Intersection Graphs of Non-Crossing Paths

Steven Chaplick

Lehrstuhl für Informatik I, Universität Würzburg, Germany
steven.chaplick@uni-wuerzburg.de

Abstract. We study graph classes modeled by families of non-crossing (NC) connected sets. Two classic graph classes in this context are disk graphs and proper interval graphs. We focus on the cases when the sets are paths and the host is a tree. Forbidden induced subgraph characterizations and linear time certifying recognition algorithms are given for intersection graphs of NC paths of a tree (and related subclasses). For intersection graphs of NC paths of a tree, the dominating set problem is shown to be solvable in linear time. Also, each such graph is shown to have a Hamiltonian cycle if and only if it is 2-connected, and to have a Hamiltonian path if and only if its block-cutpoint tree is a path.

Keywords: Clique Trees · Non-crossing Models · Domination · Hamiltonicity.

1 Introduction

Intersection models of graphs are ubiquitous in graph theory and covered in many graph theory textbooks, see, e.g., [19,28]. Generally, for a given graph $G$, a collection $\mathcal{S}$ of sets, $\{S_v\}_{v \in V(G)}$, is an intersection model of $G$ when $S_u \cap S_v \neq \emptyset$ if and only if $uv \in E(G)$. Similarly, we say that $G$ is the intersection graph of $\mathcal{S}$. One quickly sees that all graphs have intersection models (e.g., by choosing, for every $v \in V(G)$, $S_v$ as the edges incident to $v$). Thus, one often considers restrictions either on the host set (i.e., the domain from which the elements of the $S_v$’s can be chosen), collection $\mathcal{S}$, and/or on the individual sets $S_v$.

In this paper we consider classes of intersection graphs where the sets are taken from a topological space, (path) connected, and pairwise non-crossing. A set $S$ is (path) connected when any two of its points can be connected by a curve within the set (note: a curve is a homeomorphic image of a closed interval). Notice that, when the topological space is a graph, connectedness is precisely the usual connectedness of a graph and curves are precisely paths. Two connected sets $S_1, S_2$ are said to be non-crossing when both $S_1 \setminus S_2$ and $S_2 \setminus S_1$ are connected. Our focus will be on intersection graphs of non-crossing paths.

The most general case of intersection graphs of non-crossing sets which have been studied are those of non-crossing connected (NC-C) sets in the plane [22]. These were considered together with another non-crossing class, the intersection graphs of disks in the plane or simply disk graphs. The recognition of both

* Research supported by DFG grant WO 758/11-1.
NC-C graphs and disk graphs is NP-hard [22]. More recently [21], disk graph recognition was shown to complete for the existential theory of the reals ($\exists R$).

One of the simplest cases of connected sets one can consider are those which reside in $\mathbb{R}$, i.e., the intervals of $\mathbb{R}$. The corresponding intersection graphs are precisely the well studied interval graphs. Moreover, imposing the non-crossing property on these intervals leads to the proper interval graphs [2]. It has often been considered how to generalize proper interval graphs to more complicated hosts, but simple attempts to do so involving the property that the sets are proper are often uninteresting. For example, the intersection graphs of proper paths in trees or proper subtrees of a tree are easily seen as the same as their non-proper versions. We will see that the non-crossing property leads to natural new classes which generalize proper interval graphs.

We formalize the setting as follows. For graph classes $S$ and $H$, a graph $G$ is an $S$-$H$ graph when each $v \in V(G)$ has an $S_v \in S$ such that:
- the graph $H = \bigcup_{v \in V(G)} S_v$ is in $H$, and
- $uv$ is an edge of $G$ if and only if $S_u \cap S_v \neq \emptyset$.

Additionally, we say that $(\{S_v\}_{v \in V(G)}, H)$ is an $S$-$H$ model of $G$ where $H$ is the host and each $S_v$ is a guest. We further state that $G$ is a non-crossing-$S$-$H$ (NC-$S$-$H$) graph when the sets $S_v$ are pairwise non-crossing. In this context the proper interval graphs are the NC-path-path graphs.

Many classes of $S$-$H$ graphs have been studied in the literature; see, e.g., [28]. Some of these are described in the table below together with the complexity of their recognition problems and whether a forbidden induced subgraph characterization (FISC) is known. The table utilizes the following terminology. A directed tree (d.tree) is a tree in which every edge $uv$ has been assigned one direction. A rooted tree (r.tree) is a directed tree where there is exactly one source node. A survey of path-tree graph classes is given in [29].

| Graph Class       | Guest | Host   | Recognition  | FISC? |
|-------------------|-------|--------|--------------|-------|
| 1 interval        | path  | path   | $O(n + m)$   | [7]   |
| 2 rooted path tree (RPT) | path  | r.tree | $O(n + m)$   | [10]  |
| 3 directed path tree (DPT) | path  | d.tree | $O(nm)$     | [4]   |
| 4 path tree (PT)  | path  | tree   | $O(nm)$     | [35]  |
| 5 chordal         | tree  | tree   | $O(n + m)$   | [34]  |

Results and outline. We study the non-crossing classes corresponding to graph classes 1–4 given in the table. Section 2 provides background and notation concerning intersection models. In Section 3 we provide forbidden induced subgraph characterizations for the non-crossing classes corresponding to 1–4 and certifying linear time recognition algorithms for them. Interestingly, this implies that one can test whether a chordal graph contains a claw in linear time. Then, for NC-path-tree graphs, in Section 4 we solve the minimum dominating set (MDS) problem in linear time by showing that there is an independent set which is

1 Note: all $\exists R$-hard problems are NP-hard, see [27] for an introduction to $\exists R$.
2 Usually defined as having no interval strictly contained within any other.
3 A graph is chordal when it has no induced cycles of length four or more.
also an MDS and using a known algorithm [11]. In Section 5, we show that 2-connectedness implies that each plane drawing of the NC-path-tree model leads to a distinct Hamiltonian cycle (HC) which can also be found in linear time, and a similar necessary condition implies the presence of Hamiltonian path (HP). For the MDS, HC and HP problems, we use the special structure of NC-path-tree models obtained in Section 3.1. Notably, the MDS problem is NP-complete on PT graphs [2], and split graphs 4 [8], but it is polynomial time solvable on RPT graphs [2]. Also, the HC and HP problems are NP-complete on strongly chordal split graphs [30], and DPT graphs [31], but easily solved on proper interval graphs [1]. We conclude with avenues for further research.

2 Preliminaries

Notation. Unless explicitly stated otherwise, all the graphs we discuss in this work are connected, undirected, simple, and loopless. For a graph $G$ with a vertex $v$, we use $N_G(v)$ to denote the neighborhood of $v$, and $N_G[v]$ to denote the closed neighborhood of $v$, i.e., $N_G[v] = N_G(v) \cup \{v\}$. The subscript $G$ will be omitted when it is clear. For a subset $S$ of $V(G)$, we use $G[S]$ to denote the subgraph of $G$ induced by $S$. For a set of graphs $\mathcal{F}$, we say that a graph $G$ is $\mathcal{F}$-free when $G$ does not contain any $F \in \mathcal{F}$ as an induced subgraph.

For graph classes $\mathcal{S}$ and $\mathcal{H}$, and an $\mathcal{S}$-$\mathcal{H}$ model $(\{S_v\}_{v \in V(G)}, H)$ of a graph $G$, we use the following notation. We refer to elements of $V(G)$ as vertices and use symbols $u$ and $v$ to refer to them whereas we call elements of $V(H)$ nodes and use $x$, $y$, and $z$ to refer to them. For a node $x$ of $H$ we use $G_x$ to denote the set of vertices in $G$ where $S_v$ contains $x$. Observe that every set $G_x$ induces a clique in $G$. Note that Section 3.1 defines the terms terminal, junction, and mixed that are also used in later sections of the paper.

Several special graphs are named and depicted in Fig. 1 along with models of them. We will refer to these throughout this paper.

Twin-free Graphs. For a graph $G$, two vertices $x$ and $y$ are called twins when they have the same closed neighborhood, i.e., $N[x] = N[y]$. Note that, for the MDS problem, it is an easy exercise to show that it suffices to consider twin-free graphs. Also, as the vertex set of a graph can be easily partitioned into its

4 A graph is a split graph when its vertices can be partitioned into a clique and an independent set. It is easy to see that split graphs are chordal.
equivalence classes of twins in linear time, one can distill the relevant twin-free
induced subgraph of $G$ in linear time.

**Chordality and Clique Trees.** This area is deeply studied and while there
are many interesting results related to our work, we only pick out a few concepts
and results which are useful in this paper. The starting point is that the chordal
graphs are well-known to be the tree-tree graphs $[15,36]$.

For a chordal graph $G$, a *clique tree* $T$ of $G$ has the maximal cliques of $G$ as
its vertices, and for every vertex $v$ of $G$, the set $K_v$ of maximal cliques which
contain $v$ induces a subtree of $T$. In other words, it is a tree-model of $G$ whose
vertices are in bijection with the maximal cliques of $G$. Clique trees are
very useful when discussing models where the host graph is a tree. When a
graph has a tree-tree $[15]$, path-tree $[17]$, path-d-tree $[29]$, path-r-tree $[16]$, or
path-path $[13]$ model, then it also has one that is a clique tree. An overview of
such results can be found in $[28]$.

We establish similar clique tree results for the corresponding NC graphs when
the guests are paths. However, we also note that when the guests are trees, we
cannot rely on clique trees. For example, the claw is an NC-tree-tree graph, but
it does not have an NC-tree-tree model that is a clique tree (see Fig. 1). We
discuss this further in the conclusions.

An important property of clique trees for our linear time algorithms is the
following. For a chordal graph $G$, $\sum_{v \in V(G)} |K_v| \in O(n + m)$ $[19]$. This implies
that the total size of a clique tree $T$ is $O(n + m)$. So, any algorithm that is linear
in the size of $T$ is also linear in the size of $G$. Additionally, one can produce a
clique tree of a chordal graph in linear time $[14]$.

## 3 Non-crossing Paths in Trees: Structure and Recognition

In this section we discuss classes of intersection graphs of non-crossing paths
in trees; namely, NC-path-tree, NC-path-d.tree, NC-path-r.tree, and NC-path-
path. We begin by noting that the *claw* ($K_{1,3}$) is not an NC-path graph regardless
of the host.

**Observation 1** If $G$ is an NC-path graph, then $G$ is claw-free.

**Proof.** Suppose $G$ contains a claw with central vertex $u$ and pendant vertices $a,$
$b, c$. Let $P$ be a path-$H$ model of $G$ where $P = \{P_v\}_{v \in V(G)}$. Clearly, $P_a \cap P_u,$
$P_b \cap P_u$ and $P_c \cap P_u$ are disjoint. As such, at most two of them include an endpoint
of $P_u$. Thus, for some $d \in \{a, b, c\}$, $P_u \setminus P_d$ is disconnected. $\blacksquare$

This section proceeds as follows. The NC-path-tree graphs are shown to be
the claw-free chordal graphs and the structure of NC-path-tree models is de-
scribed. From this structure, we then show that NC-path-d.tree = NC-path-
r.tree = (claw,3-sun)-free chordal. This provides, as a nearly direct consequence,
the classic result that proper interval graphs are precisely the (claw,3-sun,net)-
free chordal graphs $[33]$. We conclude with linear time certifying recognition
algorithms for NC-path-tree and NC-path-r-tree graphs.
3.1 The Structure of NC-path-tree Models

In this subsection we explore the structure of NC-path-tree models and prove our FISCs along the way. We first take a slight detour to claw-free chordal graphs and prove the FISC of NC-path-tree graphs. In doing so we obtain the first insight into NC-path-tree models. Namely, that it suffices to consider clique trees and that the clique trees of these graphs are unique (see Theorem 2). We then take a closer examination of these clique NC-path-tree models and carefully describe the nodes they contain – this will be used repeatedly in the rest of the paper.

Theorem 2. A graph $G$ is claw-free chordal iff it is an NC-path-tree graph. Moreover, $G$ has a unique clique tree and it is an NC-path-tree model.

Proof. $\Leftarrow$ Observation [1] and chordal graphs being tree-tree graphs imply this. 
$\Rightarrow$ Let $T$ be a clique tree of a claw-free chordal graph $G$. We first show that every subtree $T_v$ must be a path, and then we show that these paths are non-crossing. These two claims prove the characterization. The uniqueness of the clique tree of every claw-free chordal graph has been shown previously [23].

Claim 1: For every $v \in V(G)$, $T_v$ is a path.

Suppose $T_v$ is not a path. Then $T_v$ contains some claw $x_0, x_1, x_2, x_3$ with center $x_0$. However, since $G_{x_j}$ is a maximal clique (for each $j \in \{0, 1, 2, 3\}$), for each $i \in \{1, 2, 3\}$, there is $v_i \in G_{x_i} \setminus G_{x_0}$. Thus $v, v_1, v_2, v_3$ induces a claw in $G$. \(\triangle\)

Claim 2: The set $\{T_v : v \in V(G)\}$ is non-crossing.

Suppose that $T_u$ intersects $T_v$ but does not include either end of $T_v$. Let $x_1$ and $x_2$ be the endpoints of $T_v$. Now there must be $v_1 \in G_{x_1} \setminus N_G(u)$ and $v_2 \in G_{x_2} \setminus N_G(u)$. That is, $v, u, v_1, v_2$ induces a claw in $G$. \(\triangle\) ■

We now study the structure of the clique NC-path-tree model $((P_v)_{v \in V(G)}, T)$ of a graph $G$. We introduce some terminology. A node $x$ of $T$ is called a terminal when it is a leaf of every path which contains it, i.e., $x$ is not an internal node of any $P_v$. For example, the leaves of $T$ are terminals. Similarly, a node $x$ of $T$ is a junction when it is an internal node of every path which contains it, i.e., $x$ is not a leaf of any $P_v$. A node of $T$ which is neither a terminal nor a junction is called mixed. We now present the main lemma describing $T$ in these terms and an observation connecting these terms with certain induced subgraphs of $G$.

Lemma 1. For an NC-path-tree graph $G$, let $((P_v)_{v \in V(G)}, T)$ be its clique NC-path-tree model. A node $x$ of $T$ must satisfy the following properties:

1. If $x$ is mixed, then $x$ has degree two.
2. If $x$ is a junction, then (i) $x$ has degree three and (ii) $x$’s neighbors are terminals.
3. If $x$ has degree four or more, then $x$ is a terminal.

Proof. 1.: Suppose that $x$ has degree at least 3, is a leaf of $P_v$, and is an internal node of $P_u$. Further, let $y$ be the unique neighbor of $x$ in $P_v$. We see that $P_u$ includes $y$ (otherwise, $P_v$ and $P_u$ cross). Let $y'$ be the neighbor of $x$ in $P_u \setminus P_v$ and let $y''$ be a neighbor of $x$ which is not in $P_u$. Since $G$ is connected, there exists $u' \in G_x \cap G_{y''}$. Furthermore, $x$ is not a leaf of $P_{u'}$ (otherwise, $P_u$ crosses $P_{y''}$). Thus, similarly to $P_u$, $y$ belongs to $P_{u'}$. Now, since $G_x$ and $G_y$ are maximal
cliques, there is \( u'' \in G_x \setminus G_y \). Thus for \( P_{u''} \) to neither cross \( P_u \) nor \( P_{u'} \) it must include both \( y' \) and \( y'' \). However, this means \( P_{u''} \) and \( P_v \) cross. \( \triangle \)

2.: Suppose that \( x \) is a junction and let \( y_1, \ldots, y_k \) be the neighbors of \( x \). Since \( x \) is a junction, for every \( v \in G_x \), \( P_v \) contains exactly two \( y_i \)'s. Thus, if \( k = 2 \), then \( G_x \subseteq G_{y_1} \) contradicting \( T \) being a clique tree. Now suppose \( k \geq 3 \) and consider \( v \in G_x \) where (w.l.o.g.) \( P_v \) contains \( y_1 \) and \( y_2 \). Since \( G \) is connected, there must be \( v' \in G_x \cap G_{y_2} \). Furthermore, (w.l.o.g.) \( P_{v'} \) contains \( y_1 \) (otherwise, \( P_v \) and \( P_{v'} \) cross). Now, since \( G_x \) and \( G_{y_2} \) are maximal cliques, there is \( v'' \in G_x \setminus G_{y_2} \). Notice that \( P_{v''} \) must contain \( y_2 \) and \( y_3 \) in order for \( P_{v''} \) to cross neither \( P_v \) nor \( P_{v'} \). Finally consider any \( u \in G_x \setminus \{v, v', v''\}\) Notice that, in order for \( P_u \) to not cross any of \( P_v, P_{v'}, \) or \( P_{v''} \), it must contain at least two of \( y_1, y_2, y_3 \). In particular, if \( k \geq 4 \), then \( G_x \setminus G_{y_1} = \emptyset \) contradicting \( G \) being connected. Thus, \( k = 3 \) (establishing (i)).

Now, suppose that \( y_1 \) is not a terminal. By 1. and 2.(i), \( y_1 \) is either a junction with degree 3 or mixed with degree 2.

Case 1: \( y_1 \) is a junction with neighbors \( x, z_1, z_2 \). Notice that each of \( P_v \) and \( P_{v'} \) must contain exactly one of \( z_1 \) or \( z_2 \). Moreover, (w.l.o.g.) they both must contain \( z_1 \) otherwise they will cross. However, since \( y_1 \) is a junction, we have vertices \( w, w', w'' \) such that \( P_w \supseteq \{x, y_1, z_1\} \), \( P_{w'} \supseteq \{x, y_1, z_2\} \) and \( P_{w''} \supseteq \{z_1, y_1, z_2\} \). Moreover, both \( P_w \) and \( P_{w'} \) must contain either \( y_2 \) or \( y_3 \). Regardless of this choice, we end up with a crossing between either \( P_{w''} \) and \( P_v \) or \( P_{w''} \) and \( P_{v'} \). Thus, junctions cannot be neighbors.

Case 2: \( y_1 \) has degree 2 and is mixed. Let \( z \) be the neighbor of \( y_1 \) other than \( x \) and let \( w \) be a vertex of \( G \) where \( y_1 \) is not a leaf of \( P_w \), i.e., (w.l.o.g.) \( P_w \supseteq \{z_1, y_1, x, y_2\} \). Notice that, \( P_{v'} \) must also contain \( z \) otherwise \( P_{v'} \) and \( P_w \) would cross. Similarly, since \( P_{v'} \) now contains \( z \), \( P_v \) must also contain \( z \) otherwise \( P_v \) and \( P_{v'} \) would cross. However, now a vertex \( u \in G_{y_1} \setminus G_z \) must have \( P_u = \{y_1\} \) but then \( P_u \) crosses \( P_w \). Thus, no neighbor of a junction is mixed. \( \triangle \)

3.: This follows immediately from 1. and 2.(i). \( \triangle \)

Observation 3 For an NC-path-tree graph \( G \), let \((\{P_v : v \in V(G)\}, T)\) be its clique NC-path-tree model. Let \( x \) be a node of \( T \) of degree at least three.

1. If \( x \) is a junction, then \( G \) contains a 3-sun. Also, if \( G \) is twin-free, \(|G_x| = 3\).

2. If \( x \) is a terminal, then \( G \) contains a net.

Proof. 1.: As in the proof of Lemma \( \text{II} \) 2.(i) a junction \( x \) in \( T \) has three neighbors \( y_1, y_2, y_3 \) and vertices \( v, v', v'' \in G_x \) such that \( P_v \supseteq \{y_1, x, y_2\} \), \( P_{v'} \supseteq \{y_1, x, y_3\} \) and \( P_{v''} \supseteq \{y_1, x, y_3\} \). Additionally, since \( x, y_1, y_2, y_3 \) are maximal cliques, there are vertices \( u_1, u_2, u_3 \in V(G) \) such that \( u_i \in G_x \setminus G_{y_i} \) for each \( i \in \{1, 2, 3\} \). Moreover, all of these vertices are distinct due to their paths being incomparable. Thus, by considering the 3-sun and its clique tree model given in Fig.\( \text{I} \) it is now easy to see that \( G[v, v', v'', u_1, u_2, u_3] \) is a 3-sun. Furthermore, since \( y_1, y_2, y_3 \) are terminals, the paths \( P_v, P_{v'}, P_{v''} \) are the only distinct paths which are possible for vertices in \( G_x \). In other words, every vertex in \( G_x \setminus \{v, v', v''\} \) is a twin of one of \( v, v', \) or \( v'' \). \( \triangle \)
2. Let \( y_1, y_2, y_3 \) be distinct neighbors of \( x \). Since \( G \) is connected and \( x, y_1, y_2, y_3 \) are maximal cliques, we have \( v_i \in G_x \cap G_{y_i} \) and \( u_i \in G_{y_i} \setminus G_x \) for each \( i \in \{1, 2, 3\} \). The \( v_i \)'s are distinct since \( x \) is a terminal, and the \( u_i \)'s are distinct since their paths are disjoint. Thus, by considering the net and its clique tree model given in Fig. 1, it is easy to see that \( G[v_1, v_2, v_3, u_1, u_2, u_3] \) is a net. \( \triangle \)

3.2 Restricted Host Trees

Here we relate and characterize the classes of NC-path-d.tree, NC-path-r.tree, and NC-path-path graphs as stated in the next two theorems. The proofs are in the appendix and follow from Theorem 2, Lemma 1, and Observation 3.

**Theorem 4.** A graph \( G \) is (claw,3-sun)-free chordal if and only if it is NC-path-r.tree. Moreover, a graph has an NC-path-d.tree model if and only if it has a clique NC-path-r.tree.

**Theorem 5.** A graph \( G \) is (claw,3-sun,net)-free chordal if and only if it is NC-path-path, i.e., proper interval.

3.3 Recognition Algorithms

From our characterizations, there are straightforward polynomial-time certifying algorithms for the classes of NC-path-tree and NC-path-r.tree graphs. Specifically, since these classes are characterized as chordal graphs with an additional finite set of forbidden induced subgraphs, we can apply a linear time certifying algorithm for chordal graphs [3], and then apply brute-force search for our additional forbidden induced subgraphs. If no forbidden induced subgraph is found, we can simply construct the unique clique tree of the given graph and it will be an NC-path-tree (or NC-path-r.tree) model as needed to positively certify membership in our classes. However, we can do this more carefully and obtain linear time certifying algorithms as in the next theorem.

**Theorem 6.** The classes NC-path-tree and NC-path-r.tree (= NC-path-d.tree) have linear-time certifying algorithms.

**Proof.** Recall that the size \( \sum_{v \in V(G)} |K_v| \) of a clique tree is \( O(n + m) \) (we use this implicitly throughout the following). First, we run a linear-time certifying algorithm for chordal graphs, e.g., [3]. Then, we construct a clique tree \( T \) in linear-time [14]. We then annotate the clique tree to mark, for each vertex, for each maximal clique \( K \) in \( K_v \), if \( K \) is a leaf or an internal node of the model of \( v \). If some vertex \( v \) uses \( \geq 3 \) cliques as leaves, we produce a claw as in Claim 1 of the proof of Theorem 2. If there is a mixed node \( x \) of degree \( \geq 3 \), then we proceed as in the proof of Lemma 1. This provides us a pair of paths which cross in linear time. Then, proceeding as in Claim 2 in the proof of Theorem 2, we identify a claw. Now all of the nodes of degree \( \geq 3 \) are either terminals or junctions, and we mark them as such. So, if there is a junction \( x \) with degree
≥ 4, we proceed as in Lemma 2.1 to identify a pair of paths which cross and as before to find a corresponding claw. Furthermore, if a junction \( x \) neighbors a non-terminal \( y \), we proceed as in Lemma 2.1 to identify a pair of paths which cross and (again) a corresponding claw.

Now, no crossing between two paths can involve a node of degree ≥ 3. So, it remains just to ensure no crossings occur between such nodes. In particular, since the neighbors of all junctions are terminals, such a crossing must occur on a path connecting two terminals (where all nodes in between are mixed). Let \( x_1, \ldots, x_k \) be such a path. Clearly, this path of cliques represents an interval graph. Moreover, we will find a pair of crossing paths on it precisely when this interval graph is not a proper interval graph. Conveniently, this problem is known to be solvable in linear time [9]. However, to obtain linear time in total (when processing all such paths) we need to be a bit careful. Namely, rather than simply checking whether each \( G[\bigcup_{i=1}^{k-1} G_{x_i}] \) is a proper interval graph, for each such path we create the following auxiliary graph \( G' \). The vertex set of \( G' \) is \( \{u_1, u_k\} \cup \bigcup_{i=2}^{k-1} G_{x_i} \). In \( G' \), for each \( i \in \{2, \ldots, k-1\} \), \( G_{x_i} \) is a clique. Also, \( u_1 \) is adjacent to \( G_{x_1} \cap G_{x_2} \) and \( u_k \) is adjacent to \( G_{x_{k-1}} \cap G_{x_k} \). In this way, the size of \( G' \) can easily be seen as linear in the size of \( G[\bigcup_{i=2}^{k-1} G_{x_i}] \). Moreover, since we only consider paths connecting terminals, each vertex and edge of \( G \) is contained in at most one \( G' \). Finally, observe that \( G' \) is interval and is a proper interval graph if and only if each \( G[\bigcup_{i=2}^{k-1} G_{x_i}] \) is as well. Thus, running the certifying algorithm for proper interval graphs on \( G' \) will provide a claw when \( G' \) is not a proper interval graph, and such a claw is easily mapped back to a claw in \( G \).

This completes the case of NC-path-tree graphs. For NC-path-r.tree graphs, we additionally look for junctions and proceed as in Observation 3.1.

### 4 Minimum Dominating Set

Recall that a dominating set in a graph \( G \) is a subset \( D \) of \( V(G) \) such that every vertex is either in \( D \) or adjacent to a vertex in \( D \). The MDS problem is NP-complete on PT graphs [2], and split graphs [8], and line graphs of planar graphs [37] (which are of course claw-free). Interestingly, the minimum independent dominating set (MIDS) problem can be solved on chordal graphs in linear time [11]. We will show that, for NC-path-tree graphs, the size of an MIDS is the same as the size of an MDS. Thus, by using [11], we can solve the MDS problem on NC-path-tree graphs in linear time. We assume graphs are twin-free here.

**Theorem 7.** For any NC-path-tree graph \( G \), there is an independent dominating set that is also a minimum dominating set. Moreover, such an independent dominating set can be found in linear time.

**Proof.** Let \( (\{P_x\}_{x \in V(G)}, T) \) be the clique NC-path-tree model of \( G \). We root \( T \) at a leaf \( r \) and call the result \( \overrightarrow{T} \). For each node \( x \) of \( \overrightarrow{T} \), let \( p(x) \) denote the parent of \( x \). Now, if there is an MDS \( D \) of \( G \) where each node \( x \) has \( |D \cap G_x| \leq 1 \), then \( D \) is an independent set. For an MDS \( D \), let \( \overrightarrow{T}(D) \) be the subtree of \( \overrightarrow{T} \) that
contains the root and consists strictly of nodes with \(|D \cap G_x| \leq 1\) (if \(|D \cap G_x| \geq 2\), set \(\overline{T} := \emptyset\). Let \(D\) be an MDS of \(G\) where \(\overline{T}(D)\) is maximal (\(\overline{T}(D)\) not strictly contained in \(\overline{T}(D')\) for any other MDS \(D'\)) and secondly, for each node \(x\) of \(\overline{T} \setminus \overline{T}(D)\) where \(p(x) \in V(\overline{T}(D))\), \(|D \cap G_x|\) is minimized.

Suppose that there is a node in \(\overline{T} \setminus \overline{T}(D)\), and let \(x\) be a node of \(\overline{T} \setminus \overline{T}(D)\) whose parent is in \(\overline{T}(D)\) (if \(\overline{T}(D)\) is empty, we set \(x = r\)). By our choice of \(x\), \(|G_x \cap D| \geq 2\), and \(|D \cap G_{p(x)}| \leq 1\). We consider the three cases regarding \(x\), namely, \(x\) being mixed, a terminal, or a junction.

**Case 1:** \(x\) is mixed. Note that \(x\) has exactly one child. Let \(z\) be the closest descendant of \(x\) that is a terminal, and \(P_{x,z} = (x = x_1, \ldots, x_k = z)\) be the \((x,z)\)-path in \(T\). Note that, by Lemma \(\lbrack 1 \rbrack\), each \(x_i\) \((2 \leq i < k)\) is mixed and has degree two. For each vertex \(u\) of \(D \cap G_x\), \(P_u\) contains a prefix \(P^*(u)\) of this path.

Let \(u\) be a vertex of \(D \cap G_x\) where \(|P^*(u)|\) is maximum. Since \(|G_{p(x)} \cap D| \leq 1\), \(u\) cannot belong to \(G_{p(x)}\) as otherwise replacing \(D\) by \(\{u\} \cup (D \setminus (D \cap G_x))\) would result in a smaller dominating set. Similarly, \(D \cap (G_x \setminus G_{p(x)}) = \{u\}\) as otherwise replacing \(D\) by \(\{u\} \cup (D \cap G_{p(x)})\cup (D \setminus (D \cap G_x))\) would result in a smaller dominating set. Thus, \(D \cap G_x = \{u, v\}\) where \(v\) is a vertex of \(G_{p(x)}\). Now, let \(u'\) be any vertex of \(G_{x_1 \setminus G_x}\). In order for the path \(P_{u'}\) of \(u'\) to not cross \(P_u\), \(P_{u'}\) must contain \(P^*(u) \setminus \{x_1\}\). Now, replacing \(u\) by \(u'\) in \(D\) results in an MDS \(D'\) where \(\overline{T}(D')\) strictly contains \(\overline{T}(D)\), contradicting the choice of \(D\).

**Case 2:** \(x\) is a junction. Let \(y_1\) and \(y_2\) be the two children of \(x\). Recall that, as \(G\) is twin-free, by Lemma \(\lbrack 1 \rbrack\), \(G_x\) contains exactly three vertices \(v_1, v_2, v_1, 2\) and these vertices have the paths \((p(x), x, y_1), (p(x), x, y_2)\), and \((y_1, y, y_2)\) respectively. Since \(|D \cap G_{p(x)}| \leq 1\), \(D\) does not contain both of \(v_1\) and \(v_2\). So, since \(|D \cap G_x| \geq 2\), w.l.o.g., suppose that \(D\) contains \(v_1\) and \(v_1, 2\). However, now, as \(y_2\) is a terminal, replacing \(v_1, 2\) by any vertex of \(G_{y_2} \setminus G_{x_1}\) results in a new MDS \(D'\) in which \(\overline{T}(D')\) is strictly larger than \(\overline{T}(D)\) as \(\overline{T}(D)\) \(\cup \{x\} \leq \overline{T}(D')\).

**Case 3:** \(x\) is a terminal. Note that, since \(|D \cap G_x| \geq 2\), \(D\) cannot contain a vertex \(v\) where \(P_v = (x)\) as this would contradict the minimality of \(D\). So, since \(x\) is a terminal, for every vertex \(v \in D \cap G_x\), \(P_v\) must contain exactly one neighbor of \(x\). Let \(y_1, \ldots, y_l\) be the children of \(x\) where \(|D \cap G_x \cap G_{y_l}| \geq 1\).

Suppose some \(y_i\) is a junction. Let \(z_1\) and \(z_2\) be the children of \(y_i\), and let \(u_1, u_2, u_1, 2\) be the vertices of \(G_{y_2}\) where \(P_{u_1} = (x, y_1, z_1), P_{u_2} = (x, y_1, z_2), P_{u_1, 2} = (z_1, y_2, z_2)\) respectively. Suppose, w.l.o.g., that \(u_1\) is in \(D\). Now, either there is \(v \in D \cap (G_x \setminus G_{y_2})\) or \(u_2\) is also in \(D\). In both cases replacing, \(u_1\) with \(u_1, 2\) leads to a contradiction in our choice of \(D\) (either due to the maximality of \(\overline{T}(D)\) or due to the second condition). Thus, no \(y_i\) is a junction.

So, \(y_1\) is not a junction. We observe that \(|D \cap G_x \cap G_{y_1}| = 1\) as follows. Suppose we have \(u, u' \in D \cap G_x \cap G_{y_1}\). Then, since \(x\) is a terminal and \(y_1\) is not a junction, w.l.o.g., \(P_u\) contains \(P_{u'}\), i.e., we must have \(u = u'\). Note that, for each \(u^* \in G_{y_1} \setminus G_x\), in order for \(P_{u^*}\) to not cross \(P_u\), \(P_{u^*}\) must extend as least as far down \(\overline{T}\) as \(P_u\). Now, since \(|D \cap G_x| \geq 2\) there is a vertex \(v\) which is either in \(D \cap G_x \cap G_{p(x)}\) or in \(D \cap G_x \cap G_{y_2}\). Thus, due to the presence of \(v\), by replacing \(u\) with \(u^*\) we obtain a new MDS that contradicts our choice of \(D\).
5 Hamiltonian Cycles and Paths

As mentioned before the HC and HP problems are NP-complete on DPT graphs and split graphs. They are also NP-complete on line graphs of bipartite graphs, i.e., (claw, diamond, odd-hole)-free graphs [24]. In contrast, we show that, like proper interval graphs [1], 2-connectivity suffices for Hamiltonicity in NC-path-tree graphs, but additionally, every tracing of a clique NC-path-tree model provides a distinct HC of its graph. We similarly characterize the presence of an HP.

Theorem 8. An NC-path-tree graph $G$ has a Hamiltonian cycle if and only if it is 2-connected and has at least three vertices. Also, for each plane layout of $G$’s clique NC-path-tree model $T$, we obtain a distinct a Hamiltonian cycle of $G$.

Proof. We build on the fact that 2-connected proper interval graphs are not only Hamiltonian but have an HC with quite special structure, established in [1], and described as follows. Consider a proper interval graph $G$.

Let $x_1, \ldots, x_k$ be the maximal cliques $G$ ordered according to the clique NC-path-path model of $G$. Further, let $u_1$ be a vertex of $G_{x_1} \setminus G_{x_2}$ and let $u_k$ be a vertex of $G_{x_k} \setminus G_{x_{k-1}}$. When $G$ is 2-connected there are internally disjoint $(u_1, u_k)$-paths $P_1$ and $P_2$ such that every vertex of $G$ belongs to either $P_1$ or $P_2$. In essence, we will see (through an auxiliary multigraph $Q$ constructed below) that such paths also occur in 2-connected NC-path-tree graphs by considering the proper interval graphs occurring between terminals.

Now consider a 2-connected NC-path-tree graph $G$ and its clique NC-path-tree model $T$. Recall that, as we noted when designing our certifying algorithm for NC-path-tree graphs, for a path $x_1, \ldots, x_k$ in $T$ where $x_1$ and $x_k$ are terminals and each inner node is mixed, the graph $G[\bigcup_{i=1}^{k} G_{x_i}]$ is a proper interval graph. Moreover, since $G$ is 2-connected, each such subgraph is also 2-connected. Additionally, the graph $G'$ created from $G[\bigcup_{i=1}^{k} G_{x_i}]$ as before is also 2-connected. However, there is one special case where we use a slightly different auxiliary graph (otherwise we simply use the $G'$ defined before). When $k = 2$, the graph $G'$ is the clique $G_{x_1} \cap G_{x_2}$ together with new vertices $u_1$ and $u_k$ where $N(u_1) = N(u_k) = G_{x_1} \cap G_{x_2}$. Now, it is easy to see that our graphs $G'$ are 2-connected and proper interval, and since $u_1$ and $u_k$ are not adjacent, we have two non-empty disjoint paths that both start with a vertex of $G_{x_1} \cap G_{x_2}$, and end with a vertex of $G_{x_{k-1}} \cap G_{x_k}$.

We now consider the case when a neighbor $y$ of $x$ is a junction before completing our construction of the HC. Let the other two neighbors of the junction $y$ be $x'$ and $x''$. Due to the fact that $x, x', x''$ are all terminals, the vertices of $G_y$ form three equivalence classes $A, A', A''$ of twins, where each vertex in $A$ is represented by the path $x, y, x'$, each vertex in $A'$ is represented by the path $x', y, x''$, and each vertex in $A''$ is represented by the path $x'', y, x$. Namely, using $A, A', A''$ we can “traverse” from $x$ to $x'$, from $x'$ to $x''$, and from $x''$ back to $x$.

Based on the above observations, we can now build our HCs. To do this we will trace the outline of $T$ by using the paths guaranteed by the above arguments. This trace can be described by a multigraph $Q$ formed on the terminals of $T$. 
where each Eulerian tour of $Q$ will correspond to a distinct HC of $G$. Namely, for each terminal $x$, and each neighbor $y$ of $x$:

- if $y$ is a terminal, then in $Q$, $x$ and $y$ are connected by two edges (representing the two paths present in the corresponding $G'$).
- if $y$ is a mixed node and $z$ is the terminal so that $y$ occurs on the $(x, z)$-path in $T$, then, in $Q$, $x$ and $z$ are connected by two edges (representing the two paths present in the corresponding $G'$).
- if $y$ is a junction and $x'$ and $x''$ are its two other neighbors, then in $Q$, we have the edges $xx'$ and $xx''$.
- finally, if $G_x$ contains vertices that do not belong to any other $G_{x'}$ (e.g., when $x$ is a leaf of $T$), we also add a self-loop to $x$ and map to this self-loop the vertices of $G_x \setminus (\bigcup_{x' \in N(x)} G_{x'})$.

We note the following properties of $Q$ to complete the proof. The edges of $Q$ partition the vertices of $G$ and each edge $xy$ corresponds to a path in $G$ where the first vertex belongs $G_x$ and the last vertex belongs to $G_y$. Furthermore, $Q$ is Eulerian, each Eulerian cycle $C$ provides an HC, and $C$ describes a plane layout of $T$, i.e., a cyclic order of the edges around each node of $T$ so that $C$ traces the outline of this plane layout of $T$. Note that, each such plane layout will often arise from multiple Eulerian cycles in $Q$, but no two distinct layouts arise from the same cycle.

The \textit{block-cutpoint} tree $BC(G)$ of a graph $G$ contains a node for each cut-vertex of $G$, a node for each maximal 2-connected subgraph (\textit{block}) of $G$, and its edge set is $\{cB : c$ is a cut-vertex, and $B$ is a block of $G$ containing $c\}$. It is well-known that $BC(G)$ can be computed in linear time \cite{20}, and is indeed always a tree. Clearly, if $G$ has an HP, $BC(G)$ is a path. We show that this is sufficient to have an HP in NC-path-tree graphs. The main idea is to observe where the cut-vertices occur in the model and then reuse our Eulerian structure $Q$ from the previous proof (see the appendix for the proof).

\textbf{Theorem 9.} An NC-path-tree graph $G$ contains a Hamiltonian path if and only if its block-cutpoint tree is a path.

6 Concluding Remarks

A natural next step would be to study the NC-tree-tree graphs. As we have mentioned, it is not safe to simply work with clique trees in this case as the claw requires the use of a non-clique tree model to avoid crossings. We conjecture that the NC-tree-tree graphs can be characterized as chordal graphs avoiding finite set of forbidden induced subgraphs.

Other host domains have been considered in the literature. Notice that similar to proper interval graphs being NC-path-path graphs, the proper circular arc graphs are precisely the NC-path-cycle graphs. A simple host graph class which generalizes both trees and cycles is that of \textit{cacti}. A \textit{cactus} is a connected graph in which every 2-connected component is a single vertex, a single edge or a chordless cycle. The intersection graphs of subtrees of a cactus were studied by Gavril \cite{18}. So, one might consider the NC-path/tree/cactus-cactus graphs.
Finally, an alternative view of host domains has been considered quite recently through the notion of $H$-graphs [3,5,6,12], i.e., for a fixed graph $H$, a graph $G$ is an $H$-graph when it is an intersection graph of connected subgraphs of a subdivision of $H$. Here, interval graphs are the $K_2$-graphs and circular arc graphs are the $K_3$-graphs. While there is a natural notion of proper $H$-graphs [3] and these do indeed restrict $H$-graphs (for every $H$), we believe that the more restrictive non-crossing $H$-graphs might have a nicer structure and lead to easier (and more efficient) algorithms.

References
1. Bertossi, A.A.: Finding Hamiltonian circuits in proper interval graphs. Inf. Process. Lett. 17(2), 97–101 (1983). https://doi.org/10.1016/0020-0190(83)90078-9
2. Booth, K.S., Johnson, J.H.: Dominating sets in chordal graphs. SIAM J. Comput. 11(1), 191–199 (1982). https://doi.org/10.1137/0211015
3. Chaplick, S., Fomin, F., Golovach, P., Knop, D., Zeman, P.: Kernelization of graph hamiltonicity: Proper $H$-graphs. In: WADS. LNCS, vol. 11646, p. to appear. Springer (2019)
4. Chaplick, S., Gutierrez, M., Léveque, B., Tondato, S.B.: From path graphs to directed path graphs. In: WG, LNCS, vol. 6410, pp. 256–265. Springer (2010). https://doi.org/10.1007/978-3-642-16926-7_24
5. Chaplick, S., Toepfer, M., Voborník, J., Zeman, P.: On $H$-topological intersection graphs. In: WG. LNCS, vol. 10520, pp. 167–179. Springer (2017). https://doi.org/10.1007/978-3-319-68705-6_13
6. Chaplick, S., Zeman, P.: Combinatorial problems on $H$-graphs. In: EUROCOMB. vol. 61, pp. 223–229 (2017). https://doi.org/10.1016/j.endm.2017.06.042
7. Corneil, D.G., Olariu, S., Stewart, L.: The LBFS structure and recognition of interval graphs. SIAM Journal of Discrete Mathematics 23(4), 1905–1953 (2009). https://doi.org/10.1137/08075480108373455
8. Corneil, D.G., Perl, Y.: Clustering and domination in perfect graphs. Discrete Applied Mathematics 9(1), 27–39 (1984). https://doi.org/10.1016/0166-218X(84)90088-X
9. Deng, X., Hell, P., Huang, J.: Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. SIAM J.Comput. 25(2), 390–403 (1996). https://doi.org/10.1137/S0097539792269095
10. Dietz, P.F.: Intersection graph algorithms. Ph.D. thesis, Cornell University (1984)
11. Farber, M.: Independent domination in chordal graphs. Operations Research Letters 1(4), 134 – 138 (1982). https://doi.org/10.1016/0167-6377(82)90015-3
12. Fomin, F.V., Golovach, P.A., Raymond, J.: On the tractability of optimization problems on $H$-graphs. In: ESA. LIPIcs, vol. 112, pp. 30:1–30:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018). https://doi.org/10.4230/LIPIcs.ESA.2018.30
13. Fulkerson, D., Gross, O.: Incidence matrices and interval graphs. Pacific Journal of Mathematics 15, 835–855 (1965)
14. Galinier, P., Habib, M., Paul, C.: Chordal graphs and their clique graphs. In: WG, LNCS, vol. 1017, pp. 358–371. Springer (1995). https://doi.org/10.1007/3-540-60618-1_88
15. Gavril, F.: The intersection graphs of subtrees of trees are exactly the chordal graphs. Journal of Combinatorial Theory Series B 16, 47–56 (1974)
16. Gavril, F.: A recognition algorithm for the intersection graphs of directed paths in directed trees. *Discrete Mathematics* **13**(3), 237 – 249 (1975). [https://doi.org/10.1016/0012-365X(75)90021-7](https://doi.org/10.1016/0012-365X(75)90021-7)

17. Gavril, F.: A recognition algorithm for the intersection of graphs of paths in trees. *Discrete Mathematics* **23**, 211–227 (1978)

18. Gavril, F.: Intersection graphs of helly families of subtrees. *Discrete Applied Mathematics* **66**, 45–56 (1996)

19. Golumbic, M.: Algorithmic graph theory and perfect graphs. *Annals of discrete mathematics*, Elsevier (2004)

20. Hopcroft, J., Tarjan, R.: Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM* **16**(6), 372–378 (Jun 1973). [https://doi.org/10.1145/362248.362272](https://doi.org/10.1145/362248.362272)

21. Kang, R.J., Müller, T.: Sphere and dot product representations of graphs. *Discrete & Computational Geometry* **47**(3), 548–568 (Apr 2012)

22. Kratochvíl, J.: Intersection graphs of noncrossing arc-connected sets in the plane. In: GD, LNCS, vol. 1190, pp. 257–270. Springer (1997). [https://doi.org/10.1007/3-540-62495-3_53](https://doi.org/10.1007/3-540-62495-3_53)

23. Kumar, P., Madhavan, C.: Clique tree generalization and new subclasses of chordal graphs. *Discrete Applied Mathematics* **117**(13), 109 – 131 (2002). [https://doi.org/10.1016/S0166-218X(00)00336-X](https://doi.org/10.1016/S0166-218X(00)00336-X)

24. Lai, T., Wei, S.: The edge hamiltonian path problem is NP-complete for bipartite graphs. *Inf. Process. Lett.* **46**(1), 21–26 (1993). [https://doi.org/10.1016/0020-0190(93)90191-B](https://doi.org/10.1016/0020-0190(93)90191-B)

25. Lekkerkerker, C., Boland, J.: Representation of a finite graph by a set of intervals on a real line. *Fundamenta Mathematicae* **51**(1), 45–64 (1962)

26. Lévêque, B., Maffray, F., Preissmann, M.: Characterizing path graphs by forbidden induced subgraphs. *Journal of Graph Theory* **62**, 4 (2009)

27. Matousek, J.: Intersection graphs of segments and $\mathbb{R}^3$. *CoRR* abs/1406.2636 (2014), [http://arxiv.org/abs/1406.2636](http://arxiv.org/abs/1406.2636)

28. McKee, T., McMorris, F.: Intersection Graph Theory. SIAM (1999)

29. Monma, C.L., Wei, V.K.: Intersection graphs of paths in a tree. *Journal of Combinatorial Theory, Series B* **41**(2), 141–181 (1986)

30. Müller, H.: Hamiltonian circuits in chordal bipartite graphs. *Discrete Mathematics* **156**(1-3), 291–298 (1996). [https://doi.org/10.1016/0012-365X(95)00127-7](https://doi.org/10.1016/0012-365X(95)00127-7)

31. Narasimhan, G.: A note on the hamiltonian circuit problem on directed path graphs. *Inf. Process. Lett.* **32**(4), 167–170 (1989). [https://doi.org/10.1016/0020-0190(89)90038-0](https://doi.org/10.1016/0020-0190(89)90038-0)

32. Panda, B.S.: The forbidden subgraph characterization of directed vertex graphs. *Discrete Mathematics* **196**(1-3), 239–256 (1999). [https://doi.org/10.1016/S0012-365X(98)00127-7](https://doi.org/10.1016/S0012-365X(98)00127-7)

33. Roberts, F.S.: On nontransitive indifference. *Journal of Mathematical Psychology* **7**(2), 243 – 258 (1970). [https://doi.org/10.1016/0022-2496(70)90047-7](https://doi.org/10.1016/0022-2496(70)90047-7)

34. Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.* **5**(2), 266–283 (1976). [https://doi.org/10.1137/0205021](https://doi.org/10.1137/0205021)

35. Schäffer, A.A.: A faster algorithm to recognize undirected path graphs. *Discrete Applied Mathematics* **43**, 261–295 (June 1993). [https://doi.org/10.1016/0166-218X(93)90116-6](https://doi.org/10.1016/0166-218X(93)90116-6)

36. Walter, J.R.: Representations of chordal graphs as subtrees of a tree. *Journal of Graph Theory* **2**(3), 265–267 (1978). [https://doi.org/10.1002/jgt.3190020311](https://doi.org/10.1002/jgt.3190020311)

37. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics* **38**(3), 364–372 (1980)
Proofs omitted from the main text.

Proof (of Theorem 7). \(\Leftarrow\) It is known and easy to see that the 3-sun is not a path-d.tree graph \([4]\). Thus, NC-path-d.tree is a subclass of (3-sun)-free NC-path-tree by Lemma 1.1.

\(\Rightarrow\) By Theorem 2 and Observation 3, for every (3-sun,claw)-free chordal graph \(G\), in every clique NC-path-tree model \(\{P_v\}_{v \in V(G)}\) of \(G\), \(T\) has no junctions. Thus, rooting \(T\) at any terminal results in an NC-path-r.tree model.

Proof (of Theorem 8). \(\Leftarrow\) It is known and easy to see that the net is not a path-path (interval) graph \([25]\). Thus, NC-path-path is a subclass of (net)-free NC-path-r.tree by Theorem 3.

\(\Rightarrow\) As in the proof of Theorem 4, we note that since \(G\) is a (net,3-sun)-free NC-path-tree graph, by Observation 3, its unique clique NC-path-tree model has maximum degree two. Thus, the host is a path.

Proof (of Theorem 9). As noted in Section 5 it suffices to prove the \(\Leftarrow\) direction. Let \(\{\{P_v\}_{v \in V(G)}\}, T\) be the clique NC-path-tree model of a graph \(G\). Consider a cut-vertex \(v\) of \(G\). Note that \(P_v\) must contain an edge \(xy\) which is not used by any \(P_u\) for \(u \neq v\) (otherwise, \(v\) would not be a cut-vertex). The next claim is the key to the proof.

Claim: \(P_v\) is precisely the edge \(xy\) and both \(x\) and \(y\) are terminals.

Note that \(P_v\) cannot contain any junctions as the vertices whose paths use junctions cannot be cut-vertices. Suppose that \(y\) is not an end-node of \(P_v\), and let \(z\) be the neighbor of \(y\) distinct from \(z\) (note: \(y\) is mixed and as such has degree 2 by Lemma 1.1). Now, since \(G_z\) and \(G_y\) are maximal cliques, we have a vertex \(u \in G_y \setminus G_z\), but \(u\) cannot belong to \(G_x\) since \(u \neq v\) and \(P_v\) is the only path containing \(xy\). Thus, \(P_u = y\) and \(P_u\) and \(P_v\) cross. Furthermore, since \(P_v\) is the only path that uses \(xy\) and both \(x\) and \(y\) must be terminals.

Let \(B\) be a block of \(G\) that contains a single cut-vertex \(v\). Since \(B\) is 2-connected, by Theorem 8 \(B\) has a Hamiltonian cycle \(C_B\). So, to obtain a Hamiltonian path that ends at \(v\), we just delete one edge incident to \(v\) from \(C_B\). We will now argue that each block \(B\) of \(G\) that contains two cut-vertices \(v\) and \(v'\) has a Hamiltonian path that connects \(v\) to \(v'\) within \(B\).

By the claim above, in the clique NC-path-tree model \(\{\{P_B\}_{v \in V(B)}\}, T\) of \(B\), \(P_B\) and \(P'_B\) are both single nodes, say \(x\) and \(x'\) (respectively) which are terminals in \(T\). Consider the path \(x = x_1, x_2, \ldots, x_k = x'\) in \(T\). Now, consider the Eulerian multigraph \(Q\) as in the proof of Theorem 8. Note that, since \(x_j\) is a terminal, we can use \(Q\) to construct a path \(P_1\) that starts with \(v\) and ends with a vertex of \(G_{x_1} \setminus G_{x_2}\) and visits precisely the vertices in the connected component of \(B\setminus (G_{x_1} \cap G_{x_2})\) that contains \(v\). The path \(P_2\) is defined analogously. Similarly, for each terminal \(x_i\) (\(i \in \{2, \ldots, k - 1\}\)), we can use \(Q\) to craft a path \(P_i\) that visits precisely the vertices whose paths occur strictly within the subtree of \(T\setminus \{x_{i-1}, x_{i+1}\}\) that contains \(x_i\). Moreover, this path will start and end with vertices whose paths contain \(x_i\). When \(x_i\) is a junction, let \(x'_i\) be the terminal neighbor of \(x_i\) distinct from \(x_{i-1}\) and \(x_{i+1}\). Similarly to the case of \(x_1\), we note
that there is a path $P'_i$ that visits all the vertices $U_i$ “hanging below” $x'_i$ and starts and ends with a vertex of $x'_i$. Additionally, due to the three equivalence classes of twins whose paths contain the junction $x_i$, we can extend this path $P'_i$ to a path $P_{i-1,i+1}$ that starts in a vertex of $G_{x_{i-1}} \cap G_{x_i}$, ends in a vertex of $G_{x_i} \cap G_{x_{i+1}}$, and visits every vertex of $U_i \cup G_{x_i}$.

Finally, consider two terminals $x_i$ and $x_j$ ($i < j$) where, for each $l \in \{i + 1, \ldots, j - 1\}$, $x_l$ is mixed. As in the proof of Theorem 8, we again consider the auxiliary graph $G'$ corresponding to this path. Here, we instead need a Hamiltonian path in $G'$ that starts and ends in our special vertices $u_1$ and $u_k$. Fortunately, it is known \[1\], that such a path does exist and actually only requires that $G'$ is connected. Namely, have the path $P_{i,j}$ which starts in a vertex of $G_{x_i} \cap G_{x_{i+1}}$ and ends in a vertex of $G_{x_{k-1}} \cap G_{x_k}$ and visits every vertex of $\bigcup_{j=1}^{k-1} G_{x_j}$.

Thus, to form a Hamiltonian path of $B$ that starts with $v$ and ends with $v'$, we simply glue together the paths $P_i, P_{3,i}, P_{4,i}, P_{5,i}, \ldots, P_{t,i}, P_{i,k}, P_k$ where $x_{i_1}, \ldots, x_{i_t}$ are the terminals that occur between $x_1$ and $x_k$. In particular, by forming these Hamiltonian paths for each block of $G$, we are done. \[\square\]