# Solving NP-hard Problems on GaTEx Graphs: Linear-Time Algorithms for Perfect Orderings, Cliques, Colorings, and Independent Sets*

Marc Hellmuth[1] and Guillaume E. Scholz[2]

[1] *Department of Mathematics, Faculty of Science, Stockholm University, SE-10691 Stockholm, Sweden*
`marc.hellmuth@math.su.se`
[2] *Bioinformatics Group, Department of Computer Science & Interdisciplinary Center for Bioinformatics, Universität Leipzig, Härtelstraße 16–18, D-04107 Leipzig, Germany.*

**Abstract**

The class of GAlled-Tree EXplainable (GaTEx) graphs has recently been discovered as a natural generalization of cographs. Cographs are precisely those graphs that can be uniquely represented by a rooted tree where the leaves correspond to the vertices of the graph. As a generalization, GaTEx graphs are precisely those that can be uniquely represented by a particular rooted acyclic network, called a galled-tree.

This paper explores the use of galled-trees to solve combinatorial problems on GaTEx graphs that are, in general, NP-hard. We demonstrate that finding a maximum clique, an optimal vertex coloring, a perfect order, as well as a maximum independent set in GaTEx graphs can be efficiently done in linear time. The key idea behind the linear-time algorithms is to utilize the galled-trees that explain the GaTEx graphs as a guide for computing the respective cliques, colorings, perfect orders, or independent sets.

***Keywords:*** modular decomposition galled-tree cograph NP-hard problems linear-time algorithms

## 1 Introduction

Modular decomposition is a general technique to display nested "substructures" (modules) of a given graph in the form of a rooted tree (the *modular decomposition tree of G*) whose inner vertices are labeled with "0", "1", and "prime". Cographs are precisely those graphs for which the modular decomposition tree has no prime vertices. In this case, complete structural information of the underlying cograph, i.e., the knowledge of whether two vertices are linked by an edge or not, is provided by the modular decomposition tree. As a consequence, these modular decomposition trees serve as a perfect guide for algorithms to efficiently solve many computationally hard problems on cographs (e.g., the graph-isomorphism problem or classical NP-hard problems like "minimum independent set", "maximum clique", or "minimum vertex coloring") [5, 6]. However, when encountering prime vertices, conventional modular decomposition trees do not provide full structural information about the underlying graphs and become less useful for algorithmic solutions to hard problems. To circumvent this issue, we aim at using modular decomposition *networks* instead of trees. In [15], we focused on particular networks, called galled-trees, that are obtained from the modular decomposition tree by replacing prime vertices by rooted 0/1-labeled cycles. A graph $G=(X,E)$ is GA*lled*-T*ree* EX*plainable* (GaTEx) if there is a 0/1-labeled galled-tree $(N,t)$ such that $x,y \in E$ if and only if the label $t(\mathrm{lca}_N(x,y))$ of the unique least-common ancestor of $x$ and $y$ in $N$ is "1". GaTEx graphs, thus, naturally generalize the concept of cographs. Further exploration of the class of GaTEx graphs in [17] shows that these graphs are characterized by the absence of 25 forbidden subgraphs. This, in turn, implies that GaTEx graphs are closely linked to other famous graph classes such as weakly-chordal graphs, perfect graphs with perfect order, comparability and permutation graphs, murky graphs as well as interval graphs, Meyniel graphs, or very strongly-perfect and brittle graphs. In addition, every GaTEx graph has twin-width at most 1.

Cotrees serve as a guide for algorithms on cographs to solve many combinatorial problems that are classified as "hard". In this contribution, we ask whether the galled-trees that explain GaTEx graphs can be used in a similar manner. In particular, we are interested in the following classical NP-hard problems [11]: Determining the size $\omega(G)$

*This contribution is an extended version of the COCOON'23 paper [16].

of a maximum clique and finding such a clique, the size $\chi(G)$ of an optimal vertex-coloring and finding such a coloring, the size $\alpha(G)$ of a maximum independent set of a given graph $G$ and finding such an independent set. In general, determining the invariants $\omega(G)$, $\chi(G)$, and $\alpha(G)$ for arbitrary graphs $G$, as well as finding the underlying optimal cliques, colorings, and independent sets, is an NP-hard task [11]. All these invariants are not only of interest from a theoretical point of view but also have many practical applications in case the underlying graph models real-world structures, e.g., social networks [20], gene/protein-interaction networks [1, 25], job/time-slots assignments in scheduling problems [21], and many more. In addition, we consider the problem of determining a perfect ordering of GATEX graphs, i.e., an ordering of the vertices of $G$ such that a greedy coloring algorithm with that ordering optimally colors every induced subgraph of $G$. As shown by Middendorf and Pfeiffer [23], the problem of deciding whether a graph is perfectly orderable is NP-complete. As we will argue below, the problem of finding a perfect ordering remains NP-hard even for perfectly orderable graphs.

We show here that $\omega(G)$, $\chi(G)$, $\alpha(G)$ as well as a perfect ordering can be computed in linear time for GATEX graphs $G$. The crucial idea for the linear-time algorithms is to avoid working directly on the GATEX graphs $G$ but rather to utilize the galled-trees that explain $G$ as a guide for the algorithms to compute these invariants. In particular, we show first how to employ the galled-tree structure to compute a perfect ordering of GATEX graphs. This result is then used to determine $\omega(G)$, $\chi(G)$, $\alpha(G)$. In addition, we provide algorithms to find a maximum clique, an optimal vertex coloring as well as a maximum independent set in GATEX graphs in linear-time.

## 2 Preliminaries

**Graphs.** We consider graphs $G = (V,E)$ with vertex set $V(G) := V \neq \emptyset$ and edge set $E(G) := E$. A graph $G$ is *undirected* if $E$ is a subset of the set of two-element subsets of $V$ and $G$ is *directed* if $E \subseteq V \times V \setminus \{(v,v) \mid v \in V\}$. Thus, edges $e \in E$ in an undirected graph $G$ are of the form $e = \{x,y\}$ and in directed graphs of the form $e = (x,y)$ with $x,y \in V$ being distinct. We write $H \subseteq G$ if $H$ is a subgraph of $G$ and $G[W]$ for the subgraph in $G$ that is induced by some subset $W \subseteq V$. A $P_4$ denotes an induced undirected path on four vertices. We often write $a-b-c-d$ for an induced $P_4$ with vertices $a,b,c,d$ and edges $\{a,b\},\{b,c\},\{c,d\}$. An undirected graph is *connected* if, for every two vertices $u,v \in V$, there is a path connecting $u$ and $v$. A directed graph is *connected* if its underlying undirected graph is connected. A (directed or undirected) graph $G$ is *biconnected* if it contains no vertex whose removal disconnects $G$. A *biconnected component* of a $G$ is an inclusion-maximal biconnected subgraph. If such a biconnected component is not a single vertex or an edge, then it is called *non-trivial*.

**Remark 2.1.** From here on, we will call an undirected graph simply *graph*.

For two graphs $G$ and $H$ we put $G - H := (V(G) \setminus V(H), E(G) \setminus F)$ with $F \subseteq E(G)$ being the collection of all edges incident to vertices in $V(H)$, and $G \cap H := (V(G) \cap V(H), E(G) \cap E(H))$. For two vertex-disjoint graphs $G$ and $H$, their *disjoint union* is defined as $G \mathbin{\dot\cup} H := (V(G) \mathbin{\dot\cup} V(H), E(G) \mathbin{\dot\cup} E(H))$ while their *join union* is defined as $G \bowtie H := (V(G) \mathbin{\dot\cup} V(H), E(G) \mathbin{\dot\cup} E(H) \mathbin{\dot\cup} \{\{x,y\} \mid x \in V(G), y \in V(H)\})$.

A *clique* of a graph $G$ is an inclusion-maximal complete subgraph $G$. The size of a maximum clique of $G$ is called the *clique number* and denoted by $\omega(G)$. A *coloring* of a graph $G$ is a map $\sigma\colon V(G) \to S$, where $S$ denotes a set of colors, such that $\sigma(u) \neq \sigma(v)$ for all $\{u,v\} \in E(G)$. The minimum number of colors needed for a coloring of $G$ is called the *chromatic number* of $G$ and denoted by $\chi(G)$. A subset $W \subseteq V(G)$ of pairwise non-adjacent vertices is called *independent set*. The size of a maximum independent set in $G$ is called the *independence number* of $G$ and denoted by $\alpha(G)$. In general, determining the invariants $\omega(G)$, $\chi(G)$ and $\alpha(G)$ for arbitrary graphs is an NP-hard task [11].

A graph $G$ is *perfect*, if the chromatic number of every induced subgraph equals the size of the largest clique of that subgraph. We consider total orders $\zeta = v_1 \dots v_{|V|}$ defined on the vertex set $V$ of graphs $G = (V,E)$ and assume that $v_i < v_j$ precisely if $v_i$ is left of $v_j$ in this sequence $\zeta$ (or equivalently, if $i < j$ in case indices are provided). We denote with $\zeta_{|H}$ the order $\zeta$ that is restricted to $V(H)$. Let $X$ and $Y$ be two disjoint subsets of $V(G)$. If $\zeta_1 = x_1, x_2, \dots x_l$ and $\zeta_2 = y_1, y_2, \dots y_m$ are two total orderings on $X$ and $Y$, respectively, then we denote with $\zeta_1\zeta_2$ the total ordering on $X \cup Y$ given by concatenating $\zeta_1$ and $\zeta_2$, i.e., $\zeta_1\zeta_2 = x_1, x_2, \dots x_l y_1, y_2, \dots y_m$.

For a given total order $\zeta$ of $G$, a *greedy coloring algorithm* scans the vertices in order $\zeta$ and assigns to each vertex $v$ the smallest positive integer (color) assigned to none of the vertices $w < v$ that are adjacent to $v$. A coloring of $G$ obtained with such an algorithm is called *greedy coloring*. A total order $\zeta$ of $G$ is *perfect* if, for all induced subgraphs $H$ of $G$, a greedy coloring algorithm that scans the vertices in order $\zeta_{|H}$ uses the minimum number of colors to color $H$. A graph $G$ is *perfectly orderable* if it admits a *perfect order* $\zeta$. A total order $\zeta$ on $G$ contains an *obstruction* (w.r.t. $G$) if there is an induced $P_4$ $a-b-c-d$ in $G$ such that $a < b$ and $c > d$ w.r.t. this order $\zeta$. Every perfectly orderable graph is a perfect graph [4].

**Proposition 2.2** ([4]). *A total order $\zeta$ on a graph $G$ is a perfect order if and only if $\zeta$ does not contain any obstructions.*

Perfectly orderable graphs are NP-complete to recognize [23]. By Prop. 2.2, one can test in polynomial-time as whether a given order is perfect: simply check as whether one of the $O(|V|^4)$ induced $P_4$s yields an obstruction. This, in particular, implies that the problem to find a perfect ordering of a graph remains NP-hard, even if the graph is already known to be perfectly orderable.

**Trees, Galled-trees and GATEX graphs.** (Phylogenetic) trees and galled-trees are particular directed acyclic graphs (DAGs). To be more precise, a *galled-tree* $N = (V,E)$ on $X$ is a DAG such that either

*(N0)* $V = X = \{x\}$ and, thus, $E = \emptyset$.

or $N$ satisfies the following four properties

*(N1)* There is a unique *root* $\rho_N$ with indegree 0 and outdegree at least 2; and

*(N2)* $x \in X$ if and only if $x$ has outdegree 0 and indegree 1; and

*(N3)* Every vertex $v \in V^0 := V \setminus X$ with $v \neq \rho_N$ has

(i) indegree 1 and outdegree at least 2 (*tree-vertex*) or

(ii) indegree 2 and outdegree at least 1 (*hybrid-vertex*).

*(N4)* Each biconnected component $C$ contains at most one hybrid-vertex $v$ for which the two vertices $v_1, v_2$ with $(v_1,v),(v_2,v) \in E$ belong to $C$.

We note that in [15] galled-trees have been called level-1 networks. By definition, every non-trivial biconnected component in a galled-tree $N$ forms an (rooted) "*cycle*" $C$ in $N$ [3, 19] that is composed of two directed paths $P^1(C)$ and $P^2(C)$ in $N$ (called *sides* of $C$) with the same start-vertex $\rho_C$ (the root of $C$) and end-vertex $\eta_C$ (the hybrid-vertex of $C$) and whose internal vertices, i.e., vertices in $C$ that are distinct from $\rho_C$ and $\eta_C$, are pairwise distinct. *Trees* are galled-trees without hybrid-vertices. The leaf set $L(N)$ of a galled-tree $N$ is $X$, i.e., the set of all vertices satisfying *(N2)*,

Let $N = (V,E)$ be a galled-tree on $X$. A vertex $u \in V$ is called an *ancestor* of $v \in V$ and $v$ a *descendant* of $u$, in symbols $v \preceq_N u$, if there is a directed path (possibly reduced to a single vertex) in $N$ from $u$ to $v$. We write $v \prec_N u$ if $v \preceq_N u$ and $u \neq v$. If $(u,v) \in E$, then the vertex $v$ is a *child of* $u$ and $u$ is a *parent of* $v$. The set of children, resp., parents of a vertex $w$ in $N$ is denoted by $\mathrm{child}_N(w)$, resp., $\mathrm{par}_N(w)$. For a non-empty subset of leaves $A \subseteq X$, we define $\mathrm{lca}_N(A)$, or a *lowest common ancestor of* $A$, to be a $\preceq_N$-minimal vertex of $N$ that is an ancestor of every vertex in $A$. For simplicity we put $\mathrm{lca}_N(x,y) := \mathrm{lca}_N(\{x,y\})$. By Lemma 49 and 67 in [14], galled-trees $N$ are "lca-networks", i.e., $\mathrm{lca}_N(A)$ is uniquely determined for all $A \subseteq L(N)$.

We define $N(w)$ as the subgraph of $N$ rooted at $w$, i.e., the DAG induced by $w$ and all its descendants. Morever, if the context is clear, we often write $L_w = L(N(w))$ for $w \in V(N)$.

A galled-tree $N$ on $X$ is *elementary* if it contains a single rooted cycle $C$ of length $|X|+1$ with root $\rho_C = \rho_N$ and single hybrid-vertex $\eta_C \in V(C)$ and additional edges $\{v_i, x_i\}$ such that every vertex $v_i \in V(C) \setminus \{\rho_C\}$ is adjacent to a unique vertex $x_i \in X$. A galled-tree is *strong* if it *does not* contain cycles of the following form: (i) $P^1(C)$ or $P^2(C)$ consist of $\rho_C$ and $\eta_C$ only or (ii) both $P^1(C)$ and $P^2(C)$ contain only one vertex distinct from $\rho_C$ and $\eta_C$.

The tuple $(N,t)$ denotes a galled-tree $N = (V,E)$ on $X$ that is equipped with a *(vertex-)labeling* $t$ i.e., a map $t\colon V \to \{0,1,\odot\}$ such that $t(x) = \odot$ if and only if $x \in X$. The graph $\mathcal{G}(N,t) = (X,E)$ with vertex set $X$ and edges $\{x,y\} \in E$ precisely if $t(\mathrm{lca}_N(x,y)) = 1$ is said to be *explained* by $(N,t)$. A graph $G = (X,E)$ is GA*lled*-T*ree* EX*plainable (*GATEX*)* if there is a labeled galled-tree $(N,t)$ such that $G \simeq \mathcal{G}(N,t)$. A labeling $t$ (or equivalently $(N,t)$) is *quasi-discriminating* if $t(u) \neq t(v)$ for all $(u,v) \in E$ with $v$ not being a hybrid-vertex. We note in passing, that quasi-discriminating labelings form a natural generalization of discriminating labelings $t$ that require $t(u) \neq t(v)$ for all $(u,v) \in E$ [2].

**Proposition 2.3** ([15])**.** GATEX *graphs can be recognized in linear-time and a galled-tree* $(N,t)$ *that explains a* GATEX *graphs can be constructed in linear-time as well.*

Moreover, GATEX graphs are characterized by a finite set of forbidden subgraphs [17]. GATEX graphs that are explained by labeled trees $(T,t)$ are precisely the cographs and, therefore, those graphs that do not contain induced $P_4$s [5].

**Modular Decomposition (MD).** A *module* $M$ of a graph $G = (X,E)$ is a subset $M \subseteq V(G) = X$ such that for all $x,y \in M$ it holds that $N_G(x) \setminus M = N_G(y) \setminus M$, where $N_G(x)$ is the set of all vertices of $X$ that are adjacent to $x$ in $G$. A module $M$ of $G$ is *strong* if $M$ does not *overlap* with any other module of $G$, that is, $M \cap M' \in \{M, M', \emptyset\}$ for all modules $M'$ of $G$. The set of strong modules $\mathbb{M}_{\mathrm{str}}(G) \subseteq \mathbb{M}(G)$ is uniquely determined [10, 18] and forms a hierarchy which gives rise to a unique tree representation $\mathcal{T}_G$ of $G$, known as the *modular decomposition tree* (*MDT*) of $G$. Uniqueness and the hierarchical structure of $\mathbb{M}_{\mathrm{str}}(G)$ implies that there is a unique partition $\mathbb{M}_{\max}(G) = \{M_1, \dots, M_k\}$ of $X$ into inclusion-maximal strong modules $M_j \neq X$ of $G$ [8, 9].

Similar as for galled-trees, one can equip $\mathcal{T}_G$ with a vertex-labeling $t_G$ such that, for $M \in \mathbb{M}_{\mathrm{str}}(G) = V(\mathcal{T}_G)$, we have $t_G(M) = \odot$ if $|M| = 1$; $t_G(M) = 0$ if $|M| > 1$ and $G[M]$ is disconnected; $t_G(M) = 1$ if $|M| > 1$ and $G[M]$ is connected but $\overline{G}[M]$ is disconnected; $t_G(M) = prime$ in all other cases. Strong modules of $G$ are called *series*, *parallel* and *prime* if $t_G(M) = 1$, $t_G(M) = 0$ and $t_G(M) = prime$, respectively. Efficient linear algorithms to compute $(\mathcal{T}_G, t)$ have been proposed e.g. in [7, 22, 26]. The *quotient graph* $G/\mathbb{M}_{\max}(G)$ has $\mathbb{M}_{\max}(G)$ as its vertex set and edges $\{M_i, M_j\} \in E(G/\mathbb{M}_{\max}(G))$ if and only if there are $x \in M_i$ and $y \in M_j$ that are adjacent in $G$. As argued in [12], this quotient graph is well-defined.

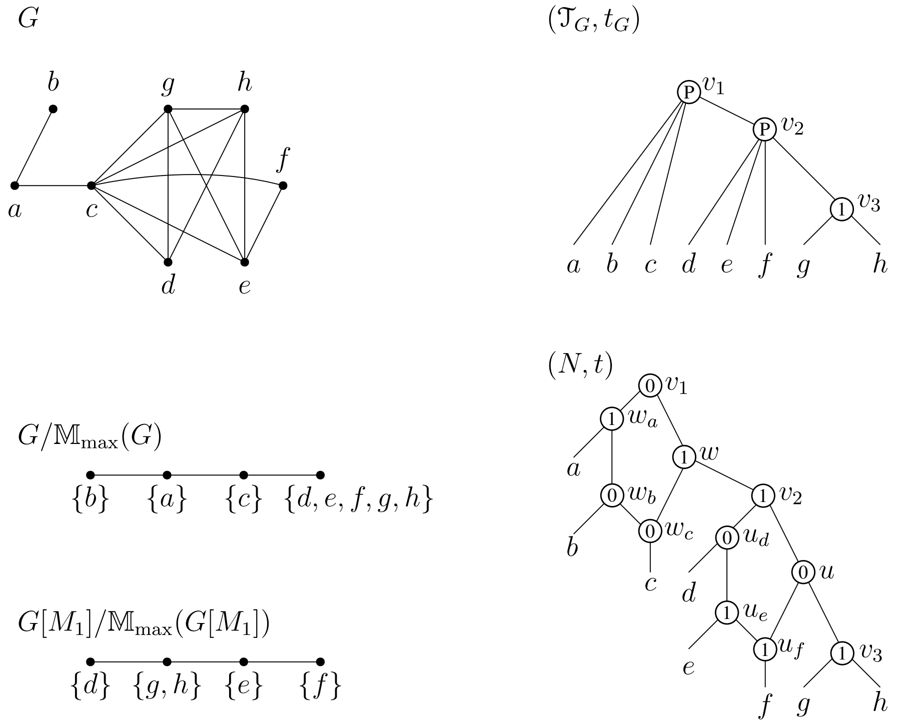


**Figure 1:** Shown is a GATEX graph $G = (V, E)$ (top left) together with its modular decomposition tree $(\mathfrak{T}_G, t_G)$ (top right) and a pvr-network $(N, t)$ (bottom right) that explains $G$. The graph $G$ has as strong modules the singletons $\{x\}$, $x \in V = \{a, b, c, \dots, g\}$, the entire vertex set $V$ and the sets $M_1 = \{d, e, f, g, h\}$ and $M_2 = \{g, h\}$. Each vertex $w$ in $\mathfrak{T}_G$ represent the strong module $L(\mathfrak{T}_G(w))$. The graph $G$ has two prime modules, namely $M_1 = L(\mathfrak{T}_G(v_2))$ and $V = L(\mathfrak{T}_G(v_1))$. The respective quotient graphs $H_1 := G/\mathbb{M}_{\max}(G)$ and $H_2 := G[M_1]/\mathbb{M}_{\max}(G[M_1])$ are shown bottom left. The pvr-network $(N, t)$ is a galled-tree that is obtained from $(\mathfrak{T}_G, t_G)$ by locally replacing the vertex $v_i$ by the strong quasi-discriminating elementary galled-tree $(N_{v_i}, t_{v_i})$ that explains $H_i$, $i \in \{1, 2\}$ (cf. Def. 2.4).

**From Modular Decomposition Trees to Galled-trees.** Galled-trees that explain a given GATEX graph $G$ can be obtained from the modular decomposition trees $(\mathfrak{T}_G, t_G)$ by replacing its prime vertices locally by simple rooted cycles. To this end, we first compute for prime vertices $v$ and the corresponding prime modules $M = L(\mathfrak{T}_G(v))$ the quotient $H = G[M]/\mathbb{M}_{\max}(G[M])$ which can be explained by a strong elementary quasi-discriminating galled-tree $(N_v, t_v)$ (cf. [15, Thm. 6.10]). We then use the rooted cycles in $(N_v, t_v)$ to replace $v$ in $(\mathfrak{T}_G, t_G)$, see Figure 2 for an illustrative example. The latter is formalized as follows.

**Definition 2.4** (prime-vertex replacement (pvr) networks)**.** *Let $G$ be a* GATEX *graph and $\mathscr{P}$ be the set of all prime vertices in $(\mathfrak{T}_G, t_G)$. A* prime-vertex replacement (pvr) *networks $(N, t)$ of $G$ (or equivalently, of $(\mathfrak{T}_G, t_G)$) is obtained by the following procedure:*

1. *For all $v \in \mathscr{P}$, let $(N_v, t_v)$ be a strong quasi-discriminating elementary galled-tree with root $v$ that explains $G[M]/\mathbb{M}_{\max}(G[M])$ with $M = L(\mathfrak{T}_G(v))$.*
2. *For all $v \in \mathscr{P}$, remove all edges $(v, u)$ with $u \in \mathrm{child}_{\mathfrak{T}_G}(v)$ from $\mathfrak{T}_G$ to obtain the forest $(T', t_G)$ and add $N_v$ to $T'$ by identifying the root of $N_v$ with $v$ in $T'$ and each leaf $M'$ of $N_v$ with the corresponding child $u \in \mathrm{child}_{\mathfrak{T}_G}(v)$ for which $M' = L(\mathfrak{T}_G(u))$.*

   This results in the *pvr-network $N$ of $G$.*
3. *Define the labeling $t\colon V(N) \to \{0, 1, \odot\}$ by putting, for all $w \in V(N)$,*

$$t(w) = \begin{cases} t_G(v) & \text{if } v \in V(\mathfrak{T}_G) \setminus \mathscr{P} \\ t_v(w) & \text{if } w \in V(N_v) \setminus X \text{ for some } v \in \mathscr{P} \end{cases}$$

Note that the leaves of the pvr-network $N$ of $G$ are the singletons $\{x\}$, $x \in V(G)$. In the remainder of this paper, we will always implicitly identify each singleton with its unique elements. In other words, we will always assume that the leaf set of $\mathfrak{T}_G$ as well as of pvr-network $N$ of $G$ is $V(G)$. By construction, we have $V(\mathfrak{T}_G) \subseteq V(N)$ given that $N$ is the pvr-network of $G$. More precisely, $V(\mathfrak{T}_G)$ is precisely the set of vertices $v$ of $N$ such that either $v$ does not belong to a cycle $C$ of $N$, or $v = \rho_C$ for some cycle $C$ of $N$. In addition, we have

**Observation 2.5.** *For a vertex $v$ in in the pvr-network $N$ of $G$, the following holds:*

*(i) If $v$ does not belong to any cycle in $N$, then all children of $v$ in $N$ are children of $v$ in $\mathcal{T}_G$.*

*(ii) If there exists a cycle $C$ of $N$ such that $v \in V(C) \setminus \{\rho_C\}$, then $v$ has a unique child $w$ in $V(N) \setminus V(C)$, and $w$ is a child of $\rho_C$ in $\mathcal{T}_C$.*

The construction of a pvr-network for a GATEX graph is well-defined and can be done in linear-time, cf. [15, Alg. 4 & Thm. 9.4]. By [15, Prop. 7.4 & 8.3], a pvr-network $(N,t)$ of a GATEX graph $G$ is a galled-tree that explains $G$. Moreover, there is a 1:1 correspondence between cycles $C$ in $N$ and prime modules $M$ of $G$. By the latter result, we can define $C_M$ as the unique cycle in $N$ corresponding to prime module $M$. For later reference, we summarize now a couple of results that are easy to verify or that have been established in [15].

**Observation 2.6.** *Let $(N,t)$ be a pvr-network of a* GATEX *graph $G$. Then,*

- *$(N,t)$ is a galled-tree that explains $G$ [15, Prop. 7.4].*
- *There is a 1:1 correspondence between the cycle $C$ in $N$ and prime modules $M$ of $G$ [15, Prop. 8.3].*

  *Hence, we can define $C_M$ as the unique cycle in $N$ corresponding to prime module $M$.*

  *Moreover, $G_1(M), G_2(M) \subseteq G[M]$ will denote the subgraphs induced by leaf-descendants of the vertices in $P^1(C_M) - \rho_{C_M}$ and $P^2(C_M) - \rho_{C_M}$, respectively.*

*Moreover, let $v$ be a prime vertex associated with the prime module $M_v = L(\mathcal{T}_G(v))$ module and let $C \coloneqq C_{M_v}$. Since we used strong elementary networks for the replacement of $v$, one easily verifies that:*

- *$C$ has a unique root $\rho_C$ and a unique hybrid-vertex $\eta_C$.*
- *$\eta_C$ has precisely one child and precisely two parents.*
- *All vertices $v \neq \eta_C$ have two children and one parent.*

  *In particular, all vertices $v \neq \eta_C, \rho_C$ have one child $u'$ located in $C$ and one child $u''$ that is not located in $C$ and these children satisfy $L(N(u')) \cap L(N(u'')) = \emptyset$ and it holds that $\operatorname{lca}_N(x,y) = w$ for all $x \in L(N(u'))$ and $y \in L(N(u''))$.*

  *Both children $u'$ and $u''$ of $\rho_C$ are located in $C$ and satisfy $L(N(u')) \cap L(N(u'')) = L(N(\eta_C))$. Moreover, $L(N(\eta_C)) \cap L(N(v_2)) = \emptyset$ for the child $v_2$ of $v \neq \eta_C, \rho_C$ that is not located in $C$.*

# 3 Perfect orderings and optimal colorings

In this section, we provide linear-time algorithms to compute the chromatic number $\chi(G)$ and an optimal coloring of a given GATEX graph $G$. For this purpose, we show first how to employ the structure of labeled galled-trees $(N,t)$ to determine a perfect ordering of GATEX graphs in linear time (cf. Alg. 1). To this end, we provide the following result for later reference.

**Lemma 3.1.** *Let $P = a - b - c - d$ be an induced $P_4$ in a* GATEX *graph $G$ and $(N,t)$ a pvr-network that explains $G$. Moreover, let $M$ be the inclusion-minimal strong module of $G$ that contains $V(P)$, i.e., $V(P) \subseteq M$ and there is no strong module $M'$ of $G$ that satisfies $V(P) \subseteq M' \subsetneq M$. Then, $M$ is a prime module of $G$. Moreover, in the unique cycle $C_M$ in $N$ that corresponds to $M$, there are vertices $u_a, u_b, u_c, u_d \in V(C_M)$ that satisfy the following conditions:*

1. *For $x \in \{a,b,c,d\}$ it holds that $x \in L(N(u'_x))$ where $u'_x$ is the unique child of $u_x$ that is not located in $C_M$.*
2. *The vertices $u_a, u_b, u_c, u_d$ are pairwise distinct.*
3. *The vertices $u_a, u_b, u_c, u_d$ do not all belong to the same side of $C_M$.*
4. *One of $u_a, u_b, u_c, u_d$ coincides with the unique hybrid $\eta_{C_M}$ of $C_M$.*

*Proof.* Let $P = a - b - c - d$ be an induced $P_4$ in a GATEX graph $G$ and $(N,t)$ a pvr-network that explains $G$. Put $Y \coloneqq \{a,b,c,d\}$. Moreover, let $M$ be the inclusion-minimal strong module of $G$ that contains $V(P)$.

We show first that $M$ is a prime module of $G$. By definition, we must show that $G[M]$ and $\overline{G}[M]$ are connected. Assume that $P = a - b - c - d$ and put $Y \coloneqq V(P)$. Observe first that $G[Y] = P$ and $\overline{G}[Y] = c - a - d - b$. Hence, both $G[Y]$ and $\overline{G}[Y]$ are connected. Assume, for contradiction, that $G[M]$ is disconnected. In this case, $P$ belongs to some connected component $H$ of $G[M]$. We show that, in this case, $V(H)$ must be a strong module of $G[M]$. Clearly, $V(H)$ is a module of $M$. Assume, for contradiction, that $V(H)$ is not strong. Hence, it overlaps with some module $M'$ of $G[M]$ and, therefore, $V(H) \cap M' \neq \emptyset$, $V(H) \setminus M' \neq \emptyset$ and $M' \setminus V(H) \neq \emptyset$. In particular, since $H$ is connected, there is a vertex $x \in V(H) \cap M'$ that is adjacent to some vertex $y \in V(H) \setminus M'$. However, since $H$ is a connected component, none of the vertices $z \in M' \setminus V(H)$ can be adjacent to $y$. Hence, $M'$ is a not a module; a contradiction. Thus, $V(H)$ is a strong module of $G[M]$ and, by [13, Lemma 3.1], $V(H)$ is a strong module of $G$; a contradiction to the choice of $M$. Thus, $G[M]$ is connected. By similar arguments and since $\overline{G}[Y] = c - a - d - b$, $\overline{G}[M]$ must be connected as well. Consequently, $M$ is a prime module of $G$.

Since $M$ is a prime module of $G$, there is a unique cycle $C_M$ in $N$ corresponding to $M$. To recall, $L_w = L(N(w))$ for $w \in V(N)$. For a vertex $w \in V(\mathcal{T}_G)$, we denote with $M_w \coloneqq L(\mathcal{T}_G(w))$ the module of $G$ "associated" with $w$. For all

**Algorithm 1** `Perfect ordering of GATEx graphs` $G$

**Input:** A GATEX graph $G=(V,E)$
**Output:** A perfect ordering $\zeta$ of the vertices of $V(G)$

1: Construct $(\mathcal{T}_G, t_G)$ and pvr-network $(N,t)$ of $G$
2: Initialize $\zeta(v) := v$ for all leaves $v$ in $\mathcal{T}_G$
3: **for all** $v \in V(\mathcal{T}_G) \setminus L(\mathcal{T}_G)$ in postorder **do**
4:   **if** $t_G(v) \in \{0,1\}$ **then**
5:     Put $\zeta(v) := \zeta(v_1)\ldots\zeta(v_k)$ arbitrarily for the $k = |\mathrm{child}_{\mathcal{T}_G}(v)|$ children $v_1,\ldots,v_k$ of $v$ in $\mathcal{T}_G$
6:   **else** ▷ $t_G(v) = \text{prime}$
7:     Let $C$ be the unique cycle in $N$ with root $\rho_C = v$.
8:     For all vertices $w \in V(C)\setminus\{\rho_C\}$, put $\zeta(w) := \zeta(w')$, where $w'$ is the unique child of $w$ in $N$ that is not a vertex of $C$.
9:     Put $\zeta^*(v) := \zeta(v_1)\ldots\zeta(v_k)$ arbitrarily for the $k = |V(C)|-2$ vertices $v_1,\ldots,v_k$ in $V(C)\setminus\{\rho_C,\eta_C\}$
10:     **if** $t(\rho_C) = 0$ **then**
11:       $\zeta(v) := \zeta(\eta_C)\zeta^*(v)$
12:     **else** ▷ $t(\rho_C) = 1$
13:       $\zeta(v) := \zeta^*(v)\zeta(\eta_C)$
14: **return** $\zeta(v)$

vertices $u \in V(C)\setminus\{\rho_C\}$, we put $L'_u = L_{u'}$, where $u'$ is the unique child of $u$ that is not in $V(C)$. By Obs. 2.6, the sets $L'_u, u \in V(C)\setminus\{\rho_C\}$ are pairwise disjoint strong modules of $G$. In particular, for all $x \in M$, there exists a unique vertex $u_x \in V(C)\setminus\{\rho_C\}$ such that $x \in L'_{u_x}$. Consequently, Condition (1) is satisfied.

We show now that the vertices $u_a, u_b, u_c$ and $u_d$ are pairwise distinct. If $u_a = u_b = u_c = u_d$, then $Y \subseteq L'_{u_a}$. Since $L'_{u_a}$ is a strong module of $G$ satisfying $L'_{u_a} \subsetneq M$, this contradicts the choice of $M$. If exactly three of $u_a, u_b, u_c, u_d$ are equal, then there exists $u \in V(C)\setminus\{\rho_C\}$ and $x \in Y$ such that $Y\setminus\{x\} \subseteq L'_u$ and $x \notin L'_u$. This and the fact that $L'_u$ is a module of $G$ implies that either all or none of of the vertices in $L'_u$ (and thus, of $Y\setminus\{x\}$) are adjacent to $x$. Consequently, $x$ has degree 0 or 3 in $G[Y]$, a contradiction since $G[Y] = P$. Finally, if two of $u_a, u_b, u_c, u_d$ are equal and distinct from the other two, then there exists $u \in V(C)\setminus\{\rho_C\}$ and $x, y \in Y$ distinct such that $Y \cap L'_u = \{x,y\}$. Since $L'_u$ is a module, then for all $z \in Y\setminus\{x,y\}$, $\{x,z\}$ is an edge of $G[Y]$ if and only if $\{y,z\}$ is an edge of $G[Y]$. However, since $G[Y] = P$, there is no pair $\{x,y\}$ of elements of $Y$ satisfying this property. Therefore, the vertices $u_a, u_b, u_c$ and $u_d$ are pairwise distinct and Condition (2) is satisfied.

This in particular implies that, for $x, y \in Y$ distinct, $\mathrm{lca}_N(x,y) \in \{u_x, u_y, \rho_C\}$. More specifically, we $\mathrm{lca}_N(x,y) = u_x$ if $u_y \prec_N u_x$, $\mathrm{lca}_N(x,y) = u_y$ if $u_x \prec_N u_y$, and $\mathrm{lca}_N(x,y) = \rho_C$ if $u_x$ and $u_y$ are $\preceq_N$-incomparable. Assume, for contradiction, that the vertices $u_a, u_b, u_c$ and $u_d$ all belong to the same side of $C$. In this case, there is a vertex $x \in Y$ such that $u_x$ is an ancestor of $u_a, u_b, u_c$ and $u_d$ in $N$. In view of the above, $\mathrm{lca}_N(x,y) = u_x$ for all $y \in Y\setminus\{x\}$. Hence, $x$ has degree 0 in $G[Y]$ if $t(u_x) = 0$, and degree 3 if $t(u_x) = 1$. Since $G[Y] = P$, none of these cases can occur. Hence, Condition (3) is satisfied.

We now show that one of $u_a, u_b, u_c$ and $u_d$ coincides with $\eta := \eta_C$. Assume, for contradiction, that this is not the case. Then two situations may occur: exactly three of $u_a, u_b, u_c$ and $u_d$ belong to the same side of $C$, or two of $u_a, u_b, u_c$ and $u_d$ belong to one side of $C$, and the other two belong to the other side. Suppose first that there exists $x \in Y$ such that $u_x$ is the only vertex on its side of $C$. Then we have $\mathrm{lca}_N(x,y) = \rho_C$ for all $y \in Y\setminus\{x\}$. Hence, $x$ has degree 0 in $G[Y]$ if $t(\rho_C) = 0$, and degree 3 if $t(\rho_C) = 1$. Since $G[Y] = P$, both cases are impossible. Suppose now that there exists $x, y \in Y$ distinct such that $u_x$ and $u_y$ belong to one side of $C$. In particular, for all $z \in Y\setminus\{x,y\}$, we have $\mathrm{lca}_N(x,z) = \mathrm{lca}_N(y,z) = \rho_C$. It follows that $G[Y]$ is disconnected if $t(\rho_C) = 0$ and $\overline{G}[Y]$ is disconnected if $t(\rho_C) = 1$. Since $G[Y] = P$, both cases are impossible. Hence, one of $u_a, u_b, u_c$ and $u_d$ coincides with $\eta$ and Condition (4) is satisfied. □

As we shall see, Algorithm 1 can be used to compute a perfect order in GATEX graphs in linear-time. Before studying Algorithm 1 in detail, we illustrate this algorithm on the example shown in Figure 2.

**Example 3.2.** *We exemplify here the main steps of Algorithm 1 using as input the* GATEX *graph $G$ as shown in Fig. 2. We first compute the modular decomposition tree $(\mathcal{T}_G, t_G)$ (as shown in Fig. 1) and the shown pvr-network $(N,t)$ that explains $G$ (Line 1). For all leaves $v$ of $\mathcal{T}_G$ (and thus, of $N$), we initialize the perfect order $\zeta(v) = v$ of the induced subgraph $G[\{v\}]$ (Line 2). We then traverse the vertices $\mathcal{T}_G$ that are not leaves in postorder and thus obtain the order $v_3, v_2, v_1$ in which the vertices are visited (Line 3). Note that postorder-traversal ensures that all children of a given vertex $v$ in $\mathcal{T}_G$ are visited before this vertex $v$ is processed. Since $v_3$ is a non-prime vertex of $\mathcal{T}_G$ (Line 4), we can choose one of the orders $\zeta(g)\zeta(h)$ or $\zeta(h)\zeta(g)$ (Line 5) and decide, in this example, to put $\zeta(v_3) = \zeta(g)\zeta(h) = gh$. We proceed with vertex $v_2$ which is a prime vertex in $\mathcal{T}_G$. We now consider the cycle $C$ with root $\rho_C = v_2$ (Line 7). This cycle $C$ refers to the subgraph in $N$ induced by $v_2, u, u_d, u_e, u_f$. In Line 8, we put $\zeta(u_x) = \zeta(x) = x$ for each $x \in \{d,e,f\}$ and $\zeta(u) = \zeta(v_3) = gh$. In Line 9 we can choose an arbitrary ordering $\zeta^*(v_2)$ and decide, in this example, for $\zeta^*(v_2) = \zeta(u_d)\zeta(u_e)\zeta(u) = degh$. Since $u_f = \eta_C$ and $t(v_2) = 1$, we put $\zeta(v_2) = \zeta^*(v_2)\zeta(\eta_C) = deghf$ (Line 13). Finally, the prime vertex $v_1$ is processed. We consider now the cycle $C$ with root $\rho_C = v_1$ that is induced by $v_1, w, w_a, w_b, w_c$. (Line 7). In Line 8, we put $\zeta(w_x) = \zeta(x) = x$ for each*

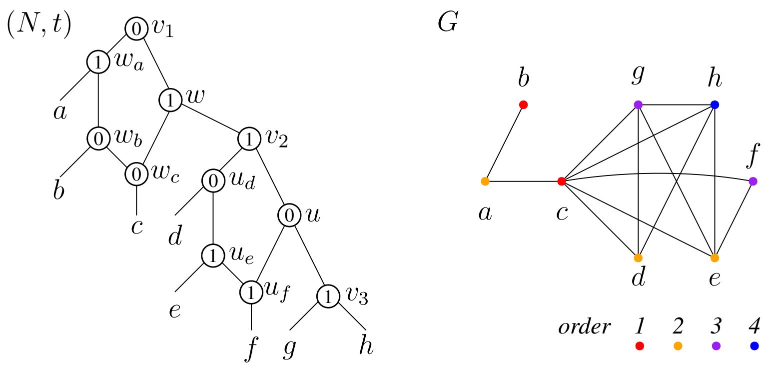


**Figure 2:** Left a galled-tree $(N,t)$ that explains the GATEX graph $G$ on the right. In addition, $G$ is equipped with a vertex coloring that is obtained with a greedy coloring based on the perfect order *cabdeghf* computed with Algorithm 1. Since $G[c,e,g,h]$ is a complete graph on four vertices, this coloring is optimal. see explanations in Example 3.2 for further details.

*$x \in \{a,b,c\}$ and $\zeta(w) = \zeta(v_2) = deghf$. In Line 9 we can choose an arbitrary ordering $\zeta^*(v_1)$ in Line 9 and decide, in this example, for $\zeta^*(v_1) = \zeta(w_a)\zeta(w_b)\zeta(w) = abdeghf$. Finally, since $t(v_1) = 0$ and $\eta_C = w_c$, we put $\zeta(v_1) = \zeta(\eta_C)\zeta^*(v_1) = cabdeghf$ (Line 11). Since $v_1$ is the root of $\mathfrak{T}_G$, the algorithm stops there, and returns the ordering $\zeta = \zeta(v_1) = cabdeghf$. As we shall show in Prop. 3.3, this ordering is a perfect ordering It is now an easy task to verify that the vertex coloring of G as shown in Fig. 2 can be obained by a greedy coloring taking the perfect order $\zeta = cabdeghf$ and the order of colors as shown in Fig. 2 (bottom right) into account.*

**Proposition 3.3.** *Algorithm 1 determines a perfect ordering of* GATEX *graphs.*

*Proof.* Let $G = (V,E)$ be a GATEX graph that serves as input for Alg. 1. We first compute $(\mathfrak{T}_G, t_G)$ and a pvr-network $(N,t)$ of $G$ (Line 1). In this proof, we put $L_w := L(N(w))$ for $w \in V(N)$. Let $\zeta(w)$ be the ordering computed with Alg. 1 for the subgraph $G[L_w]$ induced by the vertices in $L_w$. We then initialize $\zeta(v) = v$ for all leaves $v$ in $\mathfrak{T}_G$ (Line 2). Clearly, $\zeta(v)$ is a perfect ordering of $G[\{v\}]$. We then continue to traverse the remaining vertices in $\mathfrak{T}_G$ in postorder. This ensures that, whenever we reach a vertex $v$ in $\mathfrak{T}_G$, all its children have been processed and thus, that $\zeta(v)$ is well-defined in each step.

To verify that the ordering $\zeta$ returned by Alg. 1 is a perfect order of $G$, we must show that $\zeta$ does not contain any obstructions w.r.t. $G$ (cf. Prop. 2.2). If $G$ does not contain any induced $P_4$, then any ordering is perfect. Thus, assume that $G$ contains an induced $P_4$, say $P = a-b-c-d$. Put $Y = \{a,b,c,d\}$.

We first remark that Alg. 1 builds $\zeta$ by successively concatenating sub-orderings of the form $\zeta(w)$, $w \in V(\mathfrak{T}(G))$. In particular $\zeta_{|Y} = \zeta(w)_{|Y}$ holds for all $w \in V(\mathfrak{T}_G)$ for which $Y \subseteq M_w$ where $M_w := L(\mathfrak{T}_G(w))$. Let $M$ be the inclusion-minimal strong module of $G$ that contains $Y$. By Lemma 3.1, $M$ is a prime module of $G$. Hence, there is the unique cycle $C := C_M$ in $N$ corresponding to $M$. For all vertices $u \in V(C) \setminus \{\rho_C\}$, we denote with $u'$ is the unique child of $u$ that is not in $V(C)$. By Lemma 3.1, there are four vertices $u_a, u_b, u_c, u_c \in V(C)$ that satisfy the Condition (1) - (4). Hence, for $x \in \{a,b,c,d\}$ it holds that $x \in L_{u'_x}$. Moreover, the vertices $u_a, u_b, u_c, u_c$ are pairwise distinct, do not all belong to the same side of $C_M$ and one of $u_a, u_b, u_c, u_c$ coincides with the unique hybrid $\eta := \eta_C$ of $C$. The latter arguments, in particular, allow us to denote by $P^-$ (resp., $P^+$) the side of $C$ such that the set $V(P^-) \setminus \{\eta\}$ (resp., $V(P^+) \setminus \{\eta\}$) contains one (resp., two) of $u_a, u_b, u_c$ and $u_d$. In the following, let $v$ be the prime vertex in $\mathfrak{T}_G$ with $L(\mathfrak{T}_G(v)) = M$. We now distinguish between two cases: (1) $t(\rho_C) = 0$ and (2) $t(\rho_C) = 1$.

Case (1): $t(\rho_C) = 0$. Let $x \in Y$ be the vertex such that $u_x \in V(P^-) \setminus \{\eta\}$. Then, for all $y \in Y \setminus \{x\}$ with $u_y \in V(P^+) \setminus \{\eta\}$, we have $\mathrm{lca}_N(x,y) = \rho_C$ and thus, $x$ and $y$ are not joined by an edge in $G[Y]$. In particular, $x$ has degree at most one in $G[Y]$. Since $G[Y] = P$, it follows that $x$ has degree exactly one in $G[Y]$, and that the unique vertex $z \in Y$ adjacent to $x$ in $N$ satisfies $u_z = \eta$. Due to the "symmetry" of $G[Y] = P = a-b-c-d$, we can assume w.l.o.g. that $x = a$ and thus, $z = b$. By construction of $\zeta(v)$ in Line 11, we have $\zeta(v) = \zeta(\eta)\zeta^*(v)$. Since vertex $b$ appears in the order $\zeta(\eta)$ and vertex $a$ appears in the order $\zeta^*(v)$, we have in the final order $\zeta$ of $G$ always $b < a$. In this case, $P$ does not yield an obstruction of $\zeta$.

Case (2): $t(\rho_C) = 1$. Let $x \in Y$ be the vertex such that $u_x \in V(P^-) \setminus \{\eta\}$. Then for all $y \in Y \setminus \{x\}$ such that $u_y \in V(P^+) \setminus \{\eta\}$, we have $\mathrm{lca}_N(x,y) = \rho_C$ and thus, $x$ and $y$ are joined by an edge in $G[Y]$. In particular, $x$ has degree at least two in $G[Y]$. Since $G[Y] = P$, it follows that $x$ has degree exactly two in $G[Y]$, and that the unique vertex $z \in Y$ that is *not* adjacent to $x$ in $N$ satisfies $u_z = \eta$. Again, by "symmetry" of $G[Y] = P = a-b-c-d$, we can assume w.l.o.g. that $x = c$ and thus, $z = a$. Now consider the unique vertex $b$ that is adjacent to $a$ in $G[Y]$. By assumption, $b \in V(P^+) \setminus \{\eta\}$. Furthermore, by construction of $\zeta(v)$ in Line 13, we have $\zeta(v) = \zeta^*(v)\zeta(\eta)$. Since vertex $a$ appears in the order $\zeta(\eta)$ and vertex $b$ appears in the order $\zeta^*(v)$, we have in the final order $\zeta$ of $G$ always $b < a$. In this case, $P$ does not yield an obstruction of $\zeta$.

In summary, the ordering $\zeta$ returned by Alg. 1 does not contain any obstructions w.r.t. $G$. By Prop. 2.2, $\zeta$ is a perfect order of $G$. $\square$

**Proposition 3.4.** *Algorithm 1 can be implemented to run in $O(|V|+|E|)$ time where $G=(V,E)$ is the input* GATEX *graph.*

*Proof.* We show now that Algorithm 1 can be implemented to run in $O(|V|+|E|)$ time for a given GATEX graph $G=(V,E)$. The modular decomposition tree $(\mathcal{T}_G,t_G)$ can be computed in $O(|V|+|E|)$ time [12]. By [15, Thm. 9.4 and Alg. 4], the pvr-network $(N,t)$ of $G$ can be computed within the same time complexity. Thus, Line 1 takes $O(|V|+|E|)$ time. Initializing $\zeta(v) := v$ for all leaves $v$ (and thus, the vertices of $G$) in Line 2 can be done in $O(|V|)$ time.

We then traverse each of the $O(|V|)$ vertices in $(\mathcal{T}_G,t_G)$ in postorder. To compute the final perfect order, we consider an auxiliary directed graph $H$ that, initially, just consists of the vertices in $V$ and is edge-less. Whenever, we concatenate $\zeta'$ and $\zeta''$, we simply add an edge $(u,v)$ from the maximal element $u$ in $\zeta'$ to the minimal element $v$ in $\zeta''$ and define the minimal element of this now order $\zeta'''=\zeta'\zeta''$ as the minimal element of $\zeta'$ and the maximal element of $\zeta'''$ as the maximal element of $\zeta''$. Since we can keep track of these maximal and minimal elements (starting with $\zeta(v) := v$ for all leaves $v$ and defining $v$ as the maximal and minimal element of $\zeta(v)$) in each of the steps, the concatenation of two orders $\zeta'$ and $\zeta''$ and updating the maximal and minimal of $\zeta'''=\zeta'\zeta''$ can be done in constant time. The final graph $H$ then consists of a single directed path that traverses each vertex in $V$. If $t_G(v)\in\{0,1\}$, then we pick an arbitrary ordering of the children of $v$ and define $\zeta(v)=\zeta(v_1)\dots\zeta(v_k)$ by concatenating the orderings of its $k$ children $v_1,\dots,v_k$ (Line 4 - 5). By the latter arguments, this task can be done in $O(|\mathrm{child}_{\mathcal{T}_G}(v)|)$ time for each non-prime vertex $v$. Otherwise, if $t_G(v)=\mathrm{prime}$, we consider the unique cycle $C$ in $N$ that satisfies $L(N(\rho_C))=L(\mathcal{T}_G(v))$ in Line 7. We note that we can keep track of $C$ and its correspondence to $v$ when constructing the pvr-network $(N,t)$ based on $(\mathcal{T}_G,t_G)$ and thus have constant-time access to these cycles $C$ in $N$. The assignment $\zeta(w)=\zeta(w')$ for all $w\in V(C)\setminus\{\rho_C\}$ can be done in $O(|V(C)|)$ time (Line 8). By the latter arguments, construction of $\zeta^*(v)$ in Line 9 can be done in $O(|V(C)|)$ time. Note that $O(|V(C)|)=O(|\mathrm{child}_{\mathcal{T}_G}(v)|)$, since the elementary galled-tree $N_v$ that is used to replace $v$ and the edges to its children in $\mathcal{T}_G$, contains $C$ and has $2\,\mathrm{child}_{\mathcal{T}_G}(v)+1$ edges and vertices. The tasks in Line 10-13 can be done in constant time. Hence, the time-complexity of the Lines 6 to 13 is in $O(|\mathrm{child}_{\mathcal{T}_G}(v)|)$ for each prime vertex $v$.

To obtain the overall time complexity of the *for* loop starting in Line 3, observe that the degrees of vertices in $\mathcal{T}_G$ sum up to $2|E(\mathcal{T}_G)|=2(|V(\mathcal{T}_G)|-1)$. By the latter arguments and by iterating over each vertex $v\in V(\mathcal{T}_G)\setminus L(\mathcal{T}_G)$, we obtain $\sum_{v\in V(\mathcal{T}_G)\setminus L(\mathcal{T}_G)} O(|\mathrm{child}_{\mathcal{T}_G}(v)|)=O(|V(\mathcal{T}_G)|)=O(|V|)$.

Hence, the overall time-complexity of Algorithm 1 is dominated by the time-complexity to compute $(\mathcal{T}_G,t_G)$ and $(N,t)$ in Line 1 and is, therefore, in $O(|V|+|E|)$. $\square$

As an immediate consequence of Prop. 3.3 and 3.4, we obtain

**Theorem 3.5.** *Every* GATEX *graph is perfectly orderable and this ordering can be determined in linear-time.*

For a given graph $G=(V,E)$, a greedy coloring algorithm can be implemented to run in $O(|V|+|E|)$ time, see e.g. [27, Sec. 6.4]. This together with Theorem 3.5 implies

**Theorem 3.6.** *The chromatic number $\chi(G)$ and an optimal coloring of a* GATEX *graph $G$ can be determined in linear-time.*

# 4 Maximum cliques and independent sets

To recall, a clique of a graph $G$ is an inclusion-maximal complete subgraph $G$ and the maximum size of a clique of $G$ is denoted by $\omega(G)$. If $G$ is a GATEX graph, then it is explained by some labeled galled-tree $(N,t)$ whose leaf set is $V(G)$.

Since GATEX graphs $G$ are perfect, their chromatic number $\chi(G)$ and the size $\omega(G)$ of a maximum clique coincide. This together with Theorem 3.6 implies

**Theorem 4.1.** *The clique number $\omega(G)$ of a* GATEX *graph $G$ can be determined in linear-time.*

It is clear that for a given graph $G$ and integer $k=\omega(G)$, one can determine in $O(|V(G)|^k)$ time a maximum clique by examining all $O(|V(G)|^k)$ subgraphs. Since $k=\omega(G)$ can be obtained in linear time for GATEX graphs, we, therefore, immediately obtain a polynomial-time procedure to find maximum cliques in GATEX graphs. In what follows, we show that maximum cliques in GATEX graphs even can be determined in linear time.

To this end, we examine the structure and size of maximum-sized cliques induced by vertex set $L(N(v))$ in $G$ where $(N,t)$ is a galled-tree that explains $G$. In this context, it is important to take the labeling $t(v)$ of $v$ into account.

**Lemma 4.2.** *Let $G$ be a* GATEX *graph, $(N,t)$ be a labelled galled-tree explaining $G$ and $v$ be a vertex of $N$ that is not the root $\rho_C$ of any $C$ of $N$. Moreover, let $L_u=L(N(u))$ for $u\in V(N)$. Then it holds that*

1. *If $t(v)=0$, then $\omega(G[L_v])=\max_{w\in\mathrm{child}_N(v)}\{\omega(G[L_w])\}$ and any maximum clique of $G[L_v]$ is entirely contained in $G[L_w]$ for some $w\in\mathrm{child}_N(v)$.*

2. *If $t(v) = 1$, then $\omega(G[L_v]) = \sum_{w\in \mathrm{child}_N(v)} \omega(G[L_w])$ and any maximum clique of $G[L_v]$ is the join union $\Join_{w\in \mathrm{child}_N(v)} K^w$ of maximum cliques $K^w$ in $G[L_w]$.*

*Proof.* Since no cycle $C$ of $N$ satisfies $\rho_C = v$, one easily verifies that, for all distinct $z', z \in \mathrm{child}_N(v)$, it holds that $L_z \cap L_{z'} = \emptyset$ and that $\mathrm{lca}_N(x,x') = v$ for all $x \in L_z, x' \in L_{z'}$ (see also [15, Lemma 2.1]). Since $(N,t)$ explains $G$, it follows that $\{x,x'\}$ is an edge of $G$ if and only if $t(v) = 1$. In particular, the following holds:

*Case $t_G(v) = 0$:* In this case, $G[L_v]$ is the disjoint union of the graphs $G[L_w]$ with $w \in \mathrm{child}_N(v)$. Hence, every maximum clique in $G[L_v]$ must be located entirely in one of the subgraphs $G[L_w]$ of $G[L_v]$. Consequently, $\omega([G[L_v]) = \max_{w\in \mathrm{child}_N(v)}\{\omega(G[L_w])\}$ holds.

*Case $t_G(v) = 1$:* Suppose that $K$ is a maximum clique in $G[L_v]$. Since $t_G(v) = 1$, $G[L_v]$ is the join union of the graphs $G[L_w]$ with $w \in \mathrm{child}_N(v)$. In particular, $K$ can be written as the join union of cliques $K^w$ in $G[L_w]$, $w \in \mathrm{child}_N(v)$. Note that each of the cliques $K^w$ must be a maximum clique in $G[L_w]$ as otherwise we can replace $K^w$ by a larger clique in $G[L_w]$ and obtain a clique $K'$ in $G[L_v]$ that is larger than $K$. Consequently, $\omega(G[L_v]) = \sum_{w\in \mathrm{child}_N(v)} \omega(G[L_w])$ holds. □

We next investigate the case of vertices $v$ of $\mathcal{T}_G$ with $t_G(v) = prime$. To recall, we denote with $P^1(C), P^2(C)$ the sides of cycles $C \subseteq N$, i.e., the two directed paths $C$ with the same start-vertex $\rho_C$ and end-vertex $\eta_C$, and whose vertices distinct from $\rho_C$ and $\eta_C$ are pairwise distinct. Moreover $G_1(M), G_2(M) \subseteq G[M]$ will denote the subgraphs induced by leaf-descendants of the vertices in $P^1(C_M) - \rho_{C_M}$ and $P^2(C_M) - \rho_{C_M}$, respectively.

In the upcoming proofs we may need to compute the join $H' \Join H$ where $H$ is the empty graph. To avoid cumbersome case studies, we simple assume, in this case, that $H' := H' \Join H = H \Join H'$. In other words, if $H$ is empty and we argue along $H' \Join H$, then all arguments are applied to $H'$.

**Lemma 4.3.** *Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$. Put $L_\eta := L(N(\eta_{C_M}))$ and let $H \in \{G[M], G_1(M), G_2(M)\}$. If $H$ contains a maximum clique $K$ with vertices in $L_\eta$, then $V(K) \cap L_\eta$ induces a maximum clique in $G[L_\eta]$ and $(V(K) \setminus L_\eta) \,\dot\cup\, V(K')$ induces a maximum clique in $H$ for every maximum clique $K'$ in $G[L_\eta]$.*

*Proof.* Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$. Put $L_\eta := L(N(\eta_{C_M}))$, $\eta := \eta_{C_M}$ and $p = \rho_{C_M}$. In the following, let $H \in \{G[M], G_1(M), G_2(M)\}$.

Suppose that $H$ contains a maximum clique $K$ that contains vertices in $L_\eta$. Since $K$ is a clique in $H$, it must hold that $t(\mathrm{lca}_N(x,z)) = 1$ for all $x \in V(K) \cap L_\eta$ and $z \in V(K) \setminus L_\eta \subseteq V(H) \setminus L_\eta$. By definition of pvr-networks, $L_\eta$ is a module of $G$ and, therefore, $t(\mathrm{lca}_N(x,z)) = 1$ with $x \in V(K) \cap L_\eta$ and $z \in V(K) \setminus L_\eta$ implies that $t(\mathrm{lca}_N(x',z)) = 1$ for all $x' \in L_\eta$. By construction, we have $V(K) = (V(K) \setminus L_\eta) \,\dot\cup\, (V(K) \cap L_\eta)$. Assume, for contradiction, that $V(K) \cap L_\eta$ does not induce a maximum clique in $G[L_\eta]$. In this case, there is a clique $K'$ in $G[L_\eta]$ such that $|V(K')| > |V(K) \cap L_\eta|$. By the previous arguments, $t(\mathrm{lca}_N(x',z)) = 1$ for all $x' \in V(K')$ and $z \in V(K) \setminus L_\eta$. This together with $G[L_\eta] \subseteq H$ implies that $(V(K) \setminus L_\eta) \,\dot\cup\, V(K')$ induces a complete graph in $H$. However, $|(V(K) \setminus L_\eta) \,\dot\cup\, V(K')| > |(V(K) \setminus L_\eta) \,\dot\cup\, (V(K) \cap L_\eta)| = |V(K)|$; a contradiction to $K$ being a maximum clique in $H$. Therefore, $V(K) \cap L_\eta$ induces a maximum clique in $G[L_\eta]$.

Finally, let $K'$ be some maximum clique in $G[L_\eta]$ and thus, $|V(K) \cap L_\eta| = |V(K')|$. As argued before, $t(\mathrm{lca}_N(x',z)) = 1$ for all $x' \in V(K')$ and $z \in V(K) \setminus L_\eta$ which implies that $(V(K) \setminus L_\eta) \,\dot\cup\, V(K')$ induces a complete graph $K''$ in $H$ of size $|V(K'')| = |V(K) \setminus L_\eta| + |V(K')| = |V(K) \setminus L_\eta| + |V(K) \cap L_\eta| = |V(K)|$. Hence, $K''$ is a maximum clique in $H$. □

**Lemma 4.4.** *Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$ such that $t(\rho_{C_M}) = 1$. If $G[M]$ contains a maximum clique that contains vertices in $L(N(\eta_{C_M}))$, then $G_1(M)$ and $G_2(M)$ have both a maximum clique that contains vertices in $L(N(\eta_{C_M}))$.*

*Proof.* Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$ such that $t(\rho_{C_M}) = 1$. Furthermore, put $G_1 := G_1(M)$, $G_2 := G_2(M)$, $L_\eta := L(N(\eta_{C_M}))$ and $G_\eta := G[L_\eta]$. For a subgraph $H \subseteq G$ we define $|H| := |V(H)|$. Let $K$ be a maximum clique in $G[M]$ that contains vertices in $L_\eta$ and put $K^1 := (G_1 - G_\eta) \cap K$, $K^2 := (G_2 - G_\eta) \cap K$ and $K^\eta := K \cap G_\eta$. Thus, $V(K) = V(K^1) \,\dot\cup\, V(K^\eta) \,\dot\cup\, V(K^2)$.

Assume, for contradiction, that every maximum clique in $G_1$ does not contain vertices in $L_\eta$. Let $K'$ be a maximum clique in $G_1$. Since $V(K^1) \,\dot\cup\, V(K^\eta) \subseteq V(G_1)$ and $V(K^1) \,\dot\cup\, V(K^\eta)$ induce a complete graph with vertices in $L_\eta$, we can conclude that $|V(K^1) \,\dot\cup\, V(K^\eta)| = |K^1| + |K^\eta| < |K'|$. Note that $\mathrm{lca}_N(x,y) = \rho_C$ has label 1 for all $x \in V(K')$ and $y \in V(K^2)$ and thus, $K'' := K' \Join K^2$ forms a complete graph in $G[M]$ and thus, $|K'| + |K^2| = |K''| \le |K|$. This together with $|K^1| + |K^\eta| < |K'|$ yields the following contradiction:

$$|K'| + |K^2| = |K''| \le |K| = |K^1| + |K^\eta| + |K^2| < |K'| + |K^2|.$$

Hence, $G_1$ must contain a maximum clique with vertices in $L_\eta$. By similar arguments, $G_2$ must contain a maximum clique with vertices in $L_\eta$. □

**Lemma 4.5.** *Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$ such that $t(\rho_{C_M}) = 1$. Put $L_\eta := L(N(\eta_C))$ and $G_\eta = G[L_\eta]$. Furthermore, suppose that $G_1(M)$, resp., $G_2(M)$ have a maximum clique $K'$, resp., $K''$ with vertices in $L_\eta$ and such that $V(K') \cap L_\eta = V(K'') \cap L_\eta$. If $V(K') \cup V(K'')$ does not induce a maximum clique in $G[M]$, then none of the maximum cliques in $G[M]$ can have vertices in $L_\eta$.*

*Proof.* Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$ such that $t(\rho_{C_M}) = 1$. Furthermore, put $G_1 := G_1(M)$, $G_2 := G_2(M)$, $L_\eta := L(N(\eta_{C_M}))$ and $G_\eta := G[L_\eta]$. For a subgraph $H \subseteq G$ we define $|H| := |V(H)|$. Suppose that $G_1$, resp., $G_2$ contains a maximum clique $K'$, resp., $K''$ that contains vertices in $L_\eta$. Assume first that $V(K') \cap L_\eta \neq V(K'') \cap L_\eta$. By Lemma 4.3, $V(K') \cap L_\eta$ and $V(K'') \cap L_\eta$ induce a maximum clique in $G[L_\eta]$ and $(V(K') \setminus L_\eta) \,\dot\cup\, (V(K'') \cap L_\eta)$ induces a maximum clique $K'''$ in $G_1$ with vertices in $L_\eta$. In particular, $V(K''') \cap L_\eta = V(K'') \cap L_\eta$ is satisfied.

Hence, we can assume in the following w.l.o.g. that $V(K') \cap L_\eta = V(K'') \cap L_\eta$. By Lemma 4.3, $V(K') \cap L_\eta = V(K'') \cap L_\eta$ induces a maximum clique $K^\eta$ in $G[L_\eta]$.

We show first that $V(K') \cup V(K'')$ induces a complete graph in $G[M]$. Let $K^1$, resp., $K^2$ be the complete subgraph of $K'$, resp., $K''$ that is induced by $V(K') \setminus L_\eta$, resp., $V(K'') \setminus L_\eta$. Since $t(p) = 1$, all vertices in $V(K^1)$ are adjacent to all vertices in $V(K^2)$ and thus, the subgraph induced by $V(K^1) \,\dot\cup\, V(K^2)$ coincides with $K^1 \Join K^2$. Since $K^\eta$ is a complete graph, we have $K' = K^1 \Join K^\eta$ and $K'' = K^2 \Join K^\eta$, The latter two arguments imply that $K''' := K^1 \Join K^\eta \Join K^2$ is a complete graph in $G[M]$ that is induced by $V(K') \cup V(K'')$.

Suppose now that $K'''$ is not a maximum clique in $G[M]$. Let $\hat{K}$ be a maximum clique in $G[M]$ and thus, $|\hat{K}| > |K'''|$. Assume, for contradiction, that $\hat{K}$ contains vertices in $L_\eta$. Thus, we can apply Lemma 4.3 and assume w.l.o.g. that $V(\hat{K}) \cap L_\eta = V(K^\eta)$ induces the maximum clique $K^\eta$ in $G[L_\eta]$. Let $\hat{K}^i$ be the complete subgraph of $\hat{K}$ induced by the vertices $(V(G_i) \cap V(\hat{K})) \setminus L_\eta$, $i \in \{1,2\}$. By construction, $V(K''') = V(K^1) \,\dot\cup\, V(K^\eta) \,\dot\cup\, V(K^2)$ and, $V(\hat{K}) = V(\hat{K}^1) \,\dot\cup\, V(K^\eta) \,\dot\cup\, V(\hat{K}^2)$. If $|\hat{K}^1| \le |K^1|$ and $|\hat{K}^2| \le |K^2|$, then $|\hat{K}| = |\hat{K}^1| + |K^\eta| + |\hat{K}^2| \le |K^1| + |K^\eta| + |K^2| = |K'''|$, which is impossible as, by assumption, $|\hat{K}| > |K'''|$. Thus, $|\hat{K}^1| > |K^1|$ or $|\hat{K}^2| > |K^2|$ must hold. W.l.o.g. we may assume that $|\hat{K}^1| > |K^1|$. But then, $|V(\hat{K}^1) \,\dot\cup\, V(K^\eta)| > |V(K^1) \,\dot\cup\, V(K^\eta)|$ which together with the fact that $V(\hat{K}^1) \,\dot\cup\, V(K^\eta) \subseteq V(\hat{K})$ induce a complete graph in $G_1$ implies that $V(K^1) \,\dot\cup\, V(K^\eta) = V(K')$ cannot induce a maximum clique in $G_1$; a contradiction. Thus, $\hat{K}$ cannot contain vertices in $L_\eta$. □

**Proposition 4.6.** *Let $G$ be a* GATEX *graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$ where $t(\rho_{C_M}) = 1$. Put $L_\eta = L(N(\eta_{C_M}))$, $G_1 = G_1(M)$, $G_2 = G_2(M)$ and $G_\eta = G[L_\eta]$. Then,*

$$\omega(G[M]) = \max\{\omega(G_1) + \omega(G_2) - \omega(G_\eta), \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)\}.$$

*In particular, the following statements hold for $\alpha := \omega(G_1) + \omega(G_2) - \omega(G_\eta)$ and $\beta := \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)$:*

1. *If $\alpha \le \beta$, then $K = K_1 \Join K_2$ is a maximum clique in $G[M]$ for every maximum clique $K^1$ in $G_1 - G_\eta$ and $K^2$ in $G_2 - G_\eta$.*
2. *If $\alpha > \beta$, then every maximum clique in $G[M]$ contains vertices in $L_\eta$ and $V(K') \cup V(K'')$ induces a maximum clique in $G[M]$ for every maximum clique $K'$ in $G_1$ and $K''$ in $G_2$ that satisfies $V(K') \cap L_\eta = V(K'') \cap L_\eta \neq \emptyset$.*

*Proof.* Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$ where $t(\rho_{C_M}) = 1$. Put $L_\eta = L(N(\eta_{C_M}))$, $G_1 = G_1(M)$, $G_2 = G_2(M)$ and $G_\eta = G[L_\eta]$. Let $K$ be a maximum clique in $G[M]$. For a subgraph $H \subseteq G$ we define $|H| := |V(H)|$.

We start with showing that $\omega(G[M]) = \max\{\omega(G_1) + \omega(G_2) - \omega(G_\eta), \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)\}$. Since $\mathrm{lca}(x,y) = \rho_{C_M}$ and $t(\rho_{C_M}) = 1$ for all $x \in V(G_1 - G_\eta)$ and $y \in V(G_2 - G_\eta)$, every complete subgraph $K^1$ in $G_1 - G_\eta$ and $K^2$ in $G_2 - G_\eta$ yields a complete subgraph $K^1 \Join K^2$ in $G[M]$. Consider two maximum cliques $K^1$ in $G_1 - G_\eta$ and $K^2$ in $G_2 - G_\eta$. Hence, $\omega(G_1 - G_\eta) = |K^1|$ and $\omega(G_2 - G_\eta) = |K^2|$. Moreover, $\tilde{K} := K^1 \Join K^2$ forms a complete graph in $G[M \setminus L_\eta] \subseteq G[M]$. Therefore, $|K| \ge |\tilde{K}| = |K_1| + |K_2|$ and, thus, $\omega(G[M]) \ge \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)$. Hence, if $\omega(G[M]) = \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)$ we are done. Assume that $\omega(G[M]) > \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)$. Let $\tilde{K}$ be a maximum clique in $G[M \setminus L_\eta]$ and assume, for contradiction, that $\tilde{K}$ is a maximum clique in $G[M]$. By similar arguments as before, we can write $\tilde{K} = \tilde{K}^1 \Join \tilde{K}^2$ where $\tilde{K}^1 = \tilde{K} \cap (G_1 - G_\eta)$ and $\tilde{K}^2 = \tilde{K} \cap (G_2 - G_\eta)$. In particular, $\tilde{K}^1$ must be a maximum clique in $G_1 - G_\eta$ since, otherwise, there is a larger clique $\tilde{K}'$ in $G_1 - G_\eta$ and, thus, $\tilde{K}' \Join K_2$ would be larger than $\tilde{K} = \tilde{K}^1 \Join \tilde{K}^2$; a contradiction. Similarly, $\tilde{K}^2$ must be a maximum clique in $G_2 - G_\eta$. Hence, $\omega(G[M]) = |\tilde{K}| = |\tilde{K}^1| + |\tilde{K}^2| = \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta) < \omega(G[M])$; a contradiction. Hence, none of the complete graphs in $G[M \setminus L_\eta]$ are maximum cliques in $G[M]$. This and the fact that $M = V(G_1 - G_\eta) \,\dot\cup\, V(G_2 - G_\eta) \,\dot\cup\, V(G_\eta)$ implies that maximum cliques in $G[M]$ and, therefore, $K$ must contain vertices in $L_\eta$. By Lemma 4.4, $G_1$ and $G_2$ have both a maximum clique with vertices in $L_\eta$. Let $K'$, resp., $K''$ be a maximum clique in $G_1$, resp., $G_2$. Let $K^1$, resp., $K^2$ be the complete subgraph of $K'$, resp., $K''$ that is induced by $V(K') \setminus L_\eta$, resp., $V(K'') \setminus L_\eta$. By Lemma 4.3, we can assume w.l.o.g. that $V(K') \cap L_\eta = V(K'') \cap L_\eta$ induce a maximum clique $K^\eta$ in $G[L_\eta]$. Hence, $K''' = K^1 \Join K^\eta \Join K^2$ is a complete subgraph of $G[M]$. In particular, $K'''$ is induced by $V(K') \cup V(K'')$. Thus, if $K'''$ is not a maximum clique in $G[M]$, then Lemma 4.5 implies that none of the maximum cliques in $G[M]$ can have vertices in $L_\eta$; a contradiction. Hence, $K'''$ is a maximum clique in $G[M]$. Since $V(K''') = V(K') \cup V(K'')$ and $V(K') \cap V(K'') = V(K^\eta)$, we obtain $\omega(G) = |K'''| = |K'| + |K''| - |K^\eta| = \omega(G_1) + \omega(G_2) - \omega(G_\eta)$. In summary, $\omega(G[M]) = \max\{\omega(G_1) + \omega(G_2) - \omega(G_\eta), \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)\}$.

We now verify the Conditions (1) and (2) in the second statement. Consider first Condition (1) and assume that $\alpha := \omega(G_1) + \omega(G_2) - \omega(G_\eta) \le \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta) =: \beta$. By the previous arguments, $\omega(G[M]) = \beta$. Let $K^1$ be a maximum clique in $G_1(M) - G_\eta$ and $K^2$ be a maximum clique in $G_2(M) - G_\eta$ and thus, $\omega(G_1 - G_\eta) = |K^1|$ and $\omega(G_2 - G_\eta) = |K^2|$. By the arguments above, $K = K^1 \Join K^2$ is a complete subgraph in $G[M]$. In particular, $|K| = |K^1| + |K^2| = \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta) = \beta = \omega(G[M])$. Consequently, $K$ is a maximum clique in $G[M]$. Thus, Condition (1) is satisfied.

**Algorithm 2** `Computation of a maximum clique and` $\omega(G)$ `of GATEx graphs` $G$

**Input:** A GATEX graph $G = (V,E)$
**Output:** maximum clique $K$ in $G$ and its size $\omega(G)$

Compute $(\mathcal{T}_G, t_G)$ and pvr-network $(N,t)$ of $G$
put $\omega(v) := 1$ for all leaves $v$ in $L(N) = V$
**for all** $v \in V(\mathcal{T}_G) \setminus L(\mathcal{T}_G)$ in postorder **do**
  **if** $t_G(v) = 0$ **then**
    Put $\omega(v) := \max_{w \in \mathrm{child}_{\mathcal{T}_G}(v)} \{\omega(w)\}$
    Mark $w$ as `active` for precisely one $w \in \arg\max_{z \in \mathrm{child}_{\mathcal{T}_G}(v)} \{\omega(z)\}$
  **else if** $t_G(v) = 1$ **then**
    Put $\omega(v) := \sum_{w \in \mathrm{child}_{\mathcal{T}_G}(v)} \omega(w)$
    Mark all $w \in \mathrm{child}_{\mathcal{T}_G}(v)$ as `active`
  **else** ▷ $t_G(v) = \text{prime}$
    Let $C$ be the unique cycle in $N$ with root $\rho_C = v$
    ▷ *Although $\rho_C = v$, we distinguish between them to make it clearer if we are working in $\mathcal{T}_G$ or $N$* ◁
    Let $\eta$ be the unique hybrid in $C$ and $u$ be the unique child of $\eta$ in $N$
    Put $\omega(\eta) := \omega(u)$ and $\omega_{\neg\eta}(\eta) := 0$
    ▷ *Init $\omega(w)$ and $\omega_{\neg\eta}(w)$ for the vertices $w \neq \rho_c, \eta$ along the sides of $C$ bottom-up* ◁
    Let $P^1$ and $P^2$ be the two sides of $C$
    **for all** $w \in V(P^i) \setminus \{\rho_C, \eta\}$ in postorder, $i \in \{1,2\}$ **do**
      Put $u' := \mathrm{child}_N(w) \cap V(C)$ and $u'' := \mathrm{child}_N(w) \setminus V(C)$ ▷ *Note, $\mathrm{child}_N(w) = \{u', u''\}$ for $w \neq \eta$*
      **if** $t(w) = 0$ **then** Put $\omega(w) := \max\{\omega(u'), \omega(u'')\}$ and $\omega_{\neg\eta}(w) := \max\{\omega_{\neg\eta}(u'), \omega(u'')\}$
      **else** Put $\omega(w) := \omega(u') + \omega(u'')$ and $\omega_{\neg\eta}(w) := \omega_{\neg\eta}(u') + \omega(u'')$ ▷ $t(w) = 1$
    ▷ *Init $\omega(v)$. Note, $\rho_C$ corresponds to $v$ in $\mathcal{T}_G$* ◁
    Let $u'$ and $u''$ be the two children of $\rho_C$
    **if** $t(\rho_C) = 0$ **then**
      Put $\omega(v) := \max\{\omega(u'), \omega(u'')\}$
      Choose one $w \in \arg\max\{\omega(u'), \omega(u'')\}$
      Mark $w$ as `active`
      Let $P \in \{P^1, P^2\}$ be such that $w \in V(P)$.
      ACTIVATE($(N,t)$, $\{P\}$, $\omega$, $\omega_{\neg\eta}$, 0, 0, 0)
    **else** ▷ $t(\rho_C) = 1$
      Put $\alpha := \omega(u') + \omega(u'') - \omega(\eta)$ and $\beta := \omega_{\neg\eta}(u') + \omega_{\neg\eta}(u'')$
      Put $\omega(v) := \max\{\alpha, \beta\}$
      ACTIVATE($(N,t)$, $\{P^1, P^2\}$, $\omega$, $\omega_{\neg\eta}$, $\alpha$, $\beta$, 1)
$\Omega :=$ set of all leaves $x \in L(N)$ for which there is a path $P$ from $\rho_N$ to $x$ where all vertices $v \neq \rho_N$ in $P$ are `active`
**return** $G[\Omega]$ and $\omega(\rho_N)$

For Condition (2), suppose that $\alpha > \beta$ and, therefore, $\omega(G[M]) = \alpha$. As argued above, $\omega(G[M]) > \omega(G_1 - G_\eta) + \omega(G_2 - G_\eta)$ implies that every maximum clique in $G[M]$ must contain vertices in $L_\eta$. This and Lemma 4.4 implies that $G_1$ and $G_2$ have both a maximum clique that contains vertices in $L_\eta$. Let $K'$ be an arbitrary maximum clique in $G_1$ and $K''$ be an arbitrary maximum clique in $G_2$ such that $V(K') \cap L_\eta = V(K'') \cap L_\eta \neq \emptyset$ holds. By Lemma 4.3, such cliques $K'$ and $K''$ exist. Since every maximum clique in $G[M]$ must contain vertices in $L_\eta$, contraposition of Lemma 4.5 implies that $V(K') \cup V(K'')$ induce a maximum clique in $G[M]$. Thus, Condition (2) is satisfied. □

**Proposition 4.7.** *Let $G$ be a* GATEX *graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$. If $t(\rho_{C_M}) = 0$, then $\omega(G[M]) = \max\{\omega(G_1(M)), \omega(G_2(M))\}$ of* GATEX *graphs*

*Proof.* Let $G$ be a GATEX graph that is explained by the pvr-network $(N,t)$ and suppose that $G$ contains a prime module $M$ such that that $t(\rho_{C_M}) = 0$. Furthermore, put $L_\eta := L(N(\eta_{C_M}))$ and $G_\eta := G[L_\eta]$. Let $K$ be a maximum clique in $G[M]$. Note first that $K$ cannot contain vertices $x$ and $y$ such that $x \in V(G_1(M) - G_\eta)$ and $y \in V(G_2(M) - G_\eta)$ since, in this case, $\mathrm{lca}(x,y) = \rho_{C_M}$ and $t(\rho_{C_M}) = 0$ imply that $\{x,y\} \notin E(G[M])$. Hence, $K$ must be entirely contained in either $G_1(M)$ or $G_2(M)$. Moreover, any maximum clique in $G_1(M)$ and $G_2(M)$ provide a complete subgraph of $G[M]$. Taken the latter two arguments together, $\omega(G[M]) = \max\{\omega(G_1(M)), \omega(G_2(M))\}$. □

**Remark 4.8.** *For the sake of simplicity, we often put $\omega(w) := \omega(G[L_w])$ for the size of a maximum clique in the subgraph of $G$ induced by $L_w = L(N(w)) \subseteq V(G)$. where $(N,t)$ is a galled-tree that explains $G$.*

As we shall see, Algorithm 2 can be used to compute maximum cliques in GATEX graphs in linear-time. Before studying Algorithm 2 in detail, we illustrate this algorithm on the examples as shown in Figure 3.

**Procedure** ACTIVATE$((N,t),\mathscr{P},\omega,\omega_{\neg\eta},\alpha,\beta,label_\rho_C)$

**if** $label\_\rho_C = 0$ **then**
    **for** $w' \in V(P)\setminus\{\rho_C\}$ in postorder where $P\in\mathscr{P}$ **do**
        Put $u' := \mathrm{child}_N(w')\cap V(C)$ and $u'' := \mathrm{child}_N(w')\setminus V(C)$
        **if** $t(w')=0$ **then** Mark precisely one $u\in\arg\max\{\omega(u'),\omega(u'')\}$ as `active`
        **else** Mark all $u\in\mathrm{child}_N(w')$ as `active`
**else** Mark both children of $\rho_C$ as `active`
    **if** $\alpha\leq\beta$ **then**
        Let $w_1$ and $w_2$ be the unique parents of $\eta$
        Mark $u\in\mathrm{child}(w_i)\setminus\{\eta\}$ as `active` for $i\in\{1,2\}$
        **for** all $P\in\mathscr{P}$ and $w'\in V(P)\setminus\{\rho_C,\eta,w_1,w_2\}$ in postorder **do**
            Put $u' := \mathrm{child}_N(w')\cap V(C)$ and $u'' := \mathrm{child}_N(w')\setminus V(C)$
            **if** $t(w')=0$ **then** Mark exactly one $u\in\arg\max\{\omega_{\neg\eta}(u'),\omega(u'')\}$ as `active`
            **else** Mark both children $u',u''$ of $w'$ as `active` ▷ $t(w)=1$
    **else** ▷ $\alpha>\beta$
        Mark $\eta_C$ and its child as `active`
        **for** all $P\in\mathscr{P}$ and $w\in V(P)\setminus\{\rho_C,\eta_C\}$ in postorder **do**
            Mark $w$ as `active`
            **if** $t(w)=1$ **then** mark also the child of $w$ not in $P$ as `active`

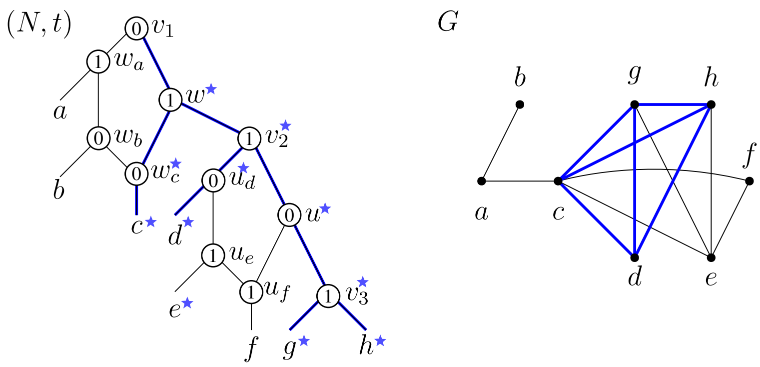


**Figure 3:** Left a galled-tree $(N,t)$ that explains the GATEX graph $G$ on the right. Algorithm 2 returns the induced subgraph $G[g,h,d,c]$ (highlighted in blue) which is a maximum clique of $G$. All vertices marked as `active` are highlighted with $\star$. Paths $P$ from $\rho_N$ to leaves $x$ in $N$ where all vertices $v\neq\rho_N$ in $P$ are `active` are highlighted in blue; see explanations in Example 4.9 for further details.

**Example 4.9.** *We exemplify here the main steps of Algorithm 2 using the* GATEX *graph $G$ as shown in Fig. 3. We first compute the modular decomposition tree $(\mathscr{T}_G,t_G)$ (as shown in Fig. 1) and the shown pvr-network $(N,t)$ that explains $G$ (Line 1). For all leaves $v$ of $\mathscr{T}_G$ (and thus, of $N$), we have $L_v=\{v\}$ and, thus, the size of a maximum clique in $G[v]$ is one and we put $\omega(v):=1$ (Line 2). We then traverse the vertices $\mathscr{T}_G$ that are not leaves in postorder and thus obtain the order $v_3,v_2,v_1$ in which the vertices are visited (Line 3). Note that postorder-traversal ensures that all children of a given vertex $v$ in $\mathscr{T}_G$ are visited before this vertex $v$ is processed. In what follows, we denote with $\omega_{\neg\eta}(v)$ the size of a maximum clique in $G[L_v\setminus L_{\eta_C}]$ given that $v$ is part of a cycle $C$ with hybrid $\eta_C$.*

*Consider now the processed vertex $v_3$. Since $t(v_3)=1$, we define $\omega(v_3)=\omega(g)+\omega(h)=2$ (Line 8). The latter is in accordance with Lemma 4.2 and refers to the fact that the maximum clique in $G[L_{v_3}]$ is precisely the edge connecting $g$ and $h$, i.e, the join union of two single vertex graphs. In addition, we mark both $g$ and $h$ as* `active` *(Line 9).*

*We have $\eta_C=u_f$, so we put $\omega(u_f)=\omega(f)=1$ and $\omega_{\neg\eta}(u_f)=0$ (Line 14). The latter refers to the fact that, in this example, any maximum clique in $G[L_\eta]$ is of size one, while any clique in $G[L_\eta\setminus L_\eta]$ is of size 0. The two sides of $C$ are $P^1=\{v_2,u,u_f\}$ and $P^2=\{v_2,u_d,u_e,u_f\}$. We first consider the unique vertex $u$ of $P^1\setminus\{\rho_C,\eta_C\}$. We have $t(u)=1$, so we put $\omega(u)=\omega(u_f)+\omega(v_3)=3$, and $\omega_{\neg\eta}(u)=\omega_{\neg\eta}(u_f)+\omega(v_3)=2$ (Line 20). Next, we consider the vertices of $P^2\setminus\{\rho_C,\eta_C\}=\{u_e,u_d\}$ in postorder, that is, $u_e$ first and $u_d$ second. We have $t(u_e)=1$, so we put $\omega(u_e)=\omega(u_f)+\omega(e)=2$, and $\omega_{\neg\eta}(u_e)=\omega_{\neg\eta}(u_f)+\omega(e)=1$ (Line 20). Since $t(u_d)=0$, we put $\omega(u_d)=\max\{\omega(u_e),\omega(d)\}=2$, and $\omega_{\neg\eta}(u_e)=\max\{\omega_{\neg\eta}(u_e),\omega(d)\}=1$ (Line 19). Afterwards, we consider the vertex $\rho_C=v_2$. We have $t(v_2)=1$, so we go to Line 29. We first define $\alpha=\omega(u_d)+\omega(u)-\omega(u_f)=3$, $\beta=\omega_{\neg\eta}(u_d)+\omega_{\neg\eta}(u)=3$ (Line 30), and $\omega(v_2)=\max\{\alpha,\beta\}=3$ (Line 31). The latter is in accordance with Prop. 4.6. In Line 32, we then call the procedure* ACTIVATE$\big((N,t),\{P^1,P^2\},\omega,\omega_{\neg\eta},3,3,1\big)$*. This precedure is used to "activate" the right vertices in such a way that, after termination of Algorithm 2, the set $\Omega$ consisting of all leaves*

*$x \in L(N)$ for which there is a path $P$ from $\rho_N$ to $x$ in $N$ with all vertices $v \neq \rho_N$ in $P$ marked as* `active` *determines the vertex set of a maximum clique in $G$.*

*We are now in the procedure* ACTIVATE. *Since label_$\rho_C = 1$, we mark both children $u$ and $u_d$ of $\rho_C = v_2$ as* `active` *(Line 9). Since $\alpha \leq \beta$, we continue with Line 7. In Line 9, we mark $v_3$ and $e$ as* `active`*. In the for-loop at Line 10, we consider the two sides $P^1, P^2 \in \mathscr{P}$ of $C$. In particular, we have $V(P^1) \setminus \{\rho_C, \eta_C, u, u_e\} = \emptyset$ and $V(P^2) \setminus \{\rho_C, \eta_C, u, u_e\} = \{u_d\}$. Thus, we only have to consider, in this run of the for-loop, the vertex $w' = u_d$. In this case, $u' = u_e$ and $u'' = d$. Since $t(u_d) = 0$, $\omega_{\neg\eta}(u_e) = 1$, and $\omega(d) = 1$, we choose one of $u_e$ or $d$ to be marked mark as* `active` *(Line 12) In this example, we decide to mark $d$* `active`*. After this, we exit the procedure* ACTIVATE.

*We are now back in Algorithm 2 proceed with the prime vertex $v_1$ and consider the cycle $C$ with root $\rho_C = v_1$ (Line 11). We have $\eta_C = w_c$, so we put $\omega(w_c) = \omega(c) = 1$ and $\omega_{\neg\eta}(w_c) = 0$ (Line 14). The two sides of $C$ are $P^1 = \{v_1, w, w_c\}$ and $P^2 = \{v_1, w_a, w_b, w_c\}$. We first consider the unique vertex $w$ of $P^1 \setminus \{\rho_C, \eta_C\}$. We have $t(w) = 1$, so we put $\omega(w) = \omega(w_c) + \omega(v_2) = 4$, and $\omega_{\neg\eta}(w) = \omega_{\neg\eta}(w_c) + \omega(v_2) = 3$ (Line 20). Next, we consider the vertices of $P^2 \setminus \{\rho_C, \eta_C\} = \{w_a, w_b\}$ in postorder, that is, $w_b$ first and $w_a$ second. We have $t(w_b) = 0$, so we put $\omega(w_b) = \max\{\omega(w_c), \omega(b)\} = 1$, and $\omega_{\neg\eta}(w_b) = \max\{\omega_{\neg\eta}(w_c), \omega(b)\} = 1$ (Line 19). Since $t(w_a) = 1$, we put $\omega(w_a) = \omega(w_b) + \omega(a) = 2$, and $\omega_{\neg\eta}(w_a) = \omega_{\neg\eta}(w_b) + \omega(a) = 2$ (Line 20). Finally, we consider the vertex $\rho_C = v_1$. We have $t(v_1) = 0$, so we go to Line 23. We first define $\omega(v_1) = \max\{\omega(w_a), \omega(w)\} = 4$ (Line 24). Since $\arg\max\{\omega(w), \omega(w_a)\} = w$ and $w \in P^1$, we mark $w$ as* `active` *(Line 26), and we call the procedure* ACTIVATE$\big((N,t), \{P^1\}, \omega, \omega_{\neg\eta}, 0,0,0\big)$ *(Line 28).*

*We are now in the procedure* ACTIVATE. *We have label$_{\rho_C} = 0$ and thus, go to the for-loop in Line 2. Here, we consider the elements of $V(P^1) \setminus \{\rho_C\} = \{w, w_c\}$ in postorder, that is $w_c$ first and $w$ second. We have $t(w_c) = 0$ and $w_c$ does not have a child in $C$, so we mark $c$ as* `active` *(Line 4). Since $t(w) = 1$, we mark both $v_2$ and $w_c$ as* `active` *(Line 5). After this, we exit the procedure.*

*We are now back in Algorithm 2. Since all vertices of $V(\mathfrak{T}_G) \setminus L(\mathfrak{T}_G)$ have now been processed, we are in Line 33 and ready to compute the set $\Omega$. The vertices marked as* `active` *are $g, h, u, u_d, v_3, e, d, w, c, v_2$ and $w_c$. Therefore, the set $\Omega$ computed at Line 33 is $\{g, h, d, c\}$. Note that although $e$ is also marked as* `active`*, its parent $u_e$ is not, so $e$ is not added to $\Omega$. The algorithm stops here and returns $G[\Omega] = G[\{g, h, d, c\}]$ and $\omega(\rho_N) = \omega(v_1) = 4$. One can verify that $G[\{g, h, d, c\}]$ is indeed a maximum clique of $G$. In particular, $\omega(v_1)$ corresponds to the size of a maximum clique in $G$.*

**Proposition 4.10.** *Algorithm 2 correctly computes the clique number $\omega(G)$ of* GATEX *graphs $G$. In particular, if $(N,t)$ is a pvr-network of $G$ used in Algorithm 2, then $\omega(v) = \omega(G[L(N(v))])$ for all $v \in V(N)$. In addition, if $v$ is contained in a cycle $C$ of $N$ and $v \neq \rho_C$, then $\omega_{\neg\eta}(v) = \omega(G[L(N(v)) \setminus L(N(\eta_C))])$.*

*Proof.* Let $G = (V,E)$ be the input GATEX graph for Algorithm 2. In order to show that $\omega(G)$ is correctly computed, we can ignore all Lines in Algorithm 2 where vertices are marked as active and where the procedure ACTIVATE is called. We start in Line 1 with computing $(\mathfrak{T}_G, t_G)$ and a pvr-network $(N,t)$ of $G$. In what follows, let $L_w := L(N(w))$ for $w \in V(N)$. Furthermore, for a vertex $w \in V(\mathfrak{T}_G)$, let $M_w := L(\mathfrak{T}_G(w))$ denote the module of $G$ "associated" with $w$. To recall, $V(\mathfrak{T}_G) \subseteq V(N)$.

In Line 2, we initialize $\omega(v) = 1$ for all leaves $v \in L(\mathfrak{T}_G) = L(N) = V$ and, thus, correctly capture the size $\omega(G[L_v]) = \omega(v)$ of a maximum clique in $G[L_v] \simeq K_1$. We then continue to traverse the remaining vertices in $\mathfrak{T}_G$ in postorder. This ensures that whenever we reach a vertex $v$ in $\mathfrak{T}_G$, all its children have been processed. We show now that $\omega(v) := \omega(G[M_v])$ is correctly computed for all $v \in V(\mathfrak{T}_G)$. Let $v$ be the currently processed vertex in Line 3. By induction, we can assume that the children $u$ of $v$ in $\mathfrak{T}_G$ satisfy $\omega(u) = \omega(G[M_u])$. We consider now the cases for $t(v) \in \{0, 1, \text{prime}\}$.

*Case $t_G(v) = 0$:* In this case, $\omega(v)$ is defined as $\max_{w \in \text{child}_{\mathfrak{T}_G}(v)}\{\omega(w)\}$ in Line 5. Lemma 4.2, together with the fact that the children of $v$ is $\mathfrak{T}_G$ are precisely the children of $v$ in $N$ (Observation 2.5), implies that $\omega(G[M_v]) = \max_{w \in \text{child}_{\mathfrak{T}_G}(v)}\{\omega(G[M_w])\}$. Therefore, $\omega(v) = \max_{w \in \text{child}_{\mathfrak{T}_G}(v)}\{\omega(w)\} = \max_{w \in \text{child}_{\mathfrak{T}_G}(v)}\{\omega(G[M_w])\} = \omega(G[M_v])$ follows.

*Case $t_G(v) = 1$:* In this case, $\omega(v)$ is defined as $\sum_{w \in \text{child}_{\mathfrak{T}_G}(v)} \omega(w)$ in Line 8. Lemma 4.2, together with the fact that the children of $v$ is $\mathfrak{T}_G$ are precisely the children of $v$ in $N$ (Observation 2.5), implies that we have $\omega(G[M_v]) = \sum_{w \in \text{child}_{\mathfrak{T}_G}(v)}\{\omega(G[M_w])\}$. Therefore, $\omega(v) = \sum_{w \in \text{child}_{\mathfrak{T}_G}(v)} \omega(w) = \sum_{w \in \text{child}_{\mathfrak{T}_G}(v)} \omega(G[M_w]) = \omega(G[M_v])$ follows.

*Case $t_G(v) = \text{prime}$:* In this case, $M := M_v$ is a prime module of $G$ and $v$ is locally replaced by a cycle $C := C_M$ with root $\rho_C = v$ according to Def. 2.4 and we have $M = L(\mathfrak{T}_G(v)) = L_{\rho_C}$ (cf. Obs. 2.6). Although $\rho_C = v$, we will distinguish between them to better keep track as whether we are working in $\mathfrak{T}_G$ or $N$. Let $P^1$ and $P^2$ be the two sides of $C$. By Obs. 2.6, all vertices $w \neq \rho_C$ in $C$ have exactly one child $u''$ that is not in $C$. By construction of $(N,t)$, each of those childs $u''$ is a child of $v$ in $\mathfrak{T}_G$. By induction assumption, we can assume that $\omega(u'')$ correctly captures the size of $\omega([G[M_{u''}]) = \omega([G[L_{u''}])$. Out task is now to determine the clique number $\omega(v) := \omega(G[M])$ of $G[M]$. In the following, we will record two values $\omega(w)$ and $\omega_{\neg\eta}(w)$ for the vertices $w \neq \rho_C$ in $C$ to capture the size $\omega(w)$ of a maximum clique in $G[L_w]$ and the size $\omega_{\neg\eta}(w)$ of a maximum clique in $G[L_w \setminus L_\eta]$.

We start in Line 13 with the the unique hybrid-vertex $\eta = \eta_C$ of $C$. By Obs. 2.6, $\eta$ has precisely one child $u$ and, therefore, $L_\eta = L_u$. Hence, $\omega(\eta) := \omega(u) = \omega(G[L_u])$ and, since $G[L_u] = G[L_\eta]$, $\omega(\eta) = \omega(G[L_\eta])$ is correctly determined in Line 14. Moreover, $\omega_{\neg\eta}(\eta) := 0$ is correctly determined as there is no clique in $G[L_\eta \setminus L_\eta]$.

In Line 17 - 20, we consider all vertices $w \in V(C) \setminus \{\rho_C, \eta\}$ in a bottom-up order. By Obs. 2.6, $w$ has precisely two children $u'$ and $u''$ where $u'$ is located on $C$ while $u''$ is not and it holds that $L_{u'} \cap L_{u''} = \emptyset$. By the post-ordering, we start with one of the parents of $\eta$ located in $C$.

Let $w$ be a parent of $\eta$ that is located in $P^i$ for some $i \in \{1,2\}$ for which $u' = \eta$. Since $w$ is a parent of $\eta$ in $C$ it holds that $L_w = L_\eta \cup L_{u''}$ and $L_w \setminus L_\eta = L_{u''}$ and thus, in particular, $\omega(G[L_w \setminus L_\eta]) = \omega(G[L_{u''}])$. Assume that $t(w) = 0$ (Line 19). In this case, we put $\omega(w) = \max\{\omega(\eta), \omega(u'')\}$ and $\omega_{\neg\eta}(w) = \max\{\omega_{\neg\eta}(eta), \omega(u'')\} = \max\{0, \omega(u'')\} = \omega(u'')$. By our induction hypothesis, $\omega(u'') = \omega(G[L_{u''}])$, so since $\omega(G[L_w \setminus L_\eta]) = \omega(G[L_{u''}])$, $\omega_{\neg\eta}(w) = \omega(G[L_w \setminus L_\eta])$ follows. Moreover, since $t(w) = 0$, Lemma 4.2 implies that $\omega(G[L_w]) = \max\{\omega(G[L_\eta]), \omega(G[L_{u''}])\}$. By our induction hypothesis, $\omega(\eta) = \omega(G[L_\eta])$ and $\omega(u'') = \omega(G[L_{u''}])$, so $\omega(w) = \omega(G[w])$ follows. Assume now that $t(w) = 1$ (Line 20). In this case, we have we put $\omega(w) = \omega(u') + \omega(u'')$ and $\omega_{\neg\eta}(w) = \omega_{\neg\eta}(u') + \omega(u'') = 0 + \omega(u'')$. As in the previous case, $\omega(u'') = \omega(G[L_{u''}])$, together with $\omega(G[L_w \setminus L_\eta]) = \omega(G[L_{u''}])$, implies that $\omega_{\neg\eta}(w) = \omega(G[L_w \setminus L_\eta])$. Moreover, since $t(w) = 1$, Lemma 4.2 implies that $\omega(G[L_w]) = \omega(G[L_\eta]) + \omega(G[L_{u''}])$. By our induction hypothesis, $\omega(\eta) = \omega(G[L_\eta])$ and $\omega(u'') = \omega(G[L_{u''}])$, so $\omega(w) = \omega(G[w])$ follows.

Suppose now that $w \in V(C) \setminus (\{\rho_C, \eta\} \cup \mathrm{par}(\eta))$ is the currently processed vertex. Note that both children $u'$ and $u''$ of $w$ have already been processed and we can assume by the latter arguments and by induction that $\omega(u') = \omega(G[L_{u'}])$, $\omega_{\neg\eta}(u') = \omega(G[L_{u'} \setminus L_\eta])$, and $\omega(u'') = \omega(G[L_{u''}])$.

Assume that $t(w) = 0$ (Line 19). Then, we put $\omega(w) = \max\{\omega(u'), \omega(u'')\}$ and by similar argument as used in the previous case, $\omega(w) = \omega(G[L_w])$ is correctly computed. Consider now $\omega_{\neg\eta}(w) = \max\{\omega_{\neg\eta}(u'), \omega(u'')\}$. By Obs. 2.6 it holds that $\mathrm{lca}_N(x,y) = w$ for all $x \in L_{u'} \setminus L_\eta \subseteq L_{u'}$ and $y \in L_{u''}$. This and $t(w) = 0$ implies that there are no edges between vertices in $G[L_{u'} \setminus L_\eta]$ and $G[L_{u''}]$. Hence, $G[L_w \setminus L_\eta] = G[L_{u'} \setminus L_\eta] \cup G[L_{u''}]$. This together with $(L_w \setminus L_\eta) \cap L_{u''} = \emptyset$ implies that $\omega(G[L_w \setminus L_\eta]) = \max\{\omega(G[L_{u'} \setminus L_\eta]), \omega(G[L_{u''}])\}$. By our induction hypothesis, $\omega_{\neg\eta}(u') = \omega(G[L_{u'} \setminus L_\eta])$ and $\omega(u'') = \omega(G[L_{u''}])$, so $\omega_{\neg\eta}(w) = \omega(G[L_w \setminus L_\eta])$ follows. Suppose now that $t(w) = 1$ (Line 20). Then, we put $\omega(w) = \omega(u') + \omega(u'')$ and by similar argument as used in the previous case ($w$ as a parent of $\eta$), $\omega(w) = \omega(G[L_w])$ is correctly computed. Consider now $\omega_{\neg\eta}(w) = \omega_{\neg\eta}(u') + \omega(u'')$. Since $\mathrm{lca}_N(x,y) = w$ for all $x \in L_{u'} \setminus L_\eta$ and $y \in L_{u''}$, and $t(w) = 1$, all vertices in $G[L_{u'} \setminus L_\eta]$ are adjacent to all vertices in $G[L_{u''}]$. Hence, $G[L_w \setminus L_\eta] = G[L_{u'} \setminus L_\eta] \bowtie G[L_{u''}]$ and therefore, $\omega(G[L_w \setminus L_\eta]) = \omega(G[L_{u'} \setminus L_\eta]) + \omega(G[L_{u''}])$. By our induction hypothesis, $\omega_{\neg\eta}(u') = \omega(G[L_{u'} \setminus L_\eta])$ and $\omega(u'') = \omega(G[L_{u''}])$, so $\omega_{\neg\eta}(w) = \omega(G[L_w \setminus L_\eta])$ follows.

In summary, in Line 17 - 20 the values $\omega(w) = \omega(G[L_w])$ and $\omega_{\neg\eta}(w) = \omega(G[L_w \setminus L_\eta])$ have been correctly computed for all $w \in V(C) \setminus \{\rho_C\}$.

In Line 22 - 32 we finally determine the value $\omega(v)$. To recall, $v = \rho_C$ is the vertex in $(\mathcal{T}_G, t_G)$ with label $t_G(v) =$ prime and $M = L(\mathcal{T}_G(v))$ is a prime module in $G$ for which $M = L(\mathcal{T}_G(v)) = L_{\rho_C}$ holds. Let $u'$ and $u''$ be the two children of $\rho_C$ (cf. Obs. 2.6). It is an easy task to verify that $G[L_{u'}] = G_i(M)$ and $G[L_{u''}] = G_j(M)$ with $\{i,j\} = \{1,2\}$. W.l.o.g. assume that $i = 1$ and $j = 2$. By induction, we can assume that $\omega(u') = \omega(G[L_{u'}])$ and $\omega(u'') = \omega(G[L_{u''}])$ and, therefore, $\omega(u') = \omega(G_1(M))$ and $\omega(u'') = \omega(G_2(M))$. Assume now that $t(\rho_C) = 0$. In this case, we put in Line 24, $\omega(v) := \max\{\omega(u'), \omega(u'')\}$. By the latter arguments, $\omega(v) = \max\{\omega(G_1(M)), \omega(G_2(M))\}$. By Prop. 4.7, $\omega(G[M]) = \max\{\omega(G_1(M)), \omega(G_2(M))\}$. Hence, $\omega(v) = \omega(G[M])$ has been correctly determined Assume now that $t(\rho_C) = 1$. Put $\alpha := \omega(u') + \omega(u'') - \omega(\eta)$ and $\beta := \omega_{\neg\eta}(u') + \omega_{\neg\eta}(u'')$. By the latter arguments and induction assumption, $\alpha = \omega(G_1(M)) + \omega(G_2(M) - \omega(G[L_\eta])$ and $\beta = \omega(G[L_{u'} \setminus L_\eta]) + \omega(G[L_{u'} \setminus L_\eta]) = \omega(G_1(M) - G[L_\eta]) + \omega(G_2(M) - G[L_\eta])$. This together with Prop. 4.6 implies that $\omega(v) = \max\{\alpha, \beta\} = \omega(G[M]$ has been correctly determined in Line 31.

Hence, by induction, $\omega(\rho_{\mathcal{T}_G})$ captures the size of a maximum clique in $G[M_{\rho_{\mathcal{T}_G}}]$. Since $M_{\rho_{\mathcal{T}_G}} = V$, we have $\omega(\rho_{\mathcal{T}_G}) = \omega(G[V]) = \omega(G)$, which completes the proof. Even more, the arguments above imply that $\omega(v) = \omega(G[L_v])$ holds for all $v \in V(N)$ and, if $v$ is contained in a cycle $C$ of $N$ and $v \neq \rho_C$, then $\omega_{\neg\eta}(v) = G[L_v \setminus L_\eta]$. $\square$

**Proposition 4.11.** *Algorithm 2 correctly computes a maximum clique in* GATEX *graphs.*

*Proof.* Let $G = (V,E)$ be the input GATEX graph for Algorithm 2 and $(N,t)$ be the pvr-network that explains $G$ and that is used in Algorithm 2. In what follows, put $L_w := L(N(w))$ for $w \in V(N)$. Furthermore, for a vertex $w \in V(\mathcal{T}_G)$, let $M_w := L(\mathcal{T}_G(w))$ denote the module of $G$ associated with $w$. By Prop. 4.10, $\omega(v) = \omega(G[L_v])$ for all $v \in V(N)$ and, if $v$ is contained in a cycle $C$ of $N$ and $v \neq \rho_C$, then $\omega_{\neg\eta}(v) = G[L_v \setminus L_{\eta_C}]$.

In the following, we call a directed path in $N$ from $w$ to some leaf in $L_w$ an `active` *$w$-path* if all vertices in $P$ distinct from $w$ are marked as `active`. Moreover, we say that *Property ($\star$) is satisfied for a vertex* $w \in V(N)$ if a maximum clique in $G[L_w]$ is induced by all those leaves in $L_w$ that can be reached from `active` $w$-paths. We show that all vertices in $V(\mathcal{T}_G) \subseteq V(N)$ satisfy *Property ($\star$)*. Note that *Property ($\star$)* is trivially satisfied for all leaves in $L(N)$. Let $v$ be the currently processed vertex in Line 3. By induction, we can assume that the children $u$ of $v$ in $\mathcal{T}_G$ satisfy *Property ($\star$)*. We consider now the cases for $t_G(v) \in \{0, 1, \text{prime}\}$.

*Case $t_G(v) = 0$:* In this case, it follows from Observation 2.5 that the children of $v$ in $N$ are precisely the children of $v$ in $\mathcal{T}_G$, that is, $\mathrm{child}_N(v) = \mathrm{child}_{\mathcal{T}_G}(v)$. By Lemma 4.2, every maximum clique in $G[L_v]$ must be located entirely in one of the subgraphs $G[L_w]$, $w \in \mathrm{child}_N(v)$, of $G[L_v]$. In this case, one of the children $w \in \mathrm{child}_N(v)$ satisfying $\omega(w) = max\{\omega(z) \mid z \in \mathrm{child}_N(v)\}$ is marked as `active` (Line 6). By induction assumption, *Property ($\star$)* holds for $w$ and, in particular, $w$ is now `active`. This and the fact that a maximum clique in $G[L_v]$ is located entirely in $G[L_w]$ implies that *Property ($\star$)* holds for $v$.

*Case $t_G(v) = 1$:* In this case, it again follows from Observation 2.5 that the children of $v$ in $N$ are precisely the children of $v$ in $\mathcal{T}_G$, that is, $\mathrm{child}_N(v) = \mathrm{child}_{\mathcal{T}_G}(v)$. By Lemma 4.2, a maximum clique in $G[L_v]$ is the join union of the maximum cliques in $G[L_w]$, $w \in \mathrm{child}_N(v)$. In this case, all children $w \in \mathrm{child}_N(v)$ are marked as `active` (Line 9). By induction assumption, *Property* ($\star$) holds for all $w \in \mathrm{child}_N(v)$ and, in particular, all $w \in \mathrm{child}_N(v)$ are now `active`. Taken the latter arguments together, *Property* ($\star$) holds for $v$.

*Case* $t_G(v) = \text{prime}$*:* In this case, $M := M_v$ is a prime module of $G$ and $v$ is locally replaced by a cycle $C := C_M$ with unique hybrid $\eta := \eta_C$ and root $\rho_C = v$ according to Def. 2.4. Let $P^1$ and $P^2$ be the two sides of $C$ and $u' \in P^1$ and $u'' \in P^2$ be the two children of $\rho_C$ in $N$. By Prop. 4.10, $\omega(u') = \omega(G[L_{u'}])$ and $\omega(u'') = \omega(G[L_{u''}])$. In $(N,t)$, we either have $t(\rho_C) = 1$ or $t(\rho_C) = 0$.

Assume first that $t(\rho_C) = 0$. In Line 24, we put $\omega(v) := \max\{\omega(u'), \omega(u'')\} = \max\{\omega(G[L_{u'}], \omega(G[L_{u''}])\}$. By Prop. 4.10, $\omega(v) = \omega(G[L_v])$. We then pick in Line 25 one of the vertices $w = u'$ or $w = u''$ for which $\omega(w) = \max\{\omega(u'), \omega(u'')\}$ is satisfied and determine in Line 27 the side $P \in \{P^1, P^2\}$ of $C$ that contains $w$. Afterwards, the procedure $\textsc{Activate}\big((N,t), \{P\}, \omega, \omega_{\neg\eta}, 0, 0, 0\big)$ is called in Line 28. We are now in the procedure $\textsc{Activate}\big((N,t), \mathscr{P}, \omega, \omega_{\neg\eta}, \alpha, \beta, label\_\rho_C\big)$. In this case, we have $label\_\rho_C = 0$ and are therefore in the *for*-loop in Line 2 of this procedure. Here, $\mathscr{P} = \{P\}$ and we traverse the vertices $w'$ in $P$ in postorder. By construction, each $w'$ has exactly two children, one of them is located in $C$ and denoted by $u'$ while the other one is the one outside of $C$ and is denoted by $u''$ (Line 3). By Observation 2.5, the child $u''$ of $w'$ is always a child of $v$ in $\mathcal{T}_G$. By our induction hypothesis, $u''$ satisfies *Property* ($\star$). We now show that $w'$ satisfies *Property* ($\star$). Note that the first vertex considered in the procedure $\textsc{Activate}$ is $w' = \eta$. By Obs. 2.6, $\eta$ has precisely one child. One easily verifies that in both cases, $t(\eta) = 0$ or $t(\eta) = 1$, the unique child $u''$ of $\eta$ is marked as `active`. Since $u''$ satisfies *Property* ($\star$), and $L_\eta = L_{u''}$, $\eta$ satifies *Property* ($\star$). Suppose now that $w'$ is distinct from $\eta$. By induction, we can assume that the child $u'$ of $w'$ in $C$ satisfies *Property* ($\star$). Note that we can use this assumption, since $u'$ is processed before $w'$ in the procedure $\textsc{Activate}$ and since $\eta$ has been processed already. By Lemma 4.2, $t(w') = 0$ implies that every maximum clique in $G[L_{w'}]$ must be located entirely in one of $G[L_{u'}]$ or $G[L_{u''}]$. In the procedure $\textsc{Activate}$ (Line 4), we mark the child $u$ of $w$ satisfying $\omega(u) = \max\{\omega(u'), \omega(u'')\}$ as `active`. By our induction hypothesis, property ($\star$) holds for $u$. This and the fact that $u$ is `active` and that there exists a maximum clique in $G[L_{w'}]$ located entirely in $G[L_u]$ implies that *Property* ($\star$) holds for $w'$. If $t(w') = 1$ then, by Lemma 4.2, a maximum clique in $G[L_{w'}]$ is the join union of a maximum cliques in $G[L_{u'}]$ and a maximal clique in $G[L_{u''}]$. In this case, both $u'$ and $u''$ are marked as `active` in the procedure $\textsc{Activate}$ (Line 5). By induction, *Property* ($\star$) holds for both $u'$ and $u''$, and both of them are now `active`. The latter two arguments imply that *Property* ($\star$) holds for $w'$. In particular, *Property* ($\star$) holds for the chosen child $w$ of $v$ in $N$. Note that $w$ was marked as `active` in Alg. 2 (Line 26), while the other child of $v$ is not. Since by choice of $w$, $G[L_v]$ admits a maximum clique entirely contained in $G[L_w]$, it follows that $v$ satisfies *Property* ($\star$).

Assume now that $t(\rho_C) = 1$. In this case, we call $\textsc{Activate}\big((N,t), \{P^1, P^2\}, \omega, \omega_{\neg\eta}, \alpha, \beta, 1\big)$ in Alg. 2 (Line 32) where $\alpha = \omega(u') + \omega(u'') - \omega(\eta)$ and $\beta = \omega_{\neg\eta}(u') + \omega_{\neg\eta}(u'')$. To recall, $u' \in P^1$ and $u'' \in P^2$ are the two children of $\rho_C$ in $N$. As argued in the proof of Prop. 4.10, $\alpha = \omega(G_1(M)) + \omega(G_2(M) - \omega(G[L_\eta])$ and $\beta = \omega(G_1(M) - G[L_\eta]) + \omega(G_2(M) - G[L_\eta])$. Since $t(\rho_C) = 1$, we continue in Line 6 of procedure $\textsc{Activate}$. There are two cases, either $\alpha \leq \beta$ or $\alpha > \beta$.

Assume first that $\alpha \leq \beta$. In this case, Proposition 4.6 implies that a maximum clique in $G[L_v]$ can be obtained by taking the join union of one maximum clique in $G[L_{v_1} \setminus L_\eta]$ and one one maximum clique in $G[L_{v_2} \setminus L_\eta]$, where $v_1$ and $v_2$ are the two children of $\rho_C = v$ in $N$. Hence, a a maximum clique in $G[L_v]$ is, in particular, a maximum clique in $G[L_v \setminus L_\eta]$ Since $\alpha \leq \beta$, we are in Line 7 of the procedure $\textsc{Activate}$. For all vertices $w'$ in $P_1$ and $P_2$ distinct from $\rho_C$ and $\eta$, Observation 2.5 implies that the unique child $u''$ of $w'$ outside of $C$ is a child of $v$ in $\mathcal{T}_G$. In particular, by our induction hypothesis, $u''$ satisfies *Property* ($\star$). We now proceed with showing that every $w'$ in $P_1$ and $P_2$ distinct from $\rho_C$ and $\eta$, satisfies the following amended version of *Property* ($\star$). Namely, we say that $w'$ satisfies *Property* ($\star\star$) if a maximum clique in $G[L_{w'} \setminus L_\eta]$ is induced by all leaves in $L_{w'}$ that can be reached from the `active` $w'$-paths. If $w'$ is a parent of $\eta$ in $N$, then $L_{w'} \setminus L_\eta = L_{u''}$, and $u''$ is marked as `active` in . Since $u''$ satisfies *Property* ($\star$) by our induction hypothesis, it follows that $w'$ satisfies Property ($\star\star$) in Line 9. All remaining vertices $w'$in $P_1$ and $P_2$. i.e., those that are distinct from $\rho_C, \eta$ and its two unique parents $w_1$ and $w_2$ are now traversed in postorder (Line 10). Suppose now that $w'$ is one of these vertices and let $u'$ be the child of $w'$ in $C$. Since the vertices of $P_1$ and $P_2$ are processed in postorder in the procedure $\textsc{Activate}$, we may assume that $u'$ satisfies *Property* ($\star\star$). The latter is justified since the parents of $\eta$ have been processed and satisfy *Property* ($\star\star$). If $t(w') = 0$, then similar arguments as in the proof of Lemma 4.2 imply that a maximum clique in $G[L_{w'} \setminus L_\eta]$ is contained either in $G[L_{u''}]$ or in $G[L_{u'} \setminus L_\eta]$. In the procedure $\textsc{Activate}$ (Line 12), we mark $u'$ as active if $\omega_{\neg\eta}(u') \geq \omega(u'')$, and we mark $u''$ as active otherwise. By induction, *Property* ($\star$) holds for $u''$, and *Property* ($\star\star$) holds for $u'$. This and the fact that there exists a maximum clique in $G[L_{u'}]$ located entirely in $G[L_{u'} \setminus L_\eta]$ (in case $\omega_{\neg\eta}(u') \geq \omega(u'')$) or in $G[L_{u''}]$ (in case $\omega(u'') \geq \omega_{\neg\eta}(u')$) implies that *Property* ($\star\star$) holds for $w'$. If $t(w') = 1$, then similar arguments as in the proof of Lemma 4.2 imply that a maximum clique in $G[L_{w'} \setminus L_\eta]$ is the join union of a maximum clique in $G[L_{u'} \setminus L_\eta]$ and a maximum clique in $G[L_{u''}]$. In this case, both $u'$ and $u''$ are marked as `active` in the procedure $\textsc{Activate}$ (Line 13). By our induction hypothesis, *Property* ($\star$) holds for $u''$ and *Property* ($\star\star$) holds for $u'$. The latter two arguments imply that *Property* ($\star\star$) holds for $w'$. In particular, *Property* ($\star\star$) holds for the children $v_1$ and $v_2$ of $v$ in $N$. Note that both $v_1$ and $v_2$ are marked as `active` (Line 6). Moreover, $\eta$ is not marked as `active`, so for $i \in \{1,2\}$, all leaves that can be reached from $v_i$ via a path of `active` vertices are not descendants of $\eta$. It follows that for all leaves $x_1 \in L_{v_1}, x_2 \in L_{v_2}$ that can be reached by such a path, $\mathrm{lca}_N(x_1, x_2) = v$. Since $t(v) = 1$ and $(N,t)$ explains $G$, it follows that $\{x_1, x_2\}$ is an edge of $G$. Together with the fact that $v_1$ and $v_2$ satisfy *Property* ($\star\star$), this implies that the set of leaves of $L_v$ that can be reached from $v$

via a path of `active` vertices induces a clique of $G[L_v]$ of size $\omega(G[L_{v_1} \setminus L_\eta]) + \omega(G[L_{v_2} \setminus L_\eta])$. By Proposition 4.10, $\omega(G[L_{v_1} \setminus L_\eta]) + \omega(G[L_{v_2} \setminus L_\eta]) = \omega_{\neg\eta}(v_1) + \omega_{\neg\eta}(v_2) = \beta$, and since $\beta \geq \alpha$, Proposition 4.6 implies that the latter clique is a maximum clique in $L_v$. Therefore, $v$ satisfies *Property* $(\star)$.

Assume now that $\alpha > \beta$. In this case, Proposition 4.6 implies that every maximum clique in $G[L_v]$ must contain vertices in $L_\eta$, i.e., we must subsequently build active parts while keeping `active` paths along $\eta$. Since $\alpha > \beta$, we are in Line 14 of the procedure ACTIVATE, and we mark $\eta$ and its unique child as `active` (Line 15) and proceed with traversing the vertices $w$ in $P_1$ and $P_2$ distinct from $\eta$ and $\rho_C$ in postorder (Line 16). By Observation 2.5, for all such $w'$, the child $u''$ of $w'$ outside of $C$ is a child of $v$ in $\mathcal{T}_G$. In particular, by our induction hypothesis, $u''$ satisfies *Property* $(\star)$. We now proceed to show that $w'$ satisfies *Property* $(\star)$. Note that since the only child $u''$ if $\eta$ is `active` (Line 15), this is true for $w' = \eta$. Since the $w' \neq \eta$ vertices of $P_1$ and $P_2$ are processed in postorder, we may therefore assume that the child $u'$ of $w'$ in $C$ satisfies *Property* $(\star)$. By Lemma 4.2, $t(w') = 0$ implies that every maximum clique in $G[L_{w'}]$ must be located entirely in one of $G[L_{u'}]$ or $G[L_{u''}]$. Moreover, by Proposition 4.6, every maximum clique in $G[L_v]$ contains vertices in $L_\eta$. As a consequence, since $L_\eta \subseteq L_{w'} \subseteq L_v$, a maximum clique in $G[L_{w'}]$ contains vertices in $L_\eta$. Since $L_{u''} \cap L_\eta = \emptyset$, the latter two arguments imply that every maximum clique in $G[L_{w'}]$ must be located entirely in $G[L_{u'}]$. Since $u'$ is marked as `active` (Line 17), and $u'$ satisfies *Property* $(\star)$, it follows that $w'$ satisfies *Property* $(\star)$. If $t(w') = 1$ then, by Lemma 4.2, a maximum clique in $G[L_{w'}]$ is the join union of a maximum clique in $G[L_{u'}]$ and a maximal clique in $G[L_{u''}]$. In this case, both $u'$ and $u''$ are marked as `active` in the procedure ACTIVATE (Lines 17 and 18). By our induction hypothesis, *Property* $(\star)$ holds for both $u'$ and $u''$, and both of them are now `active`. The latter two arguments imply that *Property* $(\star)$ holds for $w'$. In particular, *Property* $(\star)$ holds for the children $v_1$ and $v_2$ of $v$ in $N$. Note that both $v_1$ and $v_2$ are marked as `active` (Line 17). Note also that for $x \in L_\eta$, $x$ can be reached from $v_1$ via a path of `active` vertices if and only if $x$ can be reached from $v_1$ via a path of `active` vertices. Moreover, for all leaves $x_1 \in L_{v_1} \setminus L_\eta, x_2 \in L_{v_2} \setminus L_\eta$, we have $\mathrm{lca}_N(x_1, x_2) = v$. Since $t(v) = 1$ and $(N,t)$ explains $G$, it follows that $\{x_1, x_2\}$ is an edge of $G$. Together with the fact that $v_1$ and $v_2$ satisfy *Property* $(\star)$, this implies that the set of leaves of $L_v$ that can be reached from $v$ via a path of `active` vertices induces a clique of $G[L_v]$ of size $\omega(G[L_{v_1}]) + \omega(G[L_{v_2}]) - \omega(G[L_{v_1} \cap L_{v_2}]) = \omega(G[L_{v_1}]) + \omega(G[L_{v_2}]) - \omega(G[L_\eta])$. By Proposition 4.10, $\omega(G[L_{v_1}]) + \omega(G[L_{v_2}]) - \omega(G[L_\eta]) = \omega(v_1) + \omega(v_2) - \omega(\eta) = \alpha$, and since $\alpha > \beta$, Proposition 4.6 implies that the latter clique is a maximum clique in $L_v$. Therefore, $v$ satisfies *Property* $(\star)$. $\square$

**Proposition 4.12.** *Algorithm 2 can be implemented to run in $O(|V| + |E|)$ time with input $G = (V,E)$*

*Proof.* We show now that Algorithm 2 can be implemented to run in $O(|V| + |E|)$ time for a given GATEX graph $G = (V,E)$. The modular decomposition tree $(\mathcal{T}_G, t_G)$ can be computed in $O(|V| + |E|)$ time [12]. By [15, Thm. 9.4 and Alg. 4], the pvr-network $(N,t)$ of $G$ can be computed within the same time complexity. Thus, Line 1 takes $O(|V| + |E|)$ time. Initializing $\omega(v) := 1$ for all leaves $v$ (and thus, the vertices of $G$) in Line 2 can be done in $O(|V|)$ time.

Note that $V$ is the leaf set of $\mathcal{T}_G$. We then traverse each of the $O(|V|)$ non-leaf vertices in $(\mathcal{T}_G, t_G)$ in postorder starting in Line 3. To simplify the arguments and to establish the runtime, we put $W := V(\mathcal{T}_G) \setminus V$ and partition the vertices in $W$ into $W_P \,\dot\cup\, (W \setminus W_P)$ where $W_P$ contains all vertices $v$ with $t(v) =$ prime. Moreover, we denote with $\deg_H(v)$ the number of edges incident to $v$ in some DAG $H$.

Note that for $v \in W \setminus W_P$ we have $\deg_N(v) = \deg_{\mathcal{T}_G}(v)$. All vertices $v \in W \setminus W_P$ are processed in Line 5 and 6 as well as in Line 8 and 9. It is an easy task to verify that the respective two steps take $O(\deg_N(v)) = O(\deg_{\mathcal{T}_G}(v))$ time for each of the vertices in $W \setminus W_P$. Hence, processing all vertices in $W \setminus W_P$ can be done in $O(\sum_{v \in W \setminus W_P} \deg_{\mathcal{T}_G}(v)) = O(|E(\mathcal{T}_G)|) = O(|V(\mathcal{T}_G)| = O(|V|)$ time.

Now, consider the vertices in $W_P$ which are processed in Line 10-32. Note first that the sides $P^1$ and $P^2$ of $C$ can be determined in $O(|V(C)|)$ time in Line 16. Moreover, it is easy to verify that, for each $v \in W_P$, all other individual steps starting at Line 10 can be done in constant time each processed vertex has precisely two children, except execution of the procedure ACTIVATE which takes $O(|V(C)|)$ time for each individual call. For each $v \in W_P$, ACTIVATE is called once. Each $v \in W_P$ is associated with the unique cycle $C^v := C_M$ with $M = L(\mathcal{T}_G(v))$. Taken together the latter arguments, for a given prime vertex $v$, Line 10 - 32 have runtime $O(|V(C^v)| + |E(C^v)|) = |V(C^v)|$. Note that each cycle $C$ has, by definition of pvr-networks, no vertex in common with every other cycles. Hence, processing all vertices in $W_P$ can be done in $\sum_{v \in W_P} O(|V(C^v)|) = O(|V(N)|)$

By [3, Prop. 1], we have $O(|V(N)|) = O(|V|)$. Hence, the overall time-complexity of Algorithm 2 is bounded by the time-complexity to compute $(\mathcal{T}_G, t_G)$ and $(N,t)$ in Line 1 and is, therefore, $O(|V| + |E|)$ time. $\square$

We consider now the problem of determining the independence number $\alpha(G)$ as well as a maximum independent set of GATEX graphs $G$. Suppose that a GATEX graph $G$ is explained by the network $(N,t)$ and let $\bar{t}\colon V(N) \to \{0,1,\}$ where $\bar{t}(v) = \odot$ for all leaves $v$ of $N$ and $\bar{t}(v) = 1$ if and only if $t(v) = 0$. Since $L(N) = V(G)$ and by [3, Prop. 1], we have $O(|V(N)|) = O(|V(G)|)$ and thus, this labeling can be computed in $O(|V(G)|)$ time. It is easy to verify that $(N,\bar{t})$ explains the complement $\overline{G}$ of $G$. The latter arguments imply that the complement of every GATEX graph is a GATEX graph as well. Since maximum cliques in $\overline{G}$ are precisely the maximum independent sets in $G$, the latter arguments together with Prop. 4.11 and 4.12 imply

**Theorem 4.13.** *A maximum clique and a maximum independent set can be computed in linear-time for* GATEX *graphs.*

# References


[1] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999. PMID: 10582567.

[2] Sebastian Böcker and Andreas W. M. Dress. Recovering symbolically dated, rooted trees from symbolic ultrametrics. *Advances in Mathematics*, 138(1):105–125, 1998.

[3] G. Cardona, F. Rosselló, and G. Valiente. Comparison of tree-child phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6:552–569, 2007.

[4] V. Chvátal. Perfectly ordered graphs. In C. Berge and V. Chvátal, editors, *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 63–65. North-Holland, 1984.

[5] D. G. Corneil, H. Lerchs, and L K Stewart Burlingham. Complement reducible graphs. *Discr. Appl. Math.*, 3:163–174, 1981.

[6] D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.

[7] Elias Dahlhaus, Jens Gustedt, and Ross M McConnell. Efficient and practical algorithms for sequential modular decomposition. *Journal of Algorithms*, 41(2):360 – 387, 2001.

[8] A Ehrenfeucht and G Rozenberg. Theory of 2-structures, part I: Clans, basic subclasses, and morphisms. *Theor. Comp. Sci.*, 70:277–303, 1990.

[9] A Ehrenfeucht and G Rozenberg. Theory of 2-structures, part II: Representation through labeled tree families. *Theor. Comp. Sci.*, 70:305–342, 1990.

[10] Andrzej Ehrenfeucht, Harold N. Gabow, Ross M. Mcconnell, and Stephen J. Sullivan. An O($n^2$) Divide-and-Conquer Algorithm for the Prime Tree Decomposition of Two-Structures and Modular Decomposition of Graphs. *Journal of Algorithms*, 16(2):283–294, 1994.

[11] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[12] M. Habib and C. Paul. A survey of the algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41 – 59, 2010.

[13] Marc Hellmuth, Adrian Fritz, Nicolas Wieseke, and Peter F. Stadler. Cograph editing: Merging modules is equivalent to editing $P_4$'s. *Art Discr. Appl. Math.*, 3:#P2.01, 2020.

[14] Marc Hellmuth, David Schaller, and Peter F. Stadler. Clustering systems of phylogenetic networks. *Theory in Biosciences*, 142(4):301–358, 2023.

[15] Marc Hellmuth and Guillaume E. Scholz. From modular decomposition trees to level-1 networks: Pseudo-cographs, polar-cats and prime polar-cats. *Discrete Applied Mathematics*, 321:179–219, 2022.

[16] Marc Hellmuth and Guillaume E. Scholz. Linear time algorithms for NP-hard problems restricted to GaTEx graphs. In Weili Wu and Guangmo Tong, editors, *Computing and Combinatorics*, pages 115–126. Springer Nature Switzerland, Cham, 2024.

[17] Marc Hellmuth and Guillaume E. Scholz. Resolving prime modules: The structure of pseudo-cographs and galled-tree explainable graphs. *Discrete Applied Mathematics*, 343:25–43, 2024.

[18] Marc Hellmuth, Peter F Stadler, and Nicolas Wieseke. The mathematics of xenology: Di-cographs, symbolic ultrametrics, 2-structures and tree-representable systems of binary relations. *Journal of Mathematical Biology*, 75(1):199–237, 2017.

[19] K. T. Huber and G. E. Scholz. Beyond representing orthology relations with trees. *Algorithmica*, 80(1):73–103, 2018.

[20] R. Duncan Luce and Albert D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, Jun 1949.

[21] Dániel Marx. Graph colouring problems and their applications in scheduling. *Periodica Polytechnica Electrical Engineering*, 48:11–16, 2004.

[22] Ross M. McConnell and Jeremy P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1-3):189 – 241, 1999.

[23] Matthias Middendorf and Frank Pfeiffer. On the complexity of recognizing perfectly orderable graphs. *Discrete Mathematics*, 80(3):327–333, 1990.

[24] Ameera Vaheeda Shanavas, Manoj Changat, Marc Hellmuth, and Peter F. Stadler. Unique least common ancestors and clusters in directed acyclic graphs. In Subrahmanyam Kalyanasundaram and Anil Maheshwari, editors, *Algorithms and Discrete Applied Mathematics*, pages 148–161, Cham, 2024. Springer Nature Switzerland.

[25] Victor Spirin and Leonid A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100(21):12123–12128, 2003.

[26] Marc Tedder, Derek Corneil, Michel Habib, and Christophe Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In *Automata, Languages and Programming*, volume 5125 of *Lecture Notes in Computer Science*, pages 634–645. Springer Berlin Heidelberg, 2008.

[27] Volker Turau and Christoph Weyer. *Algorithmische Graphentheorie*. De Gruyter, Berlin, München, Boston, 2015.