Hypertree-Depth and Minors in Hypergraphs

Isolde Adler[a], Tomáš Gavenčiak[b,1], Tereza Klimošová[c,1]

[a]Goethe University, Frankfurt am Main
[b]Department of Applied Mathematics, Charles University, Prague
[c]Computer Science Institute, Charles University, Prague

Abstract

We introduce two new notions for hypergraphs, *hypertree-depth* and *minors in hypergraphs*. We characterise hypergraphs of bounded hypertree-depth by the ultramonotone robber and marshals game, by *vertex-hyperrankings* and by *centred hypercolourings*.

Furthermore, we show that minors in hypergraphs are 'well-behaved' with respect to hypertree-depth and other hypergraph invariants, such as generalised hypertree-depth and generalised hyperpath-width.

We work in the framework of *hypergraph pairs* (G, H), consisting of a graph G and a hypergraph H that share the same vertex set. This general framework allows us to obtain hypergraph minors, graph minors and induced graph minors as special cases.

Keywords: hypertree-depth, robber and marshals game, minors, tree-depth, hypertree-width, hyperpath-width

1. Introduction

The tree-depth of a graph is a parameter recently introduced by Nešetřil and Ossona de Mendez [11]. It has received much attention due to its connections to graph colouring and homomorphism dualities, and it plays a central role in the theory of graph classes of bounded expansion [12, 13, 14].

We introduce hypertree-depth as a parameter for hypergraphs and we investigate the structure of hypergraphs of bounded hypertree-depth. We define the *ultramonotone robber and marshals game*[2] and we show that the minimum number of steps necessary to catch the robber in this game characterises hypertree-depth. We present hypertree-depth in a setting that allows us to view it as a generalisation of graph tree-depth, and we give an equivalent characterisation using hypergraph colourings. Hypertree-depth is a hypergraph generalisation similar to generalised hypertree-width [8] and generalised hyperpath-width [10].

We propose a notion of *minors in hypergraphs*. The main desired properties of this notion are generalising induced subhypergraphs, monotonicity of the above hypergraph parameters (i.e. the parameters do not increase when taking minors), allowing hyperedge contractions, and the underlying graphs (primal graphs) to be minors in the traditional sense. We restrict the ways of deleting hyperedges, as removing arbitrary (large) hyperedges can make the hypergraph more complicated by 'uncovering' some complicated structure below (and hence break monotonicity for

Email addresses: `iadler@informatik.uni-frankfurt.de` (Isolde Adler), `gavento@kam.mff.cuni.cz` (Tomáš Gavenčiak), `tereza@iuuk.mff.cuni.cz` (Tereza Klimošová)

[1]The research was partially supported by Czech research grant GAUK 64110.

[2] The ultramonotone robber and marshals game was indeed already used by Adler as a tool for constructing hypergraph counterexamples [1].

the above parameters). In particular, we want that neither hypertree-depth nor generalised hypertree-width, nor generalised hyperpath-width can increase when taking minors. The notion of *hypergraph minors* introduced here has these properties.

We present the concepts and results both for hypergraphs and in the framework of *hypergraph pairs* (G, H), consisting of a graph G and a hypergraph H with $V(G) = V(H)$. This framework was introduced in [2] as a common generalisation of tree-width of graphs and hypertree-width of hypergraphs. Here it allows us to obtain hypergraph minors, graph minors and induced graph minors as special cases. We also propose this as a confirmation that our notion of tree-depth and minors in hypergraphs are indeed a robust generalisation.

After introducing the notation and basic definitions in Section 2, we introduce the notion of hypertree-depth and the ultramonotone robber and marshals game on hypergraphs in Section 3. Section 4 generalises these notions to the slightly more general framework of hypergraph pairs, and presents some basic properties together with an equivalence to the ultramonotone robber and marshals game. Section 5 provides several characterisations of hypertree-depth inspired by characterisations of tree-depth; namely by an inductive definition, by *vertex-hyperrankings*, and by *centred hypercolourings*. In Section 6 we introduce minors in hypergraphs and in hypergraph pairs and we discuss their basic properties. In Section 7 we put things together by showing how minors interact with generalised hypertree-width and other hypergraph invariants. Finally, we conclude with some open directions in Section 8.

2. Preliminaries

For a set A we denote the power set of A by 2^A. We denote the set of non-negative integers by ω. For $k \in \omega$ we let $[A]^k$ denote the set of all subsets of A of cardinality precisely k.

Our graphs are finite, simple and undirected. We use standard set and graph terminology as in [4]. We denote the vertex set of a graph G by $V(G)$ and the edge set by $E(G)$. The edges of G are two-element subsets of $V(G)$. A graph H is a *subgraph* of G, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For $X \subseteq V(G)$, the subgraph of G *induced* by X is the graph $G[X]$ with $V(G[X]) = X$ and $E(G[X]) = \{e \in E(G) \mid e \subseteq X\}$. A graph H is an *induced subgraph* of G, if $H = G[X]$ for some $X \subseteq V(G)$. For a graph G and a subset $X \subseteq V(G)$ we let $G \setminus X := G[V(G) \setminus X]$. For singleton sets $\{x\}$ we also write $G \setminus x$ and $V(G) \setminus x$ instead of $G \setminus \{x\}$ and $V(G) \setminus \{x\}$, respectively.

A *forest* is a graph without cycles. A *tree* is a connected forest. A *rooted* tree is a tree with a distinguished vertex, called the *root*. A *rooted forest* is a graph that is a disjoint union of rooted trees. Let F be a rooted forest. We use $\leq_F$ to denote the partial order on $V(F)$ induced by F (where the roots are the smallest elements). Let T be one of the rooted trees composing F. For $s, t \in V(T)$ we denote the *infimum* (with respect to $\leq_F$) of s and t by $s \wedge t$. A *chain* in F is a subset of $V(F)$, which is totally ordered by $\leq_F$. For $t \in V(F)$, by $\downarrow_F t$ we denote the chain of all $\leq_F$-predecessors of t in F (including t). We omit the subscript F of $\downarrow_F$, if it is clear from the context. The *height* of a node $t \in V(F)$ is the number of vertices in a path from the root (of the tree containing t) to t. The *height* of F is the maximum height of the vertices in F. For a rooted forest F and a vertex $t \in V(F)$ we let F_t be the subtree of F induced by $\{s \in V(F) \mid t \leq_F s\}$.

We will label the nodes of rooted forests by subsets of some given set. For this we introduce some notation. Let A be a set, let F be a rooted forest and let $f\colon V(F) \to 2^A$ be a mapping. For a subgraph $T \subseteq F$ we let $f(T) := \bigcup_{t \in V(T)} f(t)$. Similarly, for a set $X \subseteq V(F)$ we let $f(X) := \bigcup_{t \in X} f(t)$.

A *hypergraph* H consists of a set of vertices $V(H)$ and a set of *hyperedges* $E(H)$. The elements of $E(H)$ are non-empty subsets of $V(H)$. For a hypergraph H, let $\underline{H}$ denote the *underlying* graph (also called the *primal* graph) of H, i.e. $V(\underline{H}) = V(H)$, and $\{u, v\} \in E(\underline{H})$, whenever $u \neq v$ and there is a hyperedge $e \in E(H)$ with $\{u, v\} \subseteq e$. For a hyperedge $e \in E(H)$ let $H \setminus e = (V(H), E(H) \setminus e)$. We say that a hyperedge $e \in E(H)$ is a *subhyperedge*, if there is a hyperedge $e' \in E(H)$ such that $e \subseteq e'$. If $e \subsetneq e'$, e is a *proper subhyperedge*.

For a hypergraph H and $X \subseteq V(H)$, let $H[X]$ denote the *subhypergraph induced by* X, i.e. $H[X]$ has vertex set X and edge set $\{e \cap X \mid e \in E(H)\}$. H' is an *induced subhypergraph* of H, if $H' = H[X]$ for some $X \subseteq V(H)$. By $H \setminus X$ we denote the induced subhypergraph with vertex set $V(H) \setminus X$. *Paths* and *connectivity* of H are defined as in $\underline{H}$.

We recall the definition of *tree-width*, introduced by Robertson and Seymour [16]. A *tree decomposition* of a graph G is a pair (T, χ), consisting of a rooted tree T and a mapping $\chi \colon V(T) \to 2^{V(G)}$, such that for each $v \in V(G)$ there exists $t \in V(T)$ with $v \in \chi(t)$, for each edge $e \in E(G)$ there exists a node $t \in V(T)$ with $e \subseteq \chi(t)$, and for each $v \in V(G)$ the set $\{t \in V(T) \mid v \in \chi(t)\}$ is connected in T. The *width* of (T, χ) is defined as $\mathrm{w}(T, \chi) := \max\big\{ |\chi(t)| - 1 \ \big|\ t \in V(T)\big\}$. The *tree-width of* G is defined as

$$\mathrm{tw}(G) := \min\big\{ \mathrm{w}(T, \chi) \ \big|\ (T, \chi) \text{ is a tree decomposition of } G\big\}.$$

3. Hypertree-depth and ultramonotone marshals

The concept of hypertree-depth that we introduce in this section is a generalisation of tree-depth, a graph parameter introduced by Nešetřil et al. [11]. We recall the definition: Let F be a rooted forest. The *closure* of F, $\mathrm{clos}(F)$, is the comparability graph induced by $\leq_F$, i.e. $V(\mathrm{clos}(F)) = V(F)$ and $E(\mathrm{clos}(F)) = \big\{\{u, v\} \mid u \leq_F v,\ u \neq v\big\}$. The *tree-depth* of a graph G is the minimum height of a rooted forest F such that $G \subseteq \mathrm{clos}(F)$. Let $\mathrm{td}(G)$ denote the tree-depth of G.

Hypertree-depth is then defined as follows. Let H be a hypergraph. A *decomposition forest* of H is a tuple (F, C), where F is a rooted forest, and $C \colon V(F) \to E(H)$ is a mapping, satisfying the following conditions.

(dec1) For every vertex $v \in V(H)$ there is a node $t \in V(F)$ such that $v \in C(t)$.

(dec2) For every edge $e \in E(\underline{H})$ there are $\leq_F$-comparable nodes $s, t \in V(F)$ such that $e \subseteq C(s) \cup C(t)$.

(dec3) All $s, t \in V(F)$ satisfy: If $C(s) \cap C(t) \neq \emptyset$ then the node $s \wedge t$ exists and $C(s) \cap C(t) \subseteq \bigcup_{r \in \downarrow(s \wedge t)} C(r)$.

Definition 1 *The* hypertree-depth *of a hypergraph* H*,* $\mathrm{hd}(H)$*, is the minimum height of a rooted forest* F*, taken over all decomposition forests* (F, C) *of* H*.*

Note that if H has isolated vertices, then H does not have a decomposition forest. In this case $\mathrm{hd}(H) = \infty$.

It is easy to see that we can equivalently replace condition **(dec2)** by the following condition **(dec2′)**.

(dec2′) every hyperedge $e \in E(H)$ satisfies $e \subseteq \bigcup_{t \in Z} C(t)$ for some chain Z in F.

The trivial forest consisting of isolated vertices, each of them corresponding to one hyperedge of a given hypergraph satisfies **(dec1)** and **(dec2)**. Thus, condition

(dec3) might be viewed as the condition enforcing nontriviality of a decomposition forest and cannot be omitted. In particular, for a decomposition forest of a hypergraph H, condition **(dec3)** ensures that every tree of the decomposition forest corresponds to a union of connected components of $\underline{H}$, i.e. two hyperedges can be mapped to different trees of a decomposition forest only if their vertices are contained in two distinct connected components of $\underline{H}$.

In [7], Ganian et al. give a cop and robber game characterisation of tree-depth. We give an informal definition: In the game, the robber chooses and moves to a starting vertex. Then in every round, first the player controlling the cops selects and announces a vertex v, then the robber moves along any cop-free path to some vertex, and finally one cop is placed on v. The cops are never removed. The game ends when a cop occupies the same vertex as the robber. Then $\mathrm{td}(G) \leq k$ holds if and only if the cops have strategy to capture the robber in k rounds (after placing k cops) [7].

We propose an extension of the game to hypergraphs: The *ultramonotone*[3] robber and marshals game on a hypergraph H is a 2-player game. Player I ('the marshals') has an infinite supply of marshals at his disposal. Player II ('the robber') controls a robber. The marshals are placed on hyperedges of H and once they occupy a hyperedge, they stay there for the rest of the game. The robber moves on vertices of H. The game position is (r, M) where $r \in V(H)$ is the robber's position and $M \subseteq E(H)$ is the set of hyperedges occupied by the marshals.

In a play of the ultramonotone robber and marshals game, player II begins by choosing a vertex $r_0 \in V(H)$ and placing the robber on r_0. The initial position is $(r_0, \emptyset)$. Then, in every round of the game in position (r, M), Player I selects a hyperedge $e' \in E(H)$ and reveals it to Player II. Player II then selects a vertex $r' \in V(H)$ that is connected to r by a path in $H \setminus \bigcup M$. He moves the robber to r' and Player I places a new marshal on e'. The new position is $(r', M \cup \{e'\})$.

Player I wins if he reaches a position (r, M) with $r \in \bigcup M$. The *length* of a play is the number of rounds until the robber is captured. *Winning strategies* are defined in the usual way. A *winning strategy of length* $\leq k$ for the marshals is a winning strategy for the marshals such that every play played according to the strategy has length at most k.

This game characterises hypertree-depth. The following theorem follows directly from Lemma 8, Theorem 10 and Remark 4 below.

Theorem 2 *A hypergraph H has* $\mathrm{hd}(H) \leq k$ *if and only if the cops have a winning strategy for the ultramonotone robber and marshals' game on H of length at most k.*

[3]The word 'ultramonotone' refers to the fact that the ultramonotone robber and marshals game is 'monotone' in the sense of [8] (robber's area monotonically shrinks). Not removing the marshals is much stronger - and hence the term.

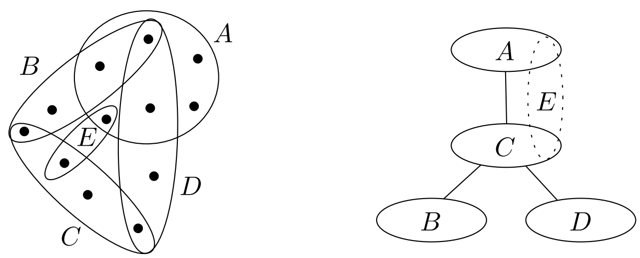


Figure 1: Example of a decomposition tree of a hypergraph

It might seem desirable for a hypergraph extension of tree-depth definition that when we view a graph G as a hypergraph, we would have $\mathrm{td}(G) = \mathrm{hd}(G)$. Our definition does not satisfy this (and neither do tree-width and generalised hypertree-width, nor path-width and hyperpath-width). The reason for this is that tree-depth counts the vertices in the decomposition, while the hypergraph measures hypertree-width, hyperpath-width and hypertree-depth count the number of edges covering certain vertex sets. This stems from the applications in database theory and constraint satisfaction problems, where tuples are counted rather that elements.

Note that in the hypergraph pair framework introduced in the next section, we can actually have $\mathrm{td}(G) = \mathrm{hd}(G, H)$ for H a hypergraph consisting of all singletons over $V(G)$ (cf. Remark 9).

However, $\mathrm{td}(G)$ and $\mathrm{hd}(G)$ are linearly bounded.

Lemma 3 *Any hypergraph H with edges of size at most k satisfies*

$$k\,\mathrm{hd}(H) \geq \mathrm{td}(\underline{H}) \geq \mathrm{hd}(H).$$

In particular, any graph G satisfies $2\,\mathrm{hd}(G) \geq \mathrm{td}(G) \geq \mathrm{hd}(G)$.

Proof. Let $G = \underline{H}$. For the first inequality, assume we have an optimal decomposition tree (F, C) of H. Define $f : V(G) \to V(F)$ by letting $f(v) = u$ with u the lowest node of F with $v \in C(u)$. This node is unique by **(dec3)**.

Now f may not be injective – up to k vertices may map to the same node (since edges of H have size at most k). We can resolve that by splitting every such node u into a path $u_1, \dots u_k$, making u_1 adjacent to the parent of u and u_k to all children of u. We also change f to map the vertices to $u_1, \dots u_k$ arbitrarily injectively. Let F' be the resulting forest and f' the resulting injective mapping.

We now have $G \subseteq \mathrm{clos}(F')$ as shown by f': let $e = xy \in E(G)$ and s, t be as in **(dec2)**. Then $f'(x) \leq s$ and $f'(y) \leq t$ and therefore $f'(x)$ and $f'(y)$ are $\leq_{F'}$-comparable. Obviously, the depth of F' is at most k times the depth of F.

For the second inequality, let F be an optimal forest with $G \subseteq \mathrm{clos}(F)$. Define $C : V(F) = V(G) \to E(G)$ with $C(u)$ an arbitrary edge of containing u. Then (F, C) trivially satisfies **(dec1)** and **(dec2)**. For **(dec3)**, have $s, t \in V(F)$ and $U = C(s) \cap C(T)$. Note that s and t are also vertices of G and $s \in C(s)$ and $t \in C(t)$. Now either s and t are comparable, in which case the required node is the lower of s, t. Or s and t are incomparable, in which case every $u \in U$ is adjacent to both s and t in $G = \underline{H}$, and therefore is below both s and t and $u \in C(u)$. This gives us that $U \subseteq \bigcup_{r \in \downarrow(s \wedge t)} C(r)$.

For graphs, the lemma follows from the case with $k = 2$. □

Note that this is tight, as we have $\mathrm{td}(K_{kn}) = 2\,\mathrm{hd}(K^k_{kn}) = kn$. (with K^k_n the complete k-uniform hypergraph on n vertices). Furthermore, for graphs we have $\mathrm{td}(P_{2^n-2}) = \mathrm{hd}(P_{2^n-2}) = n$, where P_n is the path with n edges (cf. Example 25). For $k \geq 2$, we can replace each edge of P_n with a size k hyperedge, introducing new vertices for each edge. This gives an almost tight bound (slack by an additive factor of $k - 2$).

4. Hypertree-depth and ultramonotone marshals on two game boards

In the ultramonotone robber and marshals game on a hypergraph H we restricted the robber's movement to the underlying graph $\underline{H}$. However, we can lift this restriction and instead choose any graph G with $V(G) = V(H)$ as the 'game board' for the robber. We will do this and generalise our setting from hypergraphs to hypergraph pairs; a setting used in [2] to define a common generalisation of tree-width and hypertree-width. This allows us to handle both hypertree-depth of hypergraphs and tree-depth of graphs at the same time. Moreover, we will use this slightly more flexible framework in Section 6 to capture both minors in hypergraphs and (traditional) minors in graphs.

A *hypergraph pair* is defined as a pair (G, H) consisting of a graph G and a hypergraph H with $V(G) = V(H)$. There is no other restriction on the edges or the hyperedges. A hypergraph pair (G', H') is an *(induced) subhypergraph pair* of (G, H) if G' is an (induced) subgraph of G and H' is an (induced) subhypergraph of H.

The *ultramonotone* robber and marshals game on a hypergraph pair (G, H) is played like the ultramonotone robber and marshals game on a hypergraph with the difference that Player I 'plays on H' while Player II 'plays on G', i.e. the marshals occupy the hyperedges of H and the robber moves via paths in G.

In a play of the ultramonotone robber and marshals game on a hypergraph pair (G, H), player II begins by choosing a vertex $r_0 \in V(H)$ and placing the robber on r_0. The initial position is $(r_0, \emptyset)$. Then, in every round of the game in position (r, M), Player I selects a hyperedge $e' \in E(H)$ and reveals it to Player II. Player II then selects a vertex $r' \in V(H)$ that is connected to r by a path in $G \setminus \bigcup M$. He moves the robber to r' and Player I places a new marshal on e'. The new position is $(r', M \cup \{e'\})$.

Again, player I wins if he reaches a position (r, M) with $r \in \bigcup M$. The length of the game and winning strategies are defined as in Section 3.

In a position (r, M), the component of $G \setminus \bigcup M$ containing r is called the *robber space (with respect to M)*.

This is a generalisation of the game for hypergraphs, as stated by the following straightforward observation:

Remark 4 *For a hypergraph H, the cops have a winning strategy of length at most k on H if anf only if they have a winning strategy of length at most k on $(\underline{H}, H)$.*

We also extend the definition of hypertree-depth to hypergraph pairs (G, H), and we give some intuition below the definition.

A *decomposition forest* of (G, H) is a triple (F, B, C), where F is a rooted forest, $B\colon V(F) \to 2^{V(G)}$, and $C\colon V(F) \to E(H)$ are mappings, satisfying the following conditions.

(pdec0) $B(t) \subseteq C(t)$ for every $t \in V(F)$,

(pdec1) For every vertex $v \in V(G)$ there is a node $t \in V(F)$ such that $v \in B(t)$.

(pdec2) For every edge $e \in E(G)$ there are $\leq_F$-comparable nodes $s, t \in V(F)$ such that $e \subseteq B(s) \cup B(t)$,

(pdec3) All $s, t \in V(F)$ with $s \neq t$ satisfy $B(s) \cap B(t) = \emptyset$.

Definition 5 *The* hypertree-depth *of a hypergraph pair (G, H), $\mathrm{hd}(G, H)$, is the minimum height of a rooted forest F, taken over all decomposition forests (F, B, C) of (G, H).*

If F is a tree, we also speak of F as a *decomposition tree*. Note that $\mathrm{hd}(G, H) = 1$ if and only if for every connected component $C \subseteq V(G)$ of G, there exists a hyperedge $e \in E(H)$ such that $C \subseteq e$. Moreover, $\mathrm{hd}(G, H)$ is bounded from above by the minimum number of hyperedges of H necessary to cover $V(G)$.

The forest F and the mapping C play the same roles as on hypergraphs, and intuitively, $B(s)$ indicates which of the vertices of $C(s)$ are 'covered' in the branch of F containing s. If $G \neq \underline{H}$, we may have a hyperedge e with $G[e]$ disconnected and

e present as C of multiple nodes s_i. In this case, B distinguishes which connected components of $G[V \setminus B(\downarrow t_i)]$ (with t_i the parent of s_i) are covered by the nodes above s_i.

This 'entire connected components' property of B is illustrated in the following lemma.

Lemma 6 *Let (G, H) be a hypergraph pair with a decomposition forest (F, B, C), and let $X \subseteq V(G)$ be connected in G.*

1. *There exists precisely one connected component T of F such that $X \cap B(T) \neq \emptyset$ and we have $X \subseteq B(T)$.*
2. *If T is a connected component of F and $t \in V(T)$ with $X \subseteq B(T_t) \setminus B(\downarrow t)$, then t has precisely one child s such that $X \subseteq B(T_s)$.*

Proof. *1:* If there were two trees T_1, T_2 of F with $B(T_1)$ and $B(T_2)$ intersecting X, then by **(pdec3)** we have $|X| > 1$, and there is an edge $\{u, v\} \in E(G[X])$ with $u \in B(T_1)$ and $v \in B(T_2)$, contradicting **(pdec2)**.

2: Similarly, if there are two child nodes s_1 and s_2 with $B(T_{s_1})$ and $B(T_{s_2})$ intersecting X, there is an edge $\{u, v\} \in E(G[X])$ with $u \in B(T_{s_1})$ and $v \in B(T_{s_2})$, which is impossible by **(pdec2)** and **(pdec3)**. □

We say that a decomposition tree (F, B, C) of (G, H) is *small*, if $B(t) \neq \emptyset$ for every $t \in V(F)$. Leaving out all the nodes with $B(t) = \emptyset$ (possibly contracting the edge to the parent of t) yields a small decomposition tree with smaller or equal depth.

Replacing condition **(pdec3)** by

(pdec3′) All $s, t \in V(F)$ satisfy: If $B(s) \cap B(t) \neq \emptyset$ then the node $s \wedge t$ exists and $B(s) \cap B(t) \subseteq B(\downarrow (s \wedge t))$,

we obtain another definition of decomposition forest which more closely resembles that of a decomposition forest of a (single) hypergraph. Trivially, any decomposition forest satisfying **(pdec3)** satisfies **(pdec3′)**. On the other hand we get:

Lemma 7 *For every decomposition forest (F, B, C) satisfying **(pdec0)**, **(pdec1)**, **(pdec2)** and **(pdec3′)**, but not necessarily **(pdec3)**, there is a mapping B' such that for all $t \in V(F)$, $B'(t) \subseteq B(t)$ and (F, B', C) is a decomposition forest.*

Proof. We set $B'(t) = B(t) \setminus \bigcup_{s <_F t} B(s)$. Now **(pdec3)** is satisfied: for any $s, t \in F$, $B'(s) \cap B'(t) \subseteq B(s) \cap B(t) \subseteq B(\downarrow (s \wedge t))$ and these vertices have been removed from at least one of $B'(s)$ or $B'(t)$. It is easy to see that **(pdec0)**, **(pdec1)** and **(pdec2)** are preserved by going from B to B'. □

Lemma 8 *Let H be a hypergraph. Then* $\mathrm{hd}(H) = \mathrm{hd}(\underline{H}, H)$.

Proof. We may assume that $\underline{H}$ is connected.

Towards a proof of $\mathrm{hd}(H) \leq \mathrm{hd}(\underline{H}, H)$, let (F, B, C) be a small decomposition forest of $(\underline{H}, H)$ such that the height of F is $\mathrm{hd}(\underline{H}, H)$. We claim that (F, C) is a decomposition forest of H witnessing $\mathrm{hd}(H) \leq \mathrm{hd}(\underline{H}, H)$. By **(pdec2)**, (F, C) satisfies **(dec2)**.

Towards showing that condition **(dec3)** is satisfied, let $s, t \in V(F)$ be vertices in the same component of F and let $v \in C(s) \cap C(t)$. Since (F, B, C) is small, there exists a vertex $w \in B(s) \setminus (\bigcup_{r<s} B(r))$. By **(pdec0)**, $\{v, w\} \in E(\underline{H})$. Hence by condition **(pdec2)** we must have $v \in B(T_s) \cup (\bigcup_{r \in \downarrow s} B(r))$. Applying the analogous argument to t, we must have $v \in B(T_t) \cup (\bigcup_{r \in \downarrow t} B(r))$.

Since B is a partition of $V(G)$, we have a unique u with $v \in B(u)$, $u \leq s \wedge t$, and we are done.

Conversely, if (F, C) is a decomposition forest of H of height $\mathrm{hd}(H)$, let $B(t) := C(t)$ for all $t \in V(F)$. Then (F, B, C) is a decomposition forest satisfying **(pdec3′)** instead of **(pdec3)**. By Remark 7, there is a decomposition forest (F, B', C) of $(\underline{H}, H)$ witnessing $\mathrm{hd}(\underline{H}, H) \leq \mathrm{hd}(H)$. □

The following remark is easy to see.

Remark 9 *Let (G, H) be a (1-uniform) hypergraph pair where $E(H) = \{\{v\} \mid v \in V(G)\}$. Then* $\mathrm{hd}(G, H) = \mathrm{td}(G)$.

4.1. Game characterisation of hypertree-depth

We now present and prove the equivalence of the ultramonotone robber and marshals game and the hypertree-depth on hypergraph pairs. This also finishes the proof of the equivalence on hypergraphs (Theorem 2).

Theorem 10 *A hypergraph pair (G, H) has* $\mathrm{hd}(G, H) \leq k$ *if and only if the cops have a winning strategy for the ultramonotone robber and marshals game on (G, H) of length at most k.*

Proof. If a vertex of $V(G)$ is not contained in any hyperedge of H, then $\mathrm{hd}(G, H) = \infty$ and there is no marshals' winning strategy. From now on we assume that $V(G) \subseteq \bigcup E(H)$.

First, let (F, B, C) be a decomposition tree of (G, H) of height $\mathrm{hd}(G, H)$. We now describe a strategy for the marshals. Intuitively, the marshals play along the C-labels of a branch of F, starting at a root and always following the robber.

First, the robber selects a vertex $r_0 \in V(G)$ and the play is in position $(r_0, \emptyset)$. Since the robber is only allowed to move along paths in G, he will now stay in the connected component $X_0 \subseteq V(G)$ with $r_0 \in X_0$.

By Lemma 6.1, there exists precisely one component T of F such that $X \subseteq B(T)$.

The marshals select t to be the root of T and announce their move to $C(t)$. The robber moves to a vertex r that is connected to r_0 in G and the marshals complete their move.

If $r \in C(t)$, the marshals win. Otherwise, by Lemma 6.2, there is a unique child s of t such that $r \in B(T_s)$, and the marshals announce their move to $C(s)$. Continuing in this way along a chain in T, the marshals capture the robber in at most k rounds.

Towards a proof of the other implication, suppose that the marshals have a winning strategy $\mathcal{S}$ of length at most k in the ultramonotone robber and marshals game on (G, H).

First, we observe that we may assume that the marshals' moves according to $\mathcal{S}$ depend only on the robber space, that is the component of $V(G) \setminus \bigcup M$ containing r, where (r, M) is a game position. To prove that, it is enough show we can always modify a given strategy S' of length k to a strategy of length at most k that does not depend on other components $V(G) \setminus \bigcup M$ than the robber space and on the particular vertices the robber occupies within the robber space. We can do that by replacing a substrategy of S' with an initial state (r, M) with the robber space R by a substrategy of S' with the initial state (r_1, M_1) with the robber space R, where M_1 is any inclusion-wise maximal set of hyperedges such that the state (r, M_1) occurs in S' and the robber space in the state (r, M_1) is R and r_1 is any vertex within R. The result is indeed a correct strategy because the possibilities of the robber do not depend on the robber's position but on the robber space only, and by choosing M_1 to be inclusion-wise maximal, we ensure that a new strategy uses every hyperedge at most once.

Consequently, we may assume that in each move played according to $\mathcal{S}$ (that depends only on the robber space), the robber space decreases by at least one vertex.

Now we construct the nodes of tree T while keeping track of a game position for every node constructed. We construct a separate rooted decomposition for every connected component X of G with a root vertex t_0 as follows. Suppose in the first move, the robber chooses a vertex $r \in X$. The node t_0 corresponds to the game position $(r, \emptyset)$.

Now for every node t of T corresponding to a position (r, M) we recursively define $C(t)$, $B(t)$, the set of t's children and their corresponding game positions. Let X be the robber space. Let $e = \mathcal{S}(r, M)$ be the next marshals' move according to $\mathcal{S}$. Then set $B(t) = e \cap X$ and $C(t_0) = e$. Also, let X_i be the components of $X \setminus e$. For every X_i, choose arbitrary $r_i \in X_i$ and add a node t_i to T as a child of t. The position corresponding to t_i is $(r_i, M \cup \{e\})$. Note that a node corresponding to a position winning for marshals (when $r \in \bigcup M$) has no children.

Let F be the rooted forest obtained by doing this for all connected components of G. We claim that (F, B, C) is a decomposition forest of height at most k for (G, H). Condition **(pdec0)** is satisfied by construction. Condition **(pdec1)** is satisfied, because $\mathcal{S}$ is a winning strategy and hence the robber is captured in one leaf node of F, regardless on which vertex he chooses to reside. Condition **(pdec3)** is satisfied: $B(s)$ and $B(t)$ for s, t comparable are disjoint by the construction, for s, t incomparable we have that the robber spaces X_s at s and X_t at t are disjoint. For **(pdec2)**, let $\{u, v\} \in E(G)$. Assume the robber has the strategy to play staying inside the set $\{u, v\}$ through the entire game. Since $\mathcal{S}$ is a winning strategy, there are hyperedges e and f taken by the marshals in a single play with $u \in e$ and $v \in f$. By construction of (F, B, C), there are nodes s, t with $e \in C(s)$ and $f \in C(t)$ in a chain corresponding to the play above.

Since $\mathcal{S}$ has length at most k, F has height at most k. □

5. Other characterisations of hypertree-depth

In spirit of Nešetřil et al. [11], we present alternative characterisations of hypertree-depth. We give an inductive definition together with definitions based on *vertex-hyperrankings* and *centred hypercolourings*. For brevity, we give the definitions only for hypergraph pairs, but the definitions and properties for hypergraphs follow easily by using $(\underline{H}, H)$ as the pair.

These characterisations are very similar to those of Nešetřil et al. and we also present them as an indication that our notion of hypertree-depth is the correct hypergraph generalisation of tree-depth. Note that the corresponding definitions for graphs G are obtained by using the hypergraph pair $(G, \{\{v\}, v \in V(G)\})$ (1-uniform hypergraphs from Remark 9). (Using (G, G) for graphs is not conceivable as indicated in the discussion in Section 3).

An inductive definition of hypertree-depth. Similarly to tree-depth of graphs [11], hypertree-depth has an inductive definition.

Lemma 11 *Let (G, H) be a hypergraph pair satisfying $V(G) = \bigcup E(H)$ and $|V(H)| > 0$. Let $G_1, \dots G_p$ be the connected components of G. Then*

$$\mathrm{hd}(G,H) = \begin{cases} 1, & \textit{if there exists } e \in E(H) \textit{ with } V(G) \subseteq e, \\ 1 + \min\limits_{e \in E(H)} \mathrm{hd}\big(G \setminus e, H[V(G) \setminus e]\big), & \textit{if } p = 1 \textit{ and } V(G) \setminus e \neq \emptyset \textit{ for all } e \in E(H), \\ \max\limits_{i \in \{1,\dots,p\}} \mathrm{hd}\big(G_i, H[V(G_i)]\big), & \textit{otherwise.} \end{cases}$$

Proof. We prove this by induction on $\mathrm{hd}(G, H)$. If $\mathrm{hd}(G, H) = 1$ then we are in the first case. Now let $\mathrm{hd}(G, H) = k > 1$ and assume the lemma has been proven for all hypergraph pairs with hypertree-depth at most $k - 1$. If G is connected, by Lemma 6 there exists a decomposition *tree* (T, B, C) of height k for $\mathrm{hd}(G, H)$. Let r be the root of T and let $F := T \setminus r$ be the rooted forest obtained from T be deleting r and choosing the children of r as the new roots of F. Then the restriction $(F, B \restriction V(F), C \restriction V(F))$ shows that $\mathrm{hd}\big(G \setminus C(r), H[V(G) \setminus C(r)]\big) \leq k-1$ and $\mathrm{hd}(G, H) \geq \mathrm{hd}\big(G \setminus C(r), H[V(G) \setminus C(r)]\big) + 1$. Conversely, let $e \in E(H)$ and let (F, B, C) be a decomposition forest of height $\mathrm{hd}(G \setminus e, H[V(G) \setminus e])$ for $(G \setminus e, H[V(G) \setminus e])$. Let T be the rooted tree obtained from F by introducing a new node r as the new root and connecting r to all old roots of F by edges. Letting $B(r) = C(r) = e$ we see that $\mathrm{hd}(G, H) \leq \mathrm{hd}\big(G \setminus e, H[V(G) \setminus e]\big) + 1$.

If G is not connected, it is easy to see that the third case applies. □

5.1. Vertex-hyperrankings and centred hypercolourings

Definition 12 *Let (G, H) be a hypergraph pair. A* vertex-hyperranking *of (G, H) is a (not necessarily proper) colouring* $\mathrm{col}\colon V(G) \to \{1, \ldots k\}$ *such that for every $i \leq k$ and for every connected component X of $G_i := G[\{v \in V(G) \mid \mathrm{col}(v) \geq i\}]$ there is a hyperedge $e \in E(H)$ such that $\{v \in X \mid \mathrm{col}(v) = i\} \subseteq e$.*

Definition 13 *Let (G, H) be a hypergraph pair. A* centred hypercolouring *of (G, H) is a (not necessarily proper) colouring of $V(G)$ such that for every induced subhypergraph pair (G', H') where G' is connected, there exists a colour $a = a(G', H')$ colouring at least one vertex of G' and a hyperedge $e \in E(H')$, such that $\{v \in V(G') \mid v$ has colour $a\} \subseteq e$.*

Theorem 14 *For every hypergraph pair (G, H), the following are equivalent:*

- $\mathrm{hd}(G, H)$ *has a vertex-hyperranking with labels $\{1, \ldots k\}$,*
- *(G, H) has a centred hypercolouring with at most k colours, and*
- $\mathrm{hd}(G, H) \leq k$.

Proof. We may assume that G is connected. Assume that (G, H) has a centred hypercolouring with k colours. Then there exists a colour a and a hyperedge $e \in E(H)$ such that $\emptyset \neq X := \{v \in V(G) \mid v \text{ has colour } a\} \subseteq e$.

If $k = 1$, then $V(G) \subseteq e$ and hence $\mathrm{hd}(G, H) \leq 1$.

Let $k > 1$ and assume that we have proved the theorem for all hypergraph pairs having centred hypercolourings with $k-1$ colours. Since the induced subhypergraph pair $(G \setminus X, H \setminus X)$ has a centred hypercolouring with $k - 1$ colours, we know that $\mathrm{hd}(G \setminus X, H \setminus X) \leq k-1$, witnessed by a decomposition forest (F, B, C). We create a new root r with $B(r) = X$ and $C(r) = e$, and we let the old roots of F be the children of r. In this way we obtain a decomposition tree witnessing $\mathrm{hd}(G, H) \leq k$.

Now assume that $\mathrm{hd}(G, H) = k$, witnessed by a decomposition forest (F, B, C). Rank a $v \in V(G)$ with the minimum depth of a node $t \in V(F)$ with $v \in B(t)$. We claim that this is a vertex-hyperranking: Let (G', H') be an induced subhypergraph pair of (G, H) such that G' is connected. By Lemma 6 there is a connected component $T \subseteq F$ with $V(G) \subseteq B(T)$ and T is the unique component with $V(G) \cap B(T) \neq \emptyset$. Let t in $V(T)$ be the node of minimum depth such that $B(t) \cap V(G) \neq \emptyset$. From **(pdec2)** we get that all $V(G')$ are above t and we have a proper vertex-hyperranking.

The last of the three implications follows from the fact that every vertex-hyperranking of a hypergraph pair (G, H) is a centred hypercolouring. □

6. Minors in hypergraphs and hypergraph Pairs

In this section we propose a concept of minors for hypergraphs and hypergraph pairs. We first give the hypergraph variant and then generalise it to hypergraph pairs. In Section 6.2 the hypergraph notion is shown to be equivalent to the pair definition on $(\underline{H}, H)$. We also introduce the concepts of minor maps and minor models on hypergraph pairs. Hypergraph variants of these are not introduced explicitely, as they follow from applying the definitions to pairs of the form $(\underline{H}, H)$.

6.1. Hypergraph minors

The standard and well-known minor operation is the following: Let G be a graph and let $e = \{x, y\} \in E(G)$. The graph obtained from G by *contracting* e is the graph G/e with $V(G/e) = (V(G) \setminus \{x, y\}) \dot\cup \{v_{xy}\}$, where v_{xy} is a new vertex, and $E(G/e) = \big(E(G) \setminus \{\{u, v\} \mid u \in \{x, y\},\ v \in V(G)\}\big) \cup \{\{u, v_{xy}\} \mid u \in V(G) \setminus \{x, y\},\ \{u, x\} \in E(G) \text{ or } \{u, y\} \in E(G)\}$. A graph H is a *minor* of a graph G if H can be obtained from a subgraph of G by a sequence of edge contractions. Alternatively, a minor of G is any graph obtainable from G by a series of vertex deletions, edge deletions and contractions.

For a hypergraph H, the hypergraph obtained from H by *contracting* an edge $\{x, y\} \in E(\underline{H})$ is the hypergraph H/e with $V(H/e) = V(H) \setminus \{x, y\} \cup \{v_{xy}\}$ and

$$E(H/e) = \big\{h \in E(H) \bigm| h \cap \{x, y\} = \emptyset\big\} \cup \big\{(h \setminus e) \cup \{v_{xy}\} \bigm| h \in E(H),\ h \cap e \neq \emptyset\big\}.$$

In other words, v_{xy} is the new contracted vertex and every hyperedge containing either x or y is set to contain v_{xy}.

Definition 15 *Let H and H' be hypergraphs. Then H is a minor of H', denoted $H \preccurlyeq H'$ if H can be obtained from H' by a sequence of operations of the following kinds:*

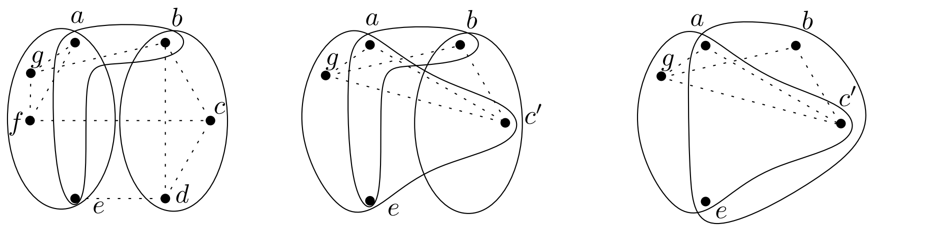
- *vertex deletion,*
- *contraction of (the edge between) two vertices that are contained in a common hyperedge,*
- *addition of a hyperedge e such that the set e induces a clique in the underlying graph, and*
- *deletion of a proper subhyperedge.*

Note that we do not allow arbitrary hyperedge removal, so a subhypergraph is not necessarily a minor. From our game and decomposition point of view, a hypergraph with a hyperedge containing all the vertices would be trivial, while removing this edge (even if preserving the underlying graph) might increase the complexity arbitrarily. While this argument is not rigorous, it is not bound to a single width parameter or game, but captures most scenarios where the hyperedges represent a unit to be taken, counted or paid for.

The first two operations are analogous to traditional graph minors. In the spirit of the previous paragraph, the third operation only decreases the 'cost' without changing the underlying graph, and the last removes hyperedges that are 'made redundant' by other edges.

The following remark shows that hypergraph minors induce graph minors on the underlying graphs.

Remark 16 *Let H and H' be hypergraphs. If $H \preccurlyeq H'$, then $\underline{H} \preccurlyeq \underline{H'}$.*



a) a hypergraph pair b) contraction of edge cf and deletion of vertex d c) addition of hyperedge $abc'e$ and deletion of its proper subhyperedges abe and bc'

Figure 2: Example of minor operations on a hypergraph pair (edges of the graph are dotted)

Note that the converse does not hold: If H is the path of length two and H' is the triangle (both viewed as hypergraphs), then $\underline{H} \preccurlyeq \underline{H'}$, but $H \not\preccurlyeq H'$ because we do not allow deleting arbitrary hyperedges. Indeed, it seems desirable to have graph minors as a special case. For this, we move one step further and generalise minors to hypergraph pairs.

6.2. Hypergraph pair minors

We now generalise the notion of graph minors to hypergraphs and hypergraph pairs. Let (G, H) be a hypergraph pair and let $v \in V(G)$. The hypergraph pair $(G \setminus v, H[V(G) \setminus v])$ is called the hypergraph pair obtained from (G, H) by *deleting* v. We denote it by $(G, H) \setminus v$. For $e \in E(G)$, the hypergraph pair $(G \setminus e, H)$ is called the hypergraph pair obtained from (G, H) by *deleting* e.

Let $e \subseteq V(H)$, then $(G, H + e)$, where $H + e = (V(H), E(H) \cup \{e\})$, is the hypergraph pair obtained from (G, H) by *adding* e. If $e, e' \in E(H)$ and $e \subsetneq e'$, then $(G, H \setminus e)$ is the hypergraph pair obtained from (G, H) by *deleting* e.

Let $e = \{x, y\} \in E(G)$. Then the hypergraph pair obtained from (G, H) by *contracting* e, denoted by $(G, H)/e$, is the hypergraph pair obtained by contracting e in both parts, i.e. $(G, H)/e := (G/e, H/e)$.

Definition 17 *A hypergraph pair (G', H') is a* minor *of (G, H), denoted by $(G', H') \preccurlyeq (G, H)$, if (G', H') can be obtained from (G, H) by a sequence of*

- *vertex deletions,*
- *graph edge contractions,*
- *graph edge deletions,*
- *hyperedge additions, and*
- *deletions of proper subhyperedges.*

Note that if H is a hypergraph and (G, H') is a hypergraph pair with $(G, H') \preccurlyeq (\underline{H}, H)$, then G is not necessarily the underlying graph of H' (for example, G could be obtained from $\underline{H}$ by deleting an edge).

The (non-rigorous) motivation for these operations is similar to hypergraph minor operations, viewing the graph as defining the complexity and the hypergraph as the means of decomposition or covering it. Namely, the first two and the last operations are the same as for hypergraphs; the third decreases the complexity of the underlying structure without decreasing the covering potential; and the fourth is similar to 'addition of hyperedges on cliques' in hypergraph minors, but here we need not restrict it by the underlying graph.

We use the symbol $\preccurlyeq$ both for the graph minor ordering and the hypergraph pair minor ordering. It will always be clear from the context which one we mean. The justification is the following lemma.

Lemma 18 *Let H and H' be hypergraphs. Then $H \preccurlyeq H'$ if and only if $(\underline{H}, H) \preccurlyeq (\underline{H'}, H')$.*

Proof. The "only if" follows from the fact that these operations are special cases of minor operations in hypergraph pairs.

For the "if" part, let $o_1, \dots o_k$ be a sequence of hypergraph pair minor operations taking $(G_0, H_0) = (\underline{H'}, H')$ to $(G_k, H_k) = (\underline{H}, H)$. We modify the sequence to only contain the operations for hypergraph minors, showing $H \preccurlyeq H'$. Notice that in any intermediate state (G_i, H_i), we have $G_i \subseteq \underline{H_i}$.

First, we make sure that all the vertex deletions come first in the sequence by "bubbling" every vertex deletion to front, adjusting the sequence locally as below to get the same result. Note that the vertex-deletion is always the second in the swapped pair. Generally, the mentioned conditions are considered in the hypergraph pair *before* the swapped operations take place, in the *current* operation sequence and intermediate states.

When swapping v'-deletion with uv-edge contraction resulting in v', drop both operations and replace them with u-deletion and v-deletion. When swapping v-deletion with uv-edge-deletion, drop the edge deletion. When swapping v-deletion with e-hyperedge-addition such that $v \in e$, replace e-addition with $e \setminus \{v\}$-addition. When swapping v-deletion with e-hyperedge-deletion such that e is a proper sub-hyperedge, we either replace it with $e \setminus \{v\}$-deletion (if $v \in e$), remove (if $e \cup \{v\}$ is a hyperedge), or leave it as-is (otherwise). Swapping two operations on disjoint vertex sets requires no change.

Second, in a similar manner, we make sure all the edge contractions come just after the vertex deletions. Assume uv-contraction resulting in vertex v'. When swapping with vw-edge-removal, replace the removal with $v'w$-removal (if uw is not an edge) or drop the edge removal (if uv is an edge). Swapping with hyperedge operations and independent edge contractions requires no changes.

Now we have a modified sequence still resulting in $(G_k, H_k) = (\underline{H}, H)$. Notice that all vetex deletions and edge contractions now preserve $G_i = \underline{H_i}$. If this is true for all the operations, it is easy to see that there are no edge deletions and the hyperedge operations satisfy the required conditions and we are done.

Otherwise, let i be first such that $G_i \neq \underline{H_i}$ and take $uv \in E(\underline{H_i}) - E(G_i)$. All subsequent operations (hyperedge addition, removal of a proper subhyperedge, edge deletion) preserve the fact that $uv \notin E(G_j)$ and $uv \in E(\underline{H_j})$ for all $j > i$, a contradiction with $(G_k, H_k) = (\underline{H}, H)$. □

Let I_n denote the graph with n vertices and no edges and let K_n denote the complete graph with n vertices and all possible edges. The following remark shows some basic properties of minors.

Example 19
1. *If H is an induced subhypergraph of H' then $H \preccurlyeq H'$.*
2. *Let $n \in \mathbb{N}$. Any hypergraph pair (G, H) with $|V(G)| \leq n$ is a minor of (K_n, I_n).*
3. *Let P_2 denote the path with two edges. Then $(P_2, P_2) \not\preccurlyeq (K_3, K_3)$.*

For a graph G let $G^\emptyset$ denote the hypergraph with $V(G^\emptyset) = V(G)$ and $E(G^\emptyset) = \emptyset$. Furthermore, let G^1 denote the hypergraph with $V(G^1) = V(G)$ and $E(G^1) = \big\{\{v\} \mid v \in V(G)\big\}$.

Minors in hypergraph pairs generalise minors in graphs, similarly to generalising tree-depth and tree-width:

Remark 20 *Let G and F be graphs. Then*

$$G \preccurlyeq F \iff (G, G^{\emptyset}) \preccurlyeq (F, F^{\emptyset}) \iff (G, G^1) \preccurlyeq (F, F^1).$$

6.3. Minor maps and models

Minor maps and minor models are well-known concepts for graph minors. We generalise them to the hypergraph pair setting. The hypergraph variants are not specified explicitly. They can be obtained by specialising the pair concepts and properties to $(\underline{H}, H)$.

Let G and G' be graphs. A *minor map* (from G to G') is a function μ defined on $V(G) \cup E(G)$, that takes a vertex $v \in V(G)$ to a non-empty connected subset $\mu(v) \subseteq V(G')$ and an edge $e \in E(G)$ to an edge $\mu(e) \in E(G')$ such that

- any vertices $u, v \in V(G)$ with $u \neq v$ satisfy $\mu(u) \cap \mu(v) = \emptyset$, and
- for every edge $\{u, v\} \in E(G)$ the image $\mu(\{u, v\})$ connects the vertex images $\mu(u)$ and $\mu(v)$.

It is straightforward to verify that there is a minor map from G to G' if and only if $G \preccurlyeq G'$.

The subgraph $\bigcup_{v \in V(G)} G'[\mu(v)] \cup \bigcup_{e \in E(G)} \mu(e)$ of G' is called a *model* of G in G'.

We extend the definition of *model* to hypergraph pairs. Let (G, H) and (G', H') be hypergraph pairs, and let μ be a minor map from G to G'. We say that μ is a *minor map* from (G, H) to (G', H'), if for every hyperedge $e \in E(H')$, the set $\{v \in V(G) \mid \mu(v) \cap e \neq \emptyset\}$ is a subset of some hyperedge of H or empty.

Remark 21 *Any two hypergraph pairs (G, H) and (G', H') satisfy*

$$(G, H) \preccurlyeq (G', H') \iff \text{there is a minor map from } (G, H) \text{ to } (G', H').$$

We just sketch the proof, which is straightforward but technical. First the 'only if' part. Maintain a minor map $(G, H) \to (G', H')$ (starting with identity), adapting it locally with every minor operation on (G, H). For the 'if' part, having a minor map μ, delete all vertices not in the image of μ, contract the individual images of μ and then remove edges and hyperedges as necessary.

We call a hypergraph pair (M, F) a *model* of (G, H) in (G', H'), if M is a model of G in G' and $F = H'[V(M)]$.

6.4. Properties of minors in hypergraphs

For graphs, a famous result by Robertson and Seymour shows that testing for a fixed minor is solvable in cubic time, i.e. for a fixed graph G, given a graph G', there is a cubic time algorithms that decides whether $G \preccurlyeq G'$ holds [15]. In contrast, testing for a fixed hypergraph minor can be NP-hard. To see this, we say that a graph G is an *induced* minor of a graph G' if G can be obtained from an induced subgraph of G' by a sequence of edge contractions. Fellows et al. [6] show that there exists a graph G such that testing for G as an induced minor is NP-complete.

Remark 22 *There exists a graph G such that deciding for a given hypergraph pair (G', H') whether $(G, G) \preccurlyeq (G', H')$ holds is NP-complete.*

Proof. For NP-hardness, observe that for any pair of graphs G and G' the graph G is an induced minor of G' if and only if $(G, G) \preccurlyeq (G', G')$, and use the result of Fellows et al. [6]. The problem is in NP because we can simply guess a minor map. □

Let us remark that the induced minor ordering is known to have infinite antichains [9]. This implies that the minor orderings of hypergraphs and hypergraph pairs have infinite antichains as well. Nevertheless, we give an explicit example of an infinite $\preccurlyeq$-antichain of hypergraph pairs.

Remark 23 *For every integer $n > 0$ let G_n be an arbitrary graph with n vertices, and let H^{n-1} be the hypergraph with $V(H^{n-1}) = V(G_n)$ and $E(H^{n-1}) = [V(H)]^{n-1}$ (all $(n-1)$-element subsets of $V(H)$).*

Then the sequence $\big((G_n, H^{n-1})\big)_{n<\omega}$ of hypergraph pairs is an infinite $\preccurlyeq$-antichain. In particular, the sequence $(H^{n-1})_{n<\omega}$ is an infinite $\preccurlyeq$-antichain.

Proof. Let $i, j \in \omega$ with $i < j$. To obtain G_i as a minor of G_j, we have to use at least one vertex deletion or edge contraction. The first operation of either kind leaves a hyperedge covering all the vertices of G_j (or such hyperedge was created before the operation). We cannot remove this hyperedge so we cannot produce (G_i, H_i) as a minor. □

7. Interaction with other hypergraph invariants

In this section we compare hypertree-depth to generalised hyperpath-width [10] and to generalised hypertree-width [8], and we show that these invariants do not increase under taking minors.

Let (G, H) be a hypergraph pair. A *generalised hypertree decomposition* [2] of (G, H) is a triple (T, χ, λ) where (T, χ) is a tree decomposition of G and $\lambda\colon V(T) \to 2^{E(H)}$ is a mapping such that

(ghd) Every $t \in V(T)$ satisfies $\chi(t) \subseteq \bigcup \lambda(t)$.

The *width* of (T, χ, λ) is defined as $\mathrm{w}(T, \chi, \lambda) := \max\big\{\, |\lambda(t)| \;\big|\; t \in V(T)\big\}$. The *generalised hypertree-width of* (G, H) is defined as

$$\mathrm{ghw}(G, H) := \min\big\{\, \mathrm{w}(T, \chi, \lambda) \;\big|\; (T, \chi, \lambda) \text{ is a generalised hypertree decomposition of } (G, H)\big\}.$$

Note that any hypergraph pair (G, H) with an edge $e \in E(H)$ such that $V(G) \subseteq e$ satisfies $\mathrm{ghw}(G, H) = 1$. Moreover, any graph G satisfies $\mathrm{tw}(G)+1 = \mathrm{ghw}(G, G^1)$.

For a hypergraph H we let $\mathrm{ghw}(H) := \mathrm{ghw}(\underline{H}, H)$, and we obtain the original notion of generalised hypertree-width of a hypergraph, as defined by Gottlob, Leone and Scarcello [8].

If we alter the definition of generalised hypertree decompositions by requiring that T be a path, we obtain the notions of *generalised hyperpath decomposition* and the *generalised hyperpath-width* of (G, H), denoted by $\mathrm{hpw}(G, H)$. This concept is also introduced in [2]. Again, for a hypergraph H we let $\mathrm{hpw}(H) := \mathrm{hpw}(\underline{H}, H)$.

Theorem 24 *Any hypergraph pair (G, H) satisfies*

$$\mathrm{ghw}(G, H) \le \mathrm{hpw}(G, H) \le \mathrm{hd}(G, H).$$

Proof. The first inequality follows immediately from the fact that every generalised hyperpath decomposition is a generalised hypertree decomposition. Towards the second inequality, let (F, B, C) be a decomposition forest of (G, H) of height k. Let $\ell_1, \dots \ell_r$ be an enumeration of the leaves of the trees in F, i.e. of the nodes of F with degree 1 that are not roots, together with all nodes of degree 0. We define a path P with vertices $V(P) = \{\ell_1, \dots \ell_r\}$, connecting ℓ_i to ℓ_{i+1} by an edge, for $1 \le i < r$. For every $1 \le i \le r$ we let $\chi(\ell_i) := B(\downarrow \ell_i)$ and $\lambda(\ell_i) := \{C(t) \mid t \in \downarrow \ell_i\}$. It is easy to see that (P, χ, λ) is a generalised hyperpath decomposition of width $\le k$ of (G, H). □

Example 25 *Let $\mathcal{T}$ be the class of all trees. It is easy to see that the generalised hypertree-width of $\mathcal{T}$ is bounded by 1, but $\mathcal{T}$ has unbounded generalised hyperpath-width. Similarly, the generalised hyperpath-width of the class $\mathcal{P}$ of all paths is bounded by 1, but $\mathcal{P}$ has unbounded hypertree-depth – the hypertree-depth of a path of length n is $\lfloor \log_2(n+2) \rfloor$. This follows from the definition and properties of vertex hyperranking by induction on the length of the path.*

We also have that none of our hypergraph invariants can increase when taking minors:

Lemma 26 *Let $(G', H') \preccurlyeq (G, H)$. Then $\mathrm{hd}(G', H') \leq \mathrm{hd}(G, H)$, $\mathrm{hpw}(G', H') \leq \mathrm{hpw}(G, H)$, and $\mathrm{ghw}(G', H') \leq \mathrm{ghw}(G, H)$.*

Proof. Let μ be a minor map from (G', H') to (G, H) and for $X \subseteq V(G)$, let $\eta(X) = \{v \in V(G') \mid \mu(v) \cap X \neq \emptyset\}$. Observe that if $\eta(X) \neq \emptyset$ and there is a hyperedge $e \in E(H)$ with $X \subseteq e$, then there exists $e' \in E(H')$ with $\eta(X) \subseteq e'$. Generally, if $\eta(X) \neq \emptyset$ and some k hyperedges of H cover X, then there are at most k hyperedges of H' covering $\eta(X)$.

We first show the inequalities for ghw and hpw. We assume that $\mathrm{ghw}(G, H) < \infty$ ($\mathrm{hpw}(G, H) < \infty$, respectively). Let (T, χ, λ) be a generalised hypertree decomposition of (G, H) with width $\mathrm{ghw}(G, H)$. Let $T' = T$ and for $t \in T$, let $\chi'(t) = \eta(\chi(t))$. Now (T', χ') is a tree decomposition of G': For $v' \in V(G')$, let S be all the nodes t' of T' with $\chi'(t') \cap \mu(v') \neq \emptyset$. Since $\mu(v')$ is connected in G, it follows from the properties of a tree decomposition that S is connected in T. Let S' be the nodes of T' containing v'. From definition, we have $S' = S$ and the occurrences of v' in T form a subtree. The other conditions for tree decompositions are satisfied trivially. We let $\lambda'(t)$ be the hyperedges covering $\eta(\lambda(t))$ from the observation above, satisfying **(ghd)** for (T', χ', λ'). Now, (T', χ', λ') is a hypertree decomposition of (G', H') showing $\mathrm{ghw}(G', H') \leq \mathrm{ghw}(G, H)$. This also proves the inequality for hpw, since if T is a path, T' is a path as well.

Finally, we show that $\mathrm{hd}(G', H') \leq \mathrm{hd}(G, H)$. Assume that $\mathrm{hd}(G, H) < \infty$. Let (F, B, C) be a decomposition forest of (G, H) with depth $\mathrm{hd}(G, H)$. Let $F' = F$ and for $t \in V(F')$, define $C'(t)$ to be the hyperedge covering $\eta(C(t))$. Define B' recursively on F' (from the roots up) as $B'(t) = \eta(B(t)) \setminus \bigcup_{s<t} B'(s)$. Now (F', B', C') trivially satisfies **(pdec0)** and **(pdec1)**. For **(pdec2)**, let $e' = (x', y')$ be an edge of G', $e = \mu(e') \in E(G)$ and take $s, t \in V(F)$ as in **(pdec2)** for (G, H) and e. From definition of B', we have $x' \in \bigcup B'(\downarrow s)$ and $y' \in \bigcup B'(\downarrow t)$ or vice-versa. For **(pdec3)**, suppose that there are $s, t \in F$ with $v' \in B'(s) \cap B'(t) \neq \emptyset$. From definition of B', s and t are incomparable in F. In (G, H), $B(s)$ and $B(t)$ are separated by $S = \bigcup B(\downarrow (s \wedge t))$. But $\mu(v)$ is connected, intersects both $B(s)$ and $B(t)$ and avoids S – a contradiction with Lemma 6.2.

Therefore, (F', B', C') is is a decomposition forest of (G', H') showing that $\mathrm{hd}(G', H') \leq \mathrm{hd}(G, H)$. □

Remark 27 *Let $H' \preccurlyeq H$. Then $\mathrm{hd}(H') \leq \mathrm{hd}(H)$, $\mathrm{hpw}(H') \leq \mathrm{hpw}(H)$, and $\mathrm{ghw}(H') \leq \mathrm{ghw}(H)$.*

This follows directly from Lemma 26 on $(\underline{H}, H)$.

8. Conclusion

We introduced the notion of hypertree-depth of hypergraphs and hypergraph pairs and we characterised it using the ultramonotone robber and marshals game, and using vertex-hyperrankings and centred hypercolourings. We believe that hypertree-depth is a very natural notion with potential applications in other areas, such as databases or constraint satisfaction.

Moreover, we introduced minors in hypergraphs and hypergraph pairs and studied their properties. In particular, neither generalised hypertree-width, nor generalised hyperpath-width, nor hypertree-depth increase under taking minors. Within the new framework of minors in hypergraph pairs, many open questions arise. Since testing for minors in hypergraph pairs may be NP-hard, a natural question to ask is, for which kinds of hypergraph pairs the problem is in PTIME. (For example, this is obviously the case for pairs as in Remark 20.)

A similar question arises for the minor ordering $\preccurlyeq$ in hypergraph pairs (cf. [5]): Which classes of hypergraph pairs have no infinite $\preccurlyeq$-antichains?

Let $\mathcal{C}$ be a class of graphs closed under taking minors, and let $\mathcal{O}$ be the obstruction set for $\mathcal{C}$ (i.e. $\mathcal{O}$ is the set of $\preccurlyeq$-minimal graphs that are not in $\mathcal{C}$). Robertson and Seymour showed that $\mathcal{O}$ is *finite*. Similarly, for a class $\mathcal{C}'$ of hypergraph pairs that is closed under taking minors, we let the *obstruction set* of $\mathcal{C}'$ be the set of all $\preccurlyeq$-minimal hypergraph pairs that are not in $\mathcal{C}'$.

Example 28 *Let $\mathcal{C}$ be a class of graphs closed under taking minors with obstruction set $\mathcal{O}$. Then (G, H), with $G \in \mathcal{C}$ and H any hypergraph satisfying $V(G) = V(H)$, is closed under taking minors, and it has a* finite *obstruction set $\{(G, H) \mid G \in \mathcal{O}, H \text{ hypergraph with } V(H) = V(G)\}$.*

We may ask which classes $\mathcal{C}'$ of hypergraph pairs that are closed under taking minors have a *finite* obstruction set? What is the obstruction set for the hypergraph pairs of generalised hypertree-width (hypertree-depth) at most 1?

Another way to get finiteness properties (at least for more classes) might be a weakening of our minor relation, e.g. by allowing more operations, while still preserving width and game monotonicities.

References

[1] Isolde Adler. Width functions for hypertree decompositions. PhD Thesis, Freiburg, 2006.

[2] Isolde Adler. Tree-related widths of graphs and hypergraphs. *SIAM J. Discrete Math.*, 22(1):102–123, 2008.

[3] Jianer Chen and Fedor V. Fomin, editors. *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, volume 5917 of *Lecture Notes in Computer Science*. Springer, 2009.

[4] Reinhard Diestel. *Graph Theory.* Springer, fourth edition, 2010.

[5] Michael R. Fellows, Danny Hermelin, and Frances A. Rosamond. Well-quasi-orders in subclasses of bounded treewidth graphs. In Chen and Fomin [3], pages 149–160.

[6] Michael R. Fellows, Jan Kratochvíl, Martin Middendorf, and Frank Pfeiffer. The complexity of induced minors and related problems. *Algorithmica*, 13(3):266–282, 1995.

[7] Robert Ganian, Petr Hliněný, Joachim Kneis, Alexander Langer, Jan Obdržálek, and Peter Rossmanith. On digraph width measures in parameterized algorithmics. In Chen and Fomin [3], pages 185–197.

[8] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width. *J. Comput. Syst. Sci.*, 66(4):775–808, 2003.

[9] Jiří Matoušek, Jaroslav Nešetřil, and Robin Thomas. On polynomial time decidability of induced-minor-closed classes. *Comment. Math. Univ. Carolin.*, 29(4):703–710, 1988.

[10] Zoltán Miklós. Understanding tractable decompositions for constraint satisfaction. PhD Thesis, University of Oxford, 2008.

[11] Jaroslav Nešetřil and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *Eur. J. Comb.*, 27(6):1022–1041, 2006.

[12] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion I. Decompositions. *Eur. J. Comb.*, 29(3):760–776, 2008.

[13] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion II. Algorithmic aspects. *Eur. J. Comb.*, 29(3):777–791, 2008.

[14] Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion III. Restricted graph homomorphism dualities. *Eur. J. Comb.*, 29(4):1012–1024, 2008.

[15] Neil Robertson and Paul D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995.

[16] Neil Robertson and P.D Seymour. Graph minors. iii. planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984.