# Tree-Visibility Orders

Dieter Kratsch, Jean-Xavier Rampon

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# Tree-Visibility Orders

Dieter Kratsch , Jean-Xavier Rampon

**N° 2683**

Mai 1995

PROGRAMME 1

Rapport de recherche

ISSN 0249-6399

# Tree-Visibility Orders

Dieter Kratsch* ** , Jean-Xavier Rampon***

Programme 1 — Architectures parallèles, bases de données, réseaux et systèmes distribués
Projet Pampa

Rapport de recherche n° 2683 — Mai 1995 — 16 pages

**Abstract:** In this paper we introduce a new class of partially ordered sets, called tree-visibility orders, extending the class of interval orders in a fashion similar to the extension of interval graphs to chordal graphs. This class contains interval orders, duals of generalized interval orders and height one orders. We give a characterization of tree-visibility orders by an infinite family of minimal forbidden suborders and present an efficient recognition algorithm.

**Key-words:** Partially ordered sets, interval orders, minimal forbidden suborders, recognition algorithm.

*(Résumé : tsvp)*

*F.-Schiller-Universität, Fakultät für Mathematik und Informatik, 07740 Jena, Germany; (kratsch@minet.uni-jena.de)
**This research was done while the author was at IRISA Rennes under CHM fellowship.
***IRISA, Campus de Beaulieu, 35042 Rennes Cédex, France; (rampon@irisa.fr)

# Ordres représentables par des sous-arborescences

**Résumé :** Nous introduisons la classe des ordres représentables par des sous-arborescences d'une arborescence. Cette classe généralise la classe des ordres d'intervalles de manière similaire à la généralisation de la classe des graphes d'intervalles par celle des graphes triangulés. Cette classe contient les ordres d'intervalles, l'ordre dual de tout ordre d'intervalles généralisé et les ordres de hauteur un. Nous proposons une caractérisation par une famille infinie de sous-ordres minimaux exclus tout en donnant un algorithme efficace de reconnaisssance.

**Mots-clé :** Ensembles ordonnés, ordres d'intervalles, sous-ordres minimaux exclus, algorithme de reconnaissance.

# Tree-Visibility Orders

## Contents

# 1 Introduction

The motivation of this work is to extend the class of interval orders in a fashion similar to the extension of interval graphs to chordal graphs. For more details on these graph classes we refer to [6]. Generalized interval orders, a class of orders extending the successor set inclusion property of interval orders, has been introduced by Faigle, Schrader and Turán in [3]. A survey about two other generalizations of interval orders, one allowing intervals to overlap with a given ratio and the second dealing with intervals of partial but no more total order, has also been done by Bogart in [1].

We have chosen the characterization of chordal graphs as intersection graphs of subtrees of a tree and the 'visibility definition' of interval orders for extending interval orders. The combination of these two concepts leads to a class of partially ordered sets defined via visibility in a rooted directed tree. By definition, the tree-visibility orders contain all interval orders. Moreover, they also contain the dual order of any generalized interval order and they contain all height one orders. Our major contributions are a characterization of tree-visibility orders by an infinite family of minimal forbidden suborders and an $\mathcal{O}(nm)$ recognition algorithm for tree-visibility orders, where $n$ denotes the number of elements of the given order $P$ and $m$ denotes the number of edges in the comparability graph of $P$.

# 2 Preliminaries

Most of the terminology on partially ordered sets (orders for short in the sequel), used in this paper, can be found in the book of Trotter [8]. However, we choose for definition of the height of an order the number of elements of a maximal sized chain minus one. For graph theoretic notions we refer to [2].

We only mention some notions concerning characterizations by forbidden suborders. Let $P = (V, \prec_P)$ be an order. The order $P'$ is a *suborder* of $P$ if there is a subset $A \subseteq V$ being the ground set of $P'$ such that $a \prec_{P'} b$ if and only if $a \prec_P b$ for any $a, b \in A$. We also say that $P'$ is the suborder of $P$ induced by $A$ and we denote $P'$ by $P[A]$. Furthermore, $P - A$ denotes the suborder $P[V \setminus A]$. An order $Q$ is *contained (as a suborder)* in the order $P$ if there is a suborder $P'$ of $P$ which is isomorphic to $Q$.

A class $\mathcal{P}$ of orders is *hereditary* if $P \in \mathcal{P}$ implies that any suborder $P'$ of $P$ belongs to $\mathcal{P}$. Many interesting classes of orders are hereditary, as e.g. interval orders and two dimensional orders. If a class $\mathcal{P}$ is hereditary then it can be characterized by the (possibly infinite) list of all its minimal forbidden suborders, where $Q$ is a *minimal forbidden suborder* of the class $\mathcal{P}$ if $Q \notin \mathcal{P}$ but any proper suborder of $Q$ belongs to $\mathcal{P}$. Then an order $P$ belongs to the hereditary class $\mathcal{P}$ if and only if none of the minimal forbidden suborders of $\mathcal{P}$ is contained as a suborder in $P$. This nice type of characterization is certainly a very powerful tool when studying a class of orders. We are going to present an infinite list of minimal forbidden tree-visibility orders.

# 3 Definition of tree-visibility orders

We introduce a new class of orders extending the class of interval orders. Notice that in a rooted directed tree each edge is directed away from the root.

**Definition 1** *An order $P = (V, \prec_P)$ is a* tree-visibility order *if there exists a rooted directed tree $\vec{T} = (V(\vec{T}), E(\vec{T}))$ and a one-to-one mapping from $V$ to a multiset $\langle \vec{T_x},\ x \in V \rangle$ of directed rooted subtrees of $\vec{T}$ such that $u \prec_P v$ if and only if*

*(i) $V(\vec{T_u}) \cap V(\vec{T_v}) = \emptyset$, and*

*(ii) there are $x_v \in V(\vec{T_v})$ and $x_u \in V(\vec{T_u})$ such that there is a directed path from $x_v$ to $x_u$ in $\vec{T}$.*

*The rooted directed tree $\vec{T}$ is said to be a* visibility tree *of $P$ and the tuple $(\vec{T}, \langle \vec{T_x},\ x \in V \rangle)$ is said to be a* tree-visibility model *of $P$.*

**Remark 1** *Replacing the condition (ii) in the definition by the condition*

*(ii') there is a $x_v \in V(\vec{T_v})$ such that for any $x_u \in V(\vec{T_u})$ there is a directed path from $x_v$ to $x_u$ in $\vec{T}$,*

*creates an equivalent definition of tree-visibility orders.*

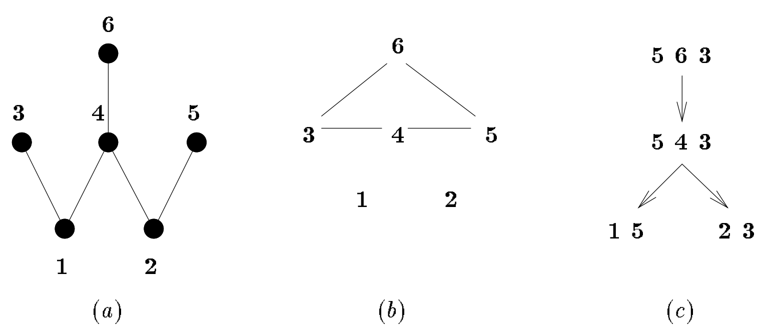


Figure 1: In $(a)$ a tree-visibility order $P = (V, \prec_P)$ is given. In $(b)$ the forcing graph (see Section 5) of $P$, that is not chordal, is depicted. In $(c)$ we give a visibility tree for $P$. The nodes of $\vec{T}$ are labeled in such a way that for any $x \in V$ the subtree $\vec{T_x}$ is induced by all nodes of $\vec{T}$ having label $x$.

Notice that several elements of a tree-visibility order $P$ may be associated to the same subtree of a visibility tree $\vec{T}$ of $P$. Clearly, a tree-visibility order may have several visibility trees and, moreover, a tree-visibility order $P$ may have several tree-visibility models $(\vec{T}, \langle \vec{T_x},\ x \in V \rangle)$ for a fixed visibility tree $\vec{T}$.

# 4 Classes of tree-visibility orders

In this section we show that tree-visibility orders extend two well-studied classes of orders, the height one orders and the interval orders. Despite the fact that our characterization of tree-visibility orders given in Section 7 directly implies the inclusion of these two classes. We give a direct proof here such that the reader becomes familiar with tree-visibility models. We denote by $x \parallel_P y$ the fact that $x$ and $y$ are incomparable in $P$.

**Theorem 1** *Any height one order is a tree-visibility order.*

**Proof:** Let $P = (V, \prec_P)$ be an height one order. Let $A = \{a_1, a_2, \ldots a_r\}$, $r \geq 1$, be the set of minimal elements of $P$ and let $B = \{b_1, b_2, \ldots b_s\}$, $s \geq 0$, be $V \setminus A$. Clearly any element of $B$ is a maximal element of $P$.

We construct a visibility tree $\vec{T}$ of $P$. The vertex set of $\vec{T}$ is $V(\vec{T}) = \{u\} \cup \{v_1, v_2, \ldots v_s\}$. The edge set of $\vec{T}$ is $E(\vec{T}) = \{(u, v_i) \ : \ i = 1, 2, \ldots r\}$. The subtrees $\vec{T_x}$ are induced subtrees of $\vec{T}$, hence it suffices to give their vertex sets. For any $a_i \in A$ we take $V(T_{a_i}) = \{v_i\}$ and for any $b_j \in B$ we take $V(T_{b_j}) = \{u\} \cup \{v_i \ : \ a_i \parallel_P b_j\}$.

Finally the constructed tuple $(\vec{T}, \langle \vec{T_x}, \ x \in V \rangle)$ is shown to be a tree-visibility model of $P$. Clearly for any $i \neq j$ $a_i \parallel_P a_j$ by condition (ii) and $b_i \parallel_P b_j$ by condition (i) of Definition 1. Now consider a pair $a_i \in A$ and $b_j \in B$. By the construction of the subtrees $\vec{T_{a_i}}$ and $\vec{T_{b_j}}$ condition (ii) is always fulfilled. By our construction of $\vec{T_{b_j}}$ we have $V(\vec{T_{a_i}}) \cap V(\vec{T_{b_j}}) \neq \emptyset$ if and only if $a_i \parallel_P b_j$. Hence $P$ is a tree-visibility order. ∎

Our definition is an extension of the 'visibility definition' of interval orders (see [8]). Thus the following theorem is expected.

**Theorem 2** *Any interval order is a tree-visibility order.*

**Proof:** Let $P = (V, \prec_P)$ be an interval order. The visibility tree $\vec{T}$ is a directed path for which the vertices correspond to the endpoints of the intervals in the interval model of $P$. The subtree $\vec{T_x}$ associated to the element $x$ of $P$ is a directed subpath and consists of all vertices associated to interval endpoints $r$ with $a(x) \leq r \leq b(x)$, where $a(x)$ (respectively $b(x)$) denotes the left endpoint (respectively right endpoint) of the interval associated to $x$. It follows immediately from the definition of an interval order that the defined tuple is a tree-visibility model of $P$. Hence $P$ is a tree-visibility order. ∎

# 5 Chordal sandwich graphs

We start with an easy observation.

**Property 1** *The class of tree-visibility orders is hereditary,*

**Proof:** Let $(\vec{T}, \langle \vec{T_x},\ x \in V\rangle)$ be a tree-visibility model of an order $P = (V, \prec_P)$ and let $P'$ be a suborder of $P$ induced by the set $A \subseteq V$. Then $(\vec{T}, \langle \vec{T_x},\ x \in A\rangle)$ is a tree-visibility model of $P'$. ∎

Hence it would be desirable to characterize the class of tree-visibility orders by giving a list of all minimal forbidden suborders. Next we present two interesting technical lemmata on tree-visibility orders.

**Lemma 1** *Let $P = (V, \prec_P)$ be a tree-visibility order and $(\vec{T}, \langle \vec{T_x},\ x \in V\rangle)$ a tree-visibility model of $P$. Then for any $u \in V$ the (unique) directed path $\mathcal{P}(u)$ from the root of $\vec{T}$ to the root of $\vec{T_u}$ contains the root of the subtree $\vec{T_v}$ for any $v \in V$ with $u \prec_P v$.*

**Proof:** Let $u \prec_P v$. Then $V(\vec{T_u}) \cap V(\vec{T_v}) = \emptyset$ and there are $x_v \in V(\vec{T_v})$ and $x_u \in V(\vec{T_u})$ such that there is a directed path from $x_v$ to $x_u$ in $\vec{T}$. Hence $x_v$ and the root of $\vec{T_u}$ are vertices on the directed path from the root of $\vec{T}$ to $x_u$. Since $V(\vec{T_u}) \cap V(\vec{T_v}) = \emptyset$ the root of $\vec{T_u}$ does not belong to $V(\vec{T_v})$ and there is a directed path from the root of $\vec{T_v}$ to the root of $\vec{T_u}$, i.e., the root of $\vec{T_v}$ is on the directed path $\mathcal{P}(u)$. ∎

**Corollary 1** *Let $P = (V, \prec_P)$ be a tree-visibility order and $(\vec{T}, \langle \vec{T_x},\ x \in V\rangle)$ a tree-visibility model of $P$. Then $u \prec_P v$ if and only if $V(\vec{T_u}) \cap V(\vec{T_v}) = \emptyset$ and the root of $\vec{T_v}$ is an ancestor of the root of $\vec{T_u}$ in the visibility tree $\vec{T}$ for each pair $u, v \in V$.*

**Lemma 2** *Let $P = (V, \prec_P)$ be a tree-visibility order and $(\vec{T}, \langle \vec{T_x},\ x \in V\rangle)$ a tree-visibility model of $P$. Then $V(\vec{T_x}) \cap V(\vec{T_y}) \neq \emptyset$ holds for any pair of incomparable elements $x, y \in V$ having a common predecessor $z$.*

**Proof:** Let $x$ and $y$ be two incomparable elements with a common predecessor $z$. By Lemma 1 we have that the root of $\vec{T_x}$ and the root of $\vec{T_y}$ occur on the unique directed path $\mathcal{P}(z)$ from the root of $\vec{T}$ to the root of $\vec{T_z}$. Hence there is either a directed path from the root of $\vec{T_x}$ to the root of $\vec{T_y}$ or vice versa. Hence $x$ and $y$ must be incomparable because of $V(\vec{T_x}) \cap V(\vec{T_y}) \neq \emptyset$. ∎

The previous lemma leads to the following concept of a forcing graph which is helpful when studying tree-visibility orders. It gives an easy tool for showing that a certain order is not a tree-visibility order.

**Definition 2** *Let $P = (V, \prec_P)$ be an order. The undirected graph $G = (V, E)$ with $E = \{\{x, y\} : x \parallel_P y$ for which $x$ and $y$ have a common predecessor$\}$ is called the* forcing graph *of $P$.*

The forcing graph of an order is a subgraph of the cocomparability graph.

**Definition 3** *Let $P = (V, \prec_P)$ be an order. The undirected graph $G = (V, E)$ with $E = \{\{x, y\} : x \parallel_P y\}$ is called the* cocomparability graph *of $P$.*

We are going to show that the existence of a tree-visiblity model for an order $P$ requires that there exists a chordal sandwich graph between the forcing graph and the cocomparability graph. The concept of a sandwich graph has been introduced and extensively studied by Golumbic, Kaplan and Shamir [7].

**Definition 4** *A graph $G$ is a* spanning subgraph *of the graph $G'$ if both graphs have the same vertex set and $G$ is a subgraph of $G'$ (i.e. $E(G) \subseteq E(G')$).*

**Definition 5** *Let $G = (V, E)$ be a spanning subgraph of the graph $G' = (V, E')$. Then $H = (V, E(H))$ is said to be a* sandwich graph *of $(G, G')$ if $G$ is a spanning subgraph of $H$ and $H$ is a spanning subgraph of $G'$ (i.e. $E(G) \subseteq E(H) \subseteq E(G')$).*

Now we will point out the relation of tree-visibility orders and sandwich graphs.

**Theorem 3** *Let $P = (V, \prec_P)$ be an order, $G$ its forcing graph and $G'$ its cocomparability graph. If $P$ is a tree-visibility order then there exists a chordal sandwich graph $H$ for $(G, G')$.*

**Proof:** Let $P = (V, \prec_P)$ be a tree-visibility order and $(\vec{T}, \langle \vec{T_x},\ x \in V \rangle)$ a tree-visibility model of $P$. Let $T$ be the underlying undirected graph of $\vec{T}$ and for any $x \in V$ let $T_x$ be the underlying undirected graph of $\vec{T_x}$. Hence $\langle T_x : x \in V \rangle$ is a collection of subtrees of the tree $T$. Let $H = (V, E(H))$ be the (vertex) intersection graph of the subtrees $T_x$, $x \in V$, i.e., $u, v \in V$ are adjacent in $H$ if and only if $V(T_u) \cap V(T_v) \neq \emptyset$. $H$ is a chordal graph since it is the intersection graph of subtrees of a tree [5]. The forcing graph $G$ of $P$ is a spanning subgraph of $H$, since $\{u, v\} \in E(G)$ implies $V(\vec{T_u}) \cap V(\vec{T_v}) \neq \emptyset$ by Lemma 2, hence $\{u, v\} \in E(H)$. $H$ is a spanning subgraph of the cocomparability graph $G'$ of $P$ since $\{u, v\} \in E(H)$ implies $V(\vec{T_u}) \cap V(\vec{T_v}) \neq \emptyset$ thus $u \parallel_P v$, by Definition 1. Consequently, $H$ is a chordal sandwich graph for $(G, G')$. ■

# 6 Forbidden suborders

We present an infinite list of minimal forbidden suborders for the class of tree-visibility orders.

**Definition 6** *Let $k \geq 1$. The order $Q_k$ has groundset $V = \{a_1, a_2, \ldots, a_k\} \cup \{b_1, b_2, \ldots, b_{k+1}\} \cup \{c_1, c_2\}$. Furthermore, $a_i \prec_{Q_k} b_j$ if and only if $j \in \{i, i+1\}$, $a_i \prec_{Q_k} c_j$ for all $i \in \{1, 2, \ldots k\}$ and $j \in \{1, 2\}$, $b_i \prec_{Q_k} c_1$ for $i \in \{1, 2, \ldots k\}$ and $b_i \prec_{Q_k} c_2$ for $i \in \{2, 3, \ldots k+1\}$. (See Fig. 2.)*

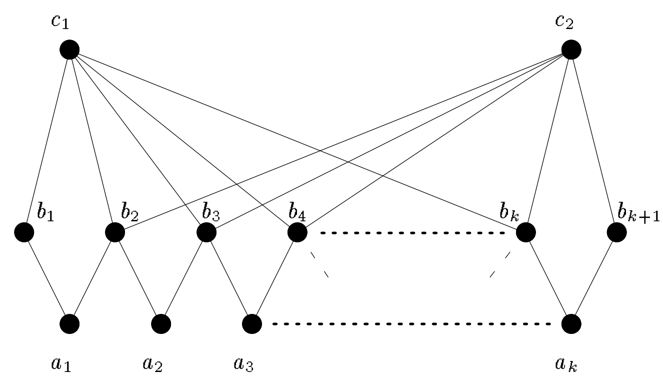


Figure 2: The forbidden order $Q_k$.

**Theorem 4** *The order $Q_k$ is a minimal forbidden suborder for the class of tree-visibility orders for any $k \geq 1$.*

**Proof:** First we show that for any $k \geq 1$ the order $Q_k$ is not a tree-visiblity order. Assume that $Q_k$ would be a tree-visibility order for some $k \geq 1$. We consider the forcing graph $G$ of $Q_k$. The graph $G$ is not chordal, since it contains the chordless cycle $C = (b_1, b_2, \ldots b_{k+1}, c_1, c_2, b_1)$. By Theorem 3, there is a chordal sandwich graph $H$ for the pair $(G, G')$ where $G'$ is the cocomparability graph of $Q_k$. Taking the vertices $c_1$ and $c_2$ of the cycle $C$ there is no vertex $b_j$, $j \in \{1, 2, \ldots, k+1\}$, in the cycle $C$ adjacent to $c_1$ and $c_2$ in $H$ since the only neighbours of $c_1$ in $G'$ are $c_2$ and $b_{k+1}$ and the only neighbours of $c_2$ in $G'$ are $c_1$ and $b_1$. Take $c_1$ and $c_2$ and the vertices of a shortest path between $c_1$ and $c_2$ in the graph obtained from $H[C]$, the graph induced in $H$ by the vertices of $C$, by deleting the edge $\{c_1, c_2\}$. This vertex set induces a chordless cycle of length at least 4 in $H$. Hence, $H$ is not chordal. Consequently, none of the order $Q_k$, $k \geq 1$, is a tree-visibility order by Theorem 3.

Finally it is a matter of routine to construct a tree-visibility model for any proper suborder $Q_k - \{x\}$, $x \in V$, of $Q_k$ and any $k \geq 1$ (see Figure 3 for a tree-visibility model of $Q_4 - \{b_3\}$). ■

The following proposition is crucial for the correctness proof of our recognition algorithm for tree-visibility orders. Moreover, it is a major step in showing that the order $Q_k$, $k \geq 1$, are exactly all minimal forbidden suborders for the class of tree-visibility orders.

We shall need some more technical concepts for the proof of the main theorem. Let $P = (V, \prec_P)$ be an order. We denote the set of all maximal (respectively, minimal) elements of $P$ by $MAX(P)$ (respectively, $MIN(P)$). $Pred(x) := \{y \in V \ : \ y \prec_P x\}$ and $Succ(x) :=$

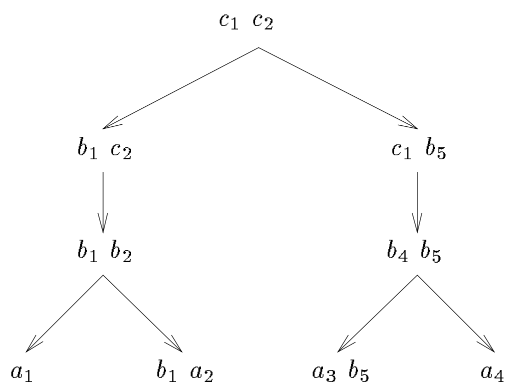


Figure 3: The tree visibility model produced by the algorithm TREE-VISIBILITY of Section 7 for the order $Q_4 - \{b_3\}$.

$\{y \in V \ : \ x \prec_P y\}$ are the predecessor set and successor set, respectively, of an element $x \in V$.

The following concept is important for the recognition algorithm. An element $x \in MAX(P)$ is said to be *universal* if its predecessor set $Pred(x) := \{y \in V \ : \ y \prec_P x\}$ is equal to $V(P) - MAX(P)$. Hence a maximal element $x$ is universal if it is only incomparable to all other maximal elements of $P$.

An order $P = (V, \prec_P)$ is said to be *connected* if its comparability graph $G(P)$ is connected. Let $u$ and $v$ be elements of a connected order $P$. Then there is a shortest $u,v$-path $(u = x_0, x_1, \ldots, x_r = v)$ in $G(P)$ such that the internal vertices $x_1, \ldots, x_{r-1}$ of the path are alternately minimal and maximal elements of $P$. Such a $u,v$-path is said to be *normalized*. To see that a normalized path exists for any pair $u, v \in V$ take any shortest $u,v$-path $(u = y_0, y_1, \ldots, y_r = v)$ in $G(P)$. Then either $y_{i-1} \prec_P y_i$ and $y_{i+1} \prec_P y_i$, or $y_i \prec_P y_{i-1}$ and $y_i \prec_P y_{i+1}$ for any $i \in \{1, \ldots, r-1\}$. If $y_i \notin MAX(P)$ in the first case then replace it by a maximal element $y_i'$ that is a successor of $y_i$. If $y_i \notin MIN(P)$ in the second case then replace it by a minimal element $y_i'$ that is a predecessor of $y_i$. This leads to a normalized path between $u$ and $v$.

In the remainder of the paper we will only consider normalized paths $(u = x_0, x_1, \ldots, x_r = v)$ between maximal elements of an order. Thus $x_i$ is a minimal element if $i$ is even and $x_i$ is a maximal element if $i$ is odd. Moreover, $r$ is even.

**Proposition 1** *Let $P = (V, \prec_P)$ be a connected order, $P - MAX(P)$ connected and assume that $P$ has no universal element in $MAX(P)$. Then $P$ contains a $Q_k$ as a suborder for suitable $k \geq 1$.*

**Proof:** Let $P$ be an order fulfilling the conditions of the theorem. We denote the connected suborder $P - MAX(P)$ by $P'$. We say that a maximal element $x$ of $P$ has a *private predecessor* $p_x$ if $p_x \prec_P x$ and $p_x \parallel_P y$ for all $y \in (MAX(P) \setminus \{x\})$.

**Case 1:** $P$ has a maximal element $x$ with a private predecessor $p_x$.
W.l.o.g. $p_x$ is a maximal element of $P'$. Since $x$ is not a universal element of $P$ there are elements $u \in MAX(P')$ with $u \parallel_P x$. We choose $t \in MAX(P') \cap \{u \ : \ u \parallel_P x\}$ such that the length of a shortest path between $t$ and $p_x$ in $G(P')$, is minimum among all elements $u \in MAX(P')$ that are incomparable to $x$. Let $(p_x = x_0, x_1, \ldots, x_{2s} = t)$, $s \geq 1$, be a normalized $p_x, t$-path in $P'$. Clearly the set $A = \{p_x = x_0, x_1, \ldots, x_{2s} = t\}$ induces a fence in $P'$. Furthermore $x_{2i} \in MAX(P')$ for all $i \in \{0, 1, \ldots, s\}$. By the choice of $t$ we have $x_{2i} \prec_P x$ for all $i \in \{0, 1, \ldots, s-1\}$.

Since $t$ is not a maximal element of $P$ there is a $y \in MAX(P)$ with $t \prec_P y$. Furthermore there is a $j$ with $x_{2j} \parallel_P y$ since $p_x$ is a private predecessor of $x = x_0$ implying $x \parallel_P y$. Now let $j$ be the largest subscript such that $x_{2j} \parallel_P y$. Then the set $\{x, x_{2j}, x_{2j+1}, \ldots, x_{2s} = t, y\}$ induces a $Q_{s-j}$ in $P$.

**Case 2:** No maximal element of $P$ has a private predecessor.
We choose among all elements of $MAX(P')$ an element $w$ having a successor set of minimum cardinality. Then let $R \subseteq MAX(P)$ be a subset of $Succ(w)$ containing all but one of the successors of $w$ in $P$. Notice that $R \neq \emptyset$. By the choice of $R$ every maximal element of $P'$ belongs to $P - R$ and has at least one successor in $P - R$. Thus, the maximal elements of the order $P - R$ are exactly the elements of $MAX(P) \setminus R$. Hence the order $(P-R)-(MAX(P-R))$ is exactly $P'$ and hence connected. Furthermore, $P - R$ has no universal element since any universal element $u \in MAX(P-R)$ of $P - R$ had to fulfil $MAX(P') \subseteq Pred(u)$ which would imply that $u$ is universal in $P$, a contradiction. Moreover $w$ is private predecessor of $x$ that is the only successor of $w$ in $P$ not belonging to $R$.

Altogether, $P - R$ fulfils the assumptions of Case 1. Hence, $P - R$ has a $Q_k$ for some $k \geq 1$ as a suborder. Hence $Q_k$ is also a suborder of $P$. ■

# 7 Recognition algorithm

The aim of this section is to present an efficient algorithm recognizing tree-visibility orders. Furthermore, if the given order $P$ is indeed a tree-visibility order then the algorithm constructs a somewhat compact tree-visibility model of $P$.

We have chosen to describe this algorithm in its natural recursive manner by giving a subroutine TREE-VISIBILITY$(K, N, INC)$. The algorithm TREE-VISIBILITY$(P)$ is started by calling TREE-VISIBILITY$(P, R, \emptyset)$ where $R$ is a reference variable pointing to the future root of the eventual tree-visibility model of the given order $P$.

Moreover, the algorithm will compute a visibility-tree $\vec{T}$ of $P$, if there is one, assigning to each node $N$ of $\vec{T}$ a label set that is going to be the set of all those vertices $u$ for which $\vec{T_u}$ contains the node $N$.

**SUBROUTINE** TREE-VISIBILITY($K, N, INC$)

$K$: /* Current order. */
$N$: /* Father of the root of the subtree representing $K$. */
$INC$: /* Set of all elements of the label set of node $N$ that */
/* are incomparable to all elements of the order $K$. */

**Begin**
Compute $MAX(K)$;
Compute the connected components $K_1, K_2, \dots, K_r$ of $K - MAX(K)$;
**If** $K - MAX(K)$ has exactly one connected component
**Then**
  Compute $U(K)$ the set of all universal elements of $K$;
  **If** $U(K) = \emptyset$;
  **Then**
    EXIT; output "$K$ is not a tree-visibility order."
  **Else**
    Create a node $C$ with father $N$ and label set $INC \cup MAX(K)$;
    TREE-VISIBILITY($K - U(K); C; INC$);
  **EndIf**
**Else**
  Create a node $C$ with father $N$ and label set $INC \cup MAX(K)$;
  **For** all connected components $K_i = (V(K_i), \prec_P)$ of $K - MAX(K)$ **Do**
    Compute $M_i := \left(\bigcup_{x \in V(K_i)} Succ(x)\right) \cap MAX(K)$;
    Compute $L_i := \{x \in M_i \ : \ V(K_i) \setminus Pred(x) \neq \emptyset\}$;
    TREE-VISIBILITY($K[V(K_i) \cup L_i]; C; INC \cup (MAX(K) \setminus M_i)$);
  **EndFor**
**EndIf**
**End**;

**Theorem 5** *Given an order $P = (V, \prec_P)$, the algorithm* TREE-VISIBILITY($P$) *decides whether $P$ is a tree-visibility order. If so, the algorithm computes a tree-visibility model of $P$. The running time of the algorithm is $\mathcal{O}(nm)$, where $n$ denotes the number of elements of $P$ and $m$ denotes the number of edges in the comparability graph of $P$.*

**Proof:** The algorithm TREE-VISIBILITY($P$) terminates in two different ways. Either it outputs "$P$ is not a tree-visibility order" since a recursive call of TREE-VISIBILITY($K, N, INC$) found a connected suborder $K$ of $P$ such that $K - MAX(K)$ is connected and has no universal vertex. Hence $K$ contains a suborder $Q_k$ for some $k \geq 1$ by Proposition 1. Consequently there is a $Q_k$ that is a suborder of $P$, thus $P$ is not a tree-visibility order by Theorem 4.

Otherwise the algorithm TREE-VISIBILITY($P$) terminates successfully with the construction of a tree $T$ such that the reference variable $R$ points to the root of $T$. This means that any subroutine TREE-VISIBILITY($K, N, INC$) recursively called during the execution of the algorithm either terminated by recursive calls of $s \geq 1$ subroutines where $s$ is the number of connected components of $K - MAX(K)$ or by only creating a leaf of the final tree $T$ if $V(K)$ is an antichain, i.e., $V(K) \setminus MAX(K) = \emptyset$.

For proving the correctness of the algorithm it suffices to show that the tree $T$ with the label sets assigned to each node of $T$ constitutes a tree-visibility model of the given order $P$. Consider $T$ as a directed tree $\vec{T}$ with the root specified by $R$. For any $v \in V$ the corresponding subtree $\vec{T_v}$ consists of those nodes of $T$ that have a label set containing $v$. Note that $\vec{T_v}$ is a connected subgraph of $\vec{T}$. Indeed, if $v$ belongs to the label set of the father of a node $N$ but $v$ does not belong to the label set of node $N$ then $v$ is not an element of the order $K$, i.e., the current order when the node $N$ is created. Moreover, since $v$ does not belong to the label set of $N$, $v$ does not belong to the current $INC$ and hence $v$ does not belong to an $INC$ for any recursive call creating a node with ancestor $N$. On the other hand, if $v$ belongs to the label set of a node $N$ and never appears before on a node in the path from the root of the tree to $N$, then $v$ is a maximal element in one of the connected components of $\tilde{K} - MAX(\tilde{K})$ where $\tilde{K}$ is the current order when $N'$ the father node of $N$ has been created. This guarantees that $v$ cannot belong to the current $INC$ when $N'$ has been created, that $v$ is not a maximal element of $\tilde{K}$, and that $v$ is not an element of any of the other order obtained when applying the subroutine to the remaining connected components of $\tilde{K} - MAX(\tilde{K})$. Thus $N$ is the root of $\vec{T_v}$.

It remains to show that the final tree indeed creates a tree-visibility model of $P$. This follows immediately when noting that our algorithm guarantees that when calling the subroutine TREE-VISIBILITY$(K; N; INC)$ the set $INC$ is indeed the set of all elements of $P$ that belong to the label set of a node in the path from the root of the tree to $N$ and that are incomparable to all elements of $K$. This is ensured by the use of the auxiliary sets $L_i$ and $M_i$ in the For loop.

Let $u \prec_P v$. Consider the first subroutine TREE-VISIBILITY$(K; N; INC)$ executed during the algorithm for which $u \in MAX(K)$ holds. Clearly $v \notin INC$ and $v \notin V(K)$. Hence $v$ is not in the label set of node $N$ and $\vec{T_u}$ and $\vec{T_v}$ have no node in common. On the other hand, there is a directed path from the root of $\vec{T_v}$ to the node $N$, i.e., the root of $u$, since $K$ is a suborder of $\tilde{K}$, the current order when creating the root of $\vec{T_v}$.

Finally, consider the execution of TREE-VISIBILITY$(K; N; INC)$ and suppose $v \in V$ is not an element in the label set of node $N$ but it appears in the label set of the father $N'$ of $N$. Then $v$ is an element that has all elements of the order $K$ as successor. This is guaranteed by the construction of the current orders for the recursive call.

It is not hard to implement the described algorithm such that the running time is $\mathcal{O}(nm)$. The important fact to notice is that the tree $T$, which is isomorphic to the recursion tree of the algorithm, has at most $n$ vertices since each node has in its label set an element which did not appear in the label set of any proper ancestor of the node. Indeed, if $N'$ is the father of $N$ then there is an element in the label set of $N$ that belongs to the maximal elements of the connected component of $K - MAX(K)$ inducing the node $N$ where $K$ is the current order when creating the node $N'$. Thus this element can appear only in the label set of nodes of the subtree of $\vec{T}$ rooted in $N$. It is a matter of routine to see that one subroutine

TREE-VISIBILITY$(K; N; INC)$ can be executed in time $O(m)$ using a linear time algorithm for the computation of the connected components of a graph. ■

From the proof of the Theorem 5, it appears that the algorithm TREE-VISIBILITY fails to construct a tree-visibility model of the given order if and only if the order contains a $Q_k$, $k \geq 1$, as suborder. This leads to one of the major results of our paper, namely the already announced characterization of the tree-visibility orders by an infinite family of forbidden suborders of height two.

**Theorem 6** *An order $P$ is a tree-visibility order if and only if it does not contain an order $Q_k$, $k \geq 1$, (see Figure 2) as a suborder.*

This characterization immediately implies Theorem 1 and Theorem 2. Furthermore, in [3] Faigle, Schrader and Turán introduced the generalized interval orders and a linear time recognition algorithm for generalized interval orders has been given by Garbe in [4]. An order $P = (V, \prec_P)$ is said to be a generalized interval order if for all $x, y \in V$ either $Succ(x) \cap Succ(y) = \emptyset$, $Succ(x) \subseteq Succ(y)$ or $Succ(y) \subseteq Succ(x)$. Since for any $Q_k$, $k \geq 1$, we have $Pred(c_1) \cap Pred(c_2) \neq \emptyset$, and neither $Pred(c_1) \subseteq Pred(c_2)$ nor $Pred(c_2) \subseteq Pred(c_1)$, and since any height one order is a tree-visibility order, we get:

**Corollary 2** *The class of the duals of generalized interval orders is a proper subclass of the tree-visibility orders.*

# 8 Optimality of the tree-visibility model

The aim of this section is to demonstrate that the tree-visibility model constructed by our algorithm is optimal in the following sense. The height of the directed tree computed by the algorithm is minimal among the height of all visibility trees $\vec{T}$ of any tree-visibility model $(\vec{T}, \langle \vec{T}_x,\ x \in V \rangle)$. Since the number of nodes of the tree is bounded by the number of elements of the given order this shows that the algorithm provides a compact tree-visibility model.

In order to prove the minimality of the height of the tree, we need the following lemma which also explains why the label set of the node created during the call of TREE-VISIBILITY on an order $K$ is chosen as $MAX(K)$.

**Lemma 3** *Let $P = (V, \prec_P)$ be a tree-visibility order, then for any tree-visibility model $(\vec{T}, \langle \vec{T}_x,\ x \in V \rangle)$ of $P$ there is another tree-visibility model of $P$ on the same visibility tree $\vec{T}$ such that the label set of the root of $\vec{T}$ is $MAX(P)$.*

**Proof:** Let $L(\vec{T})$ be the label set of the root of $\vec{T}$. Since $L(\vec{T}) \subseteq MAX(P)$ we assume w.l.o.g. $L(\vec{T}) \neq MAX(P)$. For all $x \in MAX(P) \setminus L(\vec{T})$, let $A(x)$ be the set of nodes of $\vec{T}$ which do not belong to the maximal subtree of $\vec{T}$ rooted in the root of $\vec{T}_x$. For any such $x$, the root of $\vec{T}$ belongs to $A(x)$ and since $x$ is a maximal element of $P$ any element of $P$ belonging to the label set of a node in $A(x)$ is incomparable to $x$ in $P$. By adding $x$ to the label set of any node in $A(x)$, for any $x \in MAX(P) \setminus L(\vec{T})$, we obtain a tree-visibility model fulfilling the claimed property. ■

**Theorem 7** *The visibility tree of the tree-visibility model computed by the algorithm* TREE-VISIBILITY *is of minimal height.*

**Proof:** We prove the minimality of the height of the visibility tree computed by the algorithm TREE-VISIBILITY by induction on the number of elements of the given tree-visibility order $P = (V, \prec_P)$. Note that if $P$ is an antichain then the algorithm computes a tree with a unique node whose label set is all the elements of $P$. This tree is obviously of minimal height. This settle the case $|V| = 1$ and allow to consider orders of height greater or equal to one.

Assume that $|V| = n$ with $n > 1$ and let $\vec{T}$ be the visibility tree of the tree-visibility model of $P$ computed by the algorithm TREE-VISIBILITY. Let $(\vec{\mathcal{S}}, \langle \vec{\mathcal{S}_x},\ x \in V\rangle)$ be a tree-visibility model of $P$ such that both $\vec{\mathcal{S}}$ is of minimal height and the label set of its root is $MAX(P)$. Notice that such a model exists by Lemma 3.

Let $\vec{\mathcal{R}^1}, \ldots, \vec{\mathcal{R}^k}$ be the maximal rooted subtrees of $\vec{\mathcal{S}}$ obtained by deleting the root of $\vec{\mathcal{S}}$. Hence $k$ is the number of children of the root of $\vec{\mathcal{S}}$ and $k \geq 1$. Let $P_i = (V(P_i), \prec_P)$ be any connected component of $P - MAX(P)$. Since $MAX(P_i) \cap MAX(P) = \emptyset$ the subtree $\vec{\mathcal{S}_x}$ does not contain the root of $\vec{\mathcal{S}}$ for every $x \in V(P_i)$. Hence for any $x \in V(P_i)$ there exists a unique $j \in \{1, \ldots, k\}$ such that $\vec{\mathcal{S}_x}$ is a subtree of $\vec{\mathcal{R}^j}$.

We claim that there exists a $j \in \{1, \ldots, k\}$ such that $\vec{\mathcal{S}_x}$ is a subtree of $\vec{\mathcal{R}^j}$ for all $x \in V(P_i)$. To prove the claim let us suppose there would be $x, y \in V(P_i)$ such that $\vec{\mathcal{S}_x}$ is a subtree of $\vec{\mathcal{R}^a}$ and $\vec{\mathcal{S}_y}$ is a subtree of $\vec{\mathcal{R}^b}$ with $a \neq b$. Hence $x$ and $y$ are incomparable in $P$. Take any $x, y$-path in $G(P_i)$. Then there must be adjacent vertices $u_i$ and $u_{i+1}$ in the path such that the trees $\vec{\mathcal{S}_{u_i}}$ and $\vec{\mathcal{S}_{u_{i+1}}}$ are subtrees of different $\vec{\mathcal{R}^j}$'s, contradicting the fact that $u_i$ and $u_{i+1}$ are comparable in $P_i$ and $P$. This proves the claim.

Let $H_i = \{y \in MAX(P)\ :\ Pred(y) \cap V(P_i) \neq \emptyset \textit{ and } V(P_i) \setminus Pred(y) \neq \emptyset\}$. Assume that $\vec{\mathcal{S}_x}$ is a subtree of $\vec{\mathcal{R}^j}$ for every $x \in V(P_i)$. Suppose there would exist a $y \in H_i$ not belonging to the label set of the root of $\vec{\mathcal{R}^j}$. Hence $\vec{\mathcal{S}_y}$ does not contain any node of $\vec{\mathcal{R}^j}$. $\vec{\mathcal{S}_x}$ is a subtree of $\vec{\mathcal{R}^j}$ for any $x$ in $MAX(P_i)$, hence $V(P_i) \subseteq Pred(y)$, contradicting the choice of $y \in H_i$. Therefore $H_i$ is a subset of the label set of the root of $\vec{\mathcal{R}^j}$.

Clearly $(\vec{\mathcal{R}^j}, \langle \vec{\mathcal{R}^j_x},\ x \in V(P_i) \cup H_i\rangle)$, is a tree-visibility model of $P[V(P_i) \cup H_i]$, where $\vec{\mathcal{R}^j_x}$ is $\vec{\mathcal{S}_x}$ for $x \in V(P_i)$ and $\vec{\mathcal{R}^j_x}$ is the subtree of $\vec{\mathcal{S}_x}$ on $\vec{\mathcal{R}^j}$ for $x \in H_i$. The correctness of the algorithm TREE-VISIBILITY guarantees that $Max(P) - H_i \neq \emptyset$. By induction hypothesis TREE-VISIBILITY$(P[V(P_i) \cup H_i], R, \emptyset)$ computes a tree-visibility model of $P[V(P_i) \cup H_i]$ with a visibility tree of minimal height. Thus the height of the visibility tree computed by the algorithm on the input $P[V(P_i) \cup H_i]$ is at most the height of $\vec{\mathcal{R}^j}$. The order $P[V(P_i) \cup H_i]$ is exactly the order given as parameter to the call of TREE-VISIBILITY corresponding to the connected component $P_i$ of $P - MAX(P)$ (when there is only one component then $H_i = MAX(P) \setminus U(P)$). Note that the set $INC$ is never used for the construction of the visibility tree but only for the label sets. Then for any maximal rooted tree $\vec{T^a}$ of $\vec{T}$ obtained by deleting the root of $\vec{T}$ there exists a maximal rooted tree $\vec{\mathcal{R}^b}$ of $\vec{S}$ whose height is at least the height of $\vec{T^a}$. Thus the minimality of the height of $\vec{\mathcal{S}}$ implies the minimality of the height for $\vec{T}$. ∎

## References

[1] K. P. Bogart, Intervals and Orders: What Comes After Interval Orders?, *Proceedings of the International Workshop on Orders and Algorithms ORDAL'94*, Lecture Notes in Computer Science, Volume 831, Springer-Verlag, Lyon, 1994, pp. 13–32.

[2] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications*, Macmillan, New York, 1976.

[3] U. Faigle, R. Schrader, G. Turán, *The Communication Complexity of Interval Orders*, Discrete Applied Mathematics 40 (1992) 19 –28.

[4] R. Garbe, *Algorithmic Aspects of Interval Orders*, Ph.D. Thesis, University of Twente, October 1994.

[5] F. Gavril *The Intersection Graphs of Subtrees in Trees are Exactly the Chordal Graphs*, Journal of Combinatorial Theory Serie B 16, 47–56 (1974).

[6] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[7] M.C. Golumbic, H. Kaplan, R. Shamir, Algorithms and Complexity of Sandwich Problems in Graphs, *Proceedings of the 19th International Workshop on Graph-Theoretic Concepts in Computer Science WG'93*, Lecture Notes in Computer Science, Volume 790, Springer-Verlag, Berlin, 1994, pp. 57–69.

[8] W.T. Trotter, *Combinatorics and Partially Ordered Sets: Dimension Theory*, The John Hopkins University Press, Baltimore, Maryland, 1992.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399