LINEAR EVOLUTION IS IN P

BENJAMIN BERGOUGNOUX, UNIVERSITY OF BERGEN, NORWAY
SVEIN HØGEMO, UNIVERSITY OF BERGEN, NORWAY
JAN ARNE TELLE, UNIVERSITY OF BERGEN, NORWAY
MARTIN VATSHELLE, UNIVERSITY OF BERGEN, NORWAY

Abstract. A fundamental problem in computational biology is the construction of phylogenetic trees, also called evolutionary trees, for a set of organisms. A graph-theoretic approach takes as input a similarity graph G on the set of organisms, with adjacency denoting evolutionary closeness, and asks for an edge-weighted tree T whose leaves are the set of organisms, with two vertices adjacent in G if and only if the distance between them in the tree is less than some specified bound. If this exists G is called a leaf power. In this paper we consider linear evolutionary trees, with T a caterpillar, and show that the corresponding linear leaf power recognition problem is solvable in polynomial time. Our approach introduces several new techniques and we believe it forms a significant step towards solving the long-standing open problem of recognizing leaf powers in polynomial time.

1. Introduction

The reconstruction of ancestral relationships is a fundamental problem in computational biology. In a phylogenetic tree, also called an evolutionary tree, each leaf is labeled by a distinct organism and a tree is formed by positing possible ancestors that might over time have led to this set of organisms. In 2002 Nishimura et al [14] introduced a graph-theoretic approach to the situation when the evolutionary trees are formed based on distance thresholds. We are then given a similarity graph G on the organisms, with adjacencies denoting evolutionary closeness up to some specified bound k, and ask for a tree T whose leaves are the set of organisms, with two vertices adjacent in G if and only if the distance between them in T is at most k. If such a tree exists G is called a k-leaf power, hinting at the connection to graph powers. In the last couple of decades there has been a lot of research into k-leaf powers, and the more general class of graphs called leaf powers, that is formed by taking the union of the class of k-leaf powers for all values of k, see e.g. [1, 3, 4, 6, 8, 9, 11, 13, 15, 16]. We can mention that leaf powers are known to be strongly chordal, see e.g. Lafond [12] which gives a nice overview, and more recently also to have have mim-width 1 [10].See also the survey by Calamoneri et al [5].

In their pioneering paper Nishimura et al [14] write: "In practice, phylogenetic problems involve edge-weighted trees and errors in distance estimators; our algorithms should be seen as a first step toward handling this more general situation." We consider leaf powers and allow edge weights between 0 and 1 in the tree which allows us to fix a bound of 1 for the tree distance.

The outstanding open problem in the field is whether leaf powers can be recognized in polynomial time. In other words, whether we can decide in polynomial time if a given similarity graph on a set of organisms could have come about by evolution. In this paper we introduce a natural form of linear evolution and show that the related decision problem 'could the similarity graph G have come about by linear evolution?' is solvable in polynomial time. To capture linear evolution we simply require that the tree T have a path containing all nodes of degree 2 or more, i.e. T should be a caterpillar. If such a tree exists we call G a linear leaf power. Note that T is not rooted, so in evolutionary terms distance is not in time as much as in biological differences, and the biologist can freely pick the common ancestor based on some extra information.

Our approach to show that linear leaf powers can be recognized in polynomial time has several ingredients. In Section 2 we give the main definitions. In Section 3 we first show that linear leaf powers are equivalent to what we call red-blue interval graphs. The red-blue interval graphs are interesting on their own: interval graphs can be defined by assigning each vertex v an interval by $\mathsf{left}(v) < \mathsf{right}(v)$ and with uv adjacent if and only if $\mathsf{left}(v) < \mathsf{right}(u)$ and $\mathsf{left}(u) < \mathsf{right}(v)$, whereas red-blue interval graphs can be defined by simply removing the condition $\mathsf{left}(v) < \mathsf{right}(v)$, in which case the blue vertices will be those violating the condition. We show that G is a linear leaf power if and only if it is a red-blue interval graph where the blue vertices are taken as all simplicial vertices without true twins. In evolutionary terms the simplicial vertices are the organisms where the set of similar organisms are all pairwise similar. We then reduce the red-blue interval graph recognition to the problem of deciding if an interval graph (the graph induced on red vertices) given with some blue cliques (the neighborhoods of blue vertices) have what we call a nice order, which will correspond to the ordering of the left endpoints of the intervals. In section 3.4 we show that the set of blue cliques can be enlarged to a set of so-called special cliques. In Section 4 we give the algorithm that decides if an interval graph together with a set of special cliques has a nice order. The algorithm proceeds bottom-up on the partial order on vertices given by membership in special cliques. Since we consider special cliques we are able to account for constraints, given by blue cliques on not-yet-processed vertices, at an earlier stage. See Figure 3 that captures the whole process. We end with some concluding remarks in Section 5.

2. Preliminaries

Our graph terminology is standard and we refer to [7]. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$. An edge between two vertices x and y is denoted by xy

or yx. The set of vertices that is adjacent to x is denoted by $N_G(x)$. A vertex x is simplicial if $N_G(x)$ is a clique. Two adjacent vertices x, y are true twins if $N_G(x) = N_G(y)$. For a set $U \subseteq V(G)$, we define $N_G(U) := \bigcup_{x \in U} N_G(x)$. If the underlying graph is clear, then we may remove G from the subscript.

The subgraph of G induced by a subset X of its vertex set is denoted by $G[X]$. For two disjoint subsets X and Y of $V(G)$, we denote by $G[X, Y]$ the bipartite graph with vertex set $X \cup Y$ and edge set $\{xy \in E(G) \mid x \in X \text{ and } y \in Y\}$.

Given a graph G and a weight function $\mathsf{w} : E(G) \to \mathbb{Q}$, we denote the distance between two vertices x and y by $d_G(x, y)$.

Branch decomposition. A branch decomposition of a graph G is a pair (T, δ) of a tree T and a bijective function δ between $V(G)$ and the leaves of T. Since we manipulate at the same time graphs and trees representing them, the vertices of trees will be called *nodes*. A branch decomposition (T, δ) is linear if T is a caterpillar: a tree in which there exists a path that contains every vertex of degree two or more.

Linear leaf power. Given a graph G, a branch decomposition (T, δ) of G and a weight $\mathsf{w} : E(T) \to [0, 1]$, we say that (T, δ, w) is a *leaf root* of G if, for every pair of distinct vertices (x, y), we have $d_T(\delta(x), \delta(y)) \leqslant 1$ if and only if $xy \in E(G)$. Moreover, if (T, δ) is a linear branch decomposition we call (T, δ, w) a *linear leaf root* . A graph is a *leaf power* if it admits a leaf root and it is a linear leaf power if it admits a linear leaf root. See e.g. [12] for a proof that this is equivalent to the standard definition of leaf powers.

Interval graph. For a graph G an *interval model* $\mathcal{I} = (I_v)_{v \in V(G)}$ is a collection of intervals in $\mathbb{Q}$, one for each vertex in $V(G)$. G is an *interval graph* if it admits an interval model $\mathcal{I}$ such that for every pair of vertices $u, v \in V(G)$, the intervals I_v and I_u intersect if and only if $uv \in E(G)$. For an interval I in $\mathbb{Q}$, we denote by $\mathsf{left}(I)$ its left endpoint (minimum) and by $\mathsf{right}(I)$ its right endpoint (maximum).

3. Equivalent definitions of linear leaf power

3.1. **Red-blue interval graph.** First we will define a variant of interval graphs we call red-blue interval graphs. These are just used as a stepping stone to the definition we actually use in the proof, however we find that this definition is nice in itself and provides valuable intuition to understand the problem.

Definition 1 (Red-blue interval graph). *A graph G is a* red-blue interval *graph if there exists a bipartition (R, B) of $V(G)$ and an interval model $\mathcal{I} = (I_v)_{v \in V(G)}$ such that*

$$E(G) = \{r_1 r_2 \mid r_1, r_2 \in R \text{ and } I_{r_1} \cap I_{r_2} \neq \varnothing\} \cup \{rb \mid r \in R, b \in B \text{ and } I_b \subseteq I_r\}.$$

We call $(R, B, \mathcal{I})$ a red-blue interval model of G.

Intuitively, the blue vertices induce an independent set, the red vertices induce an interval graph, and we have a blue-red edge if the blue interval is contained in the red interval.

Fact 2. *Let G be a red-blue interval graph and $(R, B, \mathcal{I})$ a red-blue interval model of G. We have that every $v \in B$ is a simplicial vertex.*

The following fact can be easily deduced from Figure 1.

Fact 3. *Let two intervals $I_1, I_2 \subseteq \mathbb{Q}$ with lengths l_1, l_2 and midpoints m_1, m_2 respectively. We have $I_1 \cap I_2 \neq \varnothing$ if and only if $|m_1 - m_2| \leqslant \frac{l_1 + l_2}{2}$. Moreover, we have $I_2 \subseteq I_1$ if and only if $|m_1 - m_2| \leqslant \frac{l_1 - l_2}{2}$.*

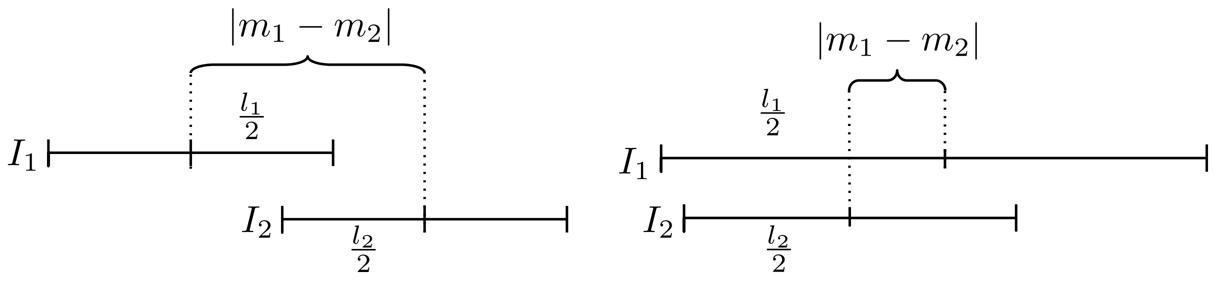

FIGURE 1. Example of two intervals overlapping and one interval containing another one.

Definition 4 (segment). *Given a graph G with an interval model $\mathcal{I} = (I_v)_{v \in V(G)}$ and $C \subseteq V(G)$, we define* $\mathsf{segment}_{\mathcal{I}}(C) = \cap_{c \in C} I_c$.

Note that if G is an interval graph and $\mathcal{I}$ an interval model of G then $\mathsf{segment}_{\mathcal{I}}(C) \neq \varnothing \iff C$ is a clique. It is clear that in any red-blue interval model $(R, B, \mathcal{I})$, for any vertex $b \in B$, we must have $I_b \subseteq \mathsf{segment}_{\mathcal{I}}(N(b))$.

Theorem 5. *A graph G is a red-blue interval graph if and only if it is a linear leaf power. Moreover, if G is a red-blue interval graph, it admits a red-blue interval model $(R, B, \mathcal{I})$ such that B is exactly the simplicial vertices of G with no true twin.*

Proof. We start with the first claim, of the equivalence between red-blue interval graphs and linear leaf powers.

($\Rightarrow$) Let G be a red-blue interval graph and $(R, B, (I_v)_{v \in V(G)})$ a red-blue interval model of G. Let $V(G) = \{v_1, \ldots, v_n\}$ such that, for every $i < j$, the midpoint of I_{v_i} is not greater than the midpoint of I_{v_j}. For every $i \in [n]$, we denote by l_i and m_i the length and the midpoint of the interval I_{v_i}. We suppose w.l.o.g. that, for every $i \in [n]$, we have $0 < l_i \leqslant 1$ as we can always divide the endpoints of all intervals by $\mathsf{max}_{i \in [n]}(l_i)$ and add some $\epsilon > 0$ to the right endpoints of the intervals of length 0.

We define (T, δ) a linear branch decomposition of G that is depicted on Figure 2. For every $i \in [n]$, we denote by f_i the edge incident to $\delta(v_i)$ and, for every $i \in [n-1]$, we define e_i the edge incident to f_i and f_{i+1}.

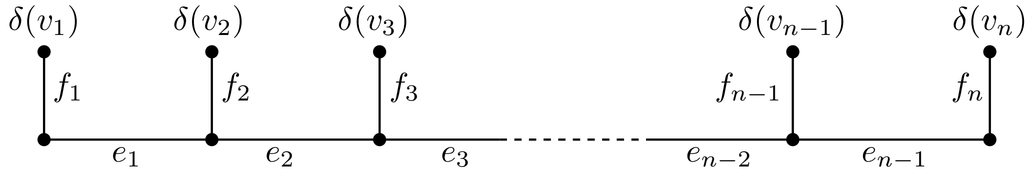

FIGURE 2. The linear branch decomposition used to prove that every red-blue interval graph is a linear leaf power.

Let w be the weight function on $E(T)$ such that, for every $i \in [n-1]$, we have $\mathsf{w}(e_i) = m_{i+1} - m_i$ and for every $i \in [n]$, we have

$$\mathsf{w}(f_i) = \begin{cases} \frac{1-l_i}{2} & \text{if } v_i \in R, \\ \frac{1+l_i}{2} & \text{if } v_i \in B \end{cases}$$

We claim that (T, δ, w) is a linear leaf root of G. Let $i, j \in [n]$ such that $i < j$. We have to prove that $v_i v_j \in E(G)$ if and only if $d_T(\delta(v_i), \delta(v_j)) \leqslant 1$.

First, observe that if v_i and v_j are blue vertices then $\mathsf{w}(f_i), \mathsf{w}(f_j) > \frac{1}{2}$ because we suppose that $l_t > 0$ for every $t \in [n]$. Consequently, we have $d_T(\delta(v_i), \delta(v_j)) > 1$ for every pair of blue vertices (v_i, v_j).

The path between v_i and v_j goes through the edges $f_i, e_i, e_{i+1}, \dots, e_{j-1}, f_j$, observe that, by construction, we have

$|\mathsf{w}(e_i)+\cdots+\mathsf{w}(e_{j-1})| = |m_{i+1}-m_i+m_{i+2}-m_{i+1}+\cdots+m_{j-1}-m_{j-2}+m_j-m_{j-1}| = |m_j-m_i|$. Hence $d_T(\delta(v_i),\delta(v_j)) = \mathsf{w}(f_i) + |m_j - m_i| + \mathsf{w}(f_j)$. Assume that v_i and v_j are red vertices. Then, $\mathsf{w}(f_i) = \frac{1-l_i}{2}$ and $\mathsf{w}(f_j) = \frac{1-l_j}{2}$. We deduce that the distance between $\delta(v_i)$ and $\delta(v_j)$ equals $|m_j - m_i| - \left(\frac{l_i+l_j}{2}\right) + 1$.

We have $v_iv_j \in E(G) \Leftrightarrow I_{v_i} \cap I_{v_j} \neq \varnothing$. By Fact 3, we have $I_{v_i} \cap I_{v_j} \neq \varnothing$ if and only if $|m_j - m_i| \leqslant \frac{l_i+l_j}{2}$. We conclude that $v_iv_j \in E(G)$ if and only if

$$d_T(\delta(v_i),\delta(v_j)) = |m_j - m_i| - \left(\frac{l_i+l_j}{2}\right) + 1 \leqslant 1.$$

Assume now that v_i and v_j do not have the same color and suppose that v_i is red (the proof is symmetric when v_i is blue). Then, the distance between $d_T(\delta(v_i),\delta(v_j)) = |m_j-m_i|-\left(\frac{l_i-l_j}{2}\right)+1$. We have $v_iv_j \in E(G) \Leftrightarrow I_{v_j} \subseteq I_{v_i}$. By Fact 3, $I_{v_j} \subseteq I_{v_i}$ if and only if $|m_j - m_i| \leqslant \frac{l_i-l_j}{2}$. We conclude that $v_iv_j \in E(G)$ if and only if

$$d_T(\delta(v_i),\delta(v_j)) = |m_j - m_i| + 1 - \left(\frac{l_i-l_j}{2}\right) \leqslant 1.$$

Hence, every red-blue interval graph is a linear leaf power.

($\Leftarrow$) Let G be a linear leaf power and (T,δ,w) a linear leaf root of G. Let $(u_1,\dots,u_t)$ be the path induced by the internal vertices of T. For every $1 \leqslant i \leqslant t$, we define $m_i = d_T(u_1,u_i)$.

We suppose w.l.o.g. that G does not contain isolated vertices as we can easily deal with such vertices by associating each of them with an interval that does not intersect the other intervals. Consequently, for every leaf a of T adjacent to some u_i, we have $d_T(a,u_i) \leqslant 1$. For every vertex $v \in V(G)$ such that $\delta(v)$ is adjacent to u_i, we associate v with an interval $I_v \subseteq \mathbb{Q}$ and a color such that the midpoint of I_v is m_i and

- if $d_T(u_i,\delta(v)) \leqslant \frac{1}{2}$, the length of I_v is $1 - 2d_T(u_i,\delta(v))$ and the color of v is red,
- otherwise ($\frac{1}{2} < d_T(u_i,\delta(v)) \leqslant 1$), the length of I_v is $2d_T(u_i,\delta(v)) - 1$ and the color of v is blue.

Let $R, B \subseteq V(G)$ be the sets of red and blue vertices respectively. By construction, we have the following equation for every vertex v such that $\delta(v)$ is adjacent to some u_i

$$d_T(u_i,\delta(v)) = \begin{cases} \frac{1-l_i}{2} & \text{if } v_i \in R, \\ \frac{1+l_i}{2} & \text{if } v_i \in B. \end{cases} \tag{1}$$

where l_i is the length of the interval I_v.

We claim that $(R,B,(I_v)_{v\in V(G)})$ is a red-blue interval model of G. Let $v_i, v_j \in E(G)$ such that $\delta(v_i)$ and $\delta(v_j)$ are adjacent to u_i and u_j respectively. By definition of the m_i's, we deduce that

$$d_T(\delta(v_i),\delta(v_j)) = |m_i - m_j| + d_T(u_i,\delta(v_i)) + d_T(u_j,\delta(v_j)).$$

First, assume that both v_i and v_j are blue. By construction, both $d_T(u_i,\delta(v_i))$ and $d_T(u_j,\delta(v_j))$ are strictly greater than $\frac{1}{2}$ and thus $d_T(\delta(v_i),\delta(v_j)) > 1$. Hence the blue vertices induce an independent set.

Now, assume that v_i or v_j is red. By Equation (1), we deduce that

$$d_T(\delta(v_i),\delta(v_j)) = \begin{cases} |m_j - m_i| + 1 - \left(\frac{l_i+l_j}{2}\right) & \text{if } v_i, v_j \in R, \\ |m_j - m_i| + 1 - \left(\frac{l_i-l_j}{2}\right) & \text{if } v_i \in B, v_j \in R, \\ |m_j - m_i| + 1 - \left(\frac{l_j-l_i}{2}\right) & \text{if } v_i \in R, v_j \in B. \end{cases}$$

As (T, δ, w) is a linear leaf root of G, we have $v_iv_j \in E(G)$ if and only if $d_T(\delta(v_i), \delta(v_j)) \leqslant 1$. By Fact 3, if v_i and v_j are red, we conclude that $v_iv_j \in E(G)$ if and only if $I_{v_i} \cap I_{v_j} \neq \varnothing$. Moreover, if v_i and v_j do not have the same color, suppose w.l.o.g. that $v_i \in R$, we have $v_iv_j \in E(G)$ if and only if $I_{v_j} \subseteq I_{v_i}$. Thus, $(R, B, (I_v)_{v\in V(G)})$ is a red-blue interval model of G.

For the second claim, we start with a simple observation: For any graph G, if there are two simplicial vertices $u, v \in V(G)$ that are neighbours, then they are also true twins. Indeed, if u had some neighbour x that v did not have, then the neighbourhood of u would not be a clique.

Now, we assume that G is a red-blue interval graph, and let X be the set of simplicial vertices in G without a true twin. Furthermore, we assume that $(R, B, \mathcal{I})$ be a red-blue interval model of G such that $|(X \setminus B) \cup (B \setminus X)|$ is minimum. We will show that the existence of a vertex in $(X \setminus B) \cup (B \setminus X)$ will lead to a contradiction.

First, we assume that there is a vertex $b \in B \setminus X$. By Fact 2, we deduce that b is a simplicial vertex with a true twin b', but since B must form an independent set, we have $b' \in R$. Since b and b' are true twins, the interval model $(R \cup \{b\}, B - b, \mathcal{I}')$ where

$$\mathcal{I}' = \{I'_v \mid I'_v = I_v \text{ for } v \neq b, I'_b = I'_{b'}\}$$

must be a red-blue interval model of G. But this contradicts the minimality of $|(X\setminus B)\cup(B\setminus X)|$.

Second, we assume that there is a vertex $x \in X \setminus B$. As x is simplicial, we know that $\mathsf{segment}(N(x)) \neq \varnothing$. Let $S = \mathsf{segment}(N(x))$. It is easy to see that also $I_x \cap S \neq \varnothing$. Also, for any $u \notin N(x)$, we have that I_u does not intersect I_x. Necessarily, $I_x \cap S \nsubseteq I_u$. Lastly, as x does not have a true twin, from our observation it cannot have any neighbours in B. This implies that the interval model $(R - x, B \cup \{x\}, \mathcal{I}')$ where

$$\mathcal{I}' = \{I'_v \mid I'_v = I_v \text{ for } v \neq x, I'_x = I_x \cap S\}$$

must be a red-blue interval model of G. But this contradicts the minimality of $|(X\setminus B)\cup(B\setminus X)|$. So we have that there must exist a red-blue interval model $(R, B, \mathcal{I})$ such that $B = X$. □

The following lemma tells us that for any red-blue interval graph G, there is a red-blue interval model $(R, B, \mathcal{I})$ where the interval of any blue vertex v is the segment of $N(v) \subseteq R$.

Lemma 6. *G is a red-blue interval graph $\iff$ G has a red-blue interval model $(R, B, \mathcal{I})$ where $\forall v \in B : I_v = \mathsf{segment}_{\mathcal{I}}(N(v))$.*

Proof. $(\Leftarrow)$ is obvious, for $(\Rightarrow)$ assume for contradiction that $(R, B, \mathcal{I})$ is the red-blue interval model with fewest blue intervals breaking this condition and that there exist $v \in B$ such that $I_v \neq \mathsf{segment}_{\mathcal{I}}(N(v))$. We must have $I_v \subseteq \mathsf{segment}_{\mathcal{I}}(N(v))$ since otherwise there must be a point $p \in I_v$ such that there exist a vertex $x \in N(v) : p \notin I_x$ (by definition of segment) which would contradict $I_v \subseteq I_x$. Assume there is $x \notin N(v) : \mathsf{segment}_{\mathcal{I}}(N(v)) \subseteq I_x$ but then we would have $I_v \subseteq I_x$ which is a contradiction. $G[R]$ is an interval graph and by Fact 2 $N(v) \subseteq R$ is a clique, hence $\forall x \in N(v) : I_x \supseteq \mathsf{segment}_{\mathcal{I}}(N(v)) \neq \varnothing$. We can replace I_v by $\mathsf{segment}_{\mathcal{I}}(N(v))$ contradicting minimality of $(R, B, \mathcal{I})$. □

3.2. **Special interval model.** By Lemma 6, finding a red-blue interval model of G is equivalent to finding an interval model for $G[R]$ satisfying some specific properties that we will specify now.

Definition 7 (special interval model)**.** *Let G be a graph, $\mathcal{B}$ a family of subsets of $V(G)$. We say an interval model $(I_v)_{v\in V(G)}$ is a special interval model for $\mathcal{B}$ if for every $C \in \mathcal{B}$ and $r \in V(G)$, we have $r \in C$ if and only if $\mathsf{segment}_{\mathcal{I}}(C) \subseteq I_r$.*

An example of a special interval model is given in Figure 3.

Fact 8. *If G is an interval graph with interval model $\mathcal{I} = (I_v)_{v\in V(G)}$, then for every maximal clique C we have $r \in C$ if and only if $\mathsf{segment}_{\mathcal{I}}(C) \subseteq I_r$.*

From Theorem 5, Lemma 6 and Fact 8 we deduce the following:

Corollary 9. *Let G be a graph, B be the set of simplicial vertices of G that do not have a true twin and $R = V(G) \setminus B$ and $\mathcal{B} = \{N(v) \mid v \in B\} \cup \{C \mid C$ is a maximal clique of $G[R]\}$. The graph G is a red-blue interval graph if and only if $G[R]$ admits a special interval model for $\mathcal{B}$.*

So deciding whether a graph is a red-blue interval graph, and hence a linear leaf power, is by Corollary 9 equivalent to solving the following problem.

Special Interval Model

Input: An interval graph G and a collection of cliques $\mathcal{B} \subseteq 2^{V(G)}$ called *blue cliques* that contains the maximal cliques of G.
Question: Does G admit a special interval model for $\mathcal{B}$?

For the remainder of the paper, we let G be an interval graph (note that G will correspond to the input graph of Linear Leaf Power induced by the red vertices) and $\mathcal{B}$ be a collection of blue cliques that contains all maximal cliques of G, and show how to solve Special Interval Model.

The following definition will be important for the rest of this paper.

Definition 10 ($\ll$)**.** *We define the predecessor relation for $(G, \mathcal{B})$ as the binary relation $\ll$ such that $u \ll v$, if for every blue clique C of $\mathcal{B}$, $v \in C$ implies $u \in C$. We define $\mathsf{Pred}(v) = \{u \in V(G) \mid u \ll v$ and $u \neq v\}$. Given a set $S \subseteq V(G)$, we denote by $\mathsf{Pred}(S) = \bigcup_{s \in S} \mathsf{Pred}(s)$.*

Observe that $\ll$ is not necessarily a partial order as this relation may be not antisymmetric. Indeed, if there exist true twins u, v such that $\{C \in \mathcal{B} \mid u \in C\} = \{C \in \mathcal{B} \mid v \in C\}$, then $u \ll v$ and $v \ll u$. In the following, we assume w.l.o.g. that there is no such pair of vertices thanks to the following fact. This assumption implies that $\ll$ is a partial order.

Fact 11. *If there is in G a pair of distinct vertices (u, v) such that $\{C \in \mathcal{B} \mid u \in C\} = \{C \in \mathcal{B} \mid v \in C\}$, then G admits a special interval model for $\mathcal{B}$ if and only if $G - v$ admits one for $\{C \setminus \{v\} \mid C \in \mathcal{B}\}$.*

Proof. Given a special interval model of G, we obtain one for $G - v$ and $\{C \setminus \{v\} \mid C \in \mathcal{B}\}$ by removing the interval associated with v. On the other hand, given a special interval model $(I_v)_{v \in V(G-v)}$ of $G - v$ for $\{C \setminus \{v\} \mid C \in \mathcal{B}\}$, we obtain one for G and $\mathcal{B}$ by adding the interval $I_v = I_u$. □

Our algorithm will decide whether G admits a special interval model by doing a bottom up traversal of the poset $(V(G), \ll)$ starting from the minimal elements, i.e. those that do not have any predecessors. The following lemma shows the main connection between the predecessor relation and intervals.

Lemma 12. *If G admits a special interval model for $\mathcal{B}$, then G admits a special interval model $(I_v)_{v \in V(G)}$ for $\mathcal{B}$ such that, for every $a, b \in V(G)$, we have $a \ll b$ if and only $I_b \subseteq I_a$.*

Proof. Assume that G admits a special interval model $\mathcal{I} = (I_v)_{v \in V(G)}$ for $\mathcal{B}$.

($\Rightarrow$)$a \ll b$ implies by definition that for every $C \in \mathcal{B}$, $b \in C$ implies $a \in C$. This means that for every $C \in \mathcal{B}$ and $a \in C$ we have $\mathsf{segment}_{\mathcal{I}}(C) \subseteq I_a \cap I_b$. Thus the model $\widehat{\mathcal{I}} = (\widehat{I_v})_{v \in V(G)}$ with $\widehat{I_b} = I_b \cap I_a$ and $\widehat{I_v} = I_v$ for every $v \neq b$ is a special interval model of G for $\mathcal{B}$. Indeed, for every $C \in \mathcal{B}$, we have $\mathsf{segment}_{\mathcal{I}}(C) = \mathsf{segment}_{\widehat{\mathcal{I}}}(C)$, thus we deduce from Definition 7 that $\widehat{\mathcal{I}}$ is a special interval model for $\mathcal{B}$.

($\Leftarrow$) For every $a, b \in V(G)$, if $I_b \subseteq I_a$, then for every $C \in \mathcal{B}$ such that $b \in C$, we have $\mathsf{segment}_{\mathcal{I}}(C) \subseteq I_b \subseteq I_a$. Thus, by Definition 7, we deduce that for every $C \in \mathcal{B}$, if $b \in C$, then $a \in C$. □

3.3. **Nice order.** In this section we will prove that G admits a special interval model for $\mathcal{B}$ if and only if G admits a total ordering on its vertices that corresponds to the order on the left endpoints of intervals.

Definition 13 (Continuous and troubling triples)**.** *Given the predecessor relation $\ll$ for $(G, \mathcal{B})$ and a partial order $\leqslant_{\mathsf{left}}$ on a subset $S \subseteq V(G)$, we say that a triple of vertices (a, b, c) in S is a troubling triple for $(C, \leqslant_{\mathsf{left}})$ with $C \subseteq V(G)$ if $a \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} c$, $a, c \in C$, $b \notin C$, and $a \not\ll b$. We say that $\leqslant_{\mathsf{left}}$ is continuous on C if there is no troubling triple for C.*

When the partial order $\leqslant_{\mathsf{left}}$ is clear from the context, we just say that a troubling triple for $(C, \leqslant_{\mathsf{left}})$ is a troubling triple for C.

Definition 14 (Nice order)**.** *A nice order $\leqslant_{\mathsf{left}}$ of $(G, \mathcal{B})$ is a linear extension of $\ll$ that is continuous on every blue clique of $\mathcal{B}$.*

The following lemma shows an important connection between nice orders and the predecessor relation.

Lemma 15. *Assuming G admits a nice order $\leqslant_{\mathsf{left}}$ for $(G, \mathcal{B})$, then for every a, b, c such that $a \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} c$, if $a \ll c$ then $a \ll b$ or $b \ll c$.*

Proof. Suppose that G admits a nice order $\leqslant_{\mathsf{left}}$ for $(G, \mathcal{B})$ and there exist $a, b, c \in V(G)$ such that $a \ll c$. Assume towards a contradiction that we do not have $a \ll c$ nor $b \ll c$. As we do not have $b \ll c$, there exists a special clique C such that $b \notin C$ and $c \in C$. Since $a \ll c$, we have $a \in C$. So $a, c \in C$ and $b \notin C$, this contradicts $\leqslant_{\mathsf{left}}$ being a nice order for $(G, \mathcal{B})$. □

Figure 3 illustrates the notions of Definitions 10 and 14.

In the following, we will prove that G admits a special interval model for $\mathcal{B}$ if and only if $(G, \mathcal{B})$ admits a nice order. For doing so, we use Algorihm 1, that given a nice order $\leqslant_{\mathsf{left}}$ for $(G, \mathcal{B})$, constructs a special interval model of G for $\mathcal{B}$. Let us explain how this algorithm proceeds. The algorithm constructs an interval model where the endpoints of the intervals are pairwise distinct. It proceeds from left to right and starts the intervals according to $\leqslant_{\mathsf{left}}$, i.e. $\mathsf{left}(I_{v_1}) < \cdots < \mathsf{left}(I_{v_n})$. The variable $\mathsf{Current}$ contains the vertices v with a left endpoint ($\mathsf{left}(v)$ is defined) and no right endpoint ($\mathsf{right}(v)$ is not defined). At each iteration of the for loop, the algorithm checks if v_{next} is adjacent to all vertices in $\mathsf{Current}$. If it is the case, then it starts the interval of v_{next}. Otherwise, it ends the interval of v_j where v_j is the minimum w.r.t. $\leqslant_{\mathsf{left}}$ among the maximal elements of $\mathsf{Current} \setminus N(v_{\mathsf{next}})$ w.r.t. $\ll$. The dummy vertex v_{n+1} is just here to force the algorithm to end every interval when $\mathsf{next} > n$.

Theorem 16. *The graph G admits a special interval model for $\mathcal{B}$ if and only if G admits a nice order for $(G, \mathcal{B})$. Moreover, if G admits a nice order for $(G, \mathcal{B})$, then, for every nice order $\leqslant_{\mathsf{left}}$ for $(G, \mathcal{B})$, Algorithm 1 returns a special interval model of G.*

Proof. ($\Rightarrow$) Assume that G admits a special interval model. Let $\mathcal{I} = (I_v)_{v \in V(G)}$ be a special interval model satisfying the property of Lemma 12. Observe that for every $a, b \in V(G)$, if $a \ll b$, then $\mathsf{left}(I_a) \leqslant \mathsf{left}(I_b)$. Let $\leqslant_{\mathsf{left}}$ be the relation on $V(G)$ such that, for every $a, b \in V(G)$, we have $a \leqslant_{\mathsf{left}} b$ if $a \ll b$ or $\mathsf{left}(I_a) < \mathsf{left}(I_b)$. One easily proves that $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$ from the following facts:

- For every $a, b \in V(G)$, if $a \ll b$, then $I_b \subseteq I_a$ and in particular $\mathsf{left}(I_a) \leqslant \mathsf{left}(I_b)$.
- For every $u, v \in V(G)$, if $\mathsf{left}(I_u) = \mathsf{left}(I_v)$, then either $I_v \subseteq I_u$ or $I_u \subseteq I_v$. We deduce that if $\mathsf{left}(I_u) = \mathsf{left}(I_v)$, then either $v \ll u$ or $u \ll v$.
- For every $u, v \in V(G)$, if $\mathsf{left}(I_u) < \mathsf{left}(I_v)$, then either $u \ll v$ or u and v are incomparable w.r.t. $\ll$.

We first prove a useful claim.

Claim 17. *Assume that $(G, \mathcal{B})$ admits a nice order $\leqslant_{\mathsf{left}}$ for $(G, \mathcal{B})$, and let $\mathcal{I} = (I_v)_{v \in V(G)}$ be the interval model generated from $(G, \leqslant_{\mathsf{left}})$ by Algorithm 1. For every $x, y \in V(G)$, if there exists $C \in \mathcal{B}$ such that $x \notin C$ and $y \in C$, then $I_y \not\subseteq I_x$.*

Proof. Assume towards a contradiction that there exist $x, y \in V(G)$ and $C \in \mathcal{B}$ such that $x \notin C$, $y \in C$ and $I_y \subseteq I_x$. Observe that $\mathsf{left}(I_x) < \mathsf{left}(I_y)$ because $I_y \subseteq I_x$ and thus we have $x \leqslant_{\mathsf{left}} y$. Let $\mathsf{Current}(y)$ and $\mathsf{next}(y)$ be the values of the variables $\mathsf{Current}$ and next in the for loop of

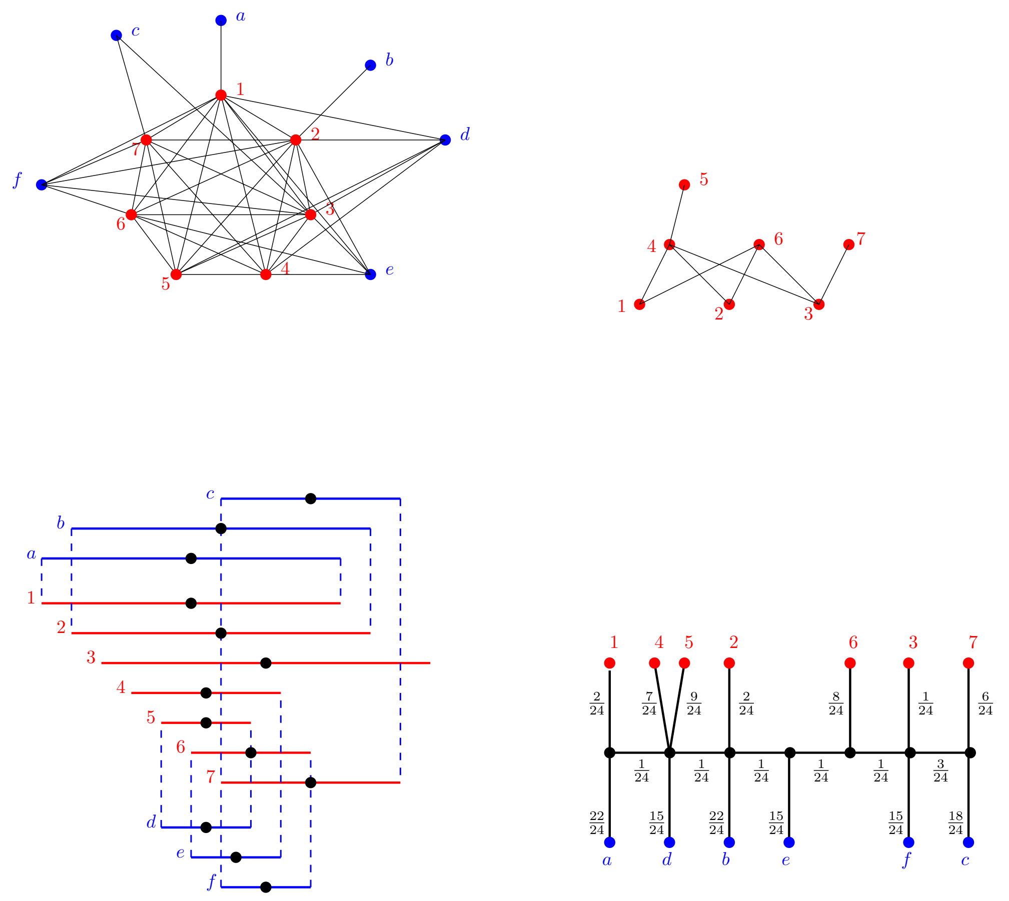

FIGURE 3. The whole process of recognizing linear leaf powers.
Top left: The input graph G, which happens to be a split graph. Vertices that are simplicial without true twins have been colored blue and the remainder red. The blue cliques are the neighborhoods of the blue vertices: $\{1\}$, $\{2\}$, $\{3,7\}$, $\{1,2,3,4,5\}$, $\{1,2,3,4,6\}$, $\{1,2,3,6,7\}$.
Top right: The Hasse diagram of the poset given by the predecessor relation on red vertices based on membership in blue cliques. The sibling classes are $\{\{1,2,3\},\{4,6\},\{7\},\{5\}\}$. Our main algorithm, featured in section 4, will show that G is a linear leaf power by finding the nice order 1234567. It does this by traversing the poset bottom-up and level by level. Let us explain the need for special cliques. When considering the minimal elements $\{1,2,3\}$ all orders on these elements are continuous on all blue cliques. However, the order 132 cannot be extended to a nice order on all red vertices, and this is captured by taking the blue clique $C = N(d) = \{1,2,3,4,5\}$ and considering the special clique $C(7,4) = \{1,2\}$. By Definition 13, if $1 \leqslant_{\mathsf{left}} 3 \leqslant_{\mathsf{left}} 2$ then $(1,2,3)$ is a troubling triple for $C(7,4)$ and the algorithm discovers right away that 132 cannot be extended to a nice order.
Bottom left: The resulting special interval model is given by the red intervals and the red-blue interval model by the red and blue intervals. Note that for red vertices the left endpoints of intervals respects the nice order and that each blue interval is equal to the intersection of its neighbourhood.
Bottom right: The caterpillar T showing that G is a linear leaf power. Note the midpoint of each interval has been marked, to better understand where each vertex is placed in T. Weights on edges is calculated as explained in Section 2 and note every edge has a weight that is a multiple of $\frac{1}{24}$.

Algorithm 1: Interval model from a nice order.

```
Data: An interval graph $G$ and a nice order $\leqslant_{\mathsf{left}}$ for $(G, \mathcal{B})$.
Result: An interval model of $G$.
1  Let $V(G) = \{v_1, \dots, v_n\}$ such that $v_1 \leqslant_{\mathsf{left}} \cdots \leqslant_{\mathsf{left}} v_n$;
2  Let $v_{n+1}$ be a dummy vertex with no neighbors;
3  $\mathsf{left}(v_1) \leftarrow 1$; $\mathsf{Current} \leftarrow \{v_1\}$; $\mathsf{next} \leftarrow 2$;
4  for $t = 2$ to $2n$ do
5      if $\mathsf{Current} \not\subseteq N(v_{\mathsf{next}})$ then
6          Let $M$ be the maximal elements of $\mathsf{Current} \setminus N(v_{\mathsf{next}})$ w.r.t. $\ll$;
7          $v_j \leftarrow \mathsf{argmin}_{v \in M} \mathsf{left}(v)$;
8          $\mathsf{right}(v_j) = t$;
9          $\mathsf{Current} \leftarrow \mathsf{Current} \setminus \{v_j\}$;
10     else
11         $\mathsf{left}(v_{\mathsf{next}}) = t$;
12         $\mathsf{Current} \leftarrow \mathsf{Current} \cup \{v_{\mathsf{next}}\}$;
13         $\mathsf{next} \leftarrow \mathsf{next} + 1$;
14     end
15 end
16 Let $(I_v)_{v \in V(G)}$ such that $I_{v_i} = [\mathsf{left}(v_i), \mathsf{right}(v_i)]$ for all $i \in [n]$;
17 return $(I_v)_{v \in V(G)}$;
```

Algorithm 1 when $t = \mathsf{right}(I_y)$ and Line 8 is executed. Let M be the set of maximal elements in $\mathsf{Current}(y) \setminus N(v_{\mathsf{next}(y)})$ w.r.t. $\ll$. We know that x and y belong to $\mathsf{Current}(y)$ and that y is the minimum element of M w.r.t. $\leqslant_{\mathsf{left}}$. Moreover, we have $x \leqslant_{\mathsf{left}} y \leqslant_{\mathsf{left}} v_{\mathsf{next}(y)}$.

Suppose that $xv_{\mathsf{next}(y)} \in E(G)$, then $v_{\mathsf{next}(y)}$ is not the dummy vertex. As $yv_{\mathsf{next}(y)} \notin E(G)$, there exists a maximal clique $\widehat{C}$ such that $x, v_{\mathsf{next}(y)} \in \widehat{C}$ and $y \notin \widehat{C}$. As $\leqslant_{\mathsf{left}}$ is a nice order, this implies $x \ll y$. This contradicts $y \in C$ and $x \notin C$.

Now assume that $xv_{\mathsf{next}(y)} \notin E(G)$. Since $x \leqslant_{\mathsf{left}} y$ and y is the minimum element of M w.r.t. $\leqslant_{\mathsf{left}}$, we know that x is not in M. Thus, x is not maximal in $\mathsf{Current}(y) \setminus N(v_{\mathsf{next}(y)})$ w.r.t. $\ll$. Consequently, there exists $z \in M$ such that $x \ll z$. Since y is minimum in M, we have $y \leqslant_{\mathsf{left}} z$. Moreover, y and z are incomparable w.r.t. $\ll$ thanks to their maximality. Thus, there exists $\widehat{C} \in \mathcal{B}$ such that $z \in \widehat{C}$ and $y \notin \widehat{C}$. As $x \ll z$, we have $x \in \widehat{C}$. From the definition of nice order, we deduce that $x \ll y$, yielding a contradiction with $y \in C$ and $x \notin C$. □

Assume towards a contradiction that $\leqslant_{\mathsf{left}}$ admits a troubling triple (a, b, c) for some $C \in \mathcal{B}$. Because $a, c \in C$, $b \notin C$ and $a \not\ll b$, we know that b is incomparable with a and c. Thus, by definition of $\leqslant_{\mathsf{left}}$, we have $\mathsf{left}(I_a) < \mathsf{left}(I_b) < \mathsf{left}(I_c)$. Moreover, the property of Claim 17 implies that I_a does not contain I_b and I_b does not contain I_c. Thus, we have $\mathsf{right}(I_a) < \mathsf{right}(I_b) < \mathsf{right}(I_c)$. We deduce that $I_a \cap I_c = [\mathsf{left}(I_c), \mathsf{right}(I_a)] \subseteq I_b$. Hence, we have $\mathsf{segment}(C) \subseteq I_a \cap I_c \subseteq I_b$. By Definition 7, this contradicts $\mathcal{I}$ being a special interval model. So, if G admits a special interval model, then $(G, \mathcal{B})$ admits a nice order.

($\Leftarrow$) Assume that $(G, \mathcal{B})$ admits a nice order. Let $\mathcal{I} = (I_v)_{v \in V(G)}$ be the output of Algorithm 1 and let $V(G) = \{v_1, \dots, v_n\}$ such that $v_1 \leqslant_{\mathsf{left}} \cdots \leqslant_{\mathsf{left}} v_n$.

We claim that $\mathcal{I}$ is a special interval model of G for $\mathcal{B}$. We start by proving that $\mathcal{I}$ is an interval model of G, i.e. $ac \in E(G)$ if and only if $I_a \cap I_c \neq \varnothing$. Line 5 of Algorithm 1 guarantees that, for every $ac \notin E(G)$, we have $I_a \cap I_c = \varnothing$. So, we need to prove that if $ac \in E(G)$, then $I_a \cap I_c \neq \varnothing$.

Assume towards a contradiction that there exists $ac \in E(G)$ such that $I_a \cap I_c = \varnothing$. Of all such edges, we pick the lexicographically leftmost one, i.e. out of $ac, bd \in E(G)$, we pick ac if $a \leqslant_{\mathsf{left}} b$ or if $a = b$ and $c \leqslant_{\mathsf{left}} d$. Furthermore, we assume w.l.o.g. that $a \leqslant_{\mathsf{left}} c$. Then, Line 5 of Algorithm 1 implies that there exists $b \in V(G)$ with $b \leqslant_{\mathsf{left}} c$, $\mathsf{right}(b) > \mathsf{right}(a)$ and $bc \notin E(G)$. It must then be that $b \not\ll a$, by Claim 17 we thus have that $a \leqslant_{\mathsf{left}} b$. As $bc \notin E(G)$ and $ac \in E(G)$, there exists a maximal clique C such that $a, c \in C$ and $b \notin C$. By definition of nice order, it must be the case that $a \ll b$. If $I_a \cap I_b \neq \varnothing$, this contradicts a being the maximal element when it was taken out of $\mathsf{Current}$. If $I_a \cap I_b = \varnothing$, this contradicts the leftmost position of ac, as $ab \in E(G)$. Hence, $\mathcal{I}$ is an interval model of G.

It remains to prove that $\mathcal{I}$ is a special interval model of G. We assume towards a contradiction that $\mathcal{I}$ is not a special interval model. So, there exist a blue clique C and a vertex $b \notin C$ such that $\mathsf{segment}(C) \subseteq I_b$. Let $a \in C$ such that $\mathsf{right}(I_a)$ is minimum and let $c \in C$ such that $\mathsf{left}(I_c)$ is maximum. We have $\mathsf{segment}(C) = I_a \cap I_c$. Since $a, c \in C$ and $b \notin C$, by Claim 17, we know that I_b does not contain I_a nor I_c.

As $I_a \cap I_c = \mathsf{segment}(C) \subseteq I_b$, we must have $I_a \not\subseteq I_c$ and $I_c \not\subseteq I_a$. Consequently, we have

$$\mathsf{left}(I_a) < \mathsf{left}(I_b) < \mathsf{left}(I_c) < \mathsf{right}(I_a) < \mathsf{right}(I_b) < \mathsf{right}(I_c).$$

Thus, we have $a \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} c$. Since $a, c \in C$ and $b \notin C$, by definition of nice order, we have $a \ll b$.

Let $\mathsf{Current}(a)$ and $\mathsf{next}(a)$ be the values of the variables $\mathsf{Current}$ and next in the for loop of Algorithm 1 when $t = \mathsf{right}(I_a)$ and Line 8 is executed. By definition, we have $av_{\mathsf{next}(a)} \notin E(G)$. Since $a \ll b$, we deduce that b is also not adjacent to $v_{\mathsf{next}(a)}$. Thus, a is not maximal in $\mathsf{Current}(a) \setminus N(v_{\mathsf{next}(a)})$. This yields a contradiction with $\mathsf{right}(I_a) \leqslant \mathsf{right}(I_b)$. □

3.4. **Special cliques.** In this section we show that we can build a set family $\mathcal{SC} \supseteq \mathcal{B}$ whose elements will be called *special cliques*, and show that $(G, \mathcal{B})$ is a yes-instance if and only if $(G, \mathcal{SC})$ is a yes-instance. Special cliques are subsets of blue cliques such that every nice order for $(G, \mathcal{B})$ is continuous on them, and are computed by Algorithm 2 using the following definition.

Definition 18. *For $C \subseteq V(G)$, $v \notin C$ and $u \in C \setminus \mathsf{Pred}(v)$, we define $C(v, u)$ as follows*

$$C(v,u) = \{x \in C'(v,u) \mid \mathsf{Pred}(x) \subseteq C'(v,u)\} \text{ where } C'(v,u) = C \setminus (\mathsf{Pred}(v) \setminus \mathsf{Pred}(u)).$$

Observe that $C(v, u)$ is obtained from C by removing the predecessors of v that are not predecessors of u and then by removing the successors of the vertices we have removed.

Lemma 19. *In time $O(n^4)$, Algorithm 2 either confirms that $(G, \mathcal{B})$ is a no-instance or computes a set $\mathcal{SC}$ such that:*

(1) If $(G, \mathcal{B})$ admits a nice order then for every nice order $\leqslant_{\mathsf{left}}$ of G, $\leqslant_{\mathsf{left}}$ is continuous on every element of $\mathcal{SC}$.
(2) $(G, \mathcal{B})$ is a yes-instance if and only if $(G, \mathcal{SC})$ is a yes-instance.
(3) For every $C \in \mathcal{SC}$, $v \notin C$ and $u \in C \setminus \mathsf{Pred}(v)$, we have $C(v, u) \in \mathcal{SC}$.
(4) For every $C \in \mathcal{SC}$ and $v \in C$, we have $\mathsf{Pred}(v) \subseteq C$.

Proof. We will first prove Property (1) and thus assume that G admits a nice order $\leqslant_{\mathsf{left}}$.

Claim 20. *Let $C \subseteq V(G)$, $v \notin C$ and $u \in C \setminus \mathsf{Pred}(v)$. If $\leqslant_{\mathsf{left}}$ is continuous on C, then $\leqslant_{\mathsf{left}}$ is continuous on $C(v, u)$.*

Proof. Assume towards a contradiction that $\leqslant_{\mathsf{left}}$ is not continuous on $C(v, u)$. That is, there exists a troubling triple (a, b, c) for $C(v, u)$. Choose such a triple with b minimal for $\ll$.

First, we prove that $b \notin C'(v, u) = C \setminus (\mathsf{Pred}(v) \setminus \mathsf{Pred}(u))$. Suppose that $b \in C'(v, u)$, by construction $C(v, u)$, this implies that $\mathsf{Pred}(b) \not\subseteq C'(v, u)$. Thus, there exists $\widehat{b} \in \mathsf{Pred}(b)$ such that $\widehat{b} \notin C'(v, u)$. We show that we can replace b with $\widehat{b}$. Since $\widehat{b} \ll b$ and $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$, we must have $\widehat{b} \leqslant_{\mathsf{left}} b$. By Lemma 15, we can not have $\widehat{b} \leqslant_{\mathsf{left}} a \leqslant_{\mathsf{left}} b$, as this would imply $\widehat{b} \ll a$ contradicting $a \in C(v, u)$. We deduce that $a \leqslant_{\mathsf{left}} \widehat{b} \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} c$. Moreover, we can not have $a \ll \widehat{b}$ because $a \not\ll b$ and $\widehat{b} \ll a$. Thus, $(a, \widehat{b}, c)$ is a troubling triple for $C(v, u)$ such that $\widehat{b} \notin C(v, u)$. This contradicts the choice of (a, b, c) and thus $b \notin C'(v, u)$.

This means that by definition of $C'(v, u)$ we have $b \in \mathsf{Pred}(v) \setminus \mathsf{Pred}(u)$. Since $b \ll v$ and $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$, we know that $b \leqslant_{\mathsf{left}} v$. We distinguish the following cases:

(1) If $a, c \notin \mathsf{Pred}(v)$, then by Lemma 15, we cannot have $b \leqslant_{\mathsf{left}} c \leqslant_{\mathsf{left}} v$, as this again would contradict the assumption that $c \in C'(v, u)$. Thus, we must have $a \leqslant_{\mathsf{left}} v \leqslant_{\mathsf{left}} c$. Observe that (a, v, c) is a troubling triple for C because $a, c \in C(v, u) \subseteq C$, $v \notin C$ and by assumption $a \not\ll v$. This contradicts $\leqslant_{\mathsf{left}}$ being continuous on C.
(2) If $a \in \mathsf{Pred}(v)$ and $c \notin \mathsf{Pred}(v)$, then $a \in C(v, u)$ and $C(v, u) \subseteq C'(v, u)$ imply that $a \ll u$ and so $a \leqslant_{\mathsf{left}} u$. By Lemma 15 we can not have $a \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} u$ because $b \not\ll u$. Similarly, we cannot have $b \leqslant_{\mathsf{left}} c \leqslant_{\mathsf{left}} v$ because $c \not\ll v$. We deduce that $a \leqslant_{\mathsf{left}} u \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} v \leqslant_{\mathsf{left}} c$. But this contradicts $\leqslant_{\mathsf{left}}$ being continuous on C as (u, v, c) is then a troubling triple for C (since $u, c \in C(v, u) \subseteq C$, $v \notin C$ and $u \not\ll v$).
(3) If $c \in \mathsf{Pred}(v)$ and $a \notin \mathsf{Pred}(v)$, then we deduce that $c \ll u$ and $c \leqslant_{\mathsf{left}} u$ similarly to the previous case. Since $b \not\ll u$ and $u \not\ll v$, Lemma 15 implies that we can not have $u \leqslant_{\mathsf{left}} v$. We conclude that we have $a \leqslant_{\mathsf{left}} c \leqslant_{\mathsf{left}} v \leqslant_{\mathsf{left}} u$. But this contradicts $\leqslant_{\mathsf{left}}$ being continuous on C as (a, v, u) is a troubling triple for C (since $a, u \in C$, $v \notin C$ and by assumption $a \not\ll v$).
(4) If $a, c \in \mathsf{Pred}(v)$, as in the preceding cases, we deduce that $a \ll u$ and $c \ll u$. But this implies that $a \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} u$, which by Lemma 15 contradicts $b \not\ll u$.

□

When the algorithm starts we have $\mathcal{SC} = \mathcal{B}$ and by definition of nice order, $\leqslant_{\mathsf{left}}$ is continuous on every element in $\mathcal{SC}$. At each step the algorithm takes some $C \in \mathcal{SC}$ and adds $C(v, u)$ to $\mathcal{SC}$. By Claim 20, $\leqslant_{\mathsf{left}}$ is still continuous on every element in $\mathcal{SC}$ after the addition, and thus we have proven Property (1).

Property (4) holds on every $C \in \mathcal{B}$ by definition of Pred and it holds on every $\mathcal{SC} \setminus \mathcal{B}$ by definition of $C(v, u)$.

Thus the predecessor relation of the instance $(G, \mathcal{SC})$ is the same the we use for $(G, \mathcal{B})$. Consequently, $(G, \mathcal{SC})$ is a yes-instance if and only if there exists a linear extension of $\ll$ that is continuous on every element in $\mathcal{SC}$. As $\mathcal{B} \subseteq \mathcal{SC}$, we conclude that Property (2) holds.

Since in Line 5, we add to $\mathsf{Current}$ and $\mathcal{SC}$ every $C(v, u)$ that can be constructed from $C \in \mathsf{Current}$, Property (3) holds as long as the algorithm does not return no.

We now prove that when the algorithm returns no, then $(G, \mathcal{B})$ is a no-instance. This follows from the following claim and Property (2).

Claim 21. *If $(G, \mathcal{B})$ is a yes-instance of* Special Interval Model *then $|\mathcal{B}| \leqslant n^2$.*

Proof. Suppose that $(G, \mathcal{B})$ is a yes-instance and let $(I_v)_{v \in V(G)}$ be a special interval model. Let C be a blue clique. Since $\mathsf{segment}(C) = \bigcap_{v \in C} I_v$, there exists $x, y \in C$ such that $\mathsf{segment}(C) = I_x \cap I_y$ (we can have $x = y$). Indeed, it is sufficient to take x such that $\mathsf{left}(I_x) = \mathsf{left}(\mathsf{segment}(C))$ and y such that $\mathsf{right}(I_y) = \mathsf{right}(\mathsf{segment}(C))$.

As C is a blue clique, there is no $v \notin C$ such that $\mathsf{segment}(C) \subseteq I_x$. So, for every vertex v, we have $v \in C$ if and only $\mathsf{segment}(C) \subseteq I_v$. Now, observe that $\mathsf{segment}(C) \subseteq I_v$ if and only if $\mathsf{left}(I_v) \leqslant \mathsf{left}(I_x)$ and $\mathsf{right}(I_y) \leqslant \mathsf{max}(I_v)$. So, C is entirely determined by x and y. We conclude that there are at most n^2 blue cliques. □

The runtime is $O(n^4)$ since $|\mathcal{SC}| \leqslant O(n^2)$ and for each $C \in \mathcal{SC}$ there are at most $O(n^2)$ vertices u and v that can be chosen. □

4. Recognition

In this section, we give in Algorithm 3 a polynomial time algorithm that decides whether $(G, \mathcal{B})$ admits a special interval model by deciding whether $(G, \mathcal{B})$ admits a nice order. We start in Line 1 by calling Algorithm 2 to compute $\mathcal{SC}$. The algorithm will then process vertices by a bottom up traversal of the poset given by the predecessor relation of $(G, \mathcal{B})$ (which is also the predecessor relation of $(G, \mathcal{SC})$). If all the vertices are incomparable for $\ll$, then the algorithm will end by a call to subroutine $\mathsf{LRorder}(V(G), \varnothing, \varnothing, \mathcal{SC})$ in Line 3 that simply checks whether the incidence matrix between vertices and special cliques has the consecutive ones property. Indeed,

Algorithm 2: Computation of $\mathcal{SC}$.

```
Data: An interval graph $G$ and a collection $\mathcal{B}$ of blue cliques.
Result: A collection $\mathcal{SC}$ or "no".
$\mathcal{SC} \leftarrow \mathcal{B}$;
Current $\leftarrow \mathcal{SC}$;
while Current $\neq \varnothing$ and $|\mathcal{SC}| \leqslant n^2$ do
    Take $C \in$ Current;
    for every $v \notin C$ and $u \in C \setminus \mathsf{Pred}(v)$ such that $\mathsf{Pred}(v) \cap C \not\subseteq \mathsf{Pred}(u)$ do
        Add $C(v,u)$ to Current and $\mathcal{SC}$;
    end
    Remove $C$ from Current;
end
if $|\mathcal{SC}| > n^2$ then return no;
else return $\mathcal{SC}$;
```

since the vertices are incomparable for $\ll$ the nice orders are exactly the permutations of the vertices that have the consecutive ones property, as 'continuous' is then simply 'consecutive'.

In the general case, the algorithm uses the LRorder subroutine to construct incrementally a total order on the processed vertices, that is continuous on every special clique, or to decide that such an order does not exist. Since we consider special cliques we are able to account for constraints, given by blue cliques on not-yet-processed vertices, at an earlier stage. In each iteration of the while loop the set of vertices called Current will be processed. These vertices are those that have all their predecessors in the set of processed vertices called Treated. To extend the computed total order $\leqslant_{\mathsf{left}}$ on Treated we use subroutine InsertAfter in Line 21 which takes a total order on the processed vertices X, together with a total order on set $S \subseteq$ Current of vertices having the same predecessors and splices these total orders as follows.

Insertion. Given two disjoint sets X, S, two total orders $\leqslant_{\mathsf{left}}, \leqslant_S$ on X and S respectively and $\widehat{s} \in X$, we denote by $\mathsf{InsertAfter}(\leqslant_{\mathsf{left}}, \leqslant_S, \widehat{s})$ the procedure that returns an ordering $\leqslant_{X\cup S}$ on $X \cup S$ satisfying the following properties:

- for every $x_1, x_2 \in X$, $x_1 \leqslant_{X\cup S} x_2$ if and only if $x_1 \leqslant_{\mathsf{left}} x_2$,
- for every $s_1, s_2 \in S$, $s_1 \leqslant_{X\cup S} s_2$ if and only if $s_1 \leqslant_S s_2$,
- for every $x \in X$ and $s \in S$, $x \leqslant_{X\cup S} s$ if and only if $x \leqslant_{\mathsf{left}} \widehat{s}$.

This operation is illustrated in Figure 4.

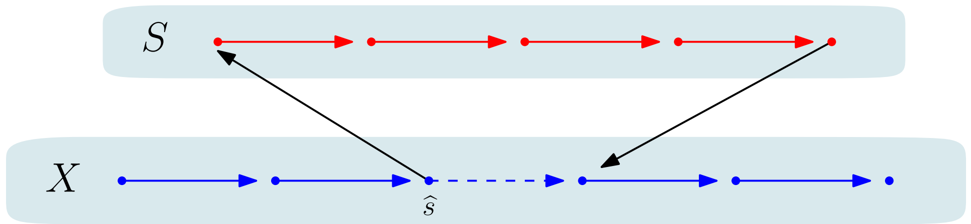

FIGURE 4. Intuitions on InsertAfter. The red and blue arrows represent respectively the Hasse diagram of $\leqslant_S$ and $\leqslant_{\mathsf{left}}$. The solid arrows represent the Hasse diagram of $\mathsf{InsertAfter}(\leqslant_{\mathsf{left}}, \leqslant_S, \widehat{s})$.

The S's are sets of pairwise incomparable vertices defined by being a sibling class, as follows.

Definition 22. *Two vertices u, v of G are siblings if* $\mathsf{Pred}(u) = \mathsf{Pred}(v)$. *The sibling relation induces an equivalence relation on $V(G)$, whose equivalence classes are called sibling classes.*

As an example, let $(G, \mathcal{B})$ be the graph and the blue cliques of Figure 3; observe that $\{4, 6\}$ is a sibling class of G with predecessors 1, 2, and 3.

We can insert a sibling class as a block because of the following fact.

Fact 23. *Assuming G admits a nice order $\leqslant_{\mathsf{left}}$, then for every sibling class S and vertex v such that v is incomparable with the vertices in S for $\ll$, we have either $s \leqslant_{\mathsf{left}} v$ for every $s \in S$, or $v \leqslant_{\mathsf{left}} s$ for every $s \in S$.*

Proof. Let S be a sibling class and v be a vertex incomparable with the vertices in S for $\ll$. Assume towards a contradiction that there exists a nice order $\leqslant_{\mathsf{left}}$, $s_1, s_2 \in S$ such that $s_1 \leqslant_{\mathsf{left}} v \leqslant_{\mathsf{left}} s_2$. Since $v \notin S$, we have $\mathsf{Pred}(S) \neq \mathsf{Pred}(v)$. Thus, we know that $\mathsf{Pred}(S) \not\subseteq \mathsf{Pred}(v)$ or $\mathsf{Pred}(v) \not\subseteq \mathsf{Pred}(S)$.

Suppose first that $\mathsf{Pred}(S) \not\subseteq \mathsf{Pred}(v)$. Let $\widehat{s} \in \mathsf{Pred}(S) \setminus \mathsf{Pred}(v)$. Since $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$, we have $\widehat{s} \leqslant_{\mathsf{left}} s_1$. We deduce that $\widehat{s} \leqslant_{\mathsf{left}} v \leqslant_{\mathsf{left}} s_2$. But $\widehat{s} \ll s_2$ and v is incomparable with both $\widehat{s}$ and s_2. This contradicts Lemma 15.

Now, suppose that $\mathsf{Pred}(v) \not\subseteq \mathsf{Pred}(S)$. Let $\widehat{v} \in \mathsf{Pred}(v) \setminus \mathsf{Pred}(S)$ such that $\widehat{v}$ is minimal for $\ll$. From Lemma 15, we deduce that we can not have $\widehat{v} \leqslant_{\mathsf{left}} s_1 \leqslant_{\mathsf{left}} v$. So we have $s_1 \leqslant_{\mathsf{left}} \widehat{v} \leqslant_{\mathsf{left}} v$.

Suppose that $\mathsf{Pred}(S) \subseteq \mathsf{Pred}(\widehat{v})$. We can not have $\mathsf{Pred}(\widehat{v}) \not\subseteq \mathsf{Pred}(S)$ because then $\widehat{v}$ is not minimal for $\ll$. Thus, $\mathsf{Pred}(S) = \mathsf{Pred}(\widehat{v})$ and $\widehat{v} \in S$. This contradicts v being incomparable with the vertices in S for $\ll$. Hence $\mathsf{Pred}(S) \not\subseteq \mathsf{Pred}(\widehat{a})$.

There exists $\widehat{s} \in \mathsf{Pred}(S) \setminus \mathsf{Pred}(\widehat{v})$. As $\leqslant_{\mathsf{left}}$ extends $\ll$, we have $\widehat{s} \leqslant_{\mathsf{left}} s_1$. Hence, we get $\widehat{s} \leqslant_{\mathsf{left}} \widehat{v} \leqslant_{\mathsf{left}} s_2$. But $\widehat{s} \ll s_2$ and $\widehat{s}$ is incomparable with $\widehat{v}$ and s_2. This is a contradiction with Lemma 15. □

The total order on a sibling class S is computed by $\mathsf{LRorder}(S, \mathsf{L}, \mathsf{R}, \mathcal{SC})$ in Line 20 where L and R are the sets of vertices in $(\mathsf{Current} \cup \mathsf{Treated}) \setminus \mathsf{Pred}(S)$ that should be before S in $\leqslant_{\mathsf{left}}$ and after it, respectively. We use these sets L and R to compute an order $\leqslant_S$ on S that is consistent with the total order we have on the processed vertices.

Lemma 24. *There exists a polynomial time algorithm* $\mathsf{LRorder}$*, that given a sibling class S, $\mathsf{L}, \mathsf{R} \subseteq V(G) \setminus S$ and a collections $\mathcal{SC} \subseteq 2^{V(G)}$ of size at most n^2, computes a total order $\leqslant_S$ on S that is continuous on every set in $\mathcal{SC}$ and such that*

- *the minimum element of $\leqslant_S$ belongs to every set in $\mathcal{SC}$ intersecting S and L, and*
- *the maximum element of $\leqslant_S$ belongs to every set in $\mathcal{SC}$ that intersects S and R.*

or the algorithm confirms that such an order does not exist.

Proof. Let S be a sibling class, $\mathcal{SC} \subseteq 2^{V(G)}$ and $\mathsf{L}, \mathsf{R} \subseteq V(G) \setminus S$. Observe that every total order on S is continuous on a set C which does not intersect S. Thus, we can assume in this proof that every set in $\mathcal{SC}$ intersects S. We define the binary matrix $\mathcal{M}$ whose columns represent the vertices in S in addition with four dummy vertices $\ell, \ell^\star, r, r^\star$ and whose rows represent the sets in $\{\mathsf{Left}, \mathsf{Right}, \mathsf{Middle}\} \cup \{C^\star \mid C \in \mathcal{SC}\}$ defined as follows. We have $\mathsf{Left} = \{\ell, \ell^\star\}$, $\mathsf{Right} = \{r, r^\star\}$, $\mathsf{Middle} = S \cup \{r, \ell\}$ and, for each $C \in \mathcal{SC}$, $C^\star$ is a superset of C that contains ℓ if $C \cap \mathsf{L} \neq \varnothing$ and $C^\star$ contains r if $C \cap \mathsf{R} \neq \varnothing$. Each entry of $\mathcal{M}$ is associated with a set X and a vertex v, this entry is 1 if $v \in X$ and 0 otherwise. Figure 5 illustrates the construction of the matrix $\mathcal{M}$.

$$
\begin{array}{l}
 \\
\mathsf{Left} \\
\mathsf{Right} \\
\mathsf{Middle} \\
C_1^\star \\
C_2^\star \\
C_3^\star
\end{array}
\begin{array}{c}
\begin{array}{cccccccc}
\ell & \ell^\star & s_1 & s_2 & s_3 & s_4 & r^\star & r
\end{array} \\
\begin{pmatrix}
1 & 1 & & & & & & \\
 & & & & & & 1 & 1 \\
 & 1 & 1 & 1 & 1 & 1 & 1 & \\
 & 1 & 1 & 1 & & & & \\
 & & & & 1 & 1 & 1 & \\
 & & & 1 & 1 & & &
\end{pmatrix}
\end{array}
$$

FIGURE 5. Example of the matrix $\mathcal{M}$ with $S = \{s_1, s_2, s_3, s_4\}$, $\mathcal{SC} = \{C_1, C_2, C_3\}$ where C_1 and C_2 intersect respectively L and R. For clarity, we only show the entries set to 1.

A $(0,1)$-matrix satisfies the *consecutive ones* property if there exists a column permutation such that the ones in each row of the resulting matrix are consecutive. We claim that $\mathcal{M}$ has the consecutive ones property if and only if there exists an order $\leqslant_S$ on the vertices in S that fulfills the conditions of this lemma.

($\Leftarrow$) Suppose that there exists an order $\leqslant_S$ on S that fulfills the conditions of this lemma. Let $\mathcal{M}'$ be the matrix obtained from $\mathcal{M}$ by removing the dummy vertices. As S is an antichain in $\ll$, it is clear that continuous on C really is the same as "consecutive": if $a \leqslant_S b \leqslant_S c$ and $a, c \in C$, then b is simply not allowed to not be in C. Thus, $\leqslant_S$ is a column permutation of $\mathcal{M}'$ with the consecutive ones property. Take π the column permutation of $\mathcal{M}$ obtained from $\leqslant_S$ such that $\ell, \ell^\star$ are the first two columns and $r^\star, r$ the two lasts. By construction, the ones in the rows $\mathsf{Left}, \mathsf{Middle}, \mathsf{Right}$ are consecutive. Take a $C \in \mathcal{SC}$ that intersects L. By assumption C intersects S and thus C contains the minimum element of $\leqslant_S$. Moreover, by construction $C^\star$ contains $\ell^\star$. Similarly, for each $C \in \mathcal{SC}$ intersecting R, $C^\star$ contains $r^\star$ and the maximum element of $\leqslant_S$. We deduce that π is a column permutation with the consecutive ones property.

($\Rightarrow$) Assume that there exists a permutation π on the columns of $\mathcal{M}$ with the consecutive ones property. This clearly means that the restriction $\leqslant_S$ of π to S is continuous on every set in $\mathcal{SC}$. What remains to show is the other property, about the minumum and maximum elements of $\leqslant_S$. The sets $\mathsf{Left}, \mathsf{Right}, \mathsf{Middle}$ guarantee that the first and last columns of π are associated with ℓ and r. As the reverse order of π has also the consecutive ones property, we can assume w.l.o.g. that the first column of π is $\ell^\star$ and the last $r^\star$. Consequently, the second and penultimate columns of π are respectively $\ell^\star$ and $r^\star$.

Let $C \in \mathcal{SC}$ that intersects L. By construction $C^\star$ contains $\ell^\star$ and by assumption $C^\star$ intersects S. Observe that C must contain the minimum element of $\leqslant_S$ as otherwise π would not satisfy the consecutive ones property. Symmetrically, each set $C \in \mathcal{SC}$ intersecting R contains the maximum element of $\leqslant_S$. Therefore $\leqslant_S$ is an order on S that fulfills the conditions of this lemma.

The last thing we must argue for is the existence of a polynomial-time algorithm that computes π on $\mathcal{M}$. Luckily, such algorithms have been long known; one can, for example, use the classical *PQ-tree* algorithm of Booth and Lueker [2], which runs in linear time in the number of ones in $\mathcal{M}$. For our purposes, this number is $O(n^3)$. □

The dual ordering. In order to know which vertices in $\mathsf{Current} \cup \mathsf{Treated}$ should appear before a sibling class in $\mathsf{Current}$ and which after it we use $\leqslant_{\mathsf{right}}$, the *dual ordering* of $\leqslant_{\mathsf{left}}$. If $(G, \mathcal{SC})$ is yes-instance then $\leqslant_{\mathsf{left}}$ represents the order on the left endpoints of processed vertices of a special interval model. In this case, the dual ordering $\leqslant_{\mathsf{right}}$ corresponds to the order on the right endpoints of the intervals.

Definition 25. *Given a linear extension $\leqslant_{\mathsf{left}}$ of $\ll$ on $S \subseteq V(G)$, we define the dual ordering $\leqslant_{\mathsf{right}}$ of $\leqslant_{\mathsf{left}}$ as the binary relation of S such that $a \leqslant_{\mathsf{right}} b$ if either:*

(1) $a \ll b$, or

(2) $b \leqslant_{\mathsf{left}} a$ and $b \not\ll a$.

For a sibling class S, $\mathsf{left\text{-}limit}(S)$ and $\mathsf{right\text{-}limit}(S)$ as defined in Lines 10 and 11 are the predecessors of S whose intervals start the latest and end the earliest, respectively. In other words, $\mathsf{segment}(\mathsf{Pred}(S))$ is equal to $I_{\mathsf{left\text{-}limit}(S)} \cap I_{\mathsf{right\text{-}limit}(S)}$, and this shows where the intervals of the vertices of S must lie. In Line 15 we use the $\mathsf{left\text{-}limit}(S)$'s and $\mathsf{right\text{-}limit}(S)$'s to define an order $\precsim$ on the sibling classes. We have $S_1 \precsim S_2$ if $\mathsf{segment}(\mathsf{Pred}(S_1))$ starts or ends before $\mathsf{segment}(\mathsf{Pred}(S_2))$. From all of this we know which vertices should appear before S and also after it.

The rest of the paper, apart from Conclusions, is a proof of the following theorem.

Theorem 26. *In polynomial time, Algorithm 3 computes a nice order for $(G, \mathcal{B})$ or confirms that such an order does not exist.*

Proof. It is sufficient to prove that it runs in polynomial time and that if $(G, \mathcal{B})$ is a yes-instance then it returns a nice order, as Line 25 guarantees that if it is a no-instance then the algorithm

Algorithm 3: Nice order.

```
Data: An interval graph $G$ and a collection $\mathcal{B}$ of special cliques.
Result: A nice order on $V(G)$ or "no".
/* This algorithm returns ''no'' if one call to Algorithm 2 or LRorder
   returns ''no'' or one of the checkings at Lines 7, 14 and 25 fails.      */
1  Let $\mathcal{SC}$ be the collection computed by Algorithm 2;
2  Let $M$ be the set of minimal elements in $V(G)$ w.r.t. $\ll$;
3  $\leqslant_{\mathsf{left}} \leftarrow \mathsf{LRorder}(M, \varnothing, \varnothing, \mathcal{SC})$;
4  $\mathsf{Treated} \leftarrow M$;
5  while $\mathsf{Treated} \neq V(G)$ do
6      Compute $\leqslant_{\mathsf{right}}$ the dual ordering of $\leqslant_{\mathsf{left}}$ as in Definition 25;
7      if $\leqslant_{\mathsf{right}}$ is not a total order then return no;
8      Let Current be the set of vertices $v \notin \mathsf{Treated}$ such that $\mathsf{Pred}(v) \subseteq \mathsf{Treated}$;
9      for every sibling class $S$ contained in Current do
10         Let $\mathsf{left\text{-}limit}(S)$ be the maximum element in $\mathsf{Pred}(S)$ for $\leqslant_{\mathsf{left}}$;
11         Let $\mathsf{right\text{-}limit}(S)$ be the maximum element in $\mathsf{Pred}(S)$ for $\leqslant_{\mathsf{right}}$;
12     end
13     Let $\precsim$ be the binary relation on the sibling classes contained in Current such that
         $A \precsim B$ if $\mathsf{right\text{-}limit}(A) \leqslant_{\mathsf{left}} \mathsf{right\text{-}limit}(B)$ and $\mathsf{left\text{-}limit}(B) \leqslant_{\mathsf{right}} \mathsf{left\text{-}limit}(A)$;
14     if $\precsim$ is not a total order then return no;
15     Let $S_1, S_2, \dots, S_t$ be the sibling classes contained in Current such that $S_1 \precsim \cdots \precsim S_t$;
16     for $i = t$ to $1$ do
17         $\mathsf{L}_{\mathsf{Treated}} \leftarrow \{v \in \mathsf{Treated} \setminus \mathsf{Pred}(S_i) \mid v \leqslant_{\mathsf{left}} \mathsf{left\text{-}limit}(S_i)\}$;
18         Let $\mathsf{L} = \mathsf{L}_{\mathsf{Treated}} \cup S_1 \cup S_2 \cup \cdots \cup S_{i-1}$;
19         Let $\mathsf{R} = (\mathsf{Current} \cup \mathsf{Treated}) \setminus (S_i \cup \mathsf{Pred}(S_i) \cup \mathsf{L})$;
           /* L contains the vertices in $(\mathsf{Treated} \cup \mathsf{Current}) \setminus \mathsf{Pred}(S_i)$ and that will
              be at the left of $S_i$, and R contains those that will be at the
              right of $S_i$.                                                   */
20         $\leqslant_{S_i} \leftarrow \mathsf{LRorder}(S_i \mathsf{L}, \mathsf{R}, \mathcal{SC})$;
21         $\leqslant_{\mathsf{left}} \leftarrow \mathsf{InsertAfter}(\leqslant_{\mathsf{left}}, \leqslant_{S_i}, \mathsf{left\text{-}limit}(S_i))$;
22     end
23     $\mathsf{Treated} \leftarrow \mathsf{Treated} \cup \mathsf{Current}$;
24 end
25 if $\leqslant_{\mathsf{left}}$ is not a nice order for $(G, \mathcal{B})$ then return no;
26 return $\leqslant_{\mathsf{left}}$;
```

returns no. In the remainder of the proof we therefore assume that $(G, \mathcal{B})$ is a yes-instance. We start by proving that the algorithm does not return no and that the following invariant holds at each iteration of the while loop.

Invariant. The order $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$ on Treated and it is continuous on every special clique in $\mathcal{SC}$.

If Algorithm 3 returns an order, then the invariant guarantees that the returned order is nice.

Claim 27. *The first call at* LRorder *returns an order and the invariant holds before the first iteration of the while loop.*

Proof. By Lemma 19, since we assume we have a yes-instance, at Line 2 Algorithm 2 will return a collection $\mathcal{SC}$. At Line 3, M is a sibling class so at Line 4 by Lemma 24 the call $\mathsf{LRorder}(M, \varnothing, \varnothing, \mathcal{SC})$ returns an order since every nice order $\leqslant_{\mathsf{left}}$ of G is continuous on every set in $\mathcal{SC}$, and subsequently the restriction of a nice order on M is also continuous on every set in $\mathcal{SC}$. Moreover $\leqslant_{\mathsf{left}}$ is obviously a linear extension of $\ll$ on M because M is an antichain in $(V(G), \ll)$. This concludes the proof of Claim 27.

$\square$

In order to prove that the algorithm returns an order, we have to prove that all checkings at Lines 7, 14 and 25 succeed and every call at $\mathsf{LRorder}$ at Line 20 returns an order.

We assume that the invariant holds at the end of the i^{th} iteration, i.e. the total order $\leqslant_{\mathsf{left}}$ computed at the i^{th} iteration is a linear extension of $\ll$ on $\mathsf{Treated}$ that is continuous on every special clique in $\mathcal{SC}$. If at the end of the i^{th} iteration we have $\mathsf{Treated} = V(G)$ then the algorithm have computed a linear extension of $\ll$ on $V(G)$ that is continuous on every special clique in $\mathcal{SC}$. So in this case, the algorithm returns a nice order and we are done. Now assume that $\mathsf{Treated} \neq V(G)$. This implies that the algorithm start a $i+1^{\text{th}}$ iteration. We will prove in the following that the invariant holds at the end of the $i+1^{\text{th}}$ iteration. For this, we use the variables $\mathsf{Treated}$, $\mathsf{Current}$ and $\leqslant_{\mathsf{left}}$ of the algorithm to denote the values of these variables as the algorithm starts the $i+1^{\text{st}}$ iteration. We start by proving that the checkings at Lines 7 and 14 succeed on the $i+1^{\text{st}}$ iteration.

Claim 28. *On the $i+1^{st}$ iteration the dual ordering $\leqslant_{\mathsf{right}}$ of $\leqslant_{\mathsf{left}}$ is a total order.*

Proof. First we prove that $\leqslant_{\mathsf{right}}$ is connexe, i.e. for every $a, b \in \mathsf{Treated}$, we have $a \leqslant_{\mathsf{right}} b$ or $b \leqslant_{\mathsf{right}} a$. If a and b are comparable for $\ll$, either $a \ll b$ or $b \ll a$ and thus $a \leqslant_{\mathsf{right}} b$ or $b \leqslant_{\mathsf{right}} a$ by definition of $\leqslant_{\mathsf{right}}$.

Now, suppose that a and b are incomparable for $\ll$. Since $\leqslant_{\mathsf{left}}$ is a total order, we have either $a \leqslant_{\mathsf{left}} b$ or $b \leqslant_{\mathsf{left}} a$. If $a \leqslant_{\mathsf{left}}$, then we have $b \leqslant_{\mathsf{right}} a$ since we suppose $b \not\ll a$. If $b \leqslant_{\mathsf{left}} a$, then we have $a \leqslant_{\mathsf{right}} b$ as $a \not\ll b$. Thus, $\leqslant_{\mathsf{right}}$ is connexe.

Now we prove that $\leqslant_{\mathsf{right}}$ is antisymmetric, i.e. if $a \leqslant_{\mathsf{right}} b$ and $b \leqslant_{\mathsf{right}} a$, then $a = b$. Let a and b such that $a \leqslant_{\mathsf{right}} b$ and $a \neq b$. We prove that we can not have $b \leqslant_{\mathsf{right}} a$, that is we can not have $b \ll a$ or ($a \not\ll b$ and $a \leqslant_{\mathsf{left}} b$).

Since $a \leqslant_{\mathsf{right}} b$, we have either $a \ll b$ or ($b \not\ll a$ and $b \leqslant_{\mathsf{left}} a$). If $a \ll b$, then $b \not\ll a$ because $\ll$ as a partial order is antisymmetric. Thus, we can not have $b \leqslant_{\mathsf{right}} a$ since $b \not\ll a$ and $a \ll b$.

Now, suppose that $b \not\ll a$ and $b \leqslant_{\mathsf{left}} a$. Since $b \leqslant_{\mathsf{left}} a$ and $a \neq b$, we know that $a \not\ll b$ since $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$ on $\mathsf{Treated}$. Moreover, we can not have $a \leqslant_{\mathsf{left}} b$ because $\leqslant_{\mathsf{left}}$ is antisymmetric. Thus we can not have $b \leqslant_{\mathsf{right}} a$. We conclude that $\leqslant_{\mathsf{right}}$ is antisymmetric.

Now we prove the transitivity of $\leqslant_{\mathsf{right}}$, i.e. if $a \leqslant_{\mathsf{right}} b$ and $b \leqslant_{\mathsf{right}} c$, then $a \leqslant_{\mathsf{right}} c$. By definition, as $a \leqslant_{\mathsf{right}} b$ we have either (1) $a \ll b$ or (2) $b \not\ll a$ and $b \leqslant_{\mathsf{left}} a$. Symmetrically, $b \leqslant_{\mathsf{right}} c$ implies that we have either (3) $b \ll c$ or (4) $c \not\ll b$ and $c \leqslant_{\mathsf{left}} b$.

If we have (1) and (3), then by transitivity of $\ll$, we have $a \ll c$. By definition of $\leqslant_{\mathsf{right}}$, if $a \ll c$, then $a \leqslant_{\mathsf{right}} c$. In the following, we assume that $a \not\ll c$.

Now, if we have (1) and (4), then $a \ll b$, $c \not\ll b$ and $c \leqslant_{\mathsf{left}} b$. Since $a \ll b$ and $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$ on $\mathsf{Treated}$, we deduce that $a \leqslant_{\mathsf{left}} b$. As $a \ll b$, $a \not\ll c$ and $c \not\ll b$, we can not have $a \leqslant_{\mathsf{left}} c \leqslant_{\mathsf{left}} b$. Otherwise, $\leqslant_{\mathsf{left}}$ would not be continuous on a special clique by Lemma 15. Thus, we have $c \leqslant_{\mathsf{left}} a$. From $a \ll b$ and $c \not\ll b$, we deduce that $c \not\ll a$ and we conclude that $a \leqslant_{\mathsf{right}} c$.

If we have (2) and (3), then $b \leqslant_{\mathsf{left}} a$, $b \not\ll a$ and $b \ll c$. As $b \not\ll a$ and $b \ll c$, we must have $c \not\ll a$. Furthermore, we assume that $a \not\ll c$, so from Lemma 15, we deduce that we can not have $b \leqslant_{\mathsf{left}} a \leqslant_{\mathsf{left}} c$. Thus, we have $b \leqslant_{\mathsf{left}} c \leqslant_{\mathsf{left}} a$. By definition, we have $a \leqslant_{\mathsf{right}} c$ since $c \not\ll a$ and $c \leqslant_{\mathsf{left}} a$.

Finally, if (2) and (4), then we have $c \leqslant_{\mathsf{left}} b \leqslant_{\mathsf{left}} a$, $b \not\ll a$ and $c \not\ll b$. Since $\leqslant_{\mathsf{left}}$ is continuous on every special clique, we can not have $c \ll a$ thanks to Lemma 15. Thus, we have $c \not\ll a$ and $c \leqslant_{\mathsf{left}} a$ and we deduce that $a \leqslant_{\mathsf{right}} c$. This concludes the proof of Claim 28. $\square$

Claim 29. *On the $i+1^{st}$ iteration the relation $\precsim$ is a total order.*

Proof. In this claim, for a sibling class X contained in $\mathsf{Current}$, we write r_X and ℓ_X to denote respectively $\mathsf{right\text{-}limit}(X)$ and $\mathsf{left\text{-}limit}(X)$ as defined at Lines 11 and 10. The following fact proves the antisymmetry of $\precsim$, i.e. for every sibling classes A, B, if $A \precsim B$ and $B \precsim A$, then $A = B$.

Fact 30. *For every pair of different sibling classes A, B contained in current, we have $r_A \neq r_B$ or $\ell_A \neq \ell_B$.*

Proof of fact. Assume towards a contradiction that there exists two different sibling classes A, B such that $r_A = r_B$ and $\ell_A = \ell_B$. Since $A \neq B$, by definition, we have $\mathsf{Pred}(A) \neq \mathsf{Pred}(B)$. So either $\mathsf{Pred}(A) \not\subseteq \mathsf{Pred}(B)$ or $\mathsf{Pred}(B) \not\subseteq \mathsf{Pred}(A)$. Suppose w.l.o.g. that $\mathsf{Pred}(A) \not\subseteq \mathsf{Pred}(B)$. Let $\widehat{a} \in \mathsf{Pred}(A) \setminus \mathsf{Pred}(B)$. By definition of r_A, we have $\widehat{a} \leqslant_{\mathsf{right}} r_A$. Moreover, we have $\widehat{a} \not\ll r_A$ since $r_A \in \mathsf{Pred}(B)$ and $\widehat{a} \notin \mathsf{Pred}(B)$ and by definition of $\leqslant_{\mathsf{right}}$, we deduce that $r_A \leqslant_{\mathsf{left}} \widehat{a}$. By definition of ℓ_A, we have $\widehat{a} \leqslant_{\mathsf{left}} \ell_A$. Thus, we have $r_A \leqslant_{\mathsf{left}} \widehat{a} \leqslant_{\mathsf{left}} \ell_A$. Since $\widehat{a} \notin \mathsf{Pred}(B)$, there exists $C_B \in \mathcal{B}$ such that $B \cap C_B \neq \varnothing$ and $\widehat{a} \notin C_B$. By assumption $r_A, \ell_A \in \mathsf{Pred}(B)$ and thus $\ell_A, r_A \in C_B$. Hence $(r_A, \widehat{a}, \ell_A)$ is a troubling triple for C_B. This is a contradiction because $r, \widehat{a}, \ell \in \mathsf{Treated}$. This concludes the proof of Fact 30. ◆

Observe that if $A \precsim B$ and $B \precsim A$ for two sibling classes, then by definition, we have $\ell_A = \ell_B$ and $r_A = r_B$. By Fact 30, this implies $A = B$ and thus the relation $\precsim$ is antisymmetric.

In the following, we write $a <_{\mathsf{left}} b$ (resp. $a <_{\mathsf{right}} b$) when $a \neq b$ and $a \leqslant_{\mathsf{left}} b$ (resp. $a \leqslant_{\mathsf{right}} b$). We use the following fact which implies the connexity of $\precsim$, i.e. for every sibling classes A, B, we have $A \precsim B$ or $B \precsim A$.

Fact 31. *For every pair of sibling classes A, B contained in current, we have either*

- *$\ell_A \leqslant_{\mathsf{left}} \ell_B$ and $r_B \leqslant_{\mathsf{right}} r_A$, or*
- *$\ell_B \leqslant_{\mathsf{left}} \ell_A$ and $r_A \leqslant_{\mathsf{right}} r_B$.*

Proof of fact. Assume towards a contradiction, that there exist two sibling classes A and B contained in $\mathsf{Current}$ such that $\ell_A <_{\mathsf{left}} \ell_B$ and $r_A <_{\mathsf{right}} r_B$. As A and B are subsets of $\mathsf{Current}$, there exists $\widehat{a} \in \mathsf{Pred}(A)$ such that the height of $\widehat{a}$ in $(V(G), \ll)$ is at least the maximum between the heights of ℓ_B and r_B. Consequently, we have $\widehat{a} \not\ll \ell_B$ and $\widehat{a} \not\ll r_B$. By definition of r_A, we have $\widehat{a} \leqslant_{\mathsf{right}} r_A$. Consequently, we have $\widehat{a} <_{\mathsf{right}} r_B$. From the definition of $\leqslant_{\mathsf{right}}$, it follows that $r_B \leqslant_{\mathsf{left}} \widehat{a}$ because $\widehat{a} \not\ll r_B$. By definition of ℓ_A, we have $\widehat{a} \leqslant_{\mathsf{left}} \ell_A$. We conclude that $r_B \leqslant_{\mathsf{left}} \widehat{a} \leqslant_{\mathsf{left}} \ell_B$.

Let $a \in A$ and $b \in B$. Since $a, b \in \mathsf{Current}$, we have $a \not\ll b$ and thus there exists $C_b \in \mathcal{B}$ such that $a \notin C_b$ and $b \in C_b$. As ℓ_B, r_B belong to $\mathsf{Pred}(B)$, they also belong to C_b. We deduce that $\ell_B, r_B \notin \mathsf{Pred}(A)$ from the definitions of ℓ_A and r_A. For example, we have $r_A <_{\mathsf{right}} r_B$ and by definition r_A is the maximum in $\mathsf{Pred}(A)$ for $\leqslant_{\mathsf{right}}$ so $r_B \notin \mathsf{Pred}(A)$. Since $r_B \not\ll a$ and $\widehat{a} \ll a$, we deduce that $r_B \not\ll \widehat{a}$. Thus, by Lemma 19, we have $C_b(a, r_B) \in \mathcal{SC}$. From the definition of $C_b(a, r_B)$, we have $\ell_B, r_B \in C(a, r_B)$ because $\ell_B, r_B \notin \mathsf{Pred}(A)$. Moreover, $\widehat{a} \notin C_b(a, r_B)$ as $\widehat{a} \in \mathsf{Pred}(a) \setminus \mathsf{Pred}(r_B)$. Hence, $(r_B, \widehat{a}, \ell_B)$ is a troubling triple for C_b. This is a contradiction since $r_B, \widehat{a}, \ell_B \in \mathsf{Treated}$. This concludes the proof of Fact 31. ◆

It remains to prove the transitivity. Let A, B, C be three sibling classes contained in $\mathsf{Current}$ and suppose that $A \precsim B$ and $B \precsim C$. We need to prove that $A \precsim C$. Since $A \precsim B$, by definition we have $\ell_A \leqslant_{\mathsf{left}} \ell_B$ and $r_B \leqslant_{\mathsf{right}} r_A$. Similarly, as $B \precsim C$, we have $\ell_B \leqslant_{\mathsf{left}} \ell_C$ and $r_C \leqslant_{\mathsf{right}} r_B$. Since $\leqslant_{\mathsf{left}}$ and $\leqslant_{\mathsf{right}}$ are total orders, they are transitive, we conclude that $\ell_A \leqslant_{\mathsf{left}} C$ and $r_C \leqslant_{\mathsf{right}} r_A$. It follows that $A \precsim C$ and thus $\precsim$ is transitive. This concludes the proof of Claim 29. □

So by Claims 28 and 29, on the $i + 1^{\text{st}}$ iteration the algorithm enters the second for loop at Line 16. We say that a partial order $\leqslant_1$ extends partial order $\leqslant_2$ if $a \leqslant_2 b$ implies $a \leqslant_1 b$. Observe that once our algorithm computes $\precsim$, we partially know how we will extend $\leqslant_{\mathsf{left}}$. Let us define rigorously this partial order $\leqslant^{\star}_{\mathsf{left}}$ extending $\leqslant_{\mathsf{left}}$ on $\mathsf{Current} \cup \mathsf{Treated}$. For every distinct pair of sibling classes A, B contained in $\mathsf{Current}$ such that $A \precsim B$, we have $a \leqslant^{\star}_{\mathsf{left}} b$ for every $a \in A$ and $b \in B$. Moreover, for every $a \in A$ and $v \in \mathsf{Treated}$, we have $v \leqslant^{\star}_{\mathsf{left}} a$ if $v \leqslant^{\star}_{\mathsf{left}} \mathsf{left\text{-}limit}(A)$ and $a \leqslant_{\mathsf{left}} v$ otherwise. Observe that the only pairs of incomparable vertices for $\leqslant^{\star}_{\mathsf{left}}$ are the pairs of vertices that belong to the same sibling class of $\mathsf{Current}$. In the following claims, we will prove that $\leqslant^\star_{\mathsf{left}}$ is an extension of $\ll$ and it is continuous on every special clique in $\mathcal{SC}$. We will use these claims to prove that every call at $\mathsf{LRorder}$ at Line 20 returns an order.

Fact 32. *The partial order $\leqslant^\star_{\mathsf{left}}$ is an extension of $\ll$.*

Proof of fact. Observe that, by definition, the restriction of $\leqslant^\star_{\mathsf{left}}$ on $\mathsf{Treated}$ is exactly $\leqslant_{\mathsf{left}}$. By assumption, the invariant holds when the algorithm starts the i^{th} iteration and $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$ on $\mathsf{Treated}$. Moreover, the vertices in $\mathsf{Current}$ are pairwise incomparable w.r.t. $\ll$. Thus, it is sufficient to prove that for every $v \in \mathsf{Current}$ and $w \in \mathsf{Pred}(v)$ we have $w \leqslant^\star_{\mathsf{left}} v$. By definition of $\leqslant^\star_{\mathsf{left}}$, we have $\mathsf{left\text{-}limit}(S) \leqslant^\star_{\mathsf{left}} v$ where S is the sibling class that contains v. By definition, $\mathsf{left\text{-}limit}$ is the maximum element in $\mathsf{Pred}(S)$ for $\leqslant_{\mathsf{left}}$. Since v belongs to S, we have $\mathsf{Pred}(S) = \mathsf{Pred}(v)$. Consequently, for every $w \in \mathsf{Pred}(v)$, we have $w \leqslant^\star_{\mathsf{left}} \mathsf{left\text{-}limit}(S) \leqslant^\star_{\mathsf{left}} v$. Hence, $\leqslant^\star_{\mathsf{left}}$ is a linear extension of $\ll$ on $\mathsf{Treated} \cup \mathsf{Current}$. This ends the proof of Fact 32. ♦

Claim 33. *The partial order $\leqslant^\star_{\mathsf{left}}$ is continuous on every special clique in $\mathcal{SC}$.*

Proof. Assume towards a contradiction that $\leqslant^\star_{\mathsf{left}}$ is not continuous on every special clique in $\mathcal{SC}$. Thus, there exists a troubling triple (a, b, c) for some $C \in \mathcal{SC}$. We choose C and a troubling triple (a, b, c) such that $|\{a, b, c\} \cap \mathsf{Current}|$ is minimum and a, b, c are minimal for $\ll$. Since $b \notin C$ and $c \in C$, Condition (4) of Lemma 19 implies that $b \not\ll c$.

By definition, the restriction of $\leqslant^\star_{\mathsf{left}}$ on $\mathsf{Treated}$ is $\leqslant_{\mathsf{left}}$. Since we assume that the invariant holds at the end of i^{th} iteration, we can not have $\{a, b, c\} \not\subseteq \mathsf{Treated}$. We start with the following facts.

Fact 34. *We have either $\mathsf{Pred}(a) \cap \mathsf{Pred}(b) \subseteq \mathsf{Pred}(c)$ or $\mathsf{Pred}(b) \cap \mathsf{Pred}(c) \subseteq \mathsf{Pred}(a)$.*

Proof of fact. Assume toward a contradiction that there exist $\widehat{a} \in \mathsf{Pred}(a) \cap \mathsf{Pred}(b)$ and $\widehat{c} \in \mathsf{Pred}(b) \cap \mathsf{Pred}(c)$ such that $\widehat{a} \notin \mathsf{Pred}(c)$ and $\widehat{c} \notin \mathsf{Pred}(a)$.

This implies that a and c are incomparable w.r.t. $\ll$ because we have $\mathsf{Pred}(a) \not\subseteq \mathsf{Pred}(c)$ and $\mathsf{Pred}(c) \not\subseteq \mathsf{Pred}(a)$. Moreover, we can not have $\widehat{a} \ll \widehat{c}$ because $\widehat{a} \not\ll c$ and $\widehat{c} \ll c$. Symmetrically, we can not have $\widehat{c} \ll \widehat{a}$. By assumption, we have $a \not\ll b$ and we already proved that $b \not\ll c$. We conclude that the poset $(V(G), \ll)$ contains the pattern presented in Figure 6. We will show that this pattern implies that G does not admit a nice order. This will contradict the assumption that $(G, \mathcal{B})$ is a yes-instance.

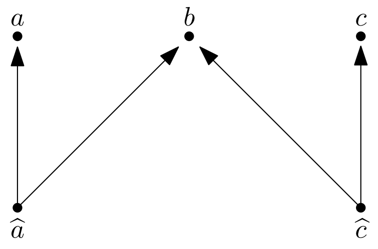

Figure 6. Forbidden pattern used to prove Fact 34.

Let $\leqslant_{\mathsf{nice}}$ be a nice order of G for $\mathcal{SC}$. We assume w.l.o.g. that $\widehat{a} \leqslant_{\mathsf{nice}} \widehat{c}$ because the other case is symmetrical. Since $\leqslant_{\mathsf{nice}}$ is a linear extension of $\ll$, we know that $\widehat{a} \leqslant_{\mathsf{nice}} a$, $\widehat{c} \leqslant_{\mathsf{nice}} b$ and $\widehat{c} \leqslant_{\mathsf{nice}} c$. As $\widehat{a} \ll a$ and $\widehat{c} \not\ll a$, we can not have $\widehat{a} \leqslant_{\mathsf{nice}} \widehat{c} \leqslant_{\mathsf{nice}} a$ as this would contradict Lemma 15. So we have $\widehat{a} \leqslant_{\mathsf{nice}} a \leqslant_{\mathsf{nice}} \widehat{c}$. Similarly, we can not have $c \leqslant_{\mathsf{nice}} b$ because we would have $\widehat{a} \leqslant_{\mathsf{nice}} c \leqslant_{\mathsf{nice}} b$. This contradicts Lemma 15 since $\widehat{a} \ll b$ and $\widehat{a} \not\ll c$.

We conclude that we have $\widehat{a} \leqslant_{\mathsf{nice}} a \leqslant_{\mathsf{nice}} \widehat{c} \leqslant_{\mathsf{nice}} b \leqslant_{\mathsf{nice}} c$. But then, $\leqslant_{\mathsf{nice}}$ is not continuous on C as $a \leqslant_{\mathsf{nice}} b \leqslant_{\mathsf{nice}} c$. This contradicts the assumption that $\leqslant_{\mathsf{nice}}$ is a nice order. This concludes the proof of Fact 34. ♦

The following fact is a consequence of Fact 34.

Fact 35. *There exists $C^\star \in \mathcal{SC}$ such that $a, c \in C^\star$ and $C^\star \cap (\mathsf{Pred}(b) \setminus \mathsf{Pred}(\{a,c\})) = \varnothing$.*

Proof of fact. By Fact 34, we have either $\mathsf{Pred}(a) \cap \mathsf{Pred}(b) \subseteq \mathsf{Pred}(c)$ or $\mathsf{Pred}(b) \cap \mathsf{Pred}(c) \subseteq \mathsf{Pred}(a)$. We suppose first that $\mathsf{Pred}(a) \cap \mathsf{Pred}(b) \subseteq \mathsf{Pred}(c)$ and take $C^\star = C(b,c)$. Condition (3) of Lemma 19 implies that $C(b,c) \in \mathcal{SC}$. By definition of $C(b,c)$, we have $c \in C(b,c)$ and $C(b,c) \cap (\mathsf{Pred}(b) \setminus \mathsf{Pred}(c)) = \varnothing$. As we suppose that $\mathsf{Pred}(a) \cap \mathsf{Pred}(b) \subseteq \mathsf{Pred}(c)$, we have $\mathsf{Pred}(b) \setminus \mathsf{Pred}(c) = \mathsf{Pred}(b) \setminus \mathsf{Pred}(\{a,c\})$. So we have $C(b,c) \cap (\mathsf{Pred}(b) \setminus \mathsf{Pred}(\{a,c\})) = \varnothing$.

Moreover, we know that $a \not\ll b$ so $a \in C'(b,c)$. We deduce that $\mathsf{Pred}(a) \subseteq C'(b,c)$ because by assumption $\mathsf{Pred}(a) \cap \mathsf{Pred}(b) \subseteq \mathsf{Pred}(c)$ and by construction $\mathsf{Pred}(c) \subseteq C'(b,c)$. Thus, we have $a \in C(b,c)$ and $C(b,c)$ satisfies all the required properties.

Symmetrically, if $\mathsf{Pred}(b) \cap \mathsf{Pred}(c) \subseteq \mathsf{Pred}(a)$, then we can take $C^\star = C(b,a)$ and prove that $a, c \in C(b,a)$ and $C(b,a) \cap (\mathsf{Pred}(b) \setminus \mathsf{Pred}(\{a,c\})) = \varnothing$. This concludes the proof of Fact 35. ⧫

Fact 36. *We have $\mathsf{Pred}(a) \subseteq \mathsf{Pred}(b)$ and for every $\widehat{c} \in \mathsf{Pred}(c)$, we have $\widehat{c} \leqslant^\star_{\mathsf{left}} b$.*

Proof of fact. Assume towards a contradiction that there exists $\widehat{a} \in \mathsf{Pred}(a) \setminus \mathsf{Pred}(b)$. So we have $\widehat{a} \not\ll b$. As $\widehat{a} \ll a$, we deduce that $\widehat{a} \leqslant^\star_{\mathsf{left}} a$ from Fact 32. Thus, we have $\widehat{a} \leqslant^\star_{\mathsf{left}} b$. Moreover, we know that $\widehat{a} \in C$ since $\mathsf{Pred}(a) \subseteq C$ thanks to Lemma 19. This contradicts the choice of a because $(\widehat{a}, b, c)$ is a troubling triple for C and $\widehat{a} \ll a$. We conclude that we have $\mathsf{Pred}(a) \subseteq \mathsf{Pred}(b)$.

Now assume towards a contradiction that there exists $\widehat{c} \in \mathsf{Pred}(c)$ such that $b \leqslant^\star_{\mathsf{left}} \widehat{c}$. From Condition (4) of Lemma 19, we know that $\widehat{c} \in C$ since $c \in C$. This contradicts the choice of c because $(a, b, \widehat{c})$ is a troubling triple for C. This concludes the proof of Fact 36. ⧫

For $x \in \{a,b,c\}$, let S_x be the sibling class containing x, $r_x = \mathsf{left\text{-}limit}(S_x)$ and $\ell_x = \mathsf{right\text{-}limit}(S_x)$. Observe that r_x and ℓ_x are the maximum elements in $\mathsf{Pred}(x)$ for respectively $\leqslant_{\mathsf{right}}$ and $\leqslant_{\mathsf{left}}$ (and also in $\leqslant^\star_{\mathsf{left}}$ by definition).

Case b is not in $\mathsf{Current}$. Suppose that $a \in \mathsf{Current}$. Observe that we can not have $\mathsf{Pred}(a) \subseteq \mathsf{Pred}(b)$ as this would contradict the assumption that $b \in \mathsf{Treated}$. Indeed, if $\mathsf{Pred}(a) \subseteq \mathsf{Pred}(b)$, then the height of b in $(V(G), \ll)$ can not be lower than the height of a. Thus, we have $\mathsf{Pred}(a) \nsubseteq \mathsf{Pred}(b)$ and this contradicts Fact 36. Hence, we deduce that $a \in \mathsf{Treated}$.

Now, suppose that $c \in \mathsf{Current}$. By definition of $\leqslant^\star_{\mathsf{left}}$, we have $b \leqslant^\star_{\mathsf{left}} \ell_c$ as $b \leqslant^\star_{\mathsf{left}} c$. This contradicts Fact 36 because $b \leqslant^\star_{\mathsf{left}} \ell_c$ and $\ell_c \ll c$.

So a and c can not belong to $\mathsf{Current}$ and consequently a, b, c belong to $\mathsf{Treated}$. But this is not possible, we already argued that we can not have $\{a,b,c\} \subseteq \mathsf{Treated}$. So we conclude that b must be in $\mathsf{Current}$.

Case a and c are not in $\mathsf{Current}$. Since $b \in \mathsf{Current}$ and $a, c \notin \mathsf{Current}$, we can not have $\mathsf{Pred}(b) \subseteq \mathsf{Pred}(\{a,c\})$ because otherwise, the height of a or c in $(V(G), \ll)$ would be at least the height of b. Let $\widehat{b} \in \mathsf{Pred}(b) \setminus \mathsf{Pred}(\{a,c\})$. Observe that we have $a \leqslant^\star_{\mathsf{left}} \ell_b \leqslant^\star_{\mathsf{left}} c$ from the definition of $\leqslant^\star_{\mathsf{left}}$. By definition of ℓ_b, we have $\widehat{b} \leqslant^\star_{\mathsf{left}} \ell_b$ and thus $\widehat{b} \leqslant^\star_{\mathsf{left}} c$.

Suppose that $\widehat{b} \leqslant^\star_{\mathsf{left}} a$. So we have $\widehat{b} \leqslant^\star_{\mathsf{left}} a \leqslant^\star_{\mathsf{left}} \ell_b$. As (a,b,c) is a troubling triple, we have $a \not\ll b$, thus there exists $C_b \in \mathcal{B}$ such that $b \in C_b$ and $a \notin C_b$. Since $\widehat{b}, \ell_b \in \mathsf{Pred}(b)$, we know that $\widehat{b}, \ell_b \in C_b$. We deduce that $(\widehat{b}, a, \ell_b)$ is a troubling triple for C_b because $\widehat{b} \leqslant^\star_{\mathsf{left}} a \leqslant^\star_{\mathsf{left}} \ell_b$ and $\widehat{b} \not\ll a$. Since $\widehat{b}, a, \ell_b \in \mathsf{Treated}$, this is a contradiction.

Now, assume that $a \leqslant^\star_{\mathsf{left}} \widehat{b}$. So we have $a \leqslant^\star_{\mathsf{left}} \widehat{b} \leqslant^\star_{\mathsf{left}} c$. Since $a \not\ll b$ by assumption, we have $a \not\ll \widehat{b}$. Furthermore, as $\leqslant^\star_{\mathsf{left}}$ is an extension of $\ll$ (Fact 32), we know that $c \not\ll \widehat{b}$. Hence, $\widehat{b}$ belongs to $\mathsf{Pred}(b) \setminus \mathsf{Pred}(\{a,c\})$. By Fact 35, there exists $C^\star$ such that $a, c \in C^\star$ and $\widehat{b} \notin C^\star$. Thus, $(a, \widehat{b}, c)$ is a troubling triple for $C^\star$. This is a contradiction since $a, \widehat{b}, c \in \mathsf{Treated}$. We conclude that at least one vertex among a and c belong to $\mathsf{Current}$.

To continue, we need the following fact.

Fact 37. *If a belongs to* $\mathsf{Current}$*, then $r_a = r_b$, $\ell_a \neq \ell_b$ and $\ell_b \not\ll a$. If c belongs to* $\mathsf{Current}$*, then $r_b \neq r_c$, $\ell_b = \ell_c$ and $r_b \not\ll c$.*

Proof of fact. Suppose that a belongs to $\mathsf{Current}$. Since $a \leqslant^\star_{\mathsf{left}} b$, we have $S_a \precsim S_b$ and thus $\ell_a \leqslant^\star_{\mathsf{left}} \ell_b$ and $r_b \leqslant_{\mathsf{right}} r_a$. By Fact 36, we know that $\mathsf{Pred}(a) \subseteq \mathsf{Pred}(b)$. So $r_a \ll b$ and from the definition of r_b, we deduce that $r_a = r_b$ because $r_b \leqslant_{\mathsf{right}} r_a$. As $a \leqslant^\star_{\mathsf{left}}$, a and b can not be in the same sibling class, i.e. $S_a \neq S_b$ and as a consequence we have that $\ell_a \neq \ell_b$ from Fact 30. Furthermore, since $\ell_a \leqslant^\star_{\mathsf{left}} \ell_b$ and $\ell_a \neq \ell_b$, we have $\ell_b \not\ll a$ from the definition of ℓ_a.

Now suppose that c belongs to $\mathsf{Current}$. Since $b \leqslant^\star_{\mathsf{left}} c$, we know that $S_b \precsim S_c$ and thus $\ell_b \leqslant^\star_{\mathsf{left}} \ell_c$ and $r_c \leqslant_{\mathsf{right}} r_b$. We can not have $\ell_b \neq \ell_c$ because we would have $b \leqslant^\star_{\mathsf{left}} c$ and that contradicts Fact 36. Thus, we have $\ell_b = \ell_c$. As $b \leqslant_l ef^\star c$, b and c are not in the same sibling class, thus by Fact 30, we have $r_b \neq r_c$. This concludes the proof of Fact 37. ⧫

Case a, b, c belong to $\mathsf{Current}$. Fact 37 implies that (1) $r_a = r_b \neq r_c$, (2) $\ell_a \neq \ell_b = \ell_c$, (3) $\ell_b \not\ll a$ and (4) $r_b \not\ll c$. Properties (2) and (3) imply that $\mathsf{Pred}(b) \cap \mathsf{Pred}(c)$ is not included in $\mathsf{Pred}(a)$. Symmetrically, Properties (1) and (4) means that $\mathsf{Pred}(a) \cap \mathsf{Pred}(b)$ is not included in $\mathsf{Pred}(c)$. Hence, we have a contradiction with Fact 34. We conclude that exactly one among a and c belong to $\mathsf{Current}$.

Case $a, b \in \mathsf{Current}$ and $c \in \mathsf{Treated}$. Due to Fact 37, we have $r_a = r_b$, $\ell_a \neq \ell_b$ and $\ell_b \not\ll a$. Observe that $r_a \in \mathsf{Pred}(a) \cap \mathsf{Pred}(b)$ and $\ell_b \in \mathsf{Pred}(b) \setminus \mathsf{Pred}(a)$.

We claim that $r_a \not\ll c$. Suppose towards a contradiction that $r_a \ll c$. Since $c \in \mathsf{Treated}$ and $b \in \mathsf{Current}$, there exists $\widehat{b} \in \mathsf{Pred}(b) \setminus \mathsf{Pred}(c)$. As $r_a = r_b$, we know that $\widehat{b} \leqslant_{\mathsf{right}} r_a$ by definition of r_b. Since $\widehat{b} \not\ll c$ and $r_a \ll c$, we deduce that $\widehat{b} \not\ll r_a$. From the definition of $\leqslant_{\mathsf{right}}$, we deduce that $r_a \leqslant^\star_{\mathsf{left}} \widehat{b}$. By the definition of ℓ_b, we have $r_a \leqslant^\star_{\mathsf{left}} \widehat{b} \leqslant^\star_{\mathsf{left}} \ell_b$ since $r_a, \widehat{b} \in \mathsf{Pred}(b)$. Moreover, as $b \leqslant^\star_{\mathsf{left}} c$, we deduce that $\ell_b \leqslant^\star_{\mathsf{left}} c$ and thus

$$r_a \leqslant^\star_{\mathsf{left}} \widehat{b} \leqslant^\star_{\mathsf{left}} c.$$

Due to $\widehat{b} \not\ll c$, there exists $C_c \in \mathcal{B}$ such that $c \in C_c$ and $\widehat{b} \notin C_c$. We know that $r_a \in C_c$ because we suppose that $r_a \ll c$. This is a contradiction as $\leqslant^\star_{\mathsf{left}}$ is not continuous on C_c and $r_a, \widehat{b}, c \in \mathsf{Treated}$. We conclude that $r_a \not\ll c$.

So $\mathsf{Pred}(a) \cap \mathsf{Pred}(b) \not\subseteq \mathsf{Pred}(c)$, by Fact 34, we deduce that $\mathsf{Pred}(b) \cap \mathsf{Pred}(c) \subseteq \mathsf{Pred}(a)$. Thus, we have $\ell_b \not\ll c$ because $\ell_b \not\ll a$. Hence, we conclude that $\ell_b \in \mathsf{Pred}(b) \setminus \mathsf{Pred}(\{a, c\})$. By Fact 35, there exists $C^\star$ such that $r_a, c \in C^\star$ and $\ell_b \notin C^\star$. From the definition of ℓ_b, we have $r_a = r_b \leqslant^\star_{\mathsf{left}} \ell_b$. Moreover, by definition of $\leqslant^\star_{\mathsf{left}}$, we have $\ell_b \leqslant^\star_{\mathsf{left}} c$ because $b \leqslant^\star_{\mathsf{left}} c$. Hence, we have $r_a \leqslant^\star_{\mathsf{left}} \ell_b \leqslant^\star_{\mathsf{left}} c$ and consequently $\leqslant^\star_{\mathsf{left}}$ is not continuous on $C^\star$. This is a contradiction because r_a, ℓ_b and c belong to $\mathsf{Treated}$.

Case $b, c \in \mathsf{Current}$ and $a \in \mathsf{Treated}$. We know from Fact 37 that $r_b \neq r_c$, $\ell_b = \ell_c$ and $r_b \not\ll c$. By definition of $\leqslant^\star_{\mathsf{left}}$, we have $a \leqslant^\star_{\mathsf{left}} b$, we have $a \leqslant^\star_{\mathsf{left}} \ell_b$. Since $\leqslant^\star_{\mathsf{left}}$ is an extension of $\ll$ (Fact 32), we deduce that $\ell_b \not\ll a$. Thus, $\mathsf{Pred}(b) \cap \mathsf{Pred}(c) \not\subseteq \mathsf{Pred}(a)$ and by Fact 34, we have $\mathsf{Pred}(a) \cap \mathsf{Pred}(b) \subseteq \mathsf{Pred}(c)$. It follows that $r_b \not\ll a$ since $r_b \not\ll c$.

Suppose that $r_b \leqslant^\star_{\mathsf{left}} a$. By assumption $a \not\ll b$ and thus there exists $C_b \in \mathcal{B}$ such that $a \notin C_b$ and $b \in C_b$. Since r_b belongs to $\mathsf{Pred}(b)$, it also belongs to C_b. Consequently, $\leqslant^\star_{\mathsf{left}}$ is not continuous on C_b because $r_b \leqslant^\star_{\mathsf{left}} a \leqslant^\star_{\mathsf{left}} b$ and $r_a \not\ll a$. This is a contradiction with the choice of C and a, b, c because $r_b, a \in \mathsf{Treated}$ and thus $|\{a, b, c\} \cap \mathsf{Treated}|$ is not minimum.

Now, suppose that $a \leqslant^\star_{\mathsf{left}} r_b$. Symmetrically to the previous case, we deduce that $r_b \leqslant^\star_{\mathsf{left}} \ell_c$ and that there exists $C^\star$ such that $a, \ell_c \in C^\star$ and $r_b \notin C^\star$. Consequently, $\leqslant^\star_{\mathsf{left}}$ is not continuous on $C^\star$ because $a \leqslant^\star_{\mathsf{left}} r_b \leqslant^\star_{\mathsf{left}} \ell_c$. This yields a contradiction because these three vertices lie in $\mathsf{Treated}$. This concludes the proof of Claim 33. □

Claim 38. *On the $i + 1^{st}$ iteration, each call at* $\mathsf{LRorder}$ *at Line 20 returns an order.*

Proof. Let S be a sibling class contained in $\mathsf{Current}$ and let L, R be the sets defined in the for loop at Line 16 when $S_i = S$. We want to prove that $\mathsf{LRorder}(S, \mathsf{L}, \mathsf{R}, \mathcal{SC})$ returns an order. This is sufficient to prove the claim. By Lemma 24, this is equivalent to prove that there exists a total order $\leqslant_S$ on S that is continuous on every special clique in $\mathcal{SC}$ and such that:

- the minimum element of $\leqslant_S$ belongs to every set in $\mathcal{SC}$ intersecting S and L, and
- the maximum element of $\leqslant_S$ belongs to every set in $\mathcal{SC}$ that intersects S and R.

Observe that every total order on S is continuous on every set C such that $S \subseteq C$ or $S \cap C = \varnothing$. Moreover, if $S \subseteq C$ then obviously C contains the minimum and the maximum element of every total order on S. So we can ignore the vertices that belong only to these kinds of special cliques. In the remainder of this proof, we assume that every vertex in $\mathsf{L} \cup \mathsf{R}$ belongs to a special clique that has non-empty intersection with S and does not contain S.

By assumption, G admits a nice order for $\mathcal{B}$. Let $\leqslant_{\mathsf{nice}}$ be a nice order for $\mathcal{B}$ and $\leqslant_S$ be the restriction of $\leqslant_{\mathsf{nice}}$ to S. As $\leqslant_{\mathsf{nice}}$ is continuous on every special clique in $\mathcal{SC}$ by definition so is $\leqslant_S$. In the following, we will prove that the existence of $\leqslant_S$ implies that $\mathsf{LRorder}(S, \mathsf{L}, \mathsf{R}, \mathcal{SC})$ returns an order.

Let s_{min} and s_{max} be respectively the minimum and maximum element for $\leqslant_S$. Let $\mathsf{L}_{\mathsf{nice}}$ and $\mathsf{R}_{\mathsf{nice}}$ be the sets defined as follows

$$\begin{aligned}\mathsf{L}_{\mathsf{nice}} =& \{v \in \mathsf{L} \cup \mathsf{R} \mid v \leqslant_{\mathsf{nice}} s_{\mathsf{min}}\}\\ \mathsf{R}_{\mathsf{nice}} =& (\mathsf{L} \cup \mathsf{R}) \setminus \mathsf{L}_{\mathsf{nice}}.\end{aligned}$$

Observe that $\{\mathsf{L}, \mathsf{R}\}$ and $\{\mathsf{L}_{\mathsf{nice}}, \mathsf{R}_{\mathsf{nice}}\}$ are both by definition partitions of the same set of vertices. We will show with the following two facts that $\{\mathsf{L}, \mathsf{R}\} = \{\mathsf{L}_{\mathsf{nice}}, \mathsf{R}_{\mathsf{nice}}\}$.

Fact 39. *For every $C \in \mathcal{SC}$ such that $C \cap S \neq \varnothing$, we have:*

- *if $C \cap \mathsf{L}_{\mathsf{nice}} \neq \varnothing$ then C contains s_{min}, and*
- *if $C \cap \mathsf{R}_{\mathsf{nice}} \neq \varnothing$ then C contains s_{max}.*

Proof of fact. Assume towards a contradiction that there exists $C \in \mathcal{SC}$ such that $C \cap \mathsf{L}_{\mathsf{nice}} \neq \varnothing$ but $s_{\mathsf{min}} \notin C$. Let $\ell \in C \cap \mathsf{L}_{\mathsf{nice}}$. Since $\mathsf{L}_{\mathsf{nice}} \subseteq \mathsf{L} \cup \mathsf{R}$, by assumption, $C \cap S \neq \varnothing$, take $c \in C \cap S$. By definition of s_{min} and $\mathsf{L}_{\mathsf{nice}}$, we have $\ell \leqslant_{\mathsf{nice}} s_{\mathsf{min}} \leqslant_{\mathsf{nice}} c$. As $\ell \in \mathsf{L} \cup \mathsf{R}$ and by definition $\mathsf{L} \cup \mathsf{R} \cap \mathsf{Pred}(S) \neq \varnothing$, we have $\ell \not\ll s_{\mathsf{min}}$. We deduce that $(\ell, s_{\mathsf{min}}, c)$ is a troubling triple for $(C, \leqslant_{\mathsf{nice}})$ that contradicts $\leqslant_{\mathsf{nice}}$ being a nice order. We conclude that $s_{\mathsf{min}} \in C$. Symmetrically, we can prove that if $C \cap \mathsf{R}_{\mathsf{nice}} \neq \varnothing$ then $s_{\mathsf{right}} \in C$. This concludes the proof of Fact 39. ⧫

Fact 40. *For every $\mathsf{X} \in \{\mathsf{L}, \mathsf{R}\}$, we have $\mathsf{X} \subseteq \mathsf{L}_{\mathsf{nice}}$ or $\mathsf{X} \subseteq \mathsf{R}_{\mathsf{nice}}$.*

Proof of fact. Let $\mathsf{X} \in \{\mathsf{L}, \mathsf{R}\}$. Assume towards a contradiction that there exist $\ell, r \in \mathsf{X}$ such that $\ell \in \mathsf{L}_{\mathsf{nice}}$ and $r \in \mathsf{R}_{\mathsf{nice}}$. In the beginning of this proof, we assume that there exists $C_{\mathsf{left}}, C_{\mathsf{right}} \in \mathcal{SC}$ such that $\ell \in C_{\mathsf{left}}$, $r \in C_{\mathsf{right}}$ and $C_{\mathsf{left}}, C_{\mathsf{right}}$ intersect S and do not contain S. By Fact 39, we have $s_{\mathsf{min}} \in C_{\mathsf{left}}$ and $s_{\mathsf{max}} \in C_{\mathsf{right}}$.

Since S is not contained in C_{left} and $\leqslant_{\mathsf{nice}}$ is continuous on C_{left}, we deduce that $s_{\mathsf{max}} \notin C_{\mathsf{left}}$ because otherwise for every $s \in S \setminus C_{\mathsf{left}}$, $(s_{\mathsf{min}}, s, s_{\mathsf{max}})$ would be a troubling triple for $(C_{\mathsf{left}}, \leqslant_{\mathsf{nice}})$. Symmetrically, we prove that $s_{\mathsf{min}} \notin C_{\mathsf{right}}$. By Fact 39, this implies that $C_{\mathsf{left}} \cap \mathsf{R}_{\mathsf{nice}} = \varnothing$ and $C_{\mathsf{right}} \cap \mathsf{L}_{\mathsf{nice}} = \varnothing$. In particular, $\ell \notin C_{\mathsf{right}}$ and $r \notin C_{\mathsf{left}}$ and thus ℓ and r are incomparable for $\ll$ since each belongs to a special clique which does not contain the other.

Since $\ell \in \mathsf{L}_{\mathsf{nice}}$ and $r \in \mathsf{R}_{\mathsf{nice}}$, we have $\ell \leqslant_{\mathsf{nice}} s_{\mathsf{min}} \leqslant_{\mathsf{nice}} r$. By definition ℓ and r are not in S thus Fact 23 implies that ℓ and r are not both in a sibling class contained in $\mathsf{Current}$. Consequently, ℓ and r are comparable w.r.t. $\leqslant^\star_{\mathsf{left}}$. Suppose w.l.o.g. that $\ell \leqslant^\star_{\mathsf{left}} r$ (the proof is symmetrical for the case $r \leqslant^\star_{\mathsf{left}} \ell$).

If $\mathsf{X} = \mathsf{L}$ then by definition of L and $\leqslant^\star_{\mathsf{left}}$, we have $r \leqslant^\star_{\mathsf{left}} s_{\mathsf{min}}$ and thus $\ell \leqslant^\star_{\mathsf{left}} r \leqslant^\star_{\mathsf{left}} s_{\mathsf{min}}$. But, then $(\ell, r, s_{\mathsf{min}})$ is a troubling triple for $(C_{\mathsf{left}}, \leqslant^\star_{\mathsf{left}})$. Symmetrically, if $\mathsf{X} = \mathsf{R}$, then we deduce that $(s_{\mathsf{max}}, \ell, r)$ is a troubling triple for $(C_{\mathsf{right}}, \leqslant^\star_{\mathsf{left}})$. In both cases, this contradicts Claim 33. This concludes the proof of Fact 40. ⧫

Since $\mathsf{L} \cup \mathsf{R} = \mathsf{L}_{\mathsf{nice}} \cup \mathsf{R}_{\mathsf{nice}}$, we deduce by Fact 40 that either (1) $\mathsf{L} = \mathsf{L}_{\mathsf{nice}}$ and $\mathsf{R} = \mathsf{R}_{\mathsf{nice}}$ or (2) $\mathsf{L} = \mathsf{R}_{\mathsf{nice}}$ and $\mathsf{R} = \mathsf{R}_{\mathsf{nice}}$. Suppose that we have (1). Let C be a special clique that has non-empty intersection with S and does not contain S. By Fact 39 if C intersects $\mathsf{L} = \mathsf{L}_{\mathsf{nice}}$ then it contains s_{min} and if C intersects $\mathsf{R} = \mathsf{R}_{\mathsf{nice}}$, then it contains s_{max}. Thus the existence of $\leqslant_S$ implies that LRorder returns an order. If we have (2) then we can easily prove that the reverse order of $\leqslant_{\mathsf{nice}}$ implies similarly that LRorder returns an order. This concludes the proof of Claim 38. ☐

Claim 38 proves that the algorithm leaves the for loop at Line 16 without returning no. Hence it successfully extends $\leqslant_{\mathsf{left}}$ on $\mathsf{Current} \cup \mathsf{Treated}$.

Claim 41. *The invariant holds at the end of the $i + 1^{st}$ iteration.*

Proof. In this proof, we denote by $\leqslant_{\mathsf{left}}$ the order computed by the algorithm at the end of the $i + 1^{\text{st}}$ iteration. Since the algorithm inserted the sibling classes in $\leqslant_{\mathsf{left}}$ from S_t to S_1 and since it inserted each sibling class S after left-limit(S), we deduce that $\leqslant_{\mathsf{left}}$ is a linear extension of $\leqslant^{\star}_{\mathsf{left}}$. By Fact 32 $\leqslant^{\star}_{\mathsf{left}}$ is a linear extension of $\ll$ thus $\leqslant_{\mathsf{left}}$ is also a linear extension of $\ll$. By Claim 33, $\leqslant^{\star}_{\mathsf{left}}$ is continuous on every special clique. Moreover, from the definition of $\leqslant^{\star}_{\leqslant}$, $u, v \in \mathsf{Current} \cup \mathsf{Treated}$ are incomparable for $\leqslant^{\star}_{\mathsf{left}}$ if and only if u, v are siblings and belong to Current. Hence it is sufficient to prove that, for every special clique C, there is no troubling triple (a, b, c) for $(C, \leqslant_{\mathsf{left}})$ such that at least two vertices among a, b, c belong to a sibling class S contained in Current.

Assume towards a contradiction that there exists such a troubling triple (a, b, c) for some $C \in \mathcal{SC}$. By Lemma 24 the order returned by LRorder for S is continuous on every special clique thus we can not have $a, b, c \in S$. Moreover, we can not have $a, c \in S$ and $b \notin S$ because the way we have inserted the vertices of Current in $\leqslant_{\mathsf{left}}$ would guarantee that a and c are either before b or after b and that would contradict the fact that (a, b, c) is a troubling triple.

Suppose that $a, b \in S$ and $c \notin C$. As $b \leqslant_{\mathsf{left}} c$ and $\leqslant_{\mathsf{left}}$ is a linear extension of $\ll$ we have $b \not\ll c$. Thus c is in $(\mathsf{Current} \cup \mathsf{Treated}) \setminus (S \cup \mathsf{Pred}(S))$ and thus $c \in \mathsf{R}$ where R is the set computed in the for loop when $S_i = S$ and by Lemma 24, C contains the maximum element s_{max} of the total order $\leqslant_S$ computed for S. Hence, (a, b, s_{max}) is a troubling triple for $(C, \leqslant_S)$ where $\leqslant_S$ is the order computed by LRorder for S. This contradicts Lemma 24.

Finally suppose that $b, c \in S$ and $a \notin C$. As $a \not\ll b$ and $\mathsf{Pred}(b) = \mathsf{Pred}(S)$, we deduce that $a \in (\mathsf{Current} \cup \mathsf{Treated}) \setminus (S \cup \mathsf{Pred}(S))$. Symmetrically to the previous case, we deduce (s_{min}, b, c) is a troubling triple for $(C, \leqslant_S)$ where s_{min} is the minimum element of $\leqslant_S$ contradicting Lemma 24. This concludes the proof of Claim 41. ☐

We assumed at the beginning of the proof that $(G, \mathcal{B})$ is a yes-instance. Then, under the assumption that the invariant holds and that $\mathsf{Treated} \neq V(G)$ at the end of the i^{st} iteration, we prove with Claims 28, 29 and 38 that the algorithm does not return no during the $i + 1^{\text{st}}$ iteration. Moreover, by Claim 27 the loop invariant holds before entering the while loop and Claim 41 proves that the invariant holds at every iteration of the while loop. By induction, we deduce that if $(G, \mathcal{B})$ is a yes-instance then the algorithm returns a nice order. As explained at the beginning of this proof, this proves the correctness of the algorithm.

The polynomial runtime follows from the following observations:

- By Lemma 19 Algorithm 2 runs in time $O(n^4)$.
- By Lemma 24 LRorder is a polynomial time algorithm.
- The while loop of Algorithm 3 makes at most n iterations because n is the maximum height of the poset given by $\ll$.
- The for loops inside this while loop do at most n iterations because the number of sibling classes is at most n.
- The checkings at Lines 7, 14 and 25 can be done in polynomial time.
- InsertAfter can be implemented so that it runs in polynomial time.

This concludes the proof of Theorem 26.

☐

5. Conclusion

Theorems 5, 16 and 26 amount to a polynomial-time algorithm deciding if an input graph is a linear leaf power. We believe that the techniques we have introduced in this paper constitute a significant step towards resolving the question if leaf powers can be recognized in polynomial time.

References

[1] Jungho Ahn, Eduard Eiben, O-joung Kwon, Sang-il Oum, et al. A polynomial kernel for 3-leaf power deletion. *arXiv preprint arXiv:1911.04249*, 2019.

[2] Kellogg S Booth and George S Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of computer and system sciences*, 13(3):335–379, 1976.

[3] Andreas Brandstädt and van Bang Le. Structure and linear time recognition of 3-leaf powers. *Information Processing Letters*, 98(4):133–138, 2006.

[4] Andreas Brandstädt, Van Bang Le, and R Sritharan. Structure and linear-time recognition of 4-leaf powers. *ACM Transactions on Algorithms (TALG)*, 5(1):1–22, 2008.

[5] Tiziana Calamoneri and Blerina Sinaimeri. Pairwise compatibility graphs: A survey. *SIAM Review*, 58(3):445–460, 2016.

[6] Zhi-Zhong Chen and Tatsuie Tsukiji. Computing bounded-degree phylogenetic roots of disconnected graphs. *J. Algorithms*, 59(2):125–148, 2006.

[7] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[8] Michael Dom, Jiong Guo, Falk Huffner, and Rolf Niedermeier. Error compensation in leaf power problems. *Algorithmica*, 44(4):363–381, 2006.

[9] David Eppstein and Elham Havvaei. Parameterized leaf power recognition via embedding into graph products. In *IPEC*, volume 18, page 16. Springer, 2018.

[10] Lars Jaffke, O-joung Kwon, Torstein J. F. Strømme, and Jan Arne Telle. Mim-width III. graph powers and generalized distance domination problems. *Theor. Comput. Sci.*, 796:216–236, 2019.

[11] William Kennedy, Guohui Lin, and Guiying Yan. Strictly chordal graphs are leaf powers. *Journal of Discrete Algorithms*, 4(4):511–525, 2006.

[12] Manuel Lafond. On strongly chordal graphs that are not leaf powers. In *Graph-Theoretic Concepts in Computer Science - 43rd International Workshop, WG 2017, Eindhoven, The Netherlands, June 21-23, 2017, Revised Selected Papers*, pages 386–398, 2017.

[13] Ragnar Nevries and Christian Rosenke. Towards a characterization of leaf powers by clique arrangements. *Graphs and Combinatorics*, 32(5):2053–2077, 2016.

[14] Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. On graph powers for leaf-labeled trees. *J. Algorithms*, 42(1):69–108, 2002.

[15] Dieter Rautenbach. Some remarks about leaf roots. *Discrete mathematics*, 306(13):1456–1461, 2006.

[16] Tatsuie Tsukiji and Zhi-Zhong Chen. Computing phylogenetic roots with bounded degrees and errors is np-complete. *Theor. Comput. Sci.*, 363(1):43–59, 2006.