \in \{0,1\}$, a_i is in a k-complex (a_{1-i})-component $\mathcal{C}_{1-i}$. If $\mathcal{C}_0 \cap \mathcal{C}_1 = \emptyset$, then* $\mathrm{ent}(\mathcal{G}) > k+1$.

Proof. The proof of this lemma is analogous to the proof of Lemma 8.6. Assume again that $\mathcal{C}_0$ and $\mathcal{C}_1$ have entanglement at most $k+1$. Otherwise the robber wins playing in the component of entanglement greater than $k+1$. Thus, $\mathcal{C}_0$-strategy ρ_0 and $\mathcal{C}_1$-strategy ρ_1 prescribe the robber to go to an exit node when all $k+1$ cops arrive in the component.

The following strategy ρ is winning for the robber in $\mathrm{EG}_{k+1}(\mathcal{G})$:

- the robber starts on any node $w \in \mathcal{C}_0$;
- $\rho(w, P) = \rho^i(w, P)$ if
  - $w \in \mathcal{C}_i$, for $i \in \{0,1\}$, and
  - $|P \cap \mathcal{C}_i| \leq k+1$, or both $|P \cap \mathcal{C}_i| < k+1$ and $w \notin \mathrm{Ex}(\mathcal{G}, \mathcal{C}_i)$;
- $\rho(w, P)$ prescribes to run along any path leading to $a_i \in \mathcal{C}_{1-i}$ until the robber enters $\mathcal{C}_{1-i}$ if $w \in \mathrm{Ex}(\mathcal{G}, \mathcal{C}_i)$ and $|P \cap \mathcal{C}_i| = k+1$, for $i \in \{0,1\}$;
- $\rho(w, P)$ prescribes to run along any path leading to $a_i \in \mathcal{C}_{1-i}$ until the robber enters $\mathcal{C}_{1-i}$ if $w \notin \mathcal{C}_0 \cup \mathcal{C}_1$ and $|P \cap \mathcal{C}_i| \neq \emptyset$, for $i \in \{0,1\}$;
- $\rho(w, P)$ prescribes to run to $\mathcal{C}_0$ in any possible way if $w \notin \mathcal{C}_0 \cup \mathcal{C}_1$ and $P \cap \mathcal{C}_i \neq \emptyset$, for both $i \in \{0,1\}$.

To see that ρ is indeed winning for the robber, and that there always is a possible path from w to $\mathcal{C}_i$ in the second and in the third cases of the definition above, let us consider a play consistent with ρ.

The robber starts on some node in $\mathcal{C}_0$ and plays ρ^0 until all $k+1$ cops are in $\mathcal{C}_0$. When the last cop moves to $\mathcal{C}_0$, she reaches an exit node u, because $\mathcal{C}_0$ is k-complex and ρ^0 was a $\mathcal{C}_0$-strategy. From u, she can run to a_0 and thus (as $a_0 \in \mathcal{C}_1$) to $\mathcal{C}_1$ (without entering $\mathcal{C}_0$ again), because $a_0 \notin \mathcal{C}_0$ and the graph is strongly connected. Now she plays according to ρ^1 until all $k+1$ cops come to $\mathcal{C}_1$, and analogously proceeds to $\mathcal{C}_0$ on a way to a_1. This goes on indefinitely, so $k+1$ cops never capture her. $\square$

Lemma 8.8. *Let $\mathcal{G}$ be a strongly connected graph. For $i \in \{0,1\}$, let a_i be two distinct vertices and let $\mathcal{C}_i$ be two k-complex a_i-components such that $\mathcal{C}_0$ is maximal with respect to $\leq_{a_0}$ and $a_1 \in \mathcal{C}_0$. If $\mathcal{C}_0 \cap \mathcal{C}_1 = \emptyset$ then* $\mathrm{ent}(\mathcal{G}) > k+1$.

Proof. It is sufficient to prove that $a_0 \in \mathcal{C}_1$; then, the desired result follows from Lemma 8.7. Towards a contradiction, assume that $a_0 \notin \mathcal{C}_1$. We distinguish three cases according to how $\mathcal{C}_1$ can be combined with k-complex a_0-components.

Case 1: There is a k-complex a_0-component $\mathcal{C}'$ and $\mathcal{C}_1 \subseteq \mathcal{C}'$. If the components $\mathcal{C}'$ and $\mathcal{C}_0$ are incomparable with respect to $\leq_{a_0}$ then Lemma 8.6 guarantees a winning strategy for the robber in the entanglement game on $\mathcal{G}$ against $k+1$ cops. Because $\mathcal{C}_0$ is maximal, we have that $\mathcal{C}' \leq_{a_0} \mathcal{C}_0$ and there is a path $\mathcal{P}_1$ from $\mathcal{C}_1$ to $\mathcal{C}_0$ with $a_0 \notin \mathcal{P}_1$ (see Figure 2).

Since $\mathcal{G}$ is strongly connected, there exists a path $\mathcal{P}_2$ from a_0 to $\mathcal{C}_1$, but no such path includes nodes of $\mathcal{C}_0$. Otherwise $\mathcal{C}_0$ and $\mathcal{C}'$ would be in the same strongly connected component of $\mathcal{G} \setminus a_0$. Furthermore, every path $\mathcal{P}_3$ from $\mathcal{C}'$ to a_0 (there is at least one) goes through a_1 (otherwise $a_0 \in \mathcal{C}_1$).

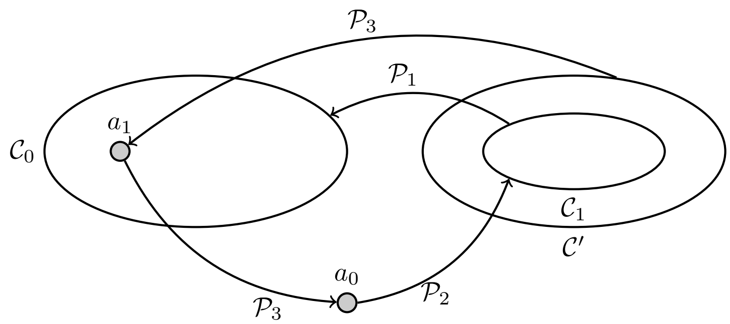

Figure 2: Case 1: $\mathcal{C}_1$ is in an a_0-component $\mathcal{C}'$.

This guarantees that the robber wins the entanglement game on $\mathcal{G}$ against $k+1$ cops switching between $\mathcal{C}'$ and $\mathcal{C}_0$, because playing according to a $\mathcal{C}_0$-strategy and being expelled from $\mathcal{C}_0$ by $k+1$ cops she can reach a_0 and then $\mathcal{C}_1$. Playing according to a $\mathcal{C}_1$-strategy and being expelled from $\mathcal{C}_1$ she can reach a_0 and thus $\mathcal{C}_0$, which on the way to a_0. Lemma 8.4 assures that it makes no difference at which node the robber enters $\mathcal{C}'$ (or $\mathcal{C}_0$): she always has a $\mathcal{C}'$-strategy (or a $\mathcal{C}_0$-strategy).

Case 2. The component $\mathcal{C}_1$ includes nodes of two different strongly connected components of $\mathcal{G} \backslash a_0$. Then there is a path in $\mathcal{C}_1$ from one such strongly connected component to the other that does not go through a_1, but through a_0. (If all such paths avoided a_0, the two strongly connected components would not be distinct.) But then we have $a_0 \in \mathcal{C}_1$.

Case 3. $\mathcal{C}_1$ does not include nodes of different a_0-components and is not a strict subset of a k-complex a_0-component. Due to our assumption, $a_0 \notin \mathcal{C}_1$, and we distinguish two subcases.

Case 3a. $\mathcal{C}_1$ consists of some nodes from an a_0-component $\mathcal{C}'$ and some nodes that are in no strongly connected component of $\mathcal{G} \setminus a_0$. In this case, these nodes must also be a part of $\mathcal{C}'$, because all nodes of $\mathcal{C}_1$ are connected by paths that contain neither a_0 nor a_1. So, in fact, this subcase is not possible.

Case 3b. $\mathcal{C}_1$ lies in a k-simple a_0-component $\mathcal{C}'$. We show that because $\mathcal{C}_1$ is k-complex, $\mathcal{C}'$ must be k-complex as well, which contradicts the assumption of this subcase. We describe a $\mathcal{C}'$-strategy for the robber. She starts in $\mathcal{C}_1$ and plays according to her $\mathcal{C}_1$-strategy. We can assume that it prescribes to wait until all $k+1$ cops come to $\mathcal{C}_1$, because otherwise $\mathrm{ent}(\mathcal{C}_1) > k+1$ and $\mathrm{ent}(\mathcal{C}') > k+1$. When all cops come to $\mathcal{C}_1$ the robber can leave $\mathcal{C}_1$. We show that she can leave $\mathcal{C}'$ as well. It suffices to show that from every $v \in \mathrm{Ex}(\mathcal{C}', \mathcal{C}_1)$ there is a path to a node $w \in \mathrm{Ex}(\mathcal{G}, \mathcal{C}')$ that avoids $\mathcal{C}_1$ (except the node v). Otherwise every path $\mathcal{P}$ from v to some w (there is such path because $\mathcal{C}'$ is strongly connected) leaves $\mathcal{C}_1$, goes through at least one node $u \in \mathcal{C}' \setminus \mathcal{C}_1$ and then goes back to $\mathcal{C}_1$. Then $a_1 \notin \mathcal{P}$ because $\mathcal{P} \subseteq \mathcal{C}'$, $a_1 \in \mathcal{C}_0$, and $\mathcal{C}'$ and $\mathcal{C}_0$ are distinct a_0-components. So we have $u \in \mathcal{C}_1$, but we assumed that $u \notin \mathcal{C}_1$. $\square$

The maximality of $\mathcal{C}_0$ in Lemma 8.8 is essential. Consider the graph in Figure 3. All requirements of Lemma 8.8 are fulfilled for this graph except the maximality of $\mathcal{C}_0$: $\mathcal{C}_0$ is a 1-complex a_0-component, $\mathcal{C}_1$ is a 1-complex a_1-component, and $a_1 \in \mathcal{C}_0$. The entanglement of the graph is two, although $\mathcal{C}_0$ and $\mathcal{C}_1$ are disjoint. The cops have the following winning strategy. We only assume moves of the robber that lead to a strongly connected cop free subgraph. The cops expel the robber from $\mathcal{C}_1$, if she is there, and place one of the cops on node a_1, which must be visited by the robber leaving $\mathcal{C}_1$. The robber visits node v and the other cop goes there. The robber proceeds to w and the cop who is not on v occupies w. Then the cop from v forces the robber to leave $\mathcal{C}_1$ and follows her to a_1. The robber visits v again, the cop from a_1 follows her there. As node w is occupied, the robber has to remain in $\mathcal{C}_0 \cup \{a_0\}$. The cop from w goes to a_1 and captures the robber.

Note that we actually have shown that all w-components are 1-simple and used the strategy for the cops described in the proof of Proposition 8.3.

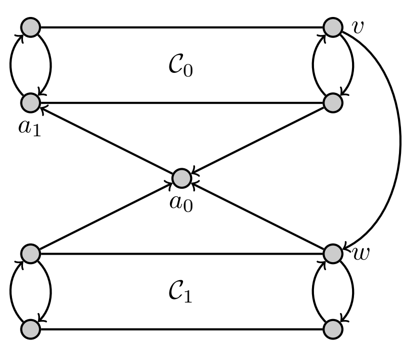

Figure 3: Importance of maximality of the components.

8.1.5. Pairwise intersecting 1*-complex components*

Lemma 8.9. *Let $\mathcal{G}$ be a strongly connected graph. Let $I = \{0, \dots, m\}$ be an index set for some $m \in \{1, \dots, |V| - 1\}$. For $i \in I$, let $a_i \in \mathcal{G}$ and let $\mathcal{C}_i$ be a* 1*-complex a_i-component such that $a_i \in \mathcal{C}_j$ for all $i \neq j$ and $j \in I$. If $\bigcap_{i \in I} \mathcal{C}_i = \emptyset$, then $\operatorname{ent}(\mathcal{G}) > 2$.*

Proof. If $m = 1$, then we have the conditions of Lemma 8.7, so assume that $m \geq 2$. We, further, can assume that $\operatorname{ent}(\mathcal{C}_i) \leq 2$ for all $i \in I$. Then $\mathcal{C}_i$-strategies prescribe the robber to wait in the component until both cops come and then to reach an exit node.

We give a winning strategy for the robber in the game $\mathrm{EG}_2(\mathcal{G})$. She starts in a cop free component $\mathcal{C}_j$ and plays according to her $\mathcal{C}_j$-strategy. When the second cop moves to $\mathcal{C}_j$ she escapes from $\mathcal{C}_j$. Now it suffices to show that she can reach a new cop free component. Let the second cop come to $\mathcal{C}_j$ on a node v, the first cop being on a node $w \in \mathcal{C}_j$. At this point, since $\bigcap_{l \in I} \mathcal{C}_l = \emptyset$, there is an a_i-component $\mathcal{C}_i$ with $w \notin \mathcal{C}_i$. If $v \in \mathcal{C}_i$, the robber plays her $\mathcal{C}_i$-strategy starting from v and assuming that a cop followed her there. If $v \notin \mathcal{C}_i$, then the robber can escape from $\mathcal{C}_j$ and reach a_j, which is in the cop free component C_i. On entering $\mathcal{C}_i$, the robber continues with a $\mathcal{C}_i$-strategy. □

8.1.6. A node having only simple components

Before we prove Theorem 8.11, we need one more lemma about possible configurations of incomparable strongly connected components.

Lemma 8.10. *Let $\mathcal{G}$ be a strongly connected graph. Let $\mathcal{C}_v$ be a v-component, and $\mathcal{C}_w$ be a w-component of $\mathcal{G}$, for distinct nodes v and w such that $\mathcal{C}_v \cap \mathcal{C}_w \neq \emptyset$ and $\mathcal{C}_v \nsubseteq \mathcal{C}_w$. If v is in $\mathcal{C}_w$, then w is in $\mathcal{C}_v$.*

Proof. Assume that the conditions of the lemma hold, but $w \notin \mathcal{C}_v$ (Figure 4). Let $u \in \mathcal{C}_v \cap \mathcal{C}_w$ and $u' \in \mathcal{C}_v \setminus \mathcal{C}_w$. Because $u', u \in \mathcal{C}_v$, which is strongly connected, there are paths from u' to u and vice versa that do not include v. None of these paths includes w (because otherwise $w \in \mathcal{C}_v$), so u' and u lie in the same w-component. But we assumed that $u' \notin \mathcal{C}_w$, and $u \in \mathcal{C}_w$, and C_w is strongly connected: contradiction. □

With the above lemma, we can finally prove the converse of Proposition 8.3.

Theorem 8.11. *On a strongly connected graph $\mathcal{G} = (V, E)$, two cops have a winning strategy in the game $\mathrm{EG}_2(\mathcal{G})$ if, and only if, there exists a node $a \in \mathcal{G}$ such that every a-component is* 1*-simple.*

Proof. The direction from right to left is proven in Proposition 8.3: if every a-component is 1-simple, then $\operatorname{ent}(\mathcal{G}) \leq 2$. We show the other direction.

Towards a contradiction, assume that the cops win $\mathrm{EG}_2(\mathcal{G})$, but, for all $a \in V$ there is a a-component $\mathcal{C}$ of $\mathcal{G}$ such that they lose $\mathrm{EG}_2^*(\mathcal{C})$.

We construct a sequence $a_0, a_1, \dots, a_m$ of nodes from V and a sequence $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_m$ of corresponding a_i-components $\mathcal{C}_i$. We require that all $\mathcal{C}_i$ are maximal 1-complex a_i-components with respect to $\leq_{a_i}$, and that $\bigcap_{i=0}^m \mathcal{C}_i \neq \emptyset$.

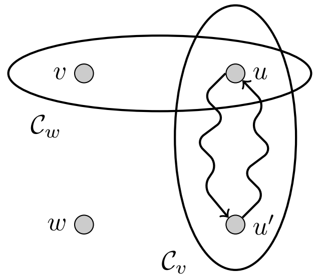

Figure 4: The w-component $\mathcal{C}_w$ contains v, but the v-component $\mathcal{C}_v$ does not include w.

Take an arbitrary node as a_0. There is a 1-complex a_0-component $\mathcal{C}_0$, due to the assumption. Choose among all such strongly connected components a maximal one with respect to $\leq_{a_0}$. In general, suppose that a_i and $\mathcal{C}_i$ are already constructed, and, for $j \leq i$, every $\mathcal{C}_j$ is maximal with respect to $\leq_{a_j}$, and $\bigcap_{j \leq i} \mathcal{C}_j \neq \emptyset$ holds. Choose a node a_{i+1} from $\bigcap_{j \leq i} \mathcal{C}_j$ and a 1-complex a_{i+1}-component $\mathcal{C}_{i+1}$ that is maximal with respect to $\leq_{a_{i+1}}$. Due to Lemma 8.8, it intersects all $\mathcal{C}_j$, for $j \leq i$ (otherwise $\operatorname{ent}(\mathcal{G}) > 2$). By Lemma 8.10, $a_i \in \mathcal{C}_j$, for all $i \neq j$. Thus, according to Lemma 8.9, $\bigcap_{j \leq i+1} \mathcal{C}_j \neq \emptyset$ (or otherwise $\operatorname{ent}(\mathcal{G}) > 2$), and we can continue the construction.

Note that for all i, $a_i \notin \mathcal{C}_i$. Finally, for some $m < |V|$, there is no corresponding 1-complex (a_{m+1})-component for a_{m+1} and the construction stops. This means that all a_{m+1}-components are 1-simple, which contradicts our assumption that for every node a there is a 1-complex a-component. Otherwise there is a 1-complex a_{m+1} component $\mathcal{C}_{m+1}$, but $\bigcap_{i=0}^{m+1} \mathcal{C}_i = \emptyset$. In this case we have $\operatorname{ent}(\mathcal{G}) > 2$, according to Lemma 8.9. $\square$

It is clear that the entanglement of a graph is at most two if, and only if, the entanglement of all its strongly connected components is at most two, so we have the following corollary.

Corollary 8.12. *Let $\mathcal{G}$ be a graph. In $\mathrm{EG}_2(\mathcal{G})$, the cops have a winning strategy if, and only if, in every strongly connected component $\mathcal{C}$ of $\mathcal{G}$, there exists a node $a \in \mathcal{C}$, such that every a-component of $\mathcal{C}$ is* 1*-simple.*

Note that the above fails for graphs of entanglement three or greater, as proven in Section 8.7.

8.2. Decompositions for entanglement two

The proof of Theorem 8.11 shows the structure of a strongly connected graph $\mathcal{G}$ of entanglement two. It has a node a_0 such that the graph $\mathcal{G} \setminus a_0$ can be decomposed in 1-simple a_0-components. We can divide them into two classes: *leaf* components, from which one cop expels the robber, and *inner* components, where one cop does not win, but blocks all exit nodes making the other cop free from guarding the simple component. It turns out that every inner component $\mathcal{C}_0$ again has a node a_1 such that $\mathcal{C}_0$ decomposes in 1-simple a_1-components an so on. We shall show that a_1 is the node where the second cop stays (blocking all exit nodes of $\mathcal{C}_0$) when the first cop leaves a_0. Let us define the decomposition for graphs of entanglement two.

Definition 8.13. An *entanglement two decomposition* of a strongly connected graph $\mathcal{G} = (V_G, E_G)$ is a triple $(\mathcal{T}, F, g)$, where $\mathcal{T}$ is a nontrivial directed tree $\mathcal{T} = (T, E)$ with root r and edges directed away from the root, and F and g are functions $F : T \to 2^{V_G}$ and $g : T \to V_G$ with the following properties:

(1) $F(r) = V_G$,
(2) $g(v) \in F(v)$ for all $v \in T$,
(3) if $(v, w_1) \in E$ and $(v, w_2) \in E$, then $F(w_1) \cap F(w_2) = \emptyset$, for $w_1 \neq w_2$,
(4) for $(v, w) \in E$, $\mathcal{G}[F(w)]$ is a strongly connected component of $\mathcal{G}[F(v)] \setminus g(v)$,
(5) the subgraph of $\mathcal{G}$ induced by the node set $\Big(F(v) \setminus g(v)\Big) \setminus \left(\bigcup_{w \in vE} F(w)\right)$ is acyclic for all $v \in T$,
(6) no node in $\mathrm{Ex}(\mathcal{G}, \mathcal{G}[F(v)])$ is reachable from $\mathcal{G}[\bigcup_{w \in vE} F(w)]$ in $\mathcal{G} \setminus g(v)$, for all $v \in T$.

We shall call tree nodes and (abusing the notation) their F-images *bags* and g-images *decomposition points*.

Note that from the definition follows that if $(v, w) \in E$ then $F(w) \subsetneq F(v)$, and that if $v \in T$ is a leaf in $\mathcal{T}$ then $\mathcal{G}[F(v)] \setminus g(v)$ is acyclic. Observe further that successors of a bag are partially ordered in the sense that, for each bag v, its successors $vE = \{w_1, \ldots, w_m\}$ form a DAG $\mathcal{D} = (vE, E_D)$ such that, for all $w_i, w_j \in vE$, w_j is reachable from w_i in $\mathcal{D}$ if, and only if, $F(w_j)$ is reachable from $F(w_i)$ in $\mathcal{G}[F(v)] \setminus g(v)$. An example of a graph and its entanglement two decomposition is given in Figure 5.

We look again at the class of trees with back-edges defined in Section 3. Let us look at decompositions of members of graph classes defined at the beginning of this section. The decomposition tree of a tree with back-edges $\mathcal{T} = (T, E_T, E_{back})$ can be given as $(T', E'_T, F, \mathrm{id}_{T'})$ where T' is T without leaves, E'_T is

$$\{(v, w) \mid (w, v) \in E_T \text{ and } v \text{ is not a leaf in } \mathcal{T}\},$$

and if $v \in T'$ then $F(v)$ is the subtree rooted at v and $g(v) = v$. It is easy to verify that $(T', E'_T, F, \mathrm{id}_{T'})$ is an entanglement two decomposition of $\mathcal{T}$.

8.3. Characterisations of graphs of entanglement two

Having defined the decomposition for entanglement two, we are ready to state our two main results characterising directed graphs of entanglement two.

Theorem 8.14. *A strongly connected graph $\mathcal{G} = (V, E)$ has entanglement at most two if, and only if, $\mathcal{G}$ has an entanglement two decomposition.*

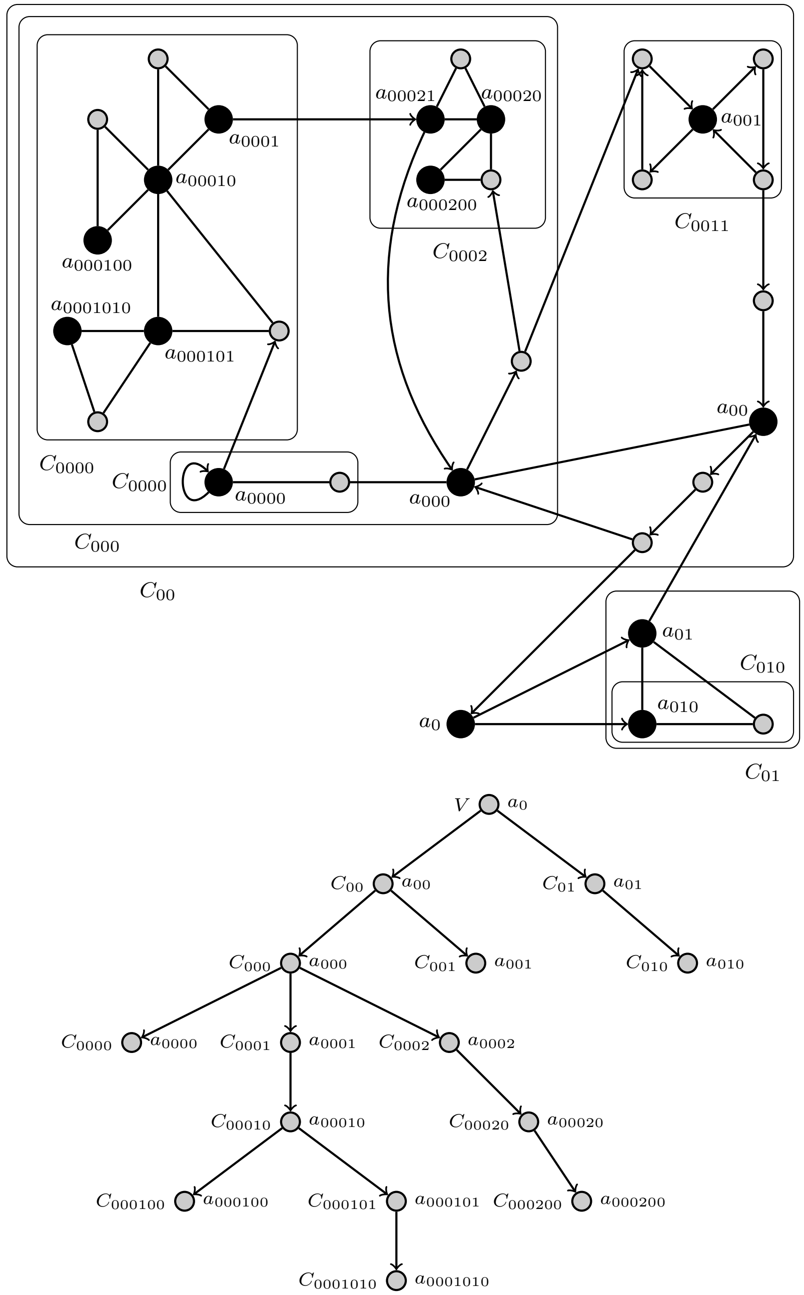

Figure 5: A typical graph of entanglement two and its entanglement two decomposition. On the upper picture, the components (images of function F) are shown as squares (only up to level 4), blocking nodes (images of function g) are shown as filled circles. On the picture below, the decomposition tree of the graph is given. The bags are labelled with images from functions F and g.

The above theorem, which we will prove in the subsequent subsections, allows us to complete the characterisation of directed graphs of entanglement two. Observe first, that there is a connection between the entanglement two decomposition and the characterisations of undirected graphs of entanglement two given by Belkhir and Santocanale [2]. They prove that an undirected graph has entanglement at most two if, and only if, each of its connected components is a tree where every edge $\{v, w\}$ may be replaced or extended by some nodes $v_1, \ldots, v_n$ with edges $\{v, v_i\}$ and $\{v_i, w\}$ for all $i = 1, \ldots, n$.

For an entanglement two decomposition of an undirected graph $\mathcal{G} = (V, E)$, consider a connected component, which is an undirected tree $\mathcal{T} = (V_T, E_T)$ with additional nodes as above. Choose an arbitrary leaf $v \in V_T$ as a root. We get a decomposition tree after orienting all edges from E_T (if an edge was deleted, restore it before orienting) away from the root and deleting all leaves other than v. We define the functions F and g as follows: $F(v)$ is V_T and $g(v)$ is v. In general, if, for a bag w, the functions F and g on w are already defined, let $\mathcal{C}$ be a strongly connected component of $\mathcal{G}[F(w)] \setminus g(w)$. Choose a node u in $\mathcal{C}$ with an edge between w and u and set $F(u) = \mathcal{C}$ and $g(u) = u$.

Recall the definition of the class $\mathcal{F}'$ at the beginning of this section for the following theorem.

Theorem 8.15. *A strongly connected directed graph $\mathcal{G}$ has entanglement at most two if, and only if, $\mathcal{G} \in \mathcal{F}'$.*

Proof. Let $\mathcal{G} = (V, E)$ be a strongly connected directed graph of entanglement at most two. We prove that $\mathcal{G}$ can be constructed using operations (1)–(4), (5′) from the definition of the class $\mathcal{F}'$. Let $\mathcal{T} = (T, E_T, F, g)$ be an entanglement two decomposition of $\mathcal{G}$. We prove by induction on the structure of $\mathcal{T}$ in a bottom-up manner that one can construct all successor bags $F(w_1), \ldots, F(w_m)$ of a bag v such that, for all $i = 1, \ldots, m$, the marked nodes of $F(w_i)$ include $g(w_i)$ and all nodes that are not reachable in $\mathcal{G}[F(v)] \setminus g(v)$ from a bag $F(w_i)$.

A leaf bag $F(v)$ becomes acyclic when the node $g(v)$ is deleted. First, we construct $\mathcal{G}[F(v)] \setminus g(v)$ such that all nodes are marked, which is possible with the operations (1)–(4). Then we apply rule (5′) adding node $g(v)$ such that the whole bag $F(v)$ is marked. This marking is possible as $\mathcal{G}[F(v)] \setminus g(v)$ is acyclic.

Having constructed all bags $F(w_1), \ldots, F(w_m)$ with marked nodes as in the induction hypothesis described above, we construct the bag $F(v)$. Let $vE_T = \{w_1, \ldots, w_m\}$. Note that $F(v)$ consists of $g(v)$, all bags $F(w_i)$ of the next lower level, and nodes of $F(v) \setminus g(v)$ not reachable from a bag $F(w_i)$ within $\mathcal{G}[F(v)] \setminus g(v)$. We denote the latter nodes by A and the induced subgraph $\mathcal{G}[A]$ by $\mathcal{A}$. Our aim is to construct $\mathcal{G}[F(v)]$ such that marked nodes are precisely $g(v)$ and the nodes of A. We first construct $\mathcal{A}$ using rules (1)–(4) such that all nodes of $\mathcal{A}$ are marked. Then we apply rule (3) to get the disjoint union of $\mathcal{A}$ and bags $\mathcal{G}[F(w_i)]$. If there are edges from $\mathcal{A}$ to a bag $F(w_i)$ we add these with rule (4), which is possible because all nodes in $\mathcal{A}$ are marked. Now we use rule (5′) to add node $g(v)$ and the edges (that exist in $\mathcal{G}$) between $g(v)$, and $F(w_i)$ and $\mathcal{A}$. We show that this is possible. There can be edges in $\mathcal{G}$ of the following kinds:

- *From* $\mathrm{Ex}(\mathcal{G}, \mathcal{G}[F(w_i)])$ *to* $g(v)$. We can add these, as nodes of $\mathrm{Ex}(\mathcal{G}, \mathcal{G}[F(w_i)])$ are not reachable from $\bigcup_{i=1}^{m} F(w_i)$ in $\mathcal{G}[F(v)] \setminus g(v)$ (due to property (6) of the entanglement two decomposition) and thus are contained in $\mathcal{A}$. But A is marked by induction hypothesis.

- *From* $\mathcal{A}$ *to* $g(v)$. We can add these edges because $\mathcal{A}$ is marked.

- *From* $g(v)$ *to any node in* $F(v)$. This is possible due to rule $(5')$.

There are no other edges in $\mathcal{G}$ between $g(v)$, $\mathcal{A}$ and $F(w_i)$ because of the definition of $\mathrm{Ex}(\mathcal{G}, \mathcal{G}[F(w_i)]))$. It remains to define marked nodes in $F(v)$. Node $g(v)$ is marked (rule $(5')$) as needed for induction hypothesis. We also let nodes in A remain marked. (This is needed because these can be exit nodes of $\mathcal{G}[F(v)]$ in $\mathcal{G}$.) Note that A is not reachable from a bag $F(w_i)$ in $\mathcal{G}[F(v)] \setminus g(v)$, so these nodes must be marked as well.

For the other direction, assume that $\mathcal{G} = (V, E)$ is strongly connected and in $\mathcal{F}'$. Note that during the construction of $\mathcal{G}$ we get a sequence of graphs with marked nodes. We show by induction on the construction of $\mathcal{G}$ according to rules (1)–(4), $(5')$ that the cops have a winning strategy in the game $\mathrm{EG}_2^*(V, E, F)$ where F is the set of marked nodes of $\mathcal{G}$. The graph consisting of one node and without edges (arising after the application of rule (1)) has entanglement zero. Applications of rules (2)–(4) do not increase entanglement because they do not introduce new cycles. Assume that two cops have a winning strategy σ on a graph $\mathcal{G}' = (V', E', F')$ with marked nodes F'. Let $\mathcal{G}''$ be the graph we get from $\mathcal{G}'$ after adding a new node v via rule $(5')$. We give a winning strategy for the cops on $\mathcal{G}''$. First, they play according to σ on $\mathcal{G}'$ thus capturing the robber or expelling her to v. When she visits v one cop follows her there. The robber runs to a strongly connected component of $\mathcal{G}'$. The cops play again according to σ using the other cop (who is not on v) first and letting the cop on v guard $\mathcal{G}'$. When σ prescribes to use the second cop in $\mathcal{G}'$ the robber cannot escape from $\mathcal{G}'$ any more (because σ is a winning strategy for the cops in $\mathrm{EG}^*(\mathcal{G}')$). So the cops capture the robber in $\mathcal{G}'$ and thus also in $\mathcal{G}''$. □

8.4. A characterisation of 1*-complex components*

Lemma 8.16. *Let $\mathcal{G} = (V, E, F)$ be a strongly connected graph with exit nodes. If, for all $v \in V$, there is a cycle $\mathcal{C}$ in $\mathcal{G} \setminus v$ from that a node in F is reachable in $\mathcal{G} \setminus v$, then $\mathcal{G}$ is* 1*-complex.*

Proof. Let $C(v)$ be a cycle in $\mathcal{G} \setminus v$ from which a node in F is reachable in $\mathcal{G} \setminus v$. Let $\mathcal{C}$ be any cycle in $\mathcal{G}$. The following strategy ρ is winning for the robber in $\mathrm{EG}_2^*(\mathcal{G})$.

- start on an arbitrary node in $\mathcal{C}$;

- $\rho(v, \emptyset)$ prescribes the robber to stay in $\mathcal{C}$;

- $\rho(v, \{w\})$ prescribes to run to a node in the cycle $C(w)$ if $v \notin C(w)$;

- $\rho(v, \{w\})$ prescribes to stay in the cycle $C(w)$ if $v \in C(w)$;

- $\rho(v, \{w, u\})$ prescribes to run to an exit node (and thus win).

By the assumption, in a position $(v, \{w, v\})$ there is a cop free path (possibly except the cop on v) to an exit node, so ρ is indeed winning for the robber. $\square$

Let $\mathcal{G}$ be a graph with exit nodes. We call a node $v \in \mathcal{G}$ a *blocking node*, if there is no strongly connected component of $\mathcal{G} \setminus v$ from which there is a path to an exit node in $\mathcal{G} \setminus v$. We denote the set of blocking nodes $B(\mathcal{G})$ and define a binary relation $\to$ on $B(\mathcal{G})$:

$$v \to w \ \text{ if, and only if, } \ w \text{ is not on a cycle in } \mathcal{G} \setminus v.$$

Lemma 8.17. *If $\mathcal{G} = (V, E, F)$ is a 1-simple graph with exit nodes then the relation $\to$ on $B(\mathcal{G})$ is a total preorder, i.e., it is transitive and total.*

Proof. For transitivity, let $u, v, w \in B(\mathcal{G})$ and assume that it is $u \to v$ and $v \to w$. Then all cycles with w contain v and all cycles with v contain u. It follows that all cycles with w contain u and w is not on a cycle in $\mathcal{G} \setminus u$.

It remains to show the totality of $\to$. Because the reflexivity is trivial, let v and w be distinct nodes in $B(\mathcal{G})$. Assume that neither $v \to w$ nor $w \to v$ holds, i.e., w is on a cycle $\mathcal{C}_v$ in $\mathcal{C} \setminus v$ and v is on a cycle $\mathcal{C}_w$ in $\mathcal{C} \setminus w$. Further, every path from $\mathcal{C}_v$ to an exit node leads through v, because v is blocking, and there is such a path, because $\mathcal{G}$ is strongly connected. Consider the part of this path from v to an exit node. Together with $\mathcal{C}_w$ it witnesses that w is not blocking, in contradiction to the choice of w. $\square$

Note that $\to$ is not necessarily antisymmetric, so we define the symmetrisation $\sim$ of $\to$ on $B(\mathcal{G})$ and extend the relation $\to$ on $B(\mathcal{G})/_{\sim}$. Let $[v]$ denote the equivalence class of v with respect to $\sim$. The binary relation $\to_\sim$ is well defined by

$$[v] \to_\sim [w] \Leftrightarrow v \to w.$$

The transitivity and the totality are inherited by $\to_\sim$ from $\to$, the antisymmetry is guaranteed by including all not antisymmetric pairs of elements into the same class, thus the following holds.

Lemma 8.18. *If $\mathcal{G}$ is a 1-simple graph with exit nodes then the relation $\to_\sim$ on $B(\mathcal{G})$ is a total order on $B(\mathcal{G})/_{\sim}$.*

If, for nodes v and w in a graph with exit nodes $\mathcal{G}$, $v \to w$ holds then we say that node v *blocks* node w. The next lemma follows from the previous one.

Lemma 8.19. *If $\mathcal{G} = (V, E, F)$ is a 1-simple graph with exit nodes such that (V, E) has entanglement two then there is a node $v \in \mathcal{G}$ that blocks all nodes from $B(\mathcal{G})$.*

Theorem 8.14 *A strongly connected graph $\mathcal{G} = (V, E)$ has entanglement at most two if, and only if, $\mathcal{G}$ has an entanglement two decomposition.*

Proof.
($\Rightarrow$) For a graph $\mathcal{G}$ with $\mathrm{ent}(\mathcal{G}) = 2$, we construct the tree $\mathcal{T} = (T, E_T)$ and the functions F and g in a top-down manner. In each step we enlarge the tree adding to a bag v that is currently a leaf some successors $\{w_1, \dots, w_m\}$ and define the functions F and g on them. We require that all $g(w_i)$-components of $\mathcal{G}[F(w_i)]$ are 1-simple.

To start with, by Theorem 8.11 there exists a node $a_0 \in V$ such that all a_0-components of $\mathcal{G}$ are 1-simple. For the root r of the tree $\mathcal{T}$ we set $F(r) = V$ and $g(r) = a_0$. In general, for every bag v that is a leaf of the already constructed part of the tree, let $C_1, \dots, C_m$ induce all strongly connected components of $F(v) \setminus g(v)$. If there are no such components (i.e., $m = 0$), skip this bag and proceed with a next one, if there is any. If $m \geq 1$, create, for each $i \in \{1, \dots, m\}$, a successor w_i of v and set $F(w_i) = C_i$. From the construction we know that each C_i induces a 1-simple $g(v)$-component. If it has a node a whose removal makes the component acyclic, i.e., the cops win $\mathrm{EG}_1(\mathcal{G}[C_i])$, then set $g(w_i) = a$. If the cops lose $\mathrm{EG}_1(\mathcal{G}[C_i])$ then, according to the definition of a 1-simple component, one cop can block all exit nodes (to win with help of the other cop), i.e., he can place himself on a blocking node of $\mathcal{G}[C_i]$. Among all blocking nodes there is a node a that blocks all nodes in $B(\mathcal{G}[C_i])$, due to Lemma 8.19. Set $g(v) = a$. Then all a-components of $\mathcal{G}[F(w_i)]$ are 1-simple. We check that all requirements of the entanglement two decomposition are fulfilled. The first four properties follow immediately from the construction. Let $vE_T = \{w_1, \dots, w_m\}$. Then the subgraph of $\mathcal{G}$ induced by the node set $\Big(F(v) \setminus g(v)\Big) \setminus \left(\bigcup_{i=1}^{m} F(w_i)\right)$ is acyclic because a cycle would induce a new strongly connected component, but $\bigcup_{i=1}^{m} F(w_i)$ includes all components of $F(v)$. Finally assume that a node $w \in \mathrm{Ex}(\mathcal{G}, \mathcal{G}[F(v)])$ is reachable from a node $u \in F(w_i)$ for some $w_i \in \{w_1, \dots, w_m\}$. Then $F(w_i)$ is a strongly connected component of $\mathcal{G}[F(v)] \setminus g(v)$ and $g(v)$ is not blocking in $\mathcal{G}[F(v)]$, but we chose it to be blocking.

($\Leftarrow$) We show that an entanglement two decomposition induces a winning strategy for two cops on $\mathcal{G}$. Observe that if a cop is on a node $g(v)$, for a bag v, and the robber is in a bag on a lower level of the tree, then the cop blocks the robber in the bags under v. Consider a node a with the robber on it. Let v be the bag with the smallest F-image (it is the lowest in the tree) among all with $a \in F(v)$ and let $vE_T = \{w_1, \dots, w_m\}$, for $m \geq 0$ (if $m = 0$ then vE_T is empty). The cops wait for the robber to enter a component $\mathcal{G}[F(w_i)]$ or to go to $g(v)$. In the first case, they play according to the same strategy with w_i instead of v. This descending along the tree is finite and on some level (w.l.o.g. already on that where v is) the robber visits $g(v)$. One cop goes there. If the robber proceeds to a component $\mathcal{G}[F(w_i)]$, the second cop continues to chase her using the same strategy. If she leaves $F(v)$ and enters a brother bag v' of v, the cop from v follows her there and so on until the robber is forced to go to $g(u)$, where u is the predecessor of v. The first cop goes to $g(u)$ as well and chases the robber in this manner upwards. This process is finite and when the robber goes downwards, the second cop plays the described strategy with the difference that the robber cannot climb so high as before. Continuing in this way the cops finally capture the robber. □

Observe that it follows that, in time $O(n^3)$, where n is the size of the input graph $\mathcal{G}$, one can not only decide whether $\mathcal{G}$ has entanglement at most two, but also compute an entanglement two decomposition of $\mathcal{G}$. The algorithm proceeds by first looking for the node a_0 by linear search. Then the a_0-components are computed. In every component the algorithm finds a node a_1 that blocks all blocking nodes of that component. If there is no such a_1, the algorithm returns "robber wins". Otherwise the procedure continues with the node a_1 instead of a_0 until there is no a_i-component for some i (i.e., the a_{i-1}-component is of entanglement one). In this case the algorithm returns "Cops win" and the computed decomposition.

8.6. DAG-width and Kelly-width for entanglement two

Entanglement two decomposition of a graph leads to winning strategies for three cops in games that correspond to DAG-width and to Kelly-width. The games characterising DAG-width and Kelly-width were discussed in Section 7.

Proposition 8.20. *For any graph $\mathcal{G}$, if $\mathrm{ent}(\mathcal{G}) \le 2$, then the DAG-width and the Kelly-width of $\mathcal{G}$ are at most 3.*

Proof. We first use the entanglement two decomposition to describe a winning strategy for the cops in the cops and visible robber game on graphs of entanglement two and then adjust this strategy to the cops and invisible inert robber game. We can assume that $\mathcal{G}$ is not acyclic. Consider an entanglement two decomposition $(\mathcal{T}, F, g)$ of $\mathcal{G}$. In the cops and visible robber game, a cop places himself on the g-image of the root of $\mathcal{T}$ at the beginning of a play. In general, assume that, for a bag v, a cop is on a blocking node $g(v)$ and the robber is on a node in $F(w)$, for a successor bag w of v. The component $F(w)$ has also a blocking node $g(w)$. A cop who is not on $g(v)$ goes to $g(w)$ and the third cop visits every node in $F(w)$ that is not in a strongly connected component of $F(w)$. Thus the robber is forced to move down the decomposition tree and finally loses.

The strategy of the cops in the cops and invisible inert robber game is similar. Assume that a cop is on a blocking node $g(v)$. The cops do not know where the robber is, so they decontaminate a strongly connected component of $F(v) \setminus g(v)$ as described for the visible robber game, move a cop back on node $g(v)$ and continue with the next strongly connected component. Note that both winning strategies are monotone. □

Proposition 8.20 gives the best possible upper bound for the number of cops needed to capture the robber in the same graph in the invisible inert robber game. Note that the third cop in the visible robber game and the invisible inert robber game is used to force the robber to move. Figure 6 shows a graph of entanglement two and both DAG-width and Kelly-width three, which is easy to verify.

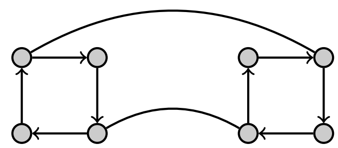

Figure 6: A graph of entanglement two, and DAG-width and Kelly-width three.

8.7. Failure of a generalisation to entanglement k

We give counterexamples to a generalisation of Corollary 8.12 to arbitrary number of cops. We show that, for every $k > 2$, there is a graph $\mathcal{G}_k$ of entanglement k in that, for every node a, there is a $(k-1)$-complex a-component. In Figure 7 such a graph is given. As the case for $k = 3$ is not obvious, a counterexample graph of entanglement three is given as well (Figure 8). Circles circumscribe parts of the graph. An arrow leading to (from) a circle denotes edges to (from) all nodes in the circle. Lines without arrows denote edges in both directions. For $m > 2$, C_m denotes an m-clique.

We show first that for nodes a_0, a_1 and a_2 there are $(k-1)$-complex components giving corresponding strategies of the robber. Note that, for all of them, the existence of a cop free path to an exit node of the component is an invariant. The a_0-component $\mathcal{C}_0$ is induced by nodes from T, U, B and the node a_2. The $\mathcal{C}_0$-strategy of the robber is to wait in U until $k-1$ cops come to U, then proceed to B and wait there for $k-1$ cops to come and so on. On the other hand, k cops can expel the robber from $\mathcal{C}_0$.

The a_1-component $\mathcal{C}_1$ is induced by a_0, a_2 and nodes of L, R, S, and F. The $\mathcal{C}_1$-strategy does not use nodes of L. The robber waits in S and R (which build a k-clique) for $k-1$ cops to come and then goes to F. Three of the cops from $S \cup R$ are needed to expel her from there. Thus a path back to $S \cup R$ becomes free for the robber and she plays further as in the beginning.

The a_2-component $\mathcal{C}_2$ is induced by a_0, T, L, R and S whereby R is not used by the robber. The $\mathcal{C}_2$-strategy is analogous to the $\mathcal{C}_1$-strategy. One can see that one of the three given strategies can be used to show that, in fact, every node a of the graph has a $(k-1)$-complex a-component.

Still, the entanglement of the graph is k. The cops have the following winning strategy in the entanglement game. One cop is placed on node a_2 and the robber is expelled from the component $\mathcal{C}_0$ defined above. If the robber visits U or F, she is captured, because a_2 is blocked by a cop. Then $k-3$ other cops occupy nodes of S. If the robber goes to R or to T, the last two cops force her to leave it, so she visits the node b. One of those two cops goes to b and the other one expels the robber from L and follows her to a_1. The robber must remain in T.

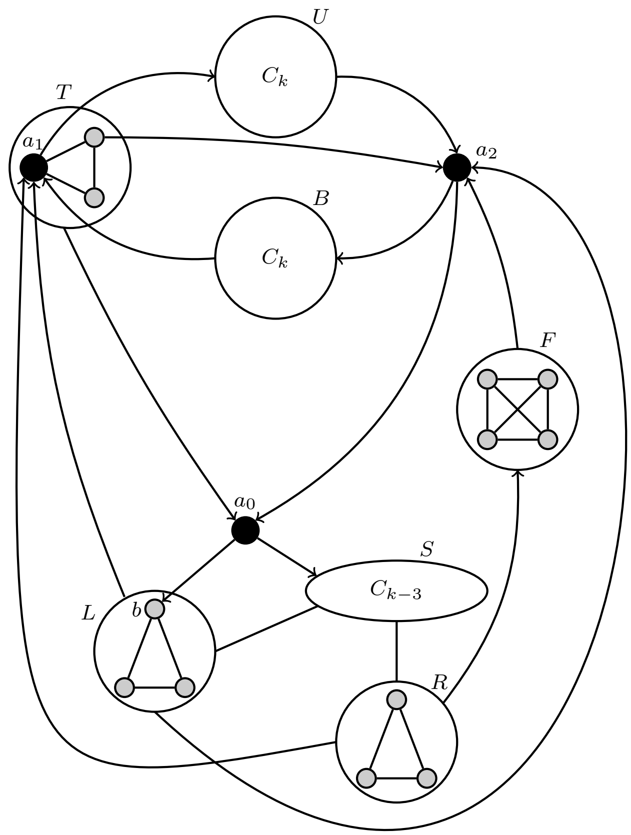

Figure 7: A graph of entanglement k with only $(k-1)$-complex components.

In this game position, one cop is on a_2, one on a_1, one on b and $k-3$ cops occupy S. At this time, the k-th cop moves from a_2 into the a_2-component $\mathcal{C}_2$ allowing the robber to leave it. The entanglement game in $\mathcal{C}_2$ with exit nodes $\mathrm{Ex}(\mathcal{G}, \mathcal{C}_2)$ would be lost by the cops, but they win the game on the whole graph. The cop from a_2 expels the robber from T. As a_0 is a dead end for her, she proceeds to a_2 and then to B. Then all cops except the one on a_1 capture her in B.

Acknowledgements.. This work has been partially supported by the ESF Networking Programme GAMES.

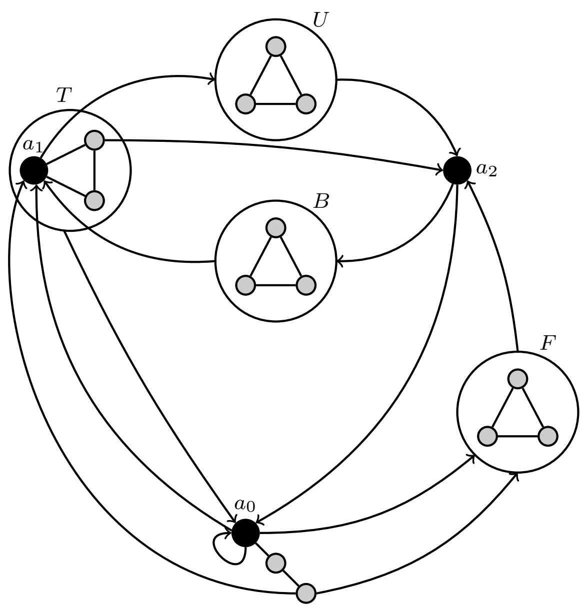

Figure 8: A graph of entanglement 3 with only 2-complex components.

References

[1] J. L. Balcazar, J. Diaz, J. Gabarro, Structural complexity 1, Springer, 1988.

[2] W. Belkhir, L. Santocanale, Undirected Graphs of Entanglement 2, in: Proceedings of FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, vol. 4855 of *LNCS*, Springer, 508–519, 2007.

[3] D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, DAG-width and parity games, in: Proceedings of STACS 2006: Symposium on Theoretical Aspects of Computer Science, vol. 3884 of *LNCS*, Springer, 524–536, 2006.

[4] D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, J. Obdrzálek, The DAG-width of directed graphs, J. Comb. Theory, Ser. B 102 (4) (2012) 900–923.

[5] D. Berwanger, E. Grädel, Entanglement – A Measure for the Complexity of Directed Graphs with Applications to Logic and Games, in: Proceedings of LPAR 2004: Logic for Programming and Automated Reasoning, vol. 3452 of *LNCS*, Springer, 209–223, 2005.

[6] D. Berwanger, E. Grädel, G. Lenzi, On the variable hierarchy of the modal mu-calculus, in: Proceedings of CSL 2002: Computer Science Logic, vol. 2471 of *LNCS*, Springer, 352–366, 2002.

[7] D. Berwanger, E. Grädel, G. Lenzi, The variable hierarchy of the μ-calculus is strict, Theory of Computing Systems 40 (2007) 437–466.

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

[9] A. K. Chandra, D. Kozen, L. J. Stockmeyer, Alternation, J. ACM 28 (1) (1981) 114–133.

[10] A. Dawar, E. Grädel, The Descriptive Complexity of Parity Games, in: Proceedings of CSL 2008: Computer Science Logic, vol. 5213 of *LNCS*, Springer, 354–368, 2008.

[11] L. C. Eggan, Transition graphs and the star-height of regular events, Michigan Math. J. 10 (1963) 385–397, ISSN 0026-2285.

[12] A. Emerson, C. Jutla, Tree automata, mu-calculus and determinacy, in: Proceedings of FOCS 1991: Foundations of Computer Science, IEEE, 368–377, 1991.

[13] E. Grädel, L. Kaiser, R. Rabinovich, Directed Graphs of Entanglement Two, in: Proceedings of FCT '09, vol. 5699 of *LNCS*, Springer, 169–181, 2009.

[14] P. Hunter, Complexity and Infinite Games on Finite Graphs, Ph.D. thesis, University of Cambridge, 2007.

[15] P. Hunter, S. Kreutzer, Digraph measures: Kelly decompositions, games, and orderings, in: Proceedings of SODA 2007: Symposium on Discrete Algorithms, SIAM, 637–644, 2007.

[16] P. Hunter, S. Kreutzer, Digraph measures: Kelly decompositions, games, and orderings, Theoretical Computer Science 399 (3) (2008) 206–219.

[17] T. Johnson, N. Robertson, P. D. Seymour, R. Thomas, Directed Tree-Width, Journal of Combinatorial Theory, Series B 82 (1) (2001) 138–154.

[18] T. Johnson, N. Robertson, P. D. Seymour, R. Thomas, Directed tree-width, Journal of Combinatorial Theory, Series B 82 (1) (2001) 138–154.

[19] N. D. Jones, Space-Bounded Reducibility among Combinatorial Problems, J. Comput. Syst. Sci. 11 (1) (1975) 68–85.

[20] M. Jurdziński, Small Progress Measures for Solving Parity Games, in: Proceedings of STACS 2000: Symposium on Theoretical Aspects of Computer Science, vol. 1770 of *LNCS*, Springer, 290–301, 2000.

[21] D. Kozen, Results on the propositional μ-calculus, Theoretical Computer Science 27 (1983) 333–354.

[22] S. Kreutzer, S. Ordyniak, Digraph Decompositions and Monotonicity in Digraph Searching, in: Proceedings of WG 2008: Graph-Theoretic Concepts in Computer Science, revised papers, Springer, 336–347, 2008.

[23] J. Obdrzálek, Fast Mu-calculus Model Checking when Tree-width is Bounded, in: Proceedings of CAV 2003: Computer-Aided Verification, vol. 2725 of *LNCS*, Springer, 80–92, 2003.

[24] J. Obdrzálek, DAG-width: connectivity measure for directed graphs, in: Proceedings of SODA 2006: Symposium on Discrete Algorithms, SIAM, 814–821, 2006.

[25] R. Rabinovich, Complexity Measures of Directed Graphs, Diploma thesis, RWTH Aachen University, 2008.

[26] N. Robertson, P. D. Seymour, Graph minors. I. Excluding a forest, J. Comb. Theory, Ser. B 35 (1) (1983) 39–61.

[27] N. Robertson, P. D. Seymour, Graph minors. III. Planar tree-width, J. Combin. Theory Ser. B 36 (1) (1984) 49–63.

[28] M. A. Safari, D-width, metric embedding, and their connections, Ph.D. thesis, University of British Columbia, Canada, 2007.

[29] P. D. Seymour, R. Thomas, Graph searching and a min-max theorem for tree-width, Journal of Combinatorial Theory, Series B 58 (1).

[30] C. Stirling, Bisimulation, Modal Logic and Model Checking Games, Logic Journal of the IGPL 7 (1) (1999) 103–124.

[31] W. Zielonka, Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees, Theoretical Computer Science 200 (1998) 135–183.