Clique trees of infinite locally finite chordal graphs

Florian Lehner* & Christoph Temmel

Abstract

We investigate clique trees of infinite, locally finite chordal graphs. Our key tool is a bijection between the set of clique trees and the product of local families of finite trees. This enables us to enumerate all clique trees of a chordal graph. It also induces a local projection onto clique trees of finite chordal graphs, allowing us to lift various classic properties of clique trees of finite graphs to infinite clique trees.

1 Introduction

A chordal graph is a graph, where every cycle of length greater than three contains a chord, i.e. an edge connecting two non-consecutive vertices along the cycle. Chordal graphs are a classic object in graph theory and computer science [BP93]. They are equivalent to the class of graphs representable as a family of subtrees of a tree [Gav74, Hal84]. Each finite and connected chordal graph has natural representations of this form, with the trees being a subclass of the spanning graphs of its clique graph. These trees are called clique trees. There are a number of characterisations of clique trees among all spanning trees of the clique graph. They relate various properties of a clique tree to minimal vertex separators of the original graph, or maximality with respect to particular edge weights in the clique graph, or properties of paths in the tree, among others.

The present paper investigates clique trees of infinite, locally finite chordal graphs. We first prove the existence of at least one clique tree. Classic proofs of the various properties of clique trees often rely heavily on the finiteness of the setting. All of the known characterising properties are either not sensible in the infinite setting (as the maximality with respect to edge weights), or are of unbounded range (running intersection property of paths), or have at least overlapping constraints.

Our core contribution is a local partition of the edge set of the clique graph and a corresponding set of constraints, one for each element of the partition, which a clique tree has to fulfil. Each constraint only depends on the edges within its partition element, whence the constraints can be satisfied or violated independently from each other. This allows a local construction of a clique tree

*Florian Lehner acknowledges the support of the Austrian Science Fund (FWF), project W1230-N13.

by fixing a satisfying subset of the edges in each element of the partition. A characterisation of all clique trees is possible via a bijection with the product of the local choices.

In the case of a finite chordal graph, our characterisation permits an enumeration of the clique trees. It turns out that this enumeration is equivalent to a prior enumeration via a local partitioning of constraints by Ho and Lee [HL89]. Their partition is indexed by the minimal vertex separators of the chordal graph. We use a different approach based on families of cliques and recover the minimal vertex separators a posteriori. Specifically, the intersections of the cliques in a clique family is a minimal vertex separator, and vice-versa.

We derive classic properties of clique trees for infinite graphs from the above local decomposition property. A finer analysis of the structures appearing in the local decomposition points out connections with minimal vertex separators and the reduced clique graph [GHP95].

The structure of this paper is as follows: Section 2 introduces basic notation, clique trees and clique families. Section 3 contains our existence and characterisation theorems for clique trees. Section 4 discusses counting and enumerating the clique trees and section 5 derives the classic properties of clique trees. Section 6 contains the proofs of the statements from section 3.

2 Notation and basic properties

2.1 Graphs

Throughout the present work, we only consider locally finite graphs. Let $G = (V, E)$ be a graph and $W \subseteq V$. Denote by $G[W]$ the induced subgraph of G with vertex set W. Contracting the set W into a single vertex yields the graph G/W. It may contain multiple edges and loops, even if the graph G did not. If $V_1, V_2, \dots, V_k$ are disjoint subsets of V, then $G/\{V_1, V_2, \dots, V_k\}$ denotes the graph resulting from G by contracting each V_i to a single vertex, where the order of contractions has no influence on the final result. For an equivalence relation $\sim$ on V, denote by $G/\sim$ the graph resulting from contracting each equivalence class with respect to $\sim$.

We call a set finite $W \subseteq V$ *complete*, iff $G[W]$ is a complete graph on W. A *clique* is a maximal complete set of vertices of G. Denote by $\mathcal{C}_G$ the set of complete subsets of V and by $\mathcal{M}_G$ the set of all cliques of G. The *clique graph* $\mathbf{M}_G$ of G has vertex set $\mathcal{M}_G$ and an edge for every pair of cliques with non-empty intersection.

A tree T is a connected and acyclic graph. For two vertices $v, w \in T$, there is a unique path $P_T(v, w)$ in T. A subgraph of G is spanning, iff it has the same vertex set as G. The *set of spanning trees* of G is $\mathcal{T}_G$. We admit the *empty graph*, which is a graph without vertices, and consider it a tree. Also, the only spanning tree of the empty graph is the empty graph.

2.2 Chordal graphs and subtree representations

Our main reference for basic facts about chordal graphs is [BP93]. A *chordal graph* has no cycle of length greater than 3. In other words, every closed path of length greater than 3 has a *chord*, an edge connecting two non-consecutive vertices of the cycle. Throughout this work, we assume that chordal graphs are connected.

Let T be a tree and denote by $\mathcal{T}$ the family of subtrees of T. A function $t\colon V \to \mathcal{T}$ is a *subtree representation of* G *on* T, iff $v_1v_2 \in E \Leftrightarrow t(v_1) \cap t(v_2) \neq \emptyset$.

A graph is chordal, iff is has a subtree representation on some tree [Gav74, Hal84]. This does not hold for general countable, non locally-finite graphs [Hal84]. If G is finite, there is combinatorial representation [Gav74], where T is a spanning tree of $\mathbf{M}_G$ and $t(v) := T[\{M \in \mathcal{M}_G \mid v \in M\}]$. We call T a *clique tree* of the chordal graph G. The *set of all clique trees* $\mathcal{TC}_G$ of G is the set of all all spanning trees $T \in \mathcal{T}_{\mathbf{M}_G}$ fulfilling

$$\forall v \in V: \quad T[\{M \in \mathcal{M}_G \mid v \in M\}] \text{ is a tree.} \tag{1}$$

2.3 The lattice of clique families

Let $W \subseteq V$. The *clique family generated by* W is

$$\mathcal{K}(W) := \{M \in \mathcal{M}_G \mid W \subseteq M\}\,.$$

The *set of clique families* associated with G is

$$\mathcal{L}_G := \{\mathcal{K}(W) \mid W \subseteq V\}\,. \tag{2}$$

Generation is *anti-monotone*:

$$W \subseteq W' \Rightarrow \mathcal{K}(W') \subseteq \mathcal{K}(W)\,. \tag{3}$$

If $W \notin \mathcal{C}_G$, then $\mathcal{K}(W) = \emptyset$. The largest clique family is $\mathcal{K}(\emptyset) = \mathcal{M}_G$. It is infinite and the only infinite clique family, iff G is infinite itself. The set of *finite clique families* is

$$\mathcal{L}_G^f := \{\mathcal{K} \in \mathcal{L}_G : |\mathcal{K}| < \infty\}\,. \tag{4}$$

For infinite G, $\mathcal{L}_G^f = \mathcal{L}_G \setminus \{\mathcal{M}_G\}$, and, for finite G, $\mathcal{L}_G^f = \mathcal{L}_G$.

By abuse of notation, we write $\mathcal{K}(v)$ instead of $\mathcal{K}(\{v\})$, for $v \in V$. These particular clique families are building blocks for all other clique families:

$$\mathcal{K}(W) = \bigcap_{v \in W} \mathcal{K}(v)\,. \tag{5}$$

For a non-empty clique family $\mathcal{K}$, every connected vertex subset $C \in \mathcal{C}_G$ with $\mathcal{K}(C) = \mathcal{K}$ is a *generator* of $\mathcal{K}$. The *set of generators* of a clique family $\mathcal{K}$ is $\mathcal{C}(\mathcal{K})$. A generator C of $\mathcal{K}$ is *minimal/maximal*, iff it is so for set inclusion in $\mathcal{C}(\mathcal{K})$. There may be more than one minimal generator (see example 2.2). There is a *unique maximal generator*:

$$C(\mathcal{K}) := \bigcap_{M\in\mathcal{K}} M = \bigcup_{C\in\mathcal{C}(\mathcal{K})} C\,. \tag{6}$$

In particular, for each non-empty clique family $\mathcal{K}$, we have

$$\mathcal{K}(C(\mathcal{K})) = \mathcal{K}\,. \tag{7}$$

Proposition 2.1. *Let $\mathcal{K}$ and $\mathcal{K}'$ be clique families. Their sets of generators coincide, iff the clique families do so, and are disjoint otherwise.*

Proof. We have the equivalence relation $C \sim C' \Leftrightarrow \mathcal{K}(C) = \mathcal{K}(C')$ on $\mathcal{C}_G$. □

Example 2.2. Let $G := (\{v_1, v_2, v_3, v_4\}, \{(v_1, v_2), (v_2, v_3), (v_1, v_3), (v_3, v_4)\})$. The cliques are $K := \{v_1, v_2, v_3\}$ and $L := \{v_3, v_4\}$. The clique families, their generators and maximal generators are:

$\mathcal{K}$	$\mathcal{C}(\mathcal{K})$	$C(\mathcal{K})$
$\{K, L\} = \mathcal{M}_G$	$\{\emptyset, \{v_3\} = K \cap L\}$	$\{v_3\}$
$\{K\}$	$\{\{v_1\}, \{v_2\}, \{v_1, v_2\}, \{v_2, v_3\}, \{v_1, v_3\}, K\}$	K
$\{L\}$	$\{\{v_4\}, L\}$	L
$\emptyset$	everything else missing from $\mathcal{P}(V)$	V

The clique family $\{K\}$ has two minimal generators.

The clique families form a lattice with respect to set inclusion. All the chains in the lattice are finite and the lattice is both atomistic and co-atomistic. We use these facts later on, to reason inductively over this lattice.

Proposition 2.3. *$\mathcal{L}_G$ is a lattice with respect to set inclusion.*

Proof. For $\mathcal{K}_1, \mathcal{K}_2 \in \mathcal{L}_G$, define

$$\mathcal{K}_1 \vee \mathcal{K}_2 := \mathcal{K}(C(\mathcal{K}_1) \cap C(\mathcal{K}_2))\,,$$
$$\mathcal{K}_1 \wedge \mathcal{K}_2 := \mathcal{K}(C(\mathcal{K}_1) \cup C(\mathcal{K}_2))\,.$$

We claim that this is indeed the supremum and infimum of $\mathcal{K}_1$ and $\mathcal{K}_2$ in $\mathcal{L}_G$ with respect to inclusion.

For the supremum property observe that each $M \in \mathcal{K}_1$ contains $C(\mathcal{K}_1)$ and hence also $C(\mathcal{K}_1) \cap C(\mathcal{K}_2)$. Thus M must also be contained in $\mathcal{K}_1 \vee \mathcal{K}_2$. The same is true for every $M \in \mathcal{K}_2$, so $\mathcal{K}_1 \vee \mathcal{K}_2$ is a common upper bound for $\mathcal{K}_1$ and $\mathcal{K}_2$. To show that it is the least upper bound let $\mathcal{K}$ be an arbitrary upper bound. Then

$$C(\mathcal{K}) = \bigcap_{M\in\mathcal{K}} M \subseteq \bigcap_{M\in\mathcal{K}_1\cup\mathcal{K}_2} M = C(\mathcal{K}_1) \cap C(\mathcal{K}_2)\,.$$

Hence, by the same argument as above, each $M \in \mathcal{K}_1 \vee \mathcal{K}_2$ must be contained in $\mathcal{K}$ showing that $\mathcal{K}_1 \vee \mathcal{K}_2 \subseteq \mathcal{K}$.

A dual argument shows that the definition of the infimum is correct. □

Note that $\mathcal{K}_1 \wedge \mathcal{K}_2 = \mathcal{K}_1 \cap \mathcal{K}_2$. In particular, $\mathcal{L}_G$ is closed under intersections. Furthermore, the lattice has the following properties:

- It is bounded with greatest element $\mathcal{M}_G = \mathcal{K}(\emptyset)$ and smallest element $\emptyset = \mathcal{K}(V)$. By the remark following (4), all intervals not containing $\mathcal{M}_G$ and, hence, all chains in $\mathcal{L}_G$ are finite.
- $\mathcal{L}_G$ is an *atomistic lattice*, the atoms being of the form $\{M\} = \mathcal{K}(M)$, for $M \in \mathcal{M}_G$. Since these sets are singletons, there is no $\mathcal{K} \in \mathcal{L}_G$ with $\emptyset \subsetneq \mathcal{K} \subsetneq \{M\}$.

  Every clique family $\mathcal{K} \neq \mathcal{M}_G$ is the supremum of a finite set of atoms (see (6) and 7):

$$\mathcal{K} = \mathcal{K}(C(\mathcal{K})) = \mathcal{K}(\bigcap_{M\in\mathcal{K}} M) = \mathcal{K}(\bigcap_{\{M\}\subseteq\mathcal{K}} M) = \bigvee_{\{M\}\subseteq\mathcal{K}} \{M\}\,.$$

- $\mathcal{L}_G$ is a *co-atomistic lattice*, the co-atoms being of the form $\mathcal{K}(v)$, for $v \in V$. There is no $\mathcal{K} \in \mathcal{L}_G$ such that $\mathcal{K}(v) \subsetneq \mathcal{K} \subsetneq \mathcal{M}_G$. In this case, $C(\mathcal{K}) = \emptyset$ would hold, implying that $\mathcal{K} = \mathcal{M}_G$. Each $\mathcal{K} \in \mathcal{L}_G$ is the infimum of finitely many co-atoms (see (5)):

$$\mathcal{K} = \mathcal{K}(C(\mathcal{K})) = \mathcal{K}(\bigcup_{v\in C(\mathcal{K})} \{v\}) = \bigwedge_{v\in C(\mathcal{K})} \mathcal{K}(v)\,.$$

3 Main result

3.1 Existence of clique trees of infinite chordal graphs

We investigate clique trees of infinite chordal graphs and extend the combinatorial construction of a subtree representation for finite graphs [Gav74]. A sensible definition of an infinite clique tree encompasses the fact that, for every induced subgraph, a corresponding induction on the cliques yields a clique tree of the induced subgraph. This gives a straightforward extension of the definition from the finite case.

Let G be infinite. A spanning tree $T \in \mathcal{T}_{\mathbf{M}_G}$ is a clique tree of G, iff it fulfils (1), i.e. if every $\mathcal{K}(v)$ induces a tree.

Proposition 3.1. *Every infinite and locally finite chordal graph has a clique tree.*

The proof of proposition 3.1 is in section 6.1.

3.2 The characterisation via clique families

Let G be a (possibly infinite) chordal graph, let $\mathcal{K} \in \mathcal{L}_G$, and denote by $\mathcal{L}_G^<(\mathcal{K})$ the *strict subfamilies of* $\mathcal{K}$, i.e. the set

$$\mathcal{L}_G^<(\mathcal{K}) := \{\mathcal{K}' \in \mathcal{L}_G : \mathcal{K}' \subsetneq \mathcal{K}\}\,. \tag{8}$$

Define a graph $\Gamma_\mathcal{K}$ with vertex set $\mathcal{K}$ and an edge $KL \in \Gamma_\mathcal{K}$, iff there is $\mathcal{K}' \in \mathcal{L}_G^<(\mathcal{K})$ with $K \cap L \in \mathcal{C}(\mathcal{K}')$. It follows, that $\Gamma_\mathcal{K}$ is a subgraph of $\mathbf{M}_G[\mathcal{K}]$.

Denote by $\sim_{\mathcal{K}}$ the equivalence relation whose classes are the connected components of $\Gamma_{\mathcal{K}}$ and by $[K]_{\sim_{\mathcal{K}}}$ the equivalence class of K with respect to the relation $\sim_{\mathcal{K}}$.

Theorem 3.2. *Let G be a locally finite chordal graph. A spanning subgraph T of $\mathbf{M}_G$ is a clique tree of G, iff it fulfils one of the following equivalent conditions:*

$$\forall \mathcal{K} \in \mathcal{L}_G : \quad T[\mathcal{K}] \text{ is a tree,} \tag{9a}$$
$$\forall \mathcal{K} \in \mathcal{L}_G : \quad T[\mathcal{K}]/\sim_{\mathcal{K}} \text{ is a tree.} \tag{9b}$$

If one takes $\mathcal{K} = \mathcal{M}_G$, then (9a) says that $T[\mathcal{M}_G] = T$ is a tree. In (9b), this fact is not so obvious, but follows from an inductive bottom-up construction over the lattice of clique families. The proof of theorem 3.2 is in section 6.2. Theorem 4.2 in the following sections shows that the conditions in (9b) are in fact non-overlapping.

4 Edge bijections and enumerating the set of clique trees

In the present section we take another look at the characterisation of clique trees via clique families in theorem 3.2 and disentangle the seemingly overlapping local conditions into disjoint local conditions. The key is differentiating between the restrictions imposed by a clique family and the restrictions imposed by its strict subfamilies. In this way, every restriction is dependent of and attached to a unique member of the lattice of clique families. We state this partition of the constraints in theorem 4.2 and apply it to counting and enumerating clique trees in the remainder of the section.

For $\mathcal{K} \in \mathcal{L}_G$, define a graph $\Xi_{\mathcal{K}}$ with vertex set $\mathcal{K}$ and an edge $KL \in \Xi_{\mathcal{K}}$, iff $KL \in \mathbf{M}_G$ and $K \cap L \in \mathcal{C}(\mathcal{K})$, equivalent to $\mathcal{K}(K \cap L) = \mathcal{K}$. The graphs $\Gamma_{\mathcal{K}}$ and $\Xi_{\mathcal{K}}$ are edge-dual subgraphs of $\mathbf{M}_G[\mathcal{K}]$: they have the same vertex set $\mathcal{K}$ and partition the edges of $\mathbf{M}_G[\mathcal{K}]$ into two disjoint sets.

Let $\Delta_{\mathcal{K}} := \Xi_{\mathcal{K}}/\sim_{\mathcal{K}}$. The edges of $\Delta_{\mathcal{K}}$ are injectively labelled by edges from $\Xi_{\mathcal{K}}$, as subgraph of $\mathbf{M}_G$.

Proposition 4.1. *The graph $\Delta_{\mathcal{K}}$ is complete, possibly with multi-edges and multi-loops. For each edge in $\Delta_{\mathcal{K}}$, its edge label KL fulfils $K \cap L = C(\mathcal{K})$, i.e. all the intersections of the edge labels coincide with the maximal generator.*

Proof. If KL is not an edge of $\Xi_{\mathcal{K}}$, then $K \sim_{\mathcal{K}} L$, whence they are identified in $\Delta_{\mathcal{K}}$. This shows the completeness of $\Delta_{\mathcal{K}}$.

By the definition of $\Xi_{\mathcal{K}}$, $K \cap L \in \mathcal{C}(\mathcal{K})$ and thus $K \cap L \subseteq C(\mathcal{K})$. On the other hand, $C(\mathcal{K}) = \bigcap_{K' \in \mathcal{K}} K' \subseteq K \cap L$. Whence, $K \cap L = C(\mathcal{K})$. □

If G is infinite, then $\mathcal{M}_G = \mathcal{K}(\emptyset)$ is the only infinite clique family and Δ_V consists of a single vertex and has no loops.

Theorem 4.2. *There is a bijection between the edges of* $\mathbf{M}_G$ *and the disjoint union over all clique families* $\mathcal{K}$ *of edges of* $\Xi_\mathcal{K}$*. Via edge-labelling, this extends to the disjoint union of edges of* $\Delta_\mathcal{K}$*.*

$$\mathbf{M}_G \overset{edges}{=} \biguplus_{\mathcal{K}\in\mathcal{L}_G} \Xi_\mathcal{K} \overset{edge\text{-}labelling}{=} \biguplus_{\mathcal{K}\in\mathcal{L}_G} \Delta_\mathcal{K}\,. \tag{10}$$

Proof of theorem 4.2. Choose $KL \in \mathbf{M}_G$ and $\mathcal{K} \in \mathcal{L}_G^f$. We know that $KL \in \Xi_\mathcal{K}$, iff $K \cap L \in \mathcal{C}(\mathcal{K})$, equivalent to $\mathcal{K}(K \cap L) = \mathcal{K}$. Proposition 2.1 and the identification between edges of $\Xi_\mathcal{K}$ and edge-labels of $\Delta_\mathcal{K}$ imply that we may partition the set of edges U according to the clique family:

$$U := \bigcup_{\mathcal{K}\in\mathcal{L}_G^f} \Xi_\mathcal{K} \overset{\text{edges}}{=} \biguplus_{\mathcal{K}\in\mathcal{L}_G^f} \Xi_\mathcal{K} \overset{\text{edge-labelling}}{=} \biguplus_{\mathcal{K}\in\mathcal{L}_G^f} \Delta_\mathcal{K}\,.$$

$\mathbf{M}_G \overset{\text{edges}}{\subseteq} U$: If $KL \in \mathbf{M}_G$, then $K \cap L \neq \emptyset$ and KL is an edge in $\Xi_{\mathcal{K}(K\cap L)}$.

$U \overset{\text{edges}}{\subseteq} \mathbf{M}_G$: If $\mathcal{K} \in \mathcal{L}_G^f$ and $KL \in \Xi_\mathcal{K}$, then $\emptyset \neq K \cap L$ and $KL \in \mathbf{M}_G$. □

Theorem 4.2 tells us that condition (9b) factorises into a series of independent conditions. The characterisation (9b) of clique trees reduces the problem of choosing a clique tree to the problem of choosing a spanning tree of $\mathbf{M}_G[\mathcal{K}]/\sim_\mathcal{K}$, for each $\mathcal{K} \in \mathcal{L}_G$. Theorem 4.2 ensures that these choices are independent of each other. This is in contrast to characterisations (1) and (9a), where each edge might be subject to constraints from several clique families.

Corollary 4.3. *Let* G *be a locally finite chordal graph. Then* $\mathcal{TC}_G$*, the set of clique trees of* G*, is in bijection with*

$$\prod_{\mathcal{K}\in\mathcal{L}_G} \mathcal{T}_{\Delta_\mathcal{K}}\,. \tag{11}$$

Proof. Using the bijection from theorem 4.2, we decompose the edges of a clique tree $T \in \mathcal{TC}_G$ into disjoint sets, indexed by $\mathcal{L}_G^f$. For $\mathcal{K} \in \mathcal{L}_G^f$, statement (9b) tell us that $T_\mathcal{K}$ must be a spanning tree of $\Delta_\mathcal{K}$.

Conversely, select a spanning tree $T_\mathcal{K} \in \mathcal{T}_{\Delta_\mathcal{K}}$, for each $\mathcal{K} \in \mathcal{L}_G^f$, and let E be the union of their edge labels. By theorem 4.2, no edge in E appears twice as an edge-label of a $T_\mathcal{K}$. Let T be the subgraph of $\mathbf{M}_G$ induced by E. By (9b) it is a clique tree. □

A similar bijection to (11) between the clique trees of a finite chordal graph and a product of trees indexed by the minimal vertex separators (see section 5.3) of the graph is known [HL89]. The bijection is in fact the same, and we make the exact relation clear in corollary 5.10. An immediate consequence of (11) is a formula for the number of clique trees of a finite chordal graph:

$$|\mathcal{TC}_G| = \prod_{\mathcal{K}\in\mathcal{L}_G} |\mathcal{T}_{\Delta_\mathcal{K}}|\,. \tag{12}$$

The value of $|\mathcal{T}_{\Delta_\mathcal{K}}|$ is explicitely given in terms of the structure of $\Delta_\mathcal{K}$ as a complete multigraph in [HL89].

Corollary 4.4. *Let G be a finite chordal graph with maximal degree D. One can loop through all clique trees of G with only $O(|V|)$ memory.*

The restriction amounts to a sequential processing of the clique trees.

Proof. As the degree is uniformly bounded, so is the size $\mathcal{K}$ and $\mathcal{T}_{\Delta_\mathcal{K}}$, for every $\mathcal{K} \in \mathcal{L}_G^f$. Furthermore, as each vertex is only contained in a uniformly bounded number of cliques and, hence, clique families, the size of $\mathcal{L}_G^f$ is linear in $|V|$. □

For infinite chordal graphs, we have a dichotomy in the number of clique trees:

Corollary 4.5. *Let G be an infinite chordal graph. It has either finitely or $\aleph_1$ many clique trees.*

Proof. We look at $\{|\mathcal{T}_{\Delta_\mathcal{K}}|\}_{\mathcal{K}\in\mathcal{L}_G^f}$. If a finite number of these numbers are greater than 1, then the number of clique trees is finite. If an unbounded number of these numbers are greater than 1, then there are at least a countable number of independent choices between more than two spanning trees and the number of clique trees is uncountable. □

5 Classic properties of clique trees

We discuss classic properties of clique trees: the running intersection property, the maximal weight spanning tree property and the relation with minimal vertex separators and the reduced clique graph. We generalise several known results for finite graphs to the infinite graphs. For $\mathcal{K} \in \mathcal{L}_G^f$, let $V(\mathcal{K}) := \{v \in K \in \mathcal{K}\}$ be the set of vertices covered by $\mathcal{K}$. We start with a projection statement, which is our tool to lift properties from the finite to the infinite setting.

Lemma 5.1. *A spanning tree $T \in \mathcal{T}_{\mathbf{M}_G}$ is a clique tree of G, iff, $T[\mathcal{K}]$ is a clique tree of $G[V(\mathcal{K})]$, for every $\mathcal{K} \in \mathcal{L}_G^f$.*

Proof. The clique families of $G[V(\mathcal{K})]$ are exactly $\{\mathcal{K}\} \uplus \mathcal{L}_G^<(\mathcal{K})$. Hence, the bijection theorem 4.2 works the same way on all graphs considered. □

5.1 The running intersection property

A tree $T \in \mathcal{T}_{\mathbf{M}_G}$ fulfils the *running intersection property*, iff

$$\forall K, L \in \mathcal{M}_G : \quad \forall K' \in P_T(K, L) : \quad K \cap L \subseteq K' . \tag{13}$$

Lemma 5.2 ([BFMY83])**.** *Let G be a finite, chordal graph and $T \in \mathcal{T}_{\mathbf{M}_G}$. Then $T \in \mathcal{TC}_G$, iff it has the running intersection property.*

Corollary 5.3. *A spanning tree T of $\mathbf{M}_G$ is a clique tree of G, iff it has the running intersection property.*

Proof. A tree $T \in \mathcal{T}_{\mathbf{M}_G}$ fulfils the running intersection property, iff it fulfils the running intersection property for cliques K, L, with $K \cap L \neq \emptyset$. In particular, for such pairs of cliques, $\mathcal{K}(K \cap L) \in \mathcal{L}_G^f$ and by (9a) the path $P_T(K, L)$ lies in $\mathbf{M}_G[\mathcal{K}(K \cap L)]$.

The corollary follows from lemma 5.1 and the statement for the finite chordal graphs $G[V(\mathcal{K})]$. □

5.2 The maximal weight spanning tree property

Let w be the weight function on the edges of $\mathbf{M}_G$ given by $w(KL) := |K \cap L|$. Extend the weight function to $T \in \mathcal{T}_{\mathbf{M}_G}$, by setting $w(T) := \sum_{KL \in T} w(KL)$. Another classic characterisation of clique trees is:

Lemma 5.4 ([BG81])**.** *Let G be a finite, chordal graph and $T \in \mathcal{T}_{\mathbf{M}_G}$. Then $T \in \mathcal{TC}_G$, iff*

$$T \in \operatorname{argmax}\{w(S) \mid S \in \mathcal{T}_{\mathbf{M}_G}\}\,. \tag{14}$$

Condition (14) makes no sense in the infinite case. We can localise (14), though:

Corollary 5.5. *The tree $T \in \mathcal{T}_{\mathbf{M}_G}$ is a clique tree, iff*

$$\forall \mathcal{K} \in \mathcal{L}_G^f : \quad T[\mathcal{K}] \in \operatorname{argmax}\{w(S) \mid S \in \mathcal{T}_{\mathbf{M}_G[\mathcal{K}]}\}\,. \tag{15}$$

Proof. Observing that $\mathbf{M}_G[\mathcal{K}] = \mathbf{M}_{G[V(\mathcal{K})]}$, we apply the projection lemma 5.1 and the statement for the finite chordal graphs $G[V(\mathcal{K})]$. □

5.3 Minimal separators and the reduced clique graphs

A non-empty subset $W \subseteq V$ is a *separator*, iff $G[V \setminus W]$ has more than one connected component. It is a *minimal separator*, iff it is minimal with respect to inclusion.

Lemma 5.6 ([Dir61])**.** *A (possibly infinite) graph is chordal, iff every minimal separator is complete.*

Every minimal separator C separates two vertices adjacent to all of C. In particular, C is a minimal separator in $G[V(\mathcal{K}(C))]$.

The *reduced clique graph* [GHP95] $\mathbf{R}_G$ of G is the subgraph of $\mathbf{M}_G$ retaining those edges KL with $K \cap L$ a minimal separator and deleting the others. The importance of $\mathbf{R}_G$ comes from:

Lemma 5.7 ([GHP95])**.** *Let G be a finite chordal graph. The union of all clique trees of G is $\mathbf{R}_G$.*

Corollary 5.8. *The union of clique trees of a chordal graph G is $\mathbf{R}_G$.*

Proof. We apply the projection lemma 5.1 and the statement for the finite case in lemma 5.7, minding the remark after lemma 5.6. □

The following lemma has been originally formulated only for finite graphs, but its proof is also valid in the infinite case:

Lemma 5.9 ([HL89])**.** *For $T \in \mathcal{TC}_G$, let $\mathcal{C}_T$ be the multiset of intersections of edge-labels of T. The multiset $\mathcal{C}_T$ is independent of T.*

Corollary 5.10. *A subset $\emptyset \neq W \subsetneq V$ is a minimal separator of G, iff W is the maximal generator of some clique family, i.e. $W = C(\mathcal{K}(W))$. In particular, W must be complete and finite.*

Proof. By corollary 5.8, every intersection of an edge label of a clique tree is a separator. By lemma 5.9, each minimal separator appears as intersection of at least one edge-label of every clique tree of T. By corollary 4.3 and proposition 4.1, the intersections of edge labels are exactly the maximal generators of finite clique families. □

6 Proofs

6.1 Proof of existence of infinite clique trees

We prove proposition 3.1 via a compactness argument, which is a rather standard approach in infinite graph theory (c.f. [Die05, Chapter 8.1]). Arguments of this type can often be used to obtain a result for infinite graphs from its finite counterpart.

Proof of proposition 3.1 by compactness. Let G be the graph. Let $(v_n)_{n\in\mathbb{N}}$ be an enumeration of the vertices of G such that v_n is connected to at least one v_i for $i < n$. Denote by G_n the subgraph of G induced by $\bigcup_{i\leq n}\bigcup_{K\in\mathcal{K}(v_i)} K$, that is, G_n contains all maximal cliques which contain at least one of $v_1,\dots,v_n$.

Since G_n is a induced subgraph of a chordal graph it must be chordal as well. It is also connected. By construction every clique in G_n corresponds to a clique in G, hence $\mathbf{M}_{G_n}$ is a subgraph of $\mathbf{M}_G$. Since G_n is finite, we know that we can find a clique tree T_n of G_n, that is, T_n is a spanning tree of $\mathbf{M}_{G_n}$ such that $v \mapsto T_n[\mathcal{K}(v)]$ defines a subtree representation of G_n.

Consider T_n as a subgraph of $\mathbf{M}_G$ and define a subgraph T of $\mathbf{M}_G$ as follows. By local finiteness of G and thus $\mathbf{M}_G$, there is an infinite subsequence T_n^1 of $(T_n)_{n\in\mathbb{N}}$ of trees which contain the same edges of $\mathbf{M}_G[\mathcal{K}(v_1)]$. Add those edges to T. Then choose an infinite sub-subsequence T_n^2 of T_n^1 such that all elements of the sequence T_n^2 contain the same edges of $\mathbf{M}_G[\mathcal{K}(v_2)]$. Proceed inductively.

We have to check that T is a tree and that $T[\mathcal{K}(v)]$ is a subtree, for every $v \in V$. The last property holds by construction. The trees corresponding to v and w overlap, iff $\mathcal{K}(v)\cap\mathcal{K}(w) \neq \emptyset$, which is the case, iff vw is an edge. Hence T is connected because G was assumed to be connected. If T contains a cycle C, then it lies in $\mathbf{M}_{G_{n_0}}$, for some n_0. Hence C is a cycle in $T_1^{n_0}$, a contradiction. □

6.2 Proof of the clique family characterisation

Recall the definition of the strict subfamilies $\Gamma_\mathcal{K}$ of a clique family $\mathcal{K}$ and the equivalence relation $\sim_\mathcal{K}$ from section 3.2. For $\emptyset \neq \mathcal{K}' \in \Gamma_\mathcal{K}$ and $K \in \mathcal{K}$, we either have $\mathcal{K}' \subseteq [K]_{\sim_\mathcal{K}}$ or $\mathcal{K}' \cap [K]_{\sim_\mathcal{K}} = \emptyset$.

The major issue in the proof of theorem 3.2 is to start from (9b). In this case we build the tree bottom up, starting with the clique families $\{M\}$, for $M \in \mathbf{M}_G$. An important issue in later stages of the construction, for bigger clique families, is that overlapping constructions on strict subfamilies play well together. As the construction only adds edges, the main problem is not connectedness, but the possible introduction of cycles. Proposition 6.1 deals with this: for every connected component $[K]_{\sim_\mathcal{K}}$ of $\Gamma_\mathcal{K}$, it asserts that there are no cycles introduced by the bottom up construction of the tree on smaller clique families.

Proposition 6.1. *Assume that T is a subgraph of $\Gamma_\mathcal{K}$ with vertex set $[K]_{\sim_\mathcal{K}}$ such that $T[\mathcal{K}']$ is a tree for every $\mathcal{K}' \in \mathcal{L}_G^<(\mathcal{K})$ with $\mathcal{K}' \subseteq [K]_{\sim_\mathcal{K}}$. Then T is a tree.*

The proof of proposition 6.1 is technical and is in section 6.2.2.

A second tool in the proof of theorem 3.2 is contracting and decontracting subtrees of trees. The following propositions, whose proofs are section 6.2.1 allow us to do the needed surgery on trees:

Proposition 6.2. *Let V be the vertex set of a finite graph and let $V_1, V_2, \dots, V_k$ be disjoint subsets of V. Every choice of two of the following statements implies the third one:*

$$G \text{ is a tree,} \tag{16a}$$

$$1 \leq i \leq k: \quad G/\{V_1, \dots, V_i\} \text{ is a tree,} \tag{16b}$$

$$\forall 1 \leq i \leq k: \quad G[V_i] \text{ is a tree.} \tag{16c}$$

Proposition 6.3. *Let $V =: V_1 \cup V_2$ be the vertex set of a tree T. If $T[V_1]$ and $T[V_2]$ are trees, then $T[V_1 \cap V_2]$ is also a tree.*

Equipped with these tools, we can prove our characterisation theorem:

Proof of theorem 3.2. (9a) $\Rightarrow$ (1): (9a) implies that $T[\mathcal{K}(v)]$ is a tree, for each $v \in G$, and that $T[\mathcal{M}_G] = T$ is a tree. This is just the definition of a clique tree.

(1) $\Rightarrow$ (9a): If $\mathcal{K} = \mathcal{M}_G$ or $\mathcal{K} = \mathcal{K}(v)$, for some vertex $v \in V$, then $T[\mathcal{K}]$ is a tree. Let $\emptyset \neq \mathcal{K} \in \mathcal{L}_G$ be arbitrary and assume that $T[\mathcal{K}]$ is not a tree. Assume that $\mathcal{K}$ is a maximal element of $\mathcal{L}_G$ with the property that $T[\mathcal{K}]$ is not a tree. Such an element exists, because there are only finitely many elements of $\mathcal{L}_G$ which are larger than $\mathcal{K}$. Hence, if $\mathcal{K}$ is not maximal with this property, choose $\mathcal{K}' \supsetneq \mathcal{K}$ such that $T[\mathcal{K}']$ is not a tree. Such a maximal family is neither empty (as $T[\emptyset]$ is a tree) nor induced by a single vertex. Let C be a minimal generator of $\mathcal{K}$, i.e., $C \subseteq C(\mathcal{K})$ and $\mathcal{K}(C) = \mathcal{K}$. The generator C contains at least two vertices. Therefore, for every $\emptyset \neq C' \subsetneq C$,

$$\emptyset \neq \mathcal{K} = \mathcal{K}(C') \cap \mathcal{K}(C \setminus C')\,.$$

Maximality of $\mathcal{K}$ implies that $T[\mathcal{K}(C')]$ and $T[\mathcal{K}(C \setminus C')]$ are trees. Proposition 6.3 implies that $T[\mathcal{K}]$ is a tree, too.

(9a) $\Rightarrow$ (9b): let $\mathcal{K} \in \mathcal{L}_G$. Proposition 6.1 together with the assumption that $T[\mathcal{K}']$ is a tree for every $\mathcal{K}'$ implies that $T[[K]_{\sim_\mathcal{K}}]$ is a tree, for every equivalence class with respect to the relation $\sim_\mathcal{K}$. If $\mathcal{K} \neq \mathcal{M}_G$, then there are only finitely many equivalence classes. Hence we can apply proposition 6.2 to show that $T[\mathcal{K}]/\sim_\mathcal{K}$ is a tree. For $\mathcal{K} = \mathcal{M}_G$, the connectedness of G implies that there is only one equivalence class. Thus proposition 6.1 implies directly (without application of proposition 6.2) that $T[\mathcal{K}]$ is a tree.

(9b) $\Rightarrow$ (9a): Assume that there is some $\mathcal{K} \in \mathcal{L}_G$ such that $T[\mathcal{K}]$ is not a tree. Choose $\mathcal{K}$ minimal with this property. This is possible because there are only finitely many elements of $\mathcal{L}_G$ which are smaller than $\mathcal{K}$. It follows from proposition 6.1 that $T[[K]_{\sim_\mathcal{K}}]$ is a tree for every equivalence class with respect to $\sim_\mathcal{K}$. Since there are only finitely many equivalence classes and $T[\mathcal{K}]/\sim_\mathcal{K}$ is a tree we can invoke proposition 6.2 to prove that $T[\mathcal{K}]$ is indeed a tree, which completes the proof of the theorem. □

6.2.1 Surgery on trees

This section contains technical results about the relation between subtrees obtained by inducing or contracting and the original tree and joining trees with common parts. The proofs of propositions 6.2 and 6.3 are also in this sections.

Proposition 6.4. *Let V be the vertex set of a finite graph G and let $W \subseteq V$. Every choice of two of the following statements implies the third one:*

$$G \text{ is a tree,} \tag{17a}$$
$$G/W \text{ is a tree,} \tag{17b}$$
$$G[W] \text{ is a tree.} \tag{17c}$$

Proof. Denote by $|G|$ and $\|G\|$ the number of vertices and edges of a graph G respectively. If G is a tree, then

$$|G| = \|G\| + 1 \tag{18}$$

holds. Contrarily, if (18) holds for a graph G and G is either acyclic or connected, then G is a tree.

For every graph G, the following identities hold:

$$|G| = |G/W| - 1 + |G[W]| \quad \text{and} \quad \|G\| = \|G/W\| + \|G[W]\| \tag{19}$$

If two of the three statements in (17) hold, then (18) holds for them. Combined with (19), this yields (18) for the third statement of (17). Thus, in all three cases, we only need to show the acyclity or connectedness of the third graph.

(17a) (and (17b)) imply (17c): Acyclicity is stable under taking inducing subgraphs.

(17a) (and (17c)) imply (17b): Connectedness is stable under contracting subgraphs.

(17c) and (17b) imply (17a): Every cycle in G is either contained in $G[W]$ or contracts to a cycle of G/W. □

Proof of proposition 6.2. Apply proposition 6.4 inductively. The key fact is that $G/\{V_1, \dots, V_l\}[V_j]$ is a tree, for all $1 \leq l < j \leq k$. □

Remark 6.5. Proposition 6.2 remains valid, if we consider locally finite graphs and countably many disjoint sets V_i. It can also be extended to nested contractions, as long as the nesting depth is finite. If the nesting depth is infinite, then the limit object is no longer a spanning tree, but a topological spanning tree.

Proof of proposition 6.3. If either one of $V_1 \setminus V_2$ or $V_2 \setminus V_1$ is empty, then $T[V_1 \cap V_2] = T[V_1]$ and $T[V_1 \cap V_2] = T[V_2]$ is a tree respectively. If $V_1 \cap V_2 = \emptyset$, then $T[V_1 \cap V_2]$ is the empty tree. Therefore, let $v \in V_1 \cap V_2$, $v_1 \in V_1 \setminus V_2$ and $v_2 \in V_2 \setminus V_1$. There is a unique path from v to v_1 and v_2 in T respectively. Thus, the edge v_1v_2 can not be in T, as it would create a cycle in T. Hence, there are no edges between $V_1 \setminus V_2$ and $V_2 \setminus V_1$ in T. As T and $T[V_2]$ are trees, proposition 6.4

implies that $T/V_2 = T[V_1]/V_1 \cap V_2$ is a tree, too. As $T[V_1]$ is a tree, another application of proposition 6.4 implies that $T[V_1][V_1 \cap V_2] = T[V_1 \cap V_2]$ is a tree, too. □

Proposition 6.6. *Let G be a graph with vertices V. Let $V =: V_1 \cup V_2$ be a non-disjoint union of V. Assume that $G[V_1]$, $G[V_2]$, and $G[V_1 \cap V_2]$ are trees, and that there are no edges connecting $V_1 \setminus V_2$ to $V_2 \setminus V_1$. Then G is a tree.*

Proof. As $G[V_1]$ and $G[V_1 \cap V_2]$ are trees, we apply proposition 6.4 to see that $G[V_1]/V_1 \cap V_2$ is a tree. There is no edge connecting $V_1 \setminus V_2$ to $V_2 \setminus V_1$, so $G/V_2 = G[V_1]/V_1 \cap V_2$. As G/V_2 and $G[V_2]$ are trees, another application of proposition 6.4 yields that G is also a tree. □

6.2.2 The proof of proposition 6.1

First, we establish an additional property of cycles in chordal graphs, needed in the proof of proposition 6.1.

Proposition 6.7. *If G is a chordal graph, then every cycle of length ≥ 4 in G contains a* 2-chord*, i.e. a chord connecting two vertices with distance* 2 *along the cycle.*

Proof. Let C be a cycle of G of length $k \geq 4$. As G is chordal, it has a chord e_1 which splits it up into two cycles. If one of those two cycles has length 3 (including the chord), then we are done. Otherwise, take one of the cycles, C_1, and split it along a chord e_2 into two cycles. Denote by C_2 the cycle from the C_1-splitting not containing e_1. Its only non-C edge is e_2. If C_2 has length 3, then we are done. Otherwise proceed recursively, with each C_i having e_i as its only non-C edge. Since the lengths of the cycles C_i are strictly decreasing, the recursion terminates. □

Proof of proposition 6.1. $\underline{T \text{ is connected}}$: If $K' \sim_{\mathcal{K}} K$, then K' is connected to K by a path in $\Gamma_{\mathcal{K}}$. Hence, we can find a sequence $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_k$ of families in $\mathcal{L}_G^{\leq}(\mathcal{K})$ such that $K \in \mathcal{K}_1$, $K' \in \mathcal{K}_k$ and $\mathcal{K}_i \cap \mathcal{K}_{i+1} \neq \emptyset$, for $1 \leq i < k$. As $K \in \mathcal{K}_1$, $\mathcal{K}_1$ is completely contained in $[K]_{\sim_{\mathcal{K}}}$. Since $\mathcal{K}_i$ and $\mathcal{K}_{i+1}$ have non-empty intersection, it follows by induction that each $\mathcal{K}_i$ contains some element of $[K]_{\sim_{\mathcal{K}}}$ and is also completely contained in $[K]_{\sim_{\mathcal{K}}}$. As $T[\mathcal{K}_i]$ is a tree, for every i, it is connected. We choose and combine paths from $T[\mathcal{K}_i]$ to obtain a path from K to K' in T.

$\underline{T \text{ is acyclic}}$: Assume that there is a cycle $C := K_1 K_2 \ldots K_n$ in T. Since K_n and K_1 are connected by an edge in $\Gamma_{\mathcal{K}}$ there must be some $\mathcal{K}' \in \mathcal{L}_G^{\leq}(\mathcal{K})$ which contains both of them, that is $\mathcal{K}(K_n \cap K_1) \subseteq \mathcal{K}' \subsetneq \mathcal{K}$. Since this $\mathcal{K}'$ has non-empty intersection with $[K]_{\sim_{\mathcal{K}}}$ it is completely contained in $[K]_{\sim_{\mathcal{K}}}$.

Define the indices $i_1, \ldots, i_r$ inductively by:

$$i_1 := \max\{i \leq n \mid \mathcal{K}(K_n \cap K_1 \cap K_2 \cap \ldots \cap K_i) \subseteq [K]_{\sim_{\mathcal{K}}}\}.$$

If $i_j < n$, then define

$$i_{j+1} = \max\{i \leq n \mid \mathcal{K}(K_{i_j} \cap K_{i_j+1} \cap \ldots \cap K_i) \subseteq [K]_{\sim_{\mathcal{K}}}\}.$$

As $1 \leq i_1 < i_2 < \ldots$, this construction stops after $r \leq n$ steps with $i_r = n$. Let

$$\begin{aligned} C_1 &:= K_n \cap K_1 \cap K_2 \cap \ldots \cap K_{i_1}\,, \\ C_2 &:= K_{i_1} \cap K_{i_1+1} \cap \ldots \cap K_{i_2}\,, \\ &\vdots \\ C_r &:= K_{i_{r-1}} \cap K_{i_{r-1}+1} \cap \ldots \cap K_n\,. \end{aligned}$$

By construction, it holds that $\mathcal{K}_j := \mathcal{K}(C_j) \subseteq [K]_{\sim_\mathcal{K}}$, for $1 \leq j \leq r$.

We choose C and its cyclic ordering minimizing the value of r.

Case $r = 1$: The tree $T[\mathcal{K}_1]$ contains the cycle C, a contradiction.

Case $r = 2$: By the minimality of r, we have $\{K_{i_1}, K_{i_2}\} \subseteq \mathcal{K}_1 \cap \mathcal{K}_2 =: \mathcal{K}_{12} \neq \emptyset$. The graph $T[\mathcal{K}_1] \cup T[\mathcal{K}_2]$ is the graph $T[\mathcal{K}_1 \cup \mathcal{K}_2]$ without the edges between $(\mathcal{K}_1 \setminus \mathcal{K}_2)$ and $(\mathcal{K}_2 \setminus \mathcal{K}_1)$. Furthermore, $T[\mathcal{K}_1] \cup T[\mathcal{K}_2]$ contains the cycle C. Because $T[\mathcal{K}_1]$, $T[\mathcal{K}_2]$ and $T[\mathcal{K}_{12}]$ are all trees we apply proposition 6.6 to $T[\mathcal{K}_1] \cup T[\mathcal{K}_2]$ and deduce that it is a tree and does not contain the cycle C.

Case $r = 3$: We claim that

$$\mathcal{K}_1 \cap \mathcal{K}_2 \cap \mathcal{K}_3 \neq \emptyset\,. \tag{20}$$

Admitting the claim for the moment, we can finish the proof of the case $r = 3$. We apply proposition 6.3 three times:

First, to the trees $T[\mathcal{K}_1]$, $T[\mathcal{K}_1]$ and $T[\mathcal{K}_1 \cap \mathcal{K}_2]$, deducing that $T[\mathcal{K}_1] \cup T[\mathcal{K}_2]$ is a tree.

Second, to the trees $T[\mathcal{K}_1 \cap \mathcal{K}_3]$, $T[\mathcal{K}_2 \cap \mathcal{K}_3]$ and $T[\mathcal{K}_1 \cap \mathcal{K}_2 \cap \mathcal{K}_3]$, deducing that $T[\mathcal{K}_1 \cap \mathcal{K}_3] \cup T[\mathcal{K}_2 \cap \mathcal{K}_3]$ is a tree.

Third, to $T[\mathcal{K}_1] \cup T[\mathcal{K}_2]$, $T[\mathcal{K}_3]$ and $T[\mathcal{K}_1 \cap \mathcal{K}_3] \cup T[\mathcal{K}_2 \cap \mathcal{K}_3]$. We check that the third tree is indeed the intersection of the first two. For the vertex sets, we have

$$(\mathcal{K}_1 \cup \mathcal{K}_2) \cap \mathcal{K}_3 = (\mathcal{K}_1 \cap \mathcal{K}_3) \cup (\mathcal{K}_2 \cap \mathcal{K}_3)\,.$$

For the edges, we have

$$\begin{aligned} & KL \in (T[\mathcal{K}_1] \cup T[\mathcal{K}_2]) \cap T[\mathcal{K}_3] \\ \Leftrightarrow\ & KL \in T[\mathcal{K}_1] \cup T[\mathcal{K}_2] \wedge KL \in T[\mathcal{K}_3] \\ \Leftrightarrow\ & (K, L \in \mathcal{K}_1 \vee K, L \in \mathcal{K}_2) \wedge K, L \in \mathcal{K}_3 \\ \Leftrightarrow\ & (K, L \in \mathcal{K}_1 \wedge K, L \in \mathcal{K}_3) \vee (K, L \in \mathcal{K}_2 \wedge K, L \in \mathcal{K}_3) \\ \Leftrightarrow\ & KL \in T[\mathcal{K}_1 \cap \mathcal{K}_3] \vee KL \in T[\mathcal{K}_2 \cap \mathcal{K}_3] \\ \Leftrightarrow\ & KL \in T[\mathcal{K}_1 \cap \mathcal{K}_3] \cup T[\mathcal{K}_2 \cap \mathcal{K}_3]\,. \end{aligned}$$

Hence $T[\mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3]$ is a tree. It contains the cycle C, a contradiction.

Proof of the claim (20): Since $C_1 \cup C_2 \in K_{i_1}$, $C_1 \cup C_3 \in K_{i_3}$ and $C_2 \cup C_3 \in K_{i_2}$, they all induce complete subgraphs of G. For a collection of subsets $\{C_i\}_{i\in[n]}$ of V, we have:

$$\forall i,j \in [n] : C_i \cup C_j \text{ is complete } \Leftrightarrow \bigcup_{i\in[n]} C_i \text{ is complete.} \tag{21}$$

Hence, $C_1 \cup C_2 \cup C_3$ is a complete subset of G and there is a clique $M \in \mathbf{M}_G$ containing $C_1 \cup C_2 \cup C_3$, establishing (20).

Case $r \geq 4$: We construct a cycle in G violating proposition 6.7. Our first observation is

$$C_{i_1} \cup C_{i_2} \text{ is complete } \Leftrightarrow |i_1 - i_2 \mod r| \leq 1\,. \tag{22}$$

To prove this observation, we assume that $1 \leq i_1 < i_2 \leq r$ and $D := C_{i_1} \cup C_{i_2}$. If $i_2 - i_1 = 1$, then $D \in M_{i_1}$. Hence D is complete. If $i_2 - i_1 \geq 2$ and $D = \emptyset$, then there is a clique M containing D with $M \in \mathcal{K}_{i_1} \cap \mathcal{K}_{i_2}$. This allows the creation of a cycle C', by shortcutting C via a connection from M_{i_1} to M in $T[\mathcal{K}_{i_1}]$ and from M to M_{i_2-1} in $T[\mathcal{K}_{i_2}]$. We can cover the cycle C' by at most $i_2 - i_1 + 1 < r$ clique families, contradicting the minimality of r.

Call a path $v_1 \dots v_l$ *two-chordless* iff

$$\forall k \in [l-2] : v_k \text{ is non-adjacent to } v_{k+2}\,, \tag{23a}$$
$$v_{l-1} \text{ is non-adjacent to } v_1\,, \tag{23b}$$
$$\forall k \in [l] : v_k \in C_{j_k}\,. \tag{23c}$$
$$j_1 < j_2 < \dots < j_l. \tag{23d}$$

We construct a path in the following way: Choose non-adjacent vertices $v_1 \in C_1$ and $v_3 \in C_3$. They exist by the observation (22). Let $j_1 = 1$, $j_2 = 2$, $j_3 = 3$ and

$$j_4 := \max\{j \leq r \mid C_j \cup \{v_3\} \in \mathcal{C}_G\,,\}\,.$$

where the maximum runs over a non-empty set, since by (22) v_3 is adjacent to each vertex in C_4. Every vertex in C_{j_4} is adjacent to v_3, but, for every $4 \leq j_4 < j \leq r$, there is a vertex in C_j not adjacent to v_3. Because of observation (22), we can choose non-adjacent vertices $v_2 \in C_2$ and $v_4 \in C_{j_4}$. The path $v_1v_2v_3v_4$ is two-chordless (23), for $l = 4$.

We extend this two-chordless path until we end up with a cycle contradicting proposition 6.7. Because the path always fulfils (23d), it has at most length r. Let $v_l \in C_{j_l}$ be the last element of the path. We have two cases:

Case v_l is adjacent to v_1: We can complete the $v_1 \dots v_l$ to a cycle. It fulfils (23a) and (23b), contradicting proposition 6.7.

Case v_l is not adjacent to v_1: We extend our path. This happens at most $r - 4$ times. By (22), we know that $j_l < r$. Hence, let

$$j_{l+1} := \max\{j \leq r \mid C_j \cup \{v_l\} \in \mathcal{C}_G\,,\}\,.$$

where the maximum is over a non-empty set, since by (22) v_l is adjacent to each vertex in C_{j_l+1}. In particular, this implies that $j_{k+1} \geq j_k + 1 > j_k$, fulfilling (23d). By the definition of j_l (with $l \geq 4$) and observation (22), there is a vertex $v_{l+1} \in C_{j_{l+1}}$ not adjacent to v_{l-1}. Thus, the extension fulfils (23a) and (23c). As v_l is not adjacent to v_1, the extension fulfils (23b). □

References

[BFMY83] Catriel Beeri, Ronald Fagin, David Maier, and Mihalis Yannakakis. On the desirability of acyclic database schemes. *J. Assoc. Comput. Mach.*, 30(3):479–513, 1983.

[BG81] Philip A. Bernstein and Nathan Goodman. Power of natural semijoins. *SIAM J. Comput.*, 10(4):751–771, 1981.

[BP93] Jean R. S. Blair and Barry Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, volume 56 of *IMA Vol. Math. Appl.*, pages 1–29. Springer, New York, 1993.

[Die05] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.

[Dir61] G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.

[Gav74] Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *J. Combinatorial Theory Ser. B*, 16:47–56, 1974.

[GHP95] Philippe Galinier, Michel Habib, and Christophe Paul. Chordal graphs and their clique graphs. In *Graph-theoretic concepts in computer science (Aachen, 1995)*, volume 1017 of *Lecture Notes in Comput. Sci.*, pages 358–371. Springer, Berlin, 1995.

[Hal84] R. Halin. On the representation of triangulated graphs in trees. *European J. Combin.*, 5(1):23–28, 1984.

[HL89] Chin Wen Ho and R. C. T. Lee. Counting clique trees and computing perfect elimination schemes in parallel. *Inform. Process. Lett.*, 31(2):61–68, 1989.