Clique trees of infinite locally finite chordal graphs

Christoph Hofer-Temmel[*]
NLDA and CWI
Den Helder and Amsterdam, Netherlands
math@temmel.me

Florian Lehner[†]
Department of Mathematics
University of Warwick
Coventry, United Kingdom
mail@florian-lehner.net

Submitted: Dec 12, 2013; Accepted: Feb 7, 2018; Published: Apr 27, 2018

Abstract

We investigate clique trees of infinite locally finite chordal graphs. Our main contribution is a bijection between the set of clique trees and the product of local finite families of finite trees. Even more, the edges of a clique tree are in bijection with the edges of the corresponding collection of finite trees. This allows us to enumerate the clique trees of a chordal graph and extend various classic characterisations of clique trees to the infinite setting.

Mathematics Subject Classifications: 05C62, 05C05, 05C30

1 Introduction

A graph is chordal, if for every cycle of length greater than three there is a chord, i.e. an edge connecting two non-consecutive vertices on the cycle. Chordal graphs are a classic object in graph theory and computer science [3]. In the finite case they are known to be equivalent to the class of graphs representable as a family of subtrees of a tree [7]. A finite and connected chordal graph has natural representations of this form: so-called clique trees, which form a subclass of the spanning trees of its clique graph.

This work investigates clique trees of infinite locally finite chordal graphs. We show their existence and extend various classic characterisations of clique trees from the finite to the infinite case.

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. This characterises the clique trees by a bijection with the product of the local choices. See Section 3.3. Each constraint only depends on the edges within its

[*] Supported by NWO grant 639.032.916.
[†] Supported by FWF project W1230-N13.

partition element, whence the constraints are satisfied or violated independently of each other. Section 3.6 presents a purely combinatorial and local construction of a clique tree by fixing a satisfying subset of the edges in each element of the partition.

In the case of a finite chordal graph, our main result gives rise to an enumeration of the clique trees, see Section 3.7. It is equivalent to a prior enumeration via a local partitioning of constraints by Ho and Lee [9]. While their partition is indexed by the minimal vertex separators of the chordal graph, ours is indexed by certain families of cliques. We recover the minimal vertex separators as intersections of the cliques within those families, thus demonstrating the equivalence of the two approaches. Section 3.8 shows this bijection. As a corollary, we identify the reduced clique graph with the union of all clique trees, extending a result in [6] to infinite graphs.

Classic characterisations [3] of clique trees of finite chordal graphs relate various properties of a clique tree to minimal vertex separators of the original graph, or demand maximality with respect to particular edge weights in the clique graph, or describe properties of paths in the tree, among others. They contain obstacles to an immediate extension to the infinite case, though. Either their range is unbounded, or the conditions overlap, or the proof depends on the finite setting or they make no sense at all in an infinite setting (such as maximality with respect to edge weights). Often these obstacles can be overcome by passing to suitable local conditions, i.e. conditions depending only on finitely many vertices or edges. The partition of the edge set allows us to do that. Consequently, in Section 4, we extend several classic characterisations or sensible versions thereof to the infinite case.

2 Notation and basics

2.1 Graphs

Throughout this paper we consider locally finite multigraphs, that is, all vertex degrees are finite. We say *graph*, if we exclude multiple edges. Let G be a multigraph with vertex set V. For $W \subseteq V$, denote by $G[W]$ the submultigraph of G induced by W. For an equivalence relation $\sim$ on V, denote by $G/\sim$ the multigraph resulting from contracting each equivalence class of $\sim$ to a single vertex. It may contain loops and multiple edges, even if G does not. For $W \subseteq V$, let G/W be the multigraph with only W contracted to a single vertex, and, for $W_1, \dots, W_k$ disjoint subsets of V, let $G/\{W_1, \dots, W_k\}$ be the multigraph resulting from G by contracting each W_i to a single vertex. If we speak of the graph $G/\sim$ (or one of the above variants), then we mean the graph underlying the multigraph $G/\sim$, including possible loops.

A multigraph is *complete*, if all vertices are adjacent to each other. We say that $W \subseteq V$ is *complete*, if $G[W]$ is complete. A *clique* is a maximal complete subset of V. Denote by $\mathcal{C}_G$ and $\mathcal{K}_G$ the set of all complete vertex subsets and cliques of G respectively. The *clique graph* $\mathbf{K}_G$ of G has vertex set $\mathcal{K}_G$ and an edge for every pair of cliques with non-empty intersection. As G is locally finite, all its cliques are finite and every vertex is contained in only a finite number of cliques. Whence, the clique graph $\mathbf{K}_G$ is locally finite, too.

A *tree* T is a connected and acyclic graph. A subgraph of G is *spanning*, if it has the same vertex set as G. Denote by $\mathcal{T}_G$ the *set of spanning trees* of G.

2.2 The lattice of clique families

For $C \in \mathcal{C}_G$, the *clique family generated by* C is

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

Clique families are always non-empty. Generation is *contravariant*, as

$$C \subseteq C' \Rightarrow F(C') \subseteq F(C)\,. \tag{1}$$

The largest clique family is $F(\varnothing) = \mathcal{K}_G$. It is infinite if and only if G is infinite, and in this case, it is the only infinite clique family. For $v \in G$, we abbreviate $F(\{v\})$ to $F(v)$. These are the building blocks of all finite clique families:

$$F(C) = \bigcap_{v \in C} F(v)\,. \tag{2}$$

Let F be a clique family. Every $C \in \mathcal{C}_G$ with $F(C) = F$ is a *generator* of F. There is a *maximal generator* of F with respect to set inclusion:

$$C(F) := \bigcap_{K \in F} K = \bigcup_{F(C)=F} C\,. \tag{3}$$

In particular, we have

$$F(C(F)) = F\,. \tag{4}$$

It is also immediate that the intersection of two clique families F_1 and F_2 is again a clique family, more precisely

$$F_1 \cap F_2 = F(C(F_1) \cup C(F_2)). \tag{5}$$

The sets of generators of two clique families coincide, if and only if the clique families are equal, and are disjoint otherwise. This follows from the equivalence relation $\sim$ on $\mathcal{C}_G$ given by $C_1 \sim C_2 \Leftrightarrow F(C_1) = F(C_2)$. An equivalence class of $\sim$ corresponds to the set of generators of a clique family.

Proposition 1. *Choose distinct* $K_1, K_2 \in \mathcal{K}_G$*. There is an edge* $K_1K_2 \in \mathbf{K}_G$*, if and only if* $\varnothing \neq K_1 \cap K_2 = C(F(K_1 \cap K_2))$*.*

Proof. We have an edge $K_1K_2 \in \mathbf{K}_G$, if and only if $K_1 \cap K_2 \neq \varnothing$. Thus, $F := F(K_1 \cap K_2)$ is finite and we have

$$\varnothing \neq K_1 \cap K_2 \subseteq \bigcup_{F(C)=F} C \overset{(3)}{=} C(F) \overset{(3)}{=} \bigcap_{K \in F} K \subseteq K_1 \cap K_2\,. \qquad \square$$

Let $\mathcal{F}_G$ be the set of clique families of G. The clique families $\mathcal{F}_G$ form a lattice with respect to set inclusion. Equation (2) implies that all chains in the lattice are finite. We use this fact to reason inductively over this lattice.

3 Infinite clique trees

3.1 Chordal graphs and subtree representations

Our main reference for basic facts about chordal graphs is [3]. A *chordal graph* contains no induced cycle of length greater than 3. In other words, the induced graph of every cycle of length greater than 3 contains a *chord*, an edge connecting two non-consecutive vertices of the cycle. Throughout this work, we assume that chordal graphs are connected, locally finite and do not contain loops.

Let T be a tree and denote by $\mathcal{T}$ the family of subtrees of T. A function $t : V \to \mathcal{T}$ is a *subtree representation of G on T*, if $v_1 v_2 \in G \Leftrightarrow t(v_1) \cap t(v_2) \neq \varnothing$. A finite graph is chordal, if and only if it has a subtree representation on some tree [7]. This remains true for locally finite infinite graphs, but there are examples of countable and non-locally finite chordal graphs which do not admit a subtree representation [8].

A special kind of subtree representation is given by clique trees. Clique trees are subtree representations on spanning trees of the clique graph $\mathbf{K}_G$. Their existence for finite chordal graphs is a classic result [7].

Definition 2. Let G be a chordal graph. A spanning tree T of $\mathbf{K}_G$ is called a *clique tree* of G, if

$$\forall\, v \in V : \quad T[F(v)] \text{ is a tree.} \tag{6}$$

A clique tree T represents G via the subtree map $v \mapsto T[F(v)]$, where $F(v)$ is the clique family generated by $\{v\}$. Let $\mathcal{CT}_G$ be the set of clique trees of G.

The following sections show not only the existence of clique trees of infinite chordal graphs, but a way of constructing them from independent local pieces. The classic recursive construction in [7] depends on the finiteness of the graph to terminate and does not give any indication of how to obtain an independent construction for non-adjacent parts of the chordal graph, a natural goal given the tree-like structure of chordal graphs.

3.2 Existence of clique trees

A first existence result stems from an implicit construction by a limiting procedure. Explicit local constructions follow in Section 3.6.

Theorem 3. *Every infinite, locally finite chordal graph has a clique tree.*

Proof. We use a compactness argument, which is a standard approach in infinite graph theory (c.f. [4, Chapter 8.1]). Arguments of this type are often useful to obtain a result for infinite graphs from its finite counterpart.

Let G be the chordal graph. Let $(v_n)_{n\in\mathbb{N}}$ be an enumeration of the vertices of G such that $V_n := \{v_1, \dots, v_n\}$ is connected for each $n \in \mathbb{N}$. Let $V'_n := \bigcup_{v\in V_n} \bigcup_{K\in F(v)} K$. In other words, V'_n contains all vertices that lie in a common clique with some vertex in V_n, or equivalently, V_n and its neighbours. The graph $G_n := G[V'_n]$ is connected and chordal.

For $v \in V_n$, the clique family $F(v)$ (with respect to G) is also a clique family of G_n, since G_n contains all vertices in cliques containing v. For $n \in \mathbb{N}$, let S_n be a clique tree of

G_n. Because every clique of G_n is also a clique of G, we may interpret S_n as a subtree T_n of $\mathbf{K}_G$.

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

By construction, $T[F(v)]$ is a tree, for each $v \in V$. It remains to verify that T is a tree as well. The trees corresponding to v and w overlap, if and only if $F(v) \cap F(w) \neq \varnothing$, equivalent to $vw \in G$. Hence T is connected, because G was assumed to be so. If T contains a cycle C, then it lies in $\mathbf{K}_{G_m}$, for some $m \in \mathbb{N}$. Hence, C is a cycle in the tree $T_1^m[\mathbf{K}_{G_m}]$, a contradiction. □

3.3 Local characterisation via clique families

We show how to construct clique trees of locally finite graphs from small local pieces. Those pieces live on domains defined in terms of the clique families.

For $F \in \mathcal{F}_G$, let Γ_F be the subgraph of $\mathbf{K}_G[F]$ with vertex set F and an edge $K_1K_2 \in \Gamma_F$, if $F(K_1 \cap K_2) \subsetneq F$, equivalent to $K_1 \cap K_2 \supsetneq C(F)$ by Proposition 1. Intuitively, the graph Γ_F connects cliques in F whose intersection is "larger than necessary", i.e. their intersection contains a vertex which is not contained in every clique in F. Let $\sim_F$ be the equivalence relation whose classes are the connected components of Γ_F, and let $[K]_{\sim_F}$ denote the equivalence class of K with respect to $\sim_F$. This permits to characterise a clique tree in a finer-grained manner than (6).

Theorem 4. *Let G be a chordal graph. A spanning subgraph T of $\mathbf{K}_G$ is a clique tree of G, if and only if it satisfies one of the following equivalent conditions:*

$$\forall F \in \mathcal{F}_G : \quad T[F] \text{ is a tree,} \tag{7a}$$

$$\forall F \in \mathcal{F}_G : \quad T[F]/\sim_F \text{ without its loops is a tree.} \tag{7b}$$

Note that, only (7a) says directly that $T = T[\mathcal{K}_G]$ is a tree. In (7b), this fact is not so obvious, but follows from an inductive bottom-up construction. The advantage of (7b) is that it allows to compose a clique tree from trees on smaller parts of the clique graph. In Section 3.6, we see that these parts do not overlap. Thus, we may pick the trees in (7b) independently of each other. Consequently, we construct parts of a clique tree locally without knowing the global structure.

Before we give a proof of Theorem 4 in Section 3.5, we formulate and prove some auxiliary results in Section 3.4.

3.4 Combining trees

Lemma 5. *Let G be a finite graph with vertex set V. Let $V_1, V_2, \ldots, V_k$ be disjoint subsets of V. Every choice of two of the following statements implies the third one:*

$$G \text{ is a tree,} \tag{8a}$$

$$\forall 1 \leqslant i \leqslant k: \quad G/\{V_1, \dots, V_i\} \text{ without its loops is a tree,} \tag{8b}$$

$$\forall 1 \leqslant i \leqslant k: \quad G[V_i] \text{ is a tree.} \tag{8c}$$

Proof. A cycle is non-trivial, if it has length greater than 2.

(8a) and (8b) $\Rightarrow$ (8c): $G[V_i]$ does not contain a cycle. It must be connected, because otherwise there would be a path between two of its connected components in G contracting to a non-trivial cycle in $G/\{V_1, \dots, V_i\}$.

(8a) and (8c) $\Rightarrow$ (8b): $G/\{V_1, \dots, V_i\}$ is connected as a contraction of a connected graph. Because we only contract connected sets, a non-trivial cycle in $G/\{V_1, \dots, V_i\}$ corresponds to a non-trivial cycle in G.

(8b) and (8c) $\Rightarrow$ (8a): G is connected, because it is obtained from the connected graph $G/\{V_1, \dots, V_k\}$ by expanding i vertices into the connected graphs $G[V_i]$. There cannot be a cycle in G, because such a cycle would either lie in some $G[V_i]$ or contract to a non-trivial cycle in $G/\{V_1, \dots, V_k\}$. □

Lemma 6. *Let T be a tree with vertex set V and $V_1, V_2 \subseteq V$ with non-empty intersection. If $T[V_1]$ and $T[V_2]$ are trees, then $T[V_1 \cap V_2]$ is also a tree.*

Proof. Obviously, there is no cycle in $T[V_1 \cap V_2]$. To see that it is connected, observe that for any two vertices $u, v \in V_1 \cap V_2$ there are unique u-v-paths in T, $T[V_1]$ and $T[V_2]$. Those paths coincide and are in $T[V_1 \cap V_2]$. □

The following lemma is specific to the situation of clique trees of chordal graphs. It contains key steps of the proof of Theorem 4.

Lemma 7. *Let G be chordal graph. Let K be a clique of G and let $F \in \mathcal{F}_G$ be a clique family containing K. Let S be a subgraph of Γ_F with vertex set $[K]_{\sim_F}$. If, for each clique family $F' \subsetneq F$ with $F' \subseteq [K]_{\sim_F}$, $S[F']$ is a tree, then S is a tree.*

Proof. First note that every clique family $F' \subsetneq F$ is either fully contained in $[K]_{\sim_F}$ or disjoint from it. Indeed, any two cliques $K_1, K_2 \in F'$ satisfy $K_1 \cap K_2 \supseteq C(F') \supsetneq C(F)$. Thus, they are connected by an edge in Γ_F.

S is connected: Let K' be a vertex of S. Because $K \sim_F K'$, there is a K'-K-path $K_0 \dots K_n$ in Γ_F. The definition of edges in Γ_F implies that $K_i \cap K_{i-1} \supsetneq C(F)$. Thus, $F_i := F(K_i \cap K_{i-1}) \subsetneq F$ and $S[F_i]$ is a tree containing a K_{i-1}-K_i-path P_i. The union of $P_1, \dots, P_n$ contains a K-K'-path in S.

S is acyclic: Let $\mathcal{M}_F$ be the set of maximal strict clique subfamilies of F. We say that $F' \in \mathcal{M}_F$ *covers an edge* $e \in S$, if $e \in S[F']$. If e is an edge of S, then it is also an edge of Γ_F. In particular, its endpoints correspond to cliques whose intersection generates a strict subfamily $F' \subsetneq F$. Hence, each edge in S is covered by some clique family in $\mathcal{M}_F$. A subset of $\mathcal{M}_F$ *covers a subgraph* of Γ_F, if each edge of the subgraph is covered by at least one element of the subset.

Assume for a contradiction that S contains a non-trivial cycle. Let $R \leqslant |\mathcal{M}_F|$ be the minimal cardinality of a subset of $\mathcal{M}_F$ covering a non-trivial cycle of S. For a non-trivial cycle Z with cover $\{F_1, \dots, F_R\}$, we say that an edge of Z is *uniquely covered* by F_i, if it is covered by F_i and not covered by any other F_i, for $j \neq i$. Denote by U_i the set of edges uniquely covered by F_i. The minimality of R implies that all U_i are non-empty. Call F_i *nice* for Z, if all edges of U_i are contained in the same connected component of $Z[F_i]$. We claim that there exists a non-trivial cycle Z with cover $\{F_1, \ldots, F_R\}$ such that

$$\forall 1 \leqslant i \leqslant R: \quad F_i \text{ is nice for } Z\,. \tag{9}$$

The statement (9) is trivial for $R = 1$. For $R \geqslant 2$, we transform Z into a non-trivial cycle with the same cover $\{F_1, \ldots, F_R\}$ and fulfilling (9) in two steps:

1. There is a cycle Z' covered by $\{F_1, \ldots, F_R\}$ with F_1 being nice for Z'.
2. If F_1 is nice, then $F_2, \ldots, F_R$ are nice, too.

Step 1: If $Z[F_1]$ is connected, then let $Z' := Z$. Otherwise, pick edges e, f in different connected components of $Z[F_1]$ such that there is a non-trivial path P in Z without an edge covered by F_1 and connecting an endpoint of e and f each. The path P is covered by $\{F_2, \ldots, F_R\}$. Because the endpoints of P are distinct and contained in F_1, and F_1 is connected, there is a non-trivial path Q with the same endpoints as P and covered by F_1. Clearly, P and Q are edge-disjoint and $Z' := P \cup Q$ is a non-trivial cycle covered by $\{F_1, \ldots, F_R\}$. As Q is connected, F_1 is nice for Z'.

Step 2: Assume that F_1 is nice but some other F_i is not. Let $e, f \in U_i$ be edges contained in different connected components of $Z[F_i]$. The graph $Z - \{e, f\}$ consists of two non-trivial paths P_1 and P_2. Without loss of generality, all of U_1 lies in P_1 and P_2 is covered by $\{F_2, \ldots, F_R\}$. There is a non-trivial path Q in $S[F_i]$ connecting the endpoints of P_1. Because P_2 contains an edge not covered by F_i and Q is covered by F_i, we have $P_2 \neq Q$. Hence, $P_2 \cup Q$ contains a non-trivial cycle covered by $\{F_2, \ldots, F_R\}$. This contradicts the minimality of R.

Let $||i - j||_R$ be the modulo R distance between i and j. We claim that we may reorder the clique families in a cover of a cycle fulfilling (9) such that

$$\forall 1 \leqslant i < j \leqslant R: \quad F_i \cap F_j \neq \varnothing \Leftrightarrow ||i - j||_R \leqslant 1 \tag{10}$$

For each i, choose an edge $e_i \in U_i$. Order the sets F_i according to the cyclic order of the edges e_i. The F_i's niceness by (9) guarantees that the order is independent of the choice of edges. Without loss of generality assume that the order is $1, 2, \ldots, R$.

First, we show that distant clique families are disjoint. Consider i, j with $||i - j||_R > 1$, whence $R \geqslant 4$ holds. Assume that $F_i \cap F_j \neq \varnothing$. The graph $Z - \{e_i, e_j\}$ consists of two disjoint non-trivial paths P_1 and P_2. Because $F_i \cap F_j \neq \varnothing$, there is a non-trivial path Q in $S[F_i \cup F_j]$ connecting e_i and e_j. Niceness implies that there are distinct clique families F_{k_1} and F_{k_2} with $k_1, k_2 \notin \{i, j\}$ such that $U_{k_1} \subseteq P_1$ and $U_{k_2} \subseteq P_2$. Hence, $P_1 \neq Q$ and $P_2 \neq Q$. Therefore, the union of P_1, Q and $\{e_i, e_j\}$ contains a non-trivial cycle covered by $\{F_1, \ldots, F_R\} \setminus \{F_{k_2}\}$. This contradicts the minimality of R.

Second, we show that close clique families overlap. For i, let F_j and F_k be the two clique families with $||i - j||_R = 1 = ||i - k||_R$. Without loss of generality, assume that

$F_i \cap F_j = \varnothing$. This can only happen if $R \geqslant 3$. The connected component of $Z[U_i]$ containing e_i is a path P. The endpoints of P are contained in F_k, because of disjointness they can not be part of F_l, with $l \notin \{i, j, k\}$, nor in F_j, by assumption. Connect these endpoints by a path Q in $S[F_k]$. The paths P and Q are distinct. Their union contains a non-trivial cycle covered by $\{F_i, F_k\}$. This contradicts the assumption that $R \geqslant 3$.

This completes the proof of the claim (10). From here on, we work with a cycle Z satisfying both (9) and (10). We proceed by case analysis on R.

Case $R = 1$: The tree $S[F_1]$ cannot contain the cycle Z.

Case $R = 2$: The fact that F_1 and F_2 are nice by (9) implies that there exist distinct $K_1, K_2 \in Z[F_1 \cap F_2]$ splitting Z into two non-trivial paths P_1 and P_2, disjoint except in $\{K_1, K_2\}$, such that $U_1 \subseteq P_1$ and $U_2 \subseteq P_2$. As $S[F_1 \cap F_2]$ is a tree, there exists a unique path Q between K_1 and K_2 in $S[F_1 \cap F_2]$. As $U_1 \subseteq P_1$, $P_1 \neq Q$. Hence, the union of P_1 and Q contains a non-trivial cycle covered by F_1. This is a contradiction to $S[F_1]$ being a tree.

Case $R = 3$: The fact that F_1, F_2 and F_3 are nice by (9) implies that there exist distinct $K_{12} \in Z[F_1 \cap F_2]$, $K_{23} \in Z[F_2 \cap F_3]$ and $K_{13} \in Z[F_1 \cap F_3]$ splitting Z into three non-trivial paths P_1, P_2 and P_3, disjoint except in $\{K_{12}, K_{23}, K_{13}\}$, such that, for all $1 \leqslant i \leqslant 3$, $U_i \subseteq P_i$.

Let $C_i := C(F_i)$ be the maximal generator of the clique family F_i. By (10), there is a clique in $F_i \cap F_j$. Hence, $C_i \cup C_j$ is complete for all $1 \leqslant i, j \leqslant 3$. Thus, $C := \bigcup_{i=1}^{3} C_i$ is complete and $\varnothing \neq F(C) \subseteq \bigcap_{i=1}^{3} F_i$.

Fix $K' \in \bigcap_{i=1}^{3} F_i$. As $S[F_1 \cap F_2]$ is a tree, there exists a unique path P_{12} between K_{12} and K' in $S[F_1 \cap F_2]$. Likewise, there is a unique path P_{23} in $S[F_2 \cap F_3]$ between K_{23} and K' and a unique path P_{13} in $S[F_1 \cap F_3]$ between K_{13} and K'. The union of P_1, P_{12} and P_{13} is covered by F_1 and contains a non-trivial cycle because $U_1 \subseteq P_1$. This is a contradiction to $S[F_1]$ being a tree.

Case $R \geqslant 4$: Again, let $C_i := C(F_i)$ and let $D_i := C_i \setminus C(F) \neq \varnothing$. For every vertex $v \in D_i$, the set $C(F) \uplus \{v\}$ generates a clique family satisfying $F_i \subseteq F(C(F) \uplus \{v\}) \subsetneq F$. Because F_i is a maximal strict subfamily of F, we infer that $F_i = F(C(F) \uplus \{v\})$. In particular the sets D_i are disjoint, because $v \in D_i \cap D_j$ implies that $F_i = F(C(F) \uplus \{v\}) = F_j$.

We investigate the edges between the sets D_i. If $||i - j||_R = 1$, then (10) implies the existence of $K \in F_i \cap F_j$. Hence, $D_i \cup D_j \subseteq C_i \cup C_j \subseteq K$ and $G[D_i \cup D_j]$ is complete. If $||i - j||_R > 1$, then assume that there is an edge $v_i v_j$ with $v_i \in D_i$ and $v_j \in D_j$. Then, $C(F) \uplus \{v_i, v_j\}$ is complete and $F_i \cap F_j$ contains a clique, a contradiction to (10). Thus, $G[D_i \cup D_j]$ contains no edges between D_i and D_j, if $||i - j||_R > 1$.

For $1 \leqslant i \leqslant R$, choose $v_i \in D_i$. The induced subgraph $G[\{v_1, \ldots, v_R\}]$ is a chordless cycle with length $R \geqslant 4$ and contradicts the chordality of G. □

3.5 Proof of Theorem 4

We prove the equivalences (6) ⇔ (7a) and (7a) ⇔ (7b). For convenience, we restate them. A spanning subgraph T of $\mathbf{K}_G$ is a clique tree of G, if and only if it satisfies the following equivalent conditions:

$$T \text{ is a tree and } \forall v \in V: \quad T[F(v)] \text{ is a tree,} \tag{6}$$

$$\forall F \in \mathcal{F}_G: \quad T[F] \text{ is a tree,} \tag{7a}$$

$$\forall F \in \mathcal{F}_G: \quad T[F]/\sim_F \text{ without loops is a tree.} \tag{7b}$$

<u>(6) ⇒ (7a)</u>: If $F = \mathcal{K}_G$ or $F = F(v)$, for some vertex $v \in V$, then $T[F]$ is a tree. Assume that (7a) does not hold. The finiteness of chains in $\mathcal{F}_G$ lets us choose a maximal $F \in \mathcal{F}_G$ such that $T[F]$ is not a tree. Furthermore, each generator of F contains at least two vertices. Let $C \subseteq C(F)$ be a minimal generator of F. For every $\varnothing \neq C' \subsetneq C$, the contravariance of clique family generation (1) implies that $F(C')$ and $F(C \setminus C')$ are strictly larger than F and $F = F(C') \cap F(C \setminus C')$. Maximality of F implies that $T[F(C')]$ and $T[F(C \setminus C')]$ are trees. Lemma 6 implies that $T[F]$ is a tree, too.

<u>(7a) ⇒ (6)</u>: Equation (7a) implies that $T[F(v)]$ is a tree, for each $v \in G$, and that $T[\mathcal{K}_G] = T$ is a tree.

<u>(7a) ⇒ (7b)</u>: Let $F \in \mathcal{F}_G$. Lemma 7 together with the assumption that $T[F']$ is a tree for every $F' \subsetneq F$ implies that $T[[K]_{\sim_F}]$ is a tree, for every equivalence class with respect to the relation $\sim_F$. If $F \neq \mathcal{K}_G$, then there are only finitely many equivalence classes. Hence, we apply Lemma 5 to show that $T[F]/\sim_F$ is a tree. For $F = \mathcal{K}_G$, we know that $\sim_F$ is the trivial relation, i.e. any two cliques are related. Whence, $T[F]/\sim_F$ is a single vertex tree.

<u>(7b) ⇒ (7a)</u>: Assume that there is some $F \in \mathcal{F}_G$ such that $T[F]$ is not a tree. Choose F minimal with this property. This is possible because chains in $\mathcal{F}_G$ are finite. Lemma 7 implies that $T[[K]_{\sim_F}]$ is a tree for every equivalence class with respect to $\sim_F$. Because there are only finitely many equivalence classes and $T[F]/\sim_F$ is a tree, Lemma 5 shows that $T[F]$ is a tree.

3.6 Edge bijections

In this section we show that the restrictions (7b) imposed by a clique family and its strict subfamilies are independent of each other. This allows us to write the set of clique trees as the product of sets of smaller trees, see Theorem 9. The product is indexed by the clique families. For a given clique family, the associated set of trees is independent of the sets of trees for subfamilies of the clique family.

Let $F \in \mathcal{F}_G$. Recall that Γ_F was defined as the subgraph of $\mathbf{K}_G[F]$ containing all edges between cliques whose intersection is strictly larger than $C(F)$. Let Ξ_F be the subgraph of $\mathbf{K}_G[F]$ containing the remaining edges. That is, Ξ_F contains an edge K_1K_2, if $F(K_1 \cap K_2) = F$, or equivalently $K_1 \cap K_2 = C(F)$, by Proposition 1. Intuitively, the graph Ξ_F connects cliques in F whose intersection is "as small as possible" within $\mathbf{K}_G[F]$. It is obvious from the definitions that Γ_F and Ξ_F partition the edges of $\mathbf{K}_G[F]$ into two disjoint sets.

Consider the multigraph $\Delta_F := \Xi_F/\sim_F$, i.e. all components of Γ_F are contracted to single points. This graph may contain (multiple) loops. We use the natural bijection between edges of Ξ_F and edges of Δ_F to label the edges of Δ_F and differentiate between them.

It is worth noting that $\mathbf{K}_G[F]/\sim_F$ can be obtained from Δ_F by adding additional loops. As a consequence, spanning trees of the two graphs are in one-to-one correspondence.

Proposition 8. *There is a bijection between the edges of $\mathbf{K}_G$ and the disjoint union over all clique families F of edges of Ξ_F. Via edge-labelling, this extends to the disjoint union of edges of Δ_F.*

$$\mathbf{K}_G \overset{\textit{edges}}{=} \biguplus_{F \in \mathcal{F}_G} \Xi_F \overset{\textit{edge-labelling}}{=} \biguplus_{F \in \mathcal{F}_G} \Delta_F\,. \tag{11}$$

Proof. For $K_1K_2 \in \mathbf{K}_G$, consider the clique family $F := F(K_1 \cap K_2)$. By Proposition 1, we have $K_1 \cap K_2 = C(F)$. The definition of Ξ_F allows K_1K_2 only as an edge in Ξ_F, but not in any other $\Xi_{F'}$ with $F' \neq F$. □

Theorem 9. *There is a bijection between the clique trees $\mathcal{CT}_G$ and a $\mathcal{F}_G$-indexed product of sets of spanning trees. For each clique tree, its edges and the edges of the spanning trees in its corresponding $\mathcal{F}_G$-indexed collection are in bijection, too.*

$$\mathcal{CT}_G \overset{\textit{edge-labelling}}{=} \prod_{F \in \mathcal{F}_G} \mathcal{T}_{\Delta_F}\,. \tag{12}$$

A similar bijection to (12) between the clique trees of a finite chordal graph and a product of trees indexed by the minimal vertex separators of the graph is already known [9]. We discuss their relationship in Section 3.8.

Proof. Using the bijection from Proposition 8, we split the edges of a clique tree $T \in \mathcal{CT}_G$ into disjoint sets $E_F := \{K_1K_2 : K_1K_2 \in T, K_1K_2 \in \Xi_F\}$, indexed by $\mathcal{F}_G$. For $F \in \mathcal{F}_G$, statement (7b) tells us that E_F labels the edges of a spanning tree of Δ_F.

Conversely, select a spanning tree $T_F \in \mathcal{T}_{\Delta_F}$, for each $F \in \mathcal{F}_G$. Let E be the union of their edge-labels. By Proposition 8, each edge in E appears exactly once as an edge-label of some T_F. By (7b), the graph $T := (\mathcal{K}_G, E)$ is a clique tree. □

3.7 Enumerating the clique trees

In this section, we enumerate the clique trees of a given chordal graph. We start with a structure statement about the auxiliary multigraphs.

Proposition 10. *The multigraph Δ_F is complete.*

Proof. Case $C(F) = \varnothing$: This only happens, if G contains disjoint cliques and $F = \mathcal{K}_G$. In this case, we have $\Gamma_{\mathcal{K}_G} = \mathbf{K}_G$ and $\Xi_{\mathcal{K}_G} = (\mathcal{K}_G, \varnothing)$. As $\mathcal{K}_G$ forms one equivalence class under $\sim_F$, $\Delta_{\mathcal{K}_G}$ is a graph with one vertex and no edges.

Case $C(F) \neq \varnothing$: This implies that F is finite. For all distinct $K_1, K_2 \in F$,

$$\varnothing \neq C(F) = \bigcap_{K \in F} K \subseteq K_1 \cap K_2\,.$$

Therefore, $\mathbf{K}_G[F]$ is complete and so is Δ_F. □

An immediate consequence of (12) is a count of clique trees of a finite chordal graph.

$$|\mathcal{CT}_G| = \prod_{F \in \mathcal{F}_G} |\mathcal{T}_{\Delta_F}| \, . \tag{13}$$

The value of $|\mathcal{T}_{\Delta_F}|$ is explicitly given in terms of the structure of Δ_F as a complete multigraph via a *matrix-tree theorem* from [9].

Corollary 11. *Fix $D \in \mathbb{N}$. For every finite chordal graph G with maximal degree D and vertices V, one can generate $\mathcal{CT}_G$ sequentially with only $O(|V|)$ working memory.*

Proof. As the degree is uniformly bounded, so are the sizes of a clique (by $D+1$, with equality if all edges incident to a vertex belong to the same clique), a non-trivial clique family F (by D, with equality if all edges incident to a vertex belong to different cliques) and its spanning trees $\mathcal{T}_{\Delta_F}$ (by D^{D-2} via Cayley's formula). Furthermore, each vertex is only contained in at most D cliques and hence in at most D clique families, so the size of $\mathcal{F}_G$ is linear in $|V|$. Generate $\mathcal{T}_{\Delta_F}$, for all $F \in \mathcal{F}_G$. This takes memory linear in $|V|$, with worst case multiplicative constants given by the bounds above which depend only on D. Iterate in lexicographic order through all the local choices of spanning trees and use (12) to obtain a clique tree from a full set of local choices. □

For infinite chordal graphs, there is a dichotomy in the number of clique trees.

Corollary 12. *Let G be an infinite chordal graph. It has either finitely or $2^{\aleph_0}$ many clique trees.*

Proof. We look at $\{|\mathcal{T}_{\Delta_F}|\}_{F \in \mathcal{F}_G}$. It is countable, because $\mathcal{F}_G$ is so. If only 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 is a countable number of independent choices between more than two spanning trees and the number of clique trees is $2^{\aleph_0}$. □

3.8 Minimal vertex separators and the reduced clique graph

As mentioned previously, a bijection indexed by minimal vertex separators and similar to (12) was given by Ho and Lee [9]. Lemma 15 shows that the minimal vertex separators correspond to the maximal generators of clique families with a non-trivial contribution to the bijection. As a consequence, the two decompositions coincide.

Following [3, Section 2.2], we call $\varnothing \neq W \subseteq V$ a *v-w-separator*, if v and w lie in different connected components of $G[V \setminus W]$. We call $\varnothing \neq W \subseteq V$ a *minimal vertex separator*, if there exist vertices v and w, such that W is a v-w-separator and no proper subset of W is a v-w-separator. Minimal vertex separators characterise chordal graphs. The proof of the following result for finite graphs from [3] generalises verbatim to infinite graphs.

Theorem 13 ([3, Theorem 2.1] after [5]). *A graph is chordal, if and only if every minimal vertex separator is complete.*

The remainder of this section shows that the minimal vertex separators form a subset of the maximal generators of the clique families.

Lemma 14. *A minimal vertex separator in a chordal graph separates two vertices adjacent to all of it.*

Proof. Let C be a minimal v_1-v_2-separator. For every $w \in C$, there exists a w-v_1-path P_w with $P_w \cap C = \{w\}$. The path P_w may be assumed to be chordless, i.e. non-successive vertices are not adjacent. For each $w \in C$, let v_1^w be the neighbour of w on P_w. Let $V_1 := \{v_1^w \mid w \in C\} \neq \varnothing$. If we show that one $u_1 \in V_1$ fulfils $C \cup \{v\} \in \mathcal{C}_G$, then a symmetric argument for a likewise u_2 on the v_2-side shows that C is a minimal u_1-u_2-separator.

For each $v \in V_1$, let $C_v := \{w \in C \mid vw \in G\}$. In particular, $w \in C_{v_1^w} \neq \varnothing$. Order V_1 by the partial order induced by the subset relation on $\{C_v \mid v \in V_1\}$. If there exists a unique maximal element v in V_1, then, for all $w \in C$, $w \in C_{v_1^w} \subseteq C_v$. Whence, $C_v = C$ and $C_v \cup \{v\} \in \mathcal{C}_G$.

If there exist more than one maximal element in V_1, then let u and v be two of them. This implies that there exist $w_u \in C_u \setminus C_v$ and $w_v \in C_v \setminus C_u$. Because w_u and w_v lie in C, they are connected. The union of P_{w_u}, the $w_u w_v$ edge and P_{w_v} contains an cycle going through u, w_u, w_v and v. As u and w_v are not connected, there must be a chord incident to w_u. Because P_{w_u} is chordless, the other end of the chord must be a vertex in $P_{w_v} \setminus \{w_v, v\}$. Let z be the neighbour of w_u in $P_{w_v} \setminus \{w_v, v\}$ which lies closest to v (measured along P_{w_v}). Consider the smaller cycle formed by the edges zw_u, $w_u w_v$ and P_{w_v} between w_v and z. It contains z, w_u, w_v and v and has length at least 4. But the vertex w_u cannot be incident to a chord, because of the minimality of z and all other vertices lie on the chordless path P_{w_v}. Thus, there cannot be a chord and there cannot be multiple maximal elements of V_1. □

Lemma 15. *A complete set of vertices $C \in \mathcal{C}_G$ is a minimal vertex separator of G, if and only if it is the maximal generator of $F(C)$ and $\Delta_{F(C)}$ contains more than one vertex.*

Proof. Let C be a minimal vertex separator. By Theorem 13, C is complete. By Lemma 14, C separates v_1 and v_2 such that $C \uplus \{v_1\}$ and $C \uplus \{v_2\}$ are complete. Hence, there are cliques $K_1, K_2 \in F(C) =: F$ with $C \uplus \{v_1\} \subseteq K_1$ and $C \uplus \{v_2\} \subseteq K_2$.

It is immediate that C is the maximal generator of F, because any generator of F is contained in both K_1 and K_2. Thus, a bigger generator would give a common neighbour of v_1 and v_2 outside of C, contradicting the fact that C is a v_1-v_2-separator.

In order to prove that Δ_F has at least two vertices, it suffices to show that there is no K_1-K_2-path in Γ_F. So assume that there was such a path P. For each edge $KK' \in P$, there is a vertex $v_{KK'} \in (K \cap K') \setminus C \neq \varnothing$. The graph $G[\{v_1, v_2\} \cup \{v_{KK'} \mid KK' \in P\}]$ contains a v_1-v_2-path. This contradicts the vertex separator property of C.

For the converse implication, let F be a clique family with Δ_F having more than two vertices. Choose two distinct vertices $[K_1]_{\sim_F}$ and $[K_2]_{\sim_F}$ of Δ_F. It follows that $K_1 \not\sim_F K_2$, implying $K_1 \cap K_2 = C(F) =: C$. Choose $v_1 \in K_1 \setminus C$ and $v_2 \in K_2 \setminus C$. We claim that C is a minimal v_1-v_2-separator.

Minimality is obvious, as, for every $v \in C$, $v_1 v v_2$ is a path in $G[V \setminus (C \setminus \{v\})]$. It remains to show that C separates v_1 and v_2. Assume for a contradiction that there is a v_1-v_2-path P in $G[V \setminus C]$. For every $w \in P$, there is a minimal v_1-v_2-separator containing $C \uplus \{w\}$. Because minimal vertex separators are complete, w is connected to all of C and $C \uplus \{w\} \subseteq K_w \in F(C)$. The sequence $(K_w)_{w \in P}$ contains a K_1-K_2-path in Γ_F, contradicting the original choice of K_1 and K_2 from different connected components. Therefore, C is a v_1-v_2-separator. □

The *reduced clique graph* [6] $\mathbf{R}_G$ of G is the subgraph of $\mathbf{K}_G$ retaining only those edges K_1K_2 with $K_1 \cap K_2$ a minimal vertex separator.

Theorem 16 (Generalisation of [6, Theorem 7]). *The set $\{K_1 \cap K_2 \mid K_1K_2 \in T\}$ is an invariant of a clique tree $T \in \mathcal{CT}_G$ and equals the set of minimal vertex separators of G. The union of the clique trees of a chordal graph G is the reduced clique graph $\mathbf{R}_G$.*

Proof. The statements are direct consequences of Lemma 15 together with the bijection in Theorem 9. □

4 Classic characterisations of clique trees

For finite chordal graphs, there exist other characterisations of clique trees besides (6). This section generalises or adapts these results to the infinite case. The characterisations are the clique intersection property in Theorem 17, the running intersection property in Theorem 18 and the maximal weight spanning tree property in Theorem 20.

A tree $T \in \mathcal{T}_{\mathbf{K}_G}$ has the *clique intersection property*, if $K_1 \cap K_2 \subseteq K_3$ holds, for every three cliques K_1, K_2, K_3 with K_3 lying on the K_1-K_2-path in T.

Theorem 17 (Generalisation of the finite case in [3, Section 3.1]). *The tree $T \in \mathcal{T}_{\mathbf{K}_G}$ is a clique tree, if and only if it fulfils the clique intersection property.*

Proof. The clique intersection property is a constraint only if $K_1 \cap K_2 \neq \varnothing$. In this case, $F(K_1 \cap K_2) =: F$ is a finite clique family.

Assume that $T \in \mathcal{T}_{\mathbf{K}_G}$ is a clique tree. Thus, $T[F]$ is a subgraph of $\mathbf{K}_G[F]$ and contains the unique K_1-K_2-path P in T. For every $K_3 \in P$, we have $K_3 \supsetneq \bigcap_{K' \in F} K' = C(F) = K_1 \cap K_2$. Thus, T fulfils the clique intersection property.

Assume that $T \in \mathcal{T}_{\mathbf{K}_G}$ fulfils the clique intersection property. It implies that $T[F]$ must be a subgraph of $\mathbf{K}_G[F]$. By Proposition 1, the set $\mathcal{C} := \{K_1 \cap K_2 : K_1K_2 \in \mathbf{K}_G\}$ is the set of maximal generators of all finite clique families of cardinality at least two. Therefore, $T[F(C)]$ is a tree, for every $C \in \mathcal{C}$. For the clique families $\mathcal{K}_G$ and $\{K\}$, for each clique K, $T[F]$ is trivially a tree. Conclude by (7a). □

An enumeration $\{K_1, K_2, \dots\}$ of $\mathcal{K}_G$ has the *running intersection property* [3, (3.1)] (after [1, Condition 3.10]), if

$$\forall\, 2 \leqslant n \in \mathbb{N} : \exists\, 1 \leqslant i < n : \quad K_n \cap \bigcup_{j=1}^{n-1} K_j \subseteq K_i\,. \tag{14}$$

A tree $T \in \mathcal{T}_{\mathbf{K}_G}$ has the *running intersection property*, if there exists an enumeration of $\mathcal{K}_G$ with the running intersection property such that the $K_n K_i$ (with $i := i(n)$ as in (14)) are the edges of T.

Theorem 18 (Generalisation of [3, Theorem 3.4])**.** *The tree $T \in \mathcal{T}_{\mathbf{K}_G}$ is a clique tree, if and only if it has the running intersection property.*

Proof. The proof of the finite case [3, Theorem 3.4] shows the equivalence to the clique intersection property. Thus, it generalises without modification to the infinite case. □

For $T \in \mathcal{CT}_G$, one obtains an enumeration of $\mathcal{K}_G$ by fixing a root, starting with it, then enumerating all its children, then their children in turn and so on recursively.

To a spanning tree $T \in \mathcal{T}_{\mathbf{K}_G}$ of the clique graph of a finite graph G assign the weight $w(T) := \sum_{K_1K_2 \in T} |K_1 \cap K_2|$. The *maximal weight spanning tree property* is another classic characterisation of finite clique trees.

Theorem 19 ([3, Theorem 3.5] after [2])**.** *Let G be a finite chordal graph. The spanning tree $T \in \mathcal{T}_{\mathbf{K}_G}$ is a clique tree, if and only if T has maximal weight with respect to w, that is*

$$T \in \operatorname{argmax}\{w(S) \mid S \in \mathcal{T}_{\mathbf{K}_G}\}\,. \tag{15}$$

Condition (15) makes no sense in the infinite case. A local version holds, though.

Theorem 20. *Let G be a chordal graph. The spanning tree $T \in \mathcal{T}_{\mathbf{K}_G}$ is a clique tree, if and only if*

$$\forall F \in \mathcal{F}_G, |F| < \infty : \quad T[F] \in \operatorname{argmax}\{w(S) \mid S \in \mathcal{T}_{\mathbf{K}_{G[F]}}\}\,. \tag{16}$$

Proof. We show the equivalence between (16) and (7b) by induction over the size of the maximal generator of a clique family. The minimal clique families are $F(K) = \{K\}$, for a clique K, and the equivalence holds trivially, as $\Delta_{\{K\}}$ contains only a single vertex $\{K\}$ and no edges. Suppose that F has minimal cardinality and violates the equivalence. Split the sum $w(T[F])$ into two parts. The first part is a sum over edges in Γ_F. By the minimality of F, the equivalence holds for all strict subfamilies of F and this sum is a constant. The second part is a sum over the edges in Ξ_F. All edges in Ξ_F have the same weight $|C(F)|$. Hence, the equivalence between maximality of the second sum and the subgraph of Δ_F induced by the edge-labels of T being spanning is obvious. □

Acknowledgements

The authors thank the anonymous referees for their detailed and constructive feedback on this work.

References

[1] 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. ISSN 0004-5411. doi:10.1145/2402.322389.

[2] Philip A. Bernstein and Nathan Goodman. Power of natural semijoins. *SIAM J. Comput.*, 10(4):751–771, 1981. ISSN 0097-5397. doi:10.1137/0210059.

[3] 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. doi:10.1007/978-1-4613-8369-7_1.

[4] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005. ISBN 978-3-540-26182-7; 3-540-26182-6; 978-3-540-26183-4.

[5] G. A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961. ISSN 0025-5858.

[6] 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. doi:10.1007/3-540-60618-1_88.

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

[8] R. Halin. On the representation of triangulated graphs in trees. *European J. Combin.*, 5(1):23–28, 1984. ISSN 0195-6698.

[9] 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. ISSN 0020-0190. doi:10.1016/0020-0190(89)90070-7.