A Fully Dynamic Algorithm for the Recognition of P_4-sparse Graphs

Stavros D. Nikolopoulos[1] Leonidas Palios[1] Charis Papadopoulos[2,*]

[1]*Department of Computer Science, University of Ioannina*
P.O.Box 1186, GR-45110 Ioannina, Greece
{stavros, palios}@cs.uoi.gr

[2]*Department of Informatics, University of Bergen*
P.B. 7800, N-5020 Bergen, Norway
charis@ii.uib.no

Abstract: In this paper, we solve the dynamic recognition problem for the class of P_4-sparse graphs: the objective is to handle edge/vertex additions and deletions, to recognize if each such modification yields a P_4-sparse graph, and if yes, to update a representation of the graph. Our approach relies on maintaining the modular decomposition tree of the graph, which we use for solving the recognition problem. We establish properties for each modification to yield a P_4-sparse graph and obtain a fully dynamic recognition algorithm which handles edge modifications in $O(1)$ time and vertex modifications in $O(d)$ time for a vertex of degree d. Thus, our algorithm implies an optimal edges-only dynamic algorithm and a new optimal incremental algorithm for P_4-sparse graphs. Moreover, by maintaining the children of each node of the modular decomposition tree in a binomial heap, we can handle vertex deletions in $O(\log n)$ time, at the expense of needing $O(\log n)$ time for each edge modification and $O(d \log n)$ time for the addition of a vertex adjacent to d vertices.

Keywords: fully dynamic algorithm, P_4-sparse graph, modular decomposition, recognition.

1 Introduction

A *dynamic graph* algorithm for a class Π of graphs is an algorithm that handles a series of on-line modifications (i.e., insertions or deletions of vertices or edges) on a graph in Π; if the modification result in a graph in Π, the algorithm performs it (updating an internal representation), otherwise it outputs `false` and does nothing. Such algorithms are categorized depending on the modification operations they support: an *incremental* (*decremental*) algorithm supports only vertex insertions (deletions); an *additions-only* (*deletions-only*) algorithm supports only edge additions (deletions); an *edges-only fully dynamic* algorithm supports both edge additions and edge deletions; a *fully dynamic* algorithm supports all edge as well as all vertex modifications.

Several authors have studied the dynamic recognition problem for graphs of specific families. Incremental recognitions algorithms have been proposed by Hsu [12] for interval graphs and by Deng *et al.* [8] for connected proper interval graphs. Ibarra [13] has given an edges-only fully dynamic algorithm for chordal graph recognition which handles each edge operation in $O(n)$ time and an edges-only fully dynamic algorithm for split graph recognition which handles each edge operation in $O(1)$ time. More recently, Hell *et al.* [10] have given a fully dynamic algorithm for recognizing proper interval graphs which works in $O(d + \log n)$ time per modification, where d is the degree of a vertex in case of a vertex modification; Shamir and Sharan [19] have developed a fully dynamic algorithm for the recognition

[*] Work by Charis Papadopoulos was carried while he was a graduate student at the University of Ioannina.

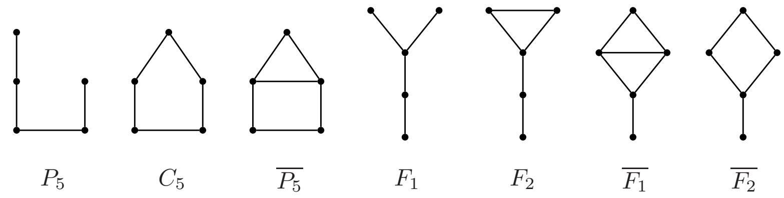

Figure 1: The seven forbidden subgraphs for the class of P_4-sparse graphs.

of cographs, threshold graphs and trivially perfect graphs which handles edge modifications in $O(1)$ time and vertex modifications in $O(d)$ time; finally, Crespelle and Paul have presented fully dynamic algorithms for directed cographs [5] and permutation graphs [6] which require $O(d)$ time if d arcs are involved, and $O(n)$ time, respectively. For the class of P_4-sparse graphs, an incremental algorithm for recognizing a P_4-sparse graph has been proposed by Jamison and Olariu [15] which handles the insertion of a vertex of degree d in $O(d)$ time.

Researchers have also considered the problem of the dynamic maintenance of the modular decomposition tree of a graph (the modular decomposition tree of a graph G is a unique (up to isomorphism) labeled tree which records all the partitions of the vertex set of G into modules and can be constructed in time and space linear in the size of the graph [4, 7, 17]): Muller and Spinrad [18] have given an incremental algorithm for the modular decomposition, which handles each vertex insertion in $O(n)$ time; Corneil *et al.* [3] have given an optimal incremental algorithm for the recognition and modular decomposition of cographs, which handles the insertion of a vertex of degree d in $O(d)$ time.

Our work in this paper focuses on P_4-sparse graphs; the *P_4-sparse graphs* are defined as the graphs for which every set of five vertices induces at most one chordless path on four vertices [11] (Figure 1 depicts the 7 forbidden subgraphs for the class of P_4-sparse graphs). They are perfect and also perfectly orderable [11], and properly contain many graph classes, such as, the cographs, the P_4-reducible graphs, etc. (see [1, 15, 16]). The P_4-sparse graphs have received considerable attention in recent years and they find applications in applied mathematics and computer science (e.g., communications, transportation, clustering, scheduling, computational semantics) in problems that deal with graphs featuring "local density" properties. Indeed, the structure of P_4-sparse graphs incorporates such local density properties since they are graphs that are unlikely to have more than a few P_4s; we note that the notion of local density is often associated with the absence of P_4s.

In this paper, we describe a fully dynamic algorithm for the class of P_4-sparse graphs. Our algorithm maintains the modular decomposition tree of the graph; it checks whether the requested edge/vertex operations yield a P_4-sparse graph, and if yes, it updates the modular decomposition tree. Edge operations are handled in $O(1)$ time while vertex operations are handled in $O(d)$ time. As a result, we obtain an optimal edges-only dynamic algorithm and a new optimal incremental algorithm for P_4-sparse graphs. Moreover, in order to improve the time complexity of the vertex deletion operation, we can maintain the children of each node of the modular decomposition tree in a binomial heap [2]. Then, we can handle vertex deletions in $O(\log n)$ time; the drawback is that then the time required for each edge modification becomes $O(\log n)$ and for the addition of a vertex adjacent to d vertices becomes $O(d \log n)$.

2 Theoretical Framework

Let G be a simple graph; we denote by $V(G)$ and $E(G)$, the vertex and edge set of G. The subgraph of G induced by a set $S \subseteq V(G)$ is denoted by $G[S]$. If a vertex u is adjacent to a vertex v, we say that u *sees* v, otherwise, we say that it *misses* v; more generally, a vertex set A sees (misses resp.) a vertex set B, if every vertex in A sees (misses resp.) every vertex in B.

A graph class Π is called complement-invariant if $G \in \Pi$ implies $\overline{G} \in \Pi$. We note that for the P_4-sparse graphs the following holds:

Lemma 2.1. *Let G be a P_4-sparse graph. Then (i) G has the complement-invariant property, and (ii) for every $v \in G$, $G' = G - v$ is a P_4-sparse graph.*

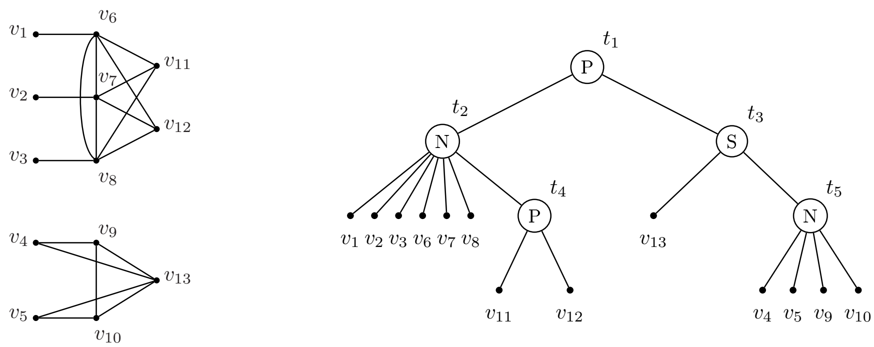

Figure 2: A disconnected P_4-sparse graph on 13 vertices and its md-tree.

Modular Decomposition and P_4-sparse Graphs. A subset M of vertices of a graph G is said to be a *module* of G, if every vertex outside M is either adjacent to all vertices in M or to none of them. The emptyset, the singletons, and the vertex set $V(G)$ are *trivial* modules and whenever G has only trivial modules it is called a *prime* (or *indecomposable*) *graph*. A module M of G is called a *strong module* if, for any module M' of G, either $M' \cap M = \emptyset$ or one module is included into the other. Furthermore, a module in G is also a module in $\overline{G}$.

The *modular decomposition* of a graph G is a linear-space representation of all the partitions of $V(G)$ where each partition class is a module. The *modular decomposition tree* $T(G)$ of the graph G (or *md-tree* for short) is a unique (up to isomorphism) labeled tree associated with the modular decomposition of G in which the leaves of $T(G)$ are the vertices of G and the set of leaves associated with the subtree rooted at an internal node induces a strong module of G (Figure 2). Thus, the md-tree $T(G)$ represents all the strong modules of G. It is known that for every graph G the md-tree $T(G)$ can be constructed in linear time [4, 7, 17].

Let t be an internal node of the md-tree $T(G)$ of a graph G. We denote by $M(t)$ the module corresponding to t which consists of the set of vertices of G associated with the subtree of $T(G)$ rooted at node t. The node t is labeled by either P (for *p*arallel module) if the subgraph $G[M(t)]$ is disconnected, S (for *s*eries module) if the complement of $G[M(t)]$ is disconnected, or N (for *n*eighborhood module) otherwise. Let $u_1, u_2, \ldots, u_p$ be the children of the node t of $T(G)$. We denote by $G(t)$ the *representative graph* of the module $M(t)$ defined as follows: $V(G(t)) = \{u_1, u_2, \ldots, u_p\}$ and $u_i u_j \in E(G(t))$ if there exists edge $v_k v_\ell \in E(G)$ such that $v_k \in M(u_i)$ and $v_\ell \in M(u_j)$; by the definition of a module, if a vertex of $M(t_i)$ is adjacent to a vertex of $M(t_j)$ then every vertex of $M(t_i)$ is adjacent to every vertex of $M(t_j)$. Thus, $G(t)$ is isomorphic to the graph induced by a subset of $M(t)$ consisting of a single vertex from each maximal strong submodule of $M(t)$ in the modular decomposition of G. Depending on whether an internal node t of $T(G)$ is a P-, S-, or N-node, the following result holds (see also [9]):

- if t is a P-node, $G(t)$ is an edgeless graph;
- if t is an S-node, $G(t)$ is complete graph;
- if t is an N-node, $G(t)$ is a prime graph.

In particular, for the class of P_4-sparse graphs, Giakoumakis and Vanherpe [9] showed that:

Lemma 2.2. *Let G be a graph and let $T(G)$ be its modular decomposition tree. The graph G is P_4-sparse iff for every N-node t of $T(G)$, $G(t)$ is a prime spider with a spider-partition (S, K, R) and no vertex of $S \cup K$ is an internal node in $T(G)$.*

A graph G is called a *spider* if the vertex set $V(G)$ of the graph G admits a partition into sets S, K, and R such that:

C1: $|S| = |K| \geq 2$, the set S is an independent (stable) set, and the set K is a clique;
C2: all the vertices in R are adjacent to all the vertices in K and to no vertex in S;
C3: there exists a bijection $f : S \longrightarrow K$ such that exactly one of the following statements holds:
  (i) for each vertex $v \in S$, $N(v) \cap K = \{f(v)\}$;
  (ii) for each vertex $v \in S$, $N(v) \cap K = K - \{f(v)\}$.

The triple (S, K, R) is called the *spider-partition*. A graph G is a *prime spider* if G is a spider with $|R| \leq 1$. If the condition of case C3(i) holds, then the spider G is called a *thin spider*, whereas if the condition of case C3(ii) holds then G is a *thick spider*; note that the complement of a thin spider is a thick spider and vice versa. A prime spider with $|S| = |K| = 2$ is simultaneously thin and thick.

3 The Fully-Dynamic Algorithm

As mentioned, our algorithm maintains the modular decomposition tree $T(G)$ of the P_4-sparse graph.

3.1 Adding an Edge

Let uv be the edge to be added and let $G' = G \cup \{uv\}$. For the two vertices u, $v \in G$ we denote by t_{uv} the least common ancestor of u and v in $T(G)$. Since u, v are non-adjacent in G, node t_{uv} is either a P-node or an N-node. Let t_u and t_v be the children of t_{uv} such that $M(t_u)$ and $M(t_v)$ contain the vertices u and v respectively. Note that if $|M(t_u)| = 1$ (resp. $|M(t_v)| = 1$) then $t_u = u$ (resp. $t_v = v$). Without loss of generality, we make the following assumption:

Assumption 3.1. *We assume that* $|M(t_v)| \geq |M(t_u)|$.

We distinguish three cases, namely, (i) $|M(t_u)| \geq 2$, (ii) $|M(t_u)| = 1$ and t_{uv} is a P-node, and (iii) $|M(t_u)| = 1$ and t_{uv} is an N-node; we prove the following lemmata.

Lemma 3.1. *Let* $|M(t_u)| \geq 2$. *Then* G' *is a* P_4*-sparse graph if and only if* t_{uv} *is a P-node and* $|M(t_u)| = |M(t_v)| = 2$.

Proof. Since $|M(t_v)| \geq |M(t_u)| \geq 2$, the node t_{uv} cannot be an N-node because at most one child of any N-node is an internal node (not a leaf) in $T(G)$ (see Figure 2). Thus, t_{uv} is a P-node; then it follows that the subgraphs $G[M(t_u)]$ and $G[M(t_v)]$ are both connected.

The "if"-part of the lemma follows from Figure 3 since the resulting graph G' is indeed P_4-sparse. For the "only if"-part, we have that G' is P_4-sparse and assume for contradiction that at least one of $M(t_u), M(t_v)$ has 3 elements; then, Assumption 3.1 implies that $|M(t_v)| \geq 3$. The connectivity of $G[M(t_u)]$ and $G[M(t_v)]$ implies that there exist vertices $u' \in M(t_u)$ and $v' \in M(t_v)$ such that $uu', vv' \in E(G)$. Then, by adding the edge uv in G, the resulting graph G' contains the P_4 $u'uvv'$. Since $G[M(t_v)]$ is connected and $|M(t_v)| \geq 3$, there exists a vertex x in $M(t_v)$ such that x sees at least one of v, v'. But then, the five vertices u', u, v, v', x induce in G' one of the following graphs: P_5, F_1, or F_2; thus, G' is not P_4-sparse, a contradiction. ∎

Lemma 3.2. *Let* $|M(t_u)| = 1$ *(i.e.,* $M(t_u) = \{u\}$*) and suppose that* t_{uv} *is a P-node. Then* G' *is a* P_4*-sparse graph if and only if one of the following (mutually exclusive) cases holds:*

(i) vertex v *sees all the vertices in* $M(t_v)$*;*

(ii) vertex v *misses exactly one vertex* $y \in M(t_v)$ *such that* y *sees only one vertex* $x \in M(t_v)$*, and only the vertex* x *sees every vertex in* $M(t_v)$*;*

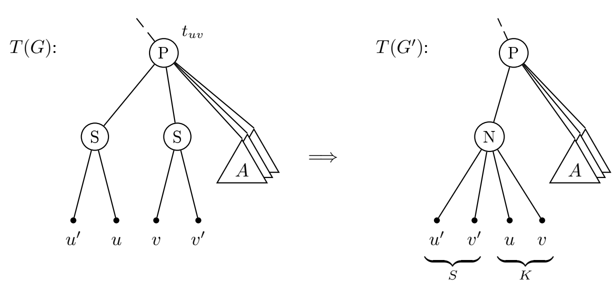

Figure 3: Illustrating the case of Lemma 3.1 and the corresponding updates of the md-tree.

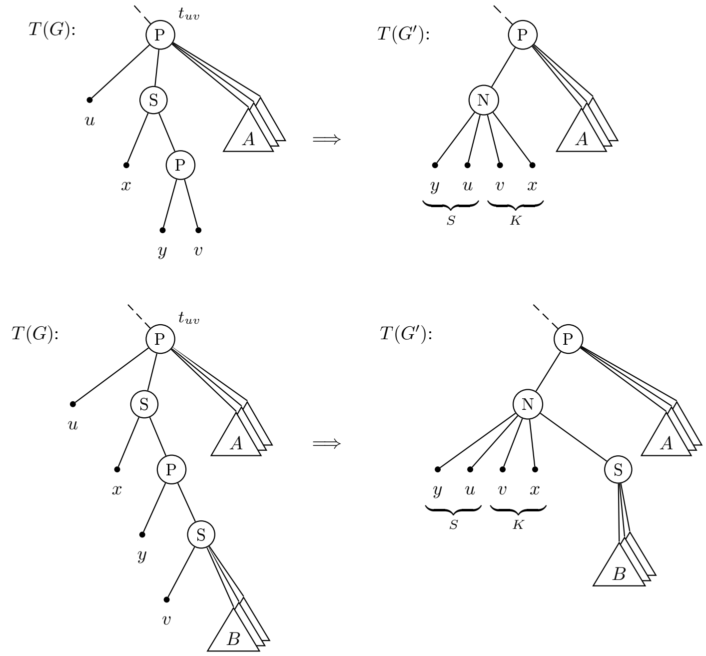

Figure 4: Illustrating case (ii) of Lemma 3.2 and the corresponding updates of the md-tree.

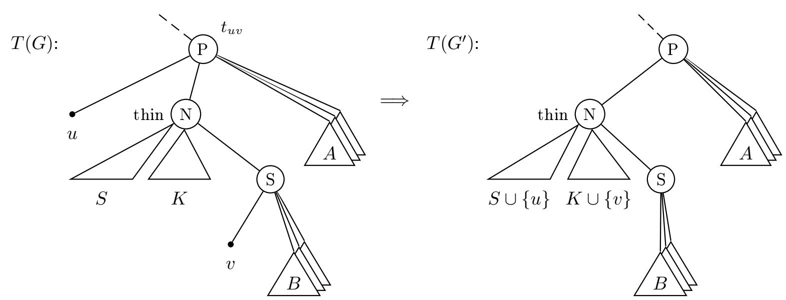

Figure 5: Illustrating case (iii) of Lemma 3.2 and the corresponding updates of the md-tree.

(iii) vertex v misses $\ell > 1$ vertices in $M(t_v)$ such that $G(t_v)$ is a thin spider (S, K, R) with $|S| = |K| = \ell$, $R = \{r\}$ and the vertex v belongs to the set $M(r)$ and sees all the vertices of $M(r)$.

Lemma 3.3. *Let $|M(t_u)| = 1$ (i.e., $M(t_u) = \{u\}$) and suppose that t_{uv} is an N-node such that (S, K, R) is the spider partition of $G(t_{uv})$. Then G' is a P_4-sparse graph if and only if either $S = \{u, v\}$ and $R = \emptyset$ or $u \in S$, $v \in K$, and $G(t_{uv})$ is a thick spider.*

Proof. The definition of the spider implies that the cases to consider are for u, v to belong both to S, or to S and K, or if $R = \{r\}$ to S and $M(r)$. Suppose that $u, v \in S$ and let $u', v' \in K$ such that $uu'v'v$ is a P_4 of G; then, G' contains the C_4 $uu'v'v$. If $R = \{r\}$ then the vertices u, v, u', v' and any vertex in $M(r)$ induce a $\overline{P}_5$ in G'; thus, $R = \emptyset$. If $|S| = |K| > 2$, then if the spider is thin, the vertices u, v, u', v', y, where $y \in K - \{u', v'\}$ induce a $\overline{P}_5$, whereas if the spider is thick, the vertices u, v, u', v', z, where $z \in S - \{u, v\}$ induce a $\overline{P}_5$ in G'. Now, let us consider the case that $u \in S$, $v \in K$ and suppose that $G(t_{uv})$ is a thin spider such that $|S| > 2$ (note that the spiders with $|S| = 2$ are also considered thick); R may or may not be $\emptyset$. Then, $v \neq f(u)$. Let $z \in K$ be such that $z \neq v$ and $z \neq f(u)$; then, the vertices $u, v, f(u), z, f^{-1}(z)$ induce a graph $\overline{F}_1$ in G'. Finally, suppose that $u \in S$ and $v \in M(r)$, and let $x \in S - \{u\}$. If $z, z' \in K$ are the vertices such that $uzz'x$ is a P_4 in $G(t_{uv})$ (and in G), then the vertices u, v, z, z, z' induce an $\overline{F_1}$ in G', which thus is not P_4-sparse. ∎

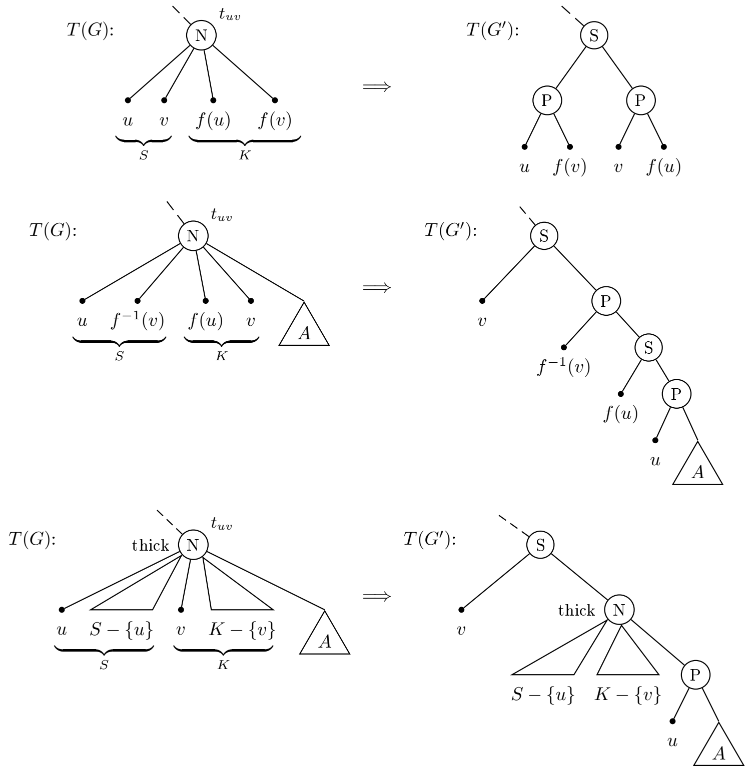

Figure 6: Illustrating the cases of Lemma 3.3 and the corresponding updates of the md-tree.

3.2 Removing an Edge

Since P_4-sparse graphs have a complement-invariant property (see Lemma 2.1), we take advantage of the following theorem [19]:

Theorem 3.1. (Shamir and Sharan [19]): *Let* Π *be a complement-invariant graph property. Let Alg be a dynamic algorithm for* Π*-recognition, which supports either edge additions only or edge deletions only, and is based on modular decomposition. Then Alg can be extended to support both operations with the same time complexity.*

3.3 Adding a Vertex

Let G be a P_4-sparse graph and a vertex $x \notin V(G)$ which is adjacent to d vertices in $V(G)$, where $d \in \{0, 1, \ldots, |V(G)|\}$. In this section, we show how to recognize if the graph G' with vertex set $V(G) \cup \{x\}$ is a P_4-sparse graph, and if so, we show how to obtain the md-tree $T(G')$ of G' from the md-tree $T(G)$ in $O(d)$ time. Let us classify the internal nodes of the md-tree $T(G)$ with respect to the vertex x into the following three categories: an internal node t is *x-fully-adjacent*, *x-partly-adjacent*, *x-non-adjacent* iff x is adjacent to all, some but not all, and none, respectively, of the vertices in the module $M(t)$. The above classification is extended to leaf-nodes: a leaf-node a is *x-fully-adjacent* or *x-non-adjacent* iff x is adjacent or non-adjacent respectively to a. For the number of x-fully-adjacent nodes of $T(G)$, we can show the following observation:

Observation 3.1. *The number of x-fully-adjacent nodes of $T(G)$ is less than $d-1$, where d is the number of vertices of $T(G)$ which are adjacent to x.*

Proof. The x-fully-adjacent nodes form a forest of subtrees of $T(G)$ whose total number of leaves is d. The observation follows from the fact that every internal node in $T(G)$ and in these subtrees has at least two children. ∎

In turn, for the x-partly-adjacent nodes, the fact that the module of an S-node induces a connected graph, the module of a P-node induces a graph whose complement is connected, and the module of an N-node induces a graph which is connected and whose complement is also connected implies:

- **P1**: if an internal node t of the md-tree $T(G)$ is x-partly-adjacent, then all its ancestors in $T(G)$ are x-partly-adjacent;
- **P2**: for every x-partly-adjacent P-node t_P of $T(G)$, the subgraph of G induced by the module $M(t_P)$ contains two non-adjacent vertices a, b such that a is adjacent and b is not adjacent to x;
- **P3**: for every x-partly-adjacent S-node t_S of $T(G)$, the subgraph of G induced by the module $M(t_S)$ contains an edge ab such that a is adjacent and b is not adjacent to x;
- **P4**: for every x-partly-adjacent N-node t_N of $T(G)$, the subgraph of G induced by the module $M(t_N)$ contains both an edge ab such that a is adjacent and b is not adjacent to x and a pair of non-adjacent vertices a', b' such that a' is adjacent and b' is not adjacent to x.

Additionally, the following very important property holds:

Theorem 3.2. *For any two x-partly-adjacent nodes of $T(G)$, the graph G' is P_4-sparse only if one of them is an ancestor of the other.*

Let $\rho_x = t_0 t_1 \cdots t_k$ denote the path in $T(G)$ containing all the x-partly-adjacent nodes (Theorem 3.2) where t_0 is the root of $T(G)$ and t_k is the x-partly-adjacent node farthest away from the root. Then, Theorem 3.2 implies that for each node t_i, $0 \le i < k$, each of t_i's children, other than t_{i+1}, is either x-fully-adjacent or x-non-adjacent; for the node t_k, each of t_k's children is either x-fully-adjacent or x-non-adjacent and there is at least one child of each kind. Additionally, for the x-partly-adjacent N-nodes, the following holds:

Lemma 3.4. *Let t be an x-partly-adjacent N-node of $T(G)$ whose corresponding spider partition of $M(t)$ is (S, K, R), and suppose that the vertex x is adjacent to a vertex in $S \cup K$. Then, the graph G' is P_4-sparse only if x is adjacent to $S \cup K$, or is adjacent to K and is not adjacent to S.*

Let us consider the partition of the vertex set $M(t_0) - M(t_k) \subset V(G)$ into the following four sets:

$$V_P = \bigcup_{t_i \text{ is a P-node}} \left(M(t_i) - M(t_{i+1})\right), \qquad V_S = \bigcup_{t_i \text{ is an S-node}} \left(M(t_i) - M(t_{i+1})\right),$$

$$V_{N_S} = \bigcup_{t_i \text{ is an N-node}} S(t_i), \qquad V_{N_K} = \bigcup_{t_i \text{ is an N-node}} K(t_i),$$

where for an N-node t_i, $S(t_i)$ and $K(t_i)$ are the independent set and the clique of the spider induced by the module $M(t_i)$. Then, every vertex in V_P (in V_S resp.) is non-adjacent (adjacent resp.) to the vertices in $M(t_k)$ since their least common ancestor t_i in $T(G)$ is a P-node (S-node resp.), while the structural properties of a spider imply that every vertex in $K(t_j)$ ($S(t_j)$ resp.) for an N-node t_j is adjacent (non-adjacent resp.) to the vertices in $M(t_k)$.

Our vertex-addition procedure relies on the following lemmata:

Lemma 3.5. *Suppose that the x-partly-adjacent nodes of the md-tree $T(G)$ lie on a path $t_0 t_1 \cdots t_k$, where t_k is the x-partly-adjacent node farthest away from the root t_0 of $T(G)$. If t_k is a P-node then G' is P_4-sparse if and only if one of the following four (mutually exclusive) cases holds:*

(i) *Vertex x sees V_S and V_{N_K}, and misses V_P and V_{N_S}.*

(ii) *Vertex x sees V_S, V_{N_K}, and exactly one vertex, say, y, in V_P, and misses V_{N_S} where*
- *(ii.1)* *vertex y is a child of node t_{k-2} (which is a P-node),*
- *(ii.2)* *node t_{k-1} is an S-node with two children, the node t_k and one vertex, say, u (which is adjacent to x), and*
- *(ii.3)* *vertex x sees all the vertices in $M(t_k)$ except for a single vertex, say, b, which is a child of t_k.*

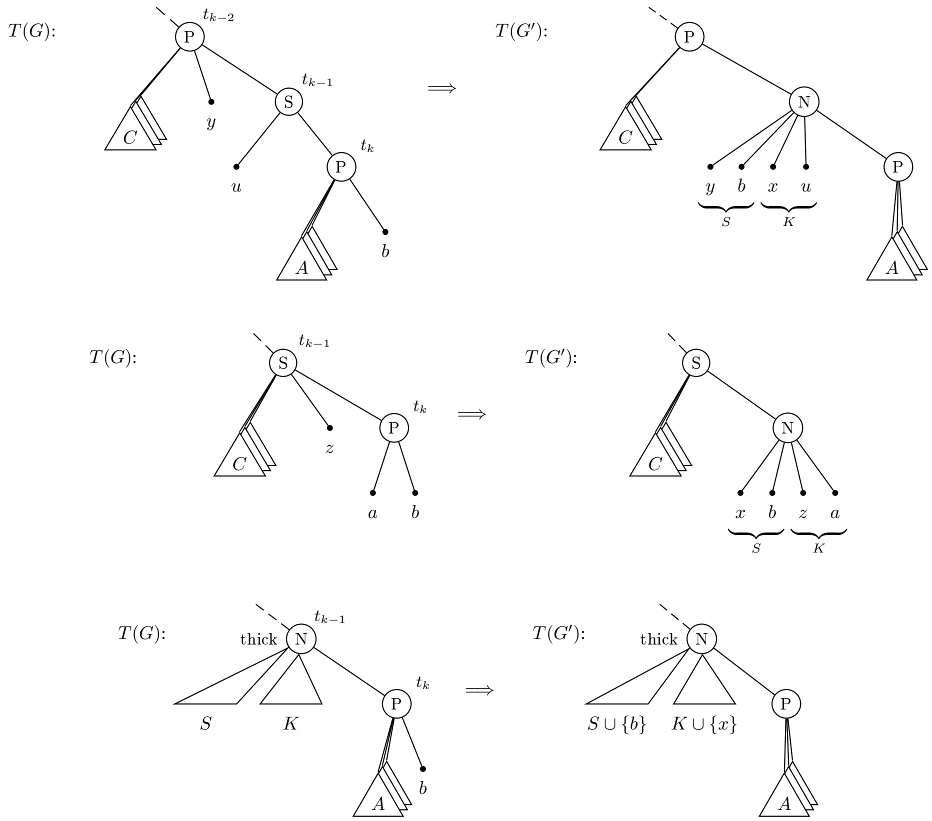

Figure 7: Illustrating cases (ii), (iii), (iv) of Lemma 3.5 and the corresponding updates of the md-tree.

(iii) *Vertex x sees V_{N_K}, all but one vertex, say, z, in V_S, and misses V_P and V_{N_S} where*
- (iii.1) *vertex z is a child of node t_{k-1} (which is an S-node), and*
- (iii.2) *node t_k has two children a, b, which are leaf-nodes such that a is adjacent and b is non-adjacent to x.*

(iv) *The node t_{k-1} is an N-node corresponding to a thick spider with independent set $S(t_{k-1})$, vertex x sees V_S, V_{N_K}, $S(t_{k-1})$, and all but one vertex, say, b, in $M(t_k)$, and misses V_P and $V_{N_S} - S(t_{k-1})$.*

The case where t_k is an S-node is precisely the complement version of Lemma 3.5: we need to exchange P- and S-nodes, thin and thick spiders, their cliques and independent sets, and what x sees/misses in the conditions of Lemma 3.5.

Lemma 3.6. *Suppose that the x-partly-adjacent nodes of the md-tree $T(G)$ lie on a path $t_0 t_1 \cdots t_k$, where t_k is the x-partly-adjacent node farthest away from the root t_0 of $T(G)$. If t_k is an S-node then G' is P_4-sparse if and only if one of the following four (mutually exclusive) cases holds:*

(i) *Vertex x sees V_S and V_{N_K}, and misses V_P and V_{N_S}.*

(ii) *Vertex x sees V_{N_K}, all but one vertex, say, y, in V_S, and misses V_P and V_{N_S} where*
- (ii.1) *vertex y is a child of node t_{k-2} (which is an S-node),*
- (ii.2) *node t_{k-1} is a P-node with two children, the node t_k and one vertex, say, u (which is non-adjacent to x), and*
- (ii.3) *vertex x sees only a single vertex of $M(t_k)$, which is a child of t_k.*

(iii) *Vertex x sees V_S, V_{N_K}, and exactly one vertex, say, z, in V_P, and misses V_{N_S} where*
- (iii.1) *vertex z is a child of node t_{k-1} (which is a P-node), and*
- (iii.2) *node t_k has two children a, b, which are leaf-nodes such that a is adjacent and b is non-adjacent to x.*

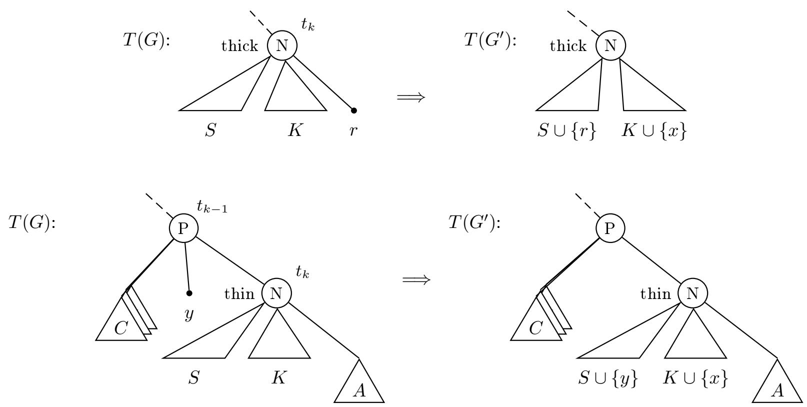

Figure 8: Illustrating cases (i) and (ii.2) of Lemma 3.7 and the corresponding updates of the md-tree.

(iv) The node t_{k-1} is an N-node corresponding to a thin spider with clique $K(t_{k-1})$, vertex x misses V_P, V_{N_S}, $K(t_{k-1})$, and all but one vertex, say, b, in $M(t_k)$, and sees V_S and $V_{N_K} - K(t_{k-1})$.

Lemma 3.7. *Suppose that the x-partly-adjacent nodes of the md-tree $T(G)$ lie on a path $t_0 t_1 \cdots t_k$, where t_k is the x-partly-adjacent node farthest away from the root t_0 of $T(G)$. If t_k is an N-node and the partition of the spider $G(t_k)$ is (S, K, R), then G' is P_4-sparse if and only if the conditions in one of the following three (mutually exclusive) cases hold:*

(i) Vertex x sees $S \cup K$ (and misses $M(r)$ where $R = \{r\}$): vertex x sees V_S and V_{N_K}, and misses V_P and V_{N_S}, the spider corresponding to t_k is a thick spider, and the node r is a leaf.

(ii) Vertex x sees K (and misses S): one of the following three cases holds:

- *(ii.1) vertex x sees V_S and V_{K_N}, and misses V_P and V_{N_S};*
- *(ii.2) vertex x sees V_S, V_{N_K}, and exactly one vertex, say, y, in V_P, and misses V_{N_S} where y is a child of t_{k-1}, the spider corresponding to t_k is thin, and all the elements of $M(r)$ (if $R = \{r\}$) are adjacent to x;*
- *(ii.3) vertex x sees V_{N_K}, all but one vertex, say, y, in V_S, and misses V_P and V_{N_S} where y is a child of t_{k-1}, the spider corresponding to t_k is thick, and all the elements of $M(r)$ (if $R = \{r\}$) are non-adjacent to x.*

(iii) Vertex x misses $S \cup K$ (and sees $M(r)$ where $R = \{r\}$): vertex x sees V_S and V_{N_K}, and misses V_P and V_{N_S}, the spider corresponding to t_k is a thin spider, and the node r is a leaf.

The procedure that handles the addition of vertex x finds the node t_k and takes advantage of Lemmata 3.5–3.7 to check and modify the tree $T(G)$. It starts form the leaves of the md-tree $T(G)$ which correspond to the neighbors of x and moving in a bottom-up fashion constructs the set A of internal nodes of $T(G)$ having at least one x-fully-adjacent child. Then, it splits A obtaining the set *Full* of x-fully-adjacent nodes of $T(G)$ and a subset *Partial* of x-partly-adjacent nodes, from which it determines t_k and all the x-partly-adjacent nodes by noting that in each case of Lemmata 3.5–3.7, x sees V_{N_K} and all but at most one of the elements of V_S, i.e., all the x-partly-adjacent N-nodes and all (but at most one) x-partly-adjacent S-nodes belong to Partial. Then, since the father of a P-node (S-node resp.) cannot be a P-node (S-node resp.), the following holds:

Observation 3.2. *For each node $t \in Partial$ at distance at least 3 from the root of the tree $T(G)$, if none of t's father, grandfather, and great-grandfather belongs to $Partial$, then the graph G' is not P_4-sparse.*

In detail, the procedure to add a vertex x works as follows:

Procedure VERTEX_ADD(vertex x)

1. $A \leftarrow \emptyset$;
   construct a queue Q whose elements are pointers to each of the leaf-nodes of $T(G)$ which correspond to the neighbors of x;
   `while` the queue Q is not empty `do`
   remove from Q an element (i.e., a pointer to a node, say, t, of $T(G)$);
   increment the *counter*-field of the father $p(t)$ of t by 1 and let its new value by val;
   `if` $val = 1$
   `then` insert in A a pointer to $p(t)$;
   `if` $val =$ number of $p(t)$'s children
   `then` insert in Q a pointer to $p(t)$; $\{t$ *is* x*-fully-adjacent*$\}$

2. $Full \leftarrow$ set of pointers to each of the leaf-nodes of $T(G)$ which correspond to the neighbors of x;
   $Partial \leftarrow \emptyset$;
   `for` each element a of the set A `do`
   let t be the node of $T(G)$ pointed by a;
   `if` the value of t's *counter*-field is equal to the number of t's children
   `then` insert a in $Full$; $\{t$ *is* x*-fully-adjacent*$\}$
   set t's *counter*-field equal to 0;
   `else` insert a in $Partial$; $\{t$ *is* x*-partly-adjacent*$\}$

3. `for` each element a of the set $Partial$ `do`
   let t be the node of $T(G)$ pointed by a;
   `if` none of t's father, grandfather, and great-grandfather (if they exist) $\in Partial$
   `then` output *false* (i.e., G' is not P_4-sparse); `return`;
   mark t's father, grandfather, and great-grandfather (if they exist) as "covered;"
   traverse the set $Partial$ and check the following:
   `if` there exist two or more elements of $Partial$ pointing to nodes which are not "covered"
   `then` output *false* (i.e., G' is not P_4-sparse); `return`;
   let the unique node in $Partial$ which is not "covered" be t';

4. Depending on whether t' is a P-node, S-node, or N-node, we check whether one of the cases of Lemma 3.5, 3.6, and 3.7, respectively, holds, and we appropriately modify $T(G)$;
   `if` none of the cases of the corresponding lemma applies
   `then` output *false* (i.e., G' is not P_4-sparse); `return`;

5. `for` each element a of the set $Partial$ `do`
   unmark the node of $T(G)$ pointed to by a and set its *counter*-field to 0;

The correctness of the procedure follows from Lemmata 3.5–3.7, Observation 3.2, and from the following facts:

- the set of nodes of the tree $T(G)$ pointed to by the elements of the set $Full$ is precisely the set of x-fully-adjacent nodes;
- the set of nodes of the tree $T(G)$ pointed to by the elements of the set $Partial$ are the x-partly-adjacent nodes of $T(G)$ with at least one x-fully-adjacent child (note that $t_k \in Partial$);
- the node t' found in Step 3 is precisely the x-partly-adjacent node t_k farthest away from the root.

3.4 Deleting a Vertex

Let $v \in V(G)$ be a vertex with d incident edges in G which has to be deleted. Clearly, the graph G' which results after the deletion of v is a P_4-sparse graph as it is an induced subgraph of G (see Lemma 2.1). Hence we focus on properly updating the md-tree $T(G)$ so that we obtain the md-tree $T(G')$.

Let us first consider the case that $v \in S \cup K$ for some N-node t such that the spider partition of $G(t)$ is (S, K, R). We distinguish the following cases:

(i) $v \in S$: First suppose that $S = \{v, v'\}$, $K = \{k, k'\}$, and let v be adjacent to k: then, the spider is replaced by an S-node with children the vertex k' and a P-node; if $R = \emptyset$, then this P-node has as children the vertices v' and k, else if $R = \{r\}$, it has as children the vertex v' and an S-node with children the vertex k and the node r. Now, suppose that $|S| = |K| \geq 3$ and let $f(v) = k \in K$. If the spider is thin then: if $R = \emptyset$, then after the removal of v, k is removed from K and is linked at the pointer for R; if $R = \{r\}$, then k is removed from K and if r is an S-node then k is linked as a child, otherwise the place of r is taken by an S-node with k and r as children. If the spider is thick, then after the removal of v, vertex k sees all the remaining vertices in $M(t)$; thus, the N-node t is replaced by an S-node with children the vertex k and the node t after we have removed the vertices v, k.

(ii) $v \in K$: Since the complement of a thin spider is a thick spider (and vice versa) with the clique and independent sets swapped (and if $R = \{r\}$, the P- and S-nodes in the subtree rooted at r swapped as well), this is the complement version of the previous case and takes the same time to handle.

(iii) $R = \{v\}$: In this case, v is deleted, and we obtain a spider with $R = \emptyset$.

Next, we consider the case where the father-node $p(v)$ of v in $T(G)$ is a P- or S-node; if $p(v)$ has more than 2 children, it suffices to simply delete v. However, caution is needed if $p(v)$ has only two children, in which case the sibling u of v needs to be linked to the grandfather $p(p(v))$ of v; furthermore, if u and $p(p(v))$ are both P- or S-nodes, then the children of u are placed as children of $p(p(v))$. Finally, in either of the remaining two cases when the father-node $p(v)$ has 2 children, i.e., if the sibling u of v is an N-node or if the grandfather $p(p(v))$ is an N-node, then u is linked as a child of $p(p(v))$ (there is no problem having an N-node with $R = \{r\}$ where r is an N-node as well).

3.5 Time complexity

Lemmata 3.1–3.3 and Figures 3–6 show that the addition of an edge requires local changes in at most 4 levels higher than v or u. For the vertex addition, we observe that the size of the set A as well as the length of the path $t_0 \cdots t_k$ are $O(d)$, while Lemmata 3.5–3.7 and Figures 7 and 8 also show that local changes are needed; note that the value of the *counter*-field of a node at the end of Step 2 is equal to the number of its children that are x-fully-adjacent. Finally, the vertex deletion requires $O(1)$ local changes except when we union the children of two P- or S-nodes.

Therefore, in addition to the auxiliary fields *counter* and *mark* (initialized to 0), we store in each node of the md-tree $T(G)$ its type (P, S, or N) and the number of its children, as well as ways to access its parent and its children. Additionally, each N-node stores the type of spider (thin or thick); the independent set S and the clique K of the spider are stored in pairs of corresponding (through the function f) vertices, while there exists a separate pointer to R which is null if $R = \emptyset$. Moreover:

- If each P- or S-node stores pointers to its parent and to a list of its children, then edge additions (and deletions, by Theorem 3.1) are handled in $O(1)$ time, whereas vertex additions/deletions are handled in $O(d)$ time.
- Alternatively, the children of a P- or S-node may be in a binomial heap [2] in which the pointer to the parent of these children in $T(G)$ is stored at the minimum element (which appears in the root list) of the heap. Then, finding the parent of such a child requires moving to the root list of the heap and visiting its elements; unioning the children of two nodes requires merging two heaps. Since all these operations, as well as additions and deletions, take time logarithmic in the size of the heap [2], then edge additions (and deletions) and vertex deletions are handled in $O(\log n)$ time, whereas vertex additions are handled in $O(d \log n)$ time.

Our results are summarized in the following theorem.

Theorem 3.3. *We have described a fully dynamic algorithm for recognizing P_4-sparse graphs and maintaining their modular decomposition tree, which handles additions and deletions of vertices and edges. Edge modifications can be handled in $O(1)$ time while vertex modifications can be handled in $O(d)$ time; alternatively, edge modifications and vertex deletions can be handled in $O(\log n)$ time and vertex additions in $O(d \log n)$ time.*

References

[1] A. Brandstädt, V.B. Le, and J. Spinrad, *Graph Classes – a Survey*, SIAM Monographs in Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms* (2nd edition), MIT Press, Inc., 2001.

[3] D.G. Corneil, Y. Perl, and L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Comput.* **14** (1985) 926–984.

[4] A. Cournier and M. Habib, A new linear algorithm for modular decomposition, *Proc. 19th Int'l Colloquium on Trees in Algebra and Programming (CAAP'94)*, LNCS **787** (1994) 68–84.

[5] C. Crespelle and C. Paul, Fully-dynamic recognition algorithm and certificate for directed cographs, *Proc. 30th Int'l Workshop on Graph-Theoretic Concepts in Computer Science (WG'04)*, LNCS **3353** (2004) 93–104.

[6] C. Crespelle and C. Paul, Fully-dynamic algorithm for modular decomposition and recognition of permutation graphs, *Proc. 31st Int'l Workshop on Graph-Theoretic Concepts in Computer Science (WG'05)*, LNCS **3787** (2005) 38–48.

[7] E. Dalhaus, J. Gustedt, and R.M. McConnell, Efficient and practical algorithms for sequential modular decomposition, *J. Algorithms* **41** (2001) 360–387.

[8] X. Deng, P. Hell, and J. Huang, Linear time representation algorithms for proper circular arc graphs and proper interval graphs, *SIAM J. Comput.* **25** (1996) 390–403.

[9] V. Giakoumakis and J.-M. Vanherpe, On extended P_4-reducible and P_4-sparse graphs, *Theoret. Comput. Sci.* **180** (1997) 269–286.

[10] P. Hell, R. Shamir, and R. Sharan, A fully dynamic algorithm for recognizing and representing proper interval graphs, *SIAM J. Comput.* **31** (2002) 289–305.

[11] C. Hoàng, Perfect graphs, Ph.D. Thesis, McGill University, Montreal, Canada, 1985.

[12] W.-L. Hsu, On-line recognition of interval graphs in $O(m + n \log n)$ time, *Combinatorics and Computer Science* 1995, LNCS **1120** (1996) 27–38.

[13] L. Ibarra, Fully dynamic algorithms for chordal graphs, *Proc. 10th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'99)*, (1999) 923–924.

[14] L. Ibarra, A fully dynamic algorithm for recognizing interval graphs using the clique-separator graph, Technical Report, DCS-263-IR, University of Victoria, 2001.

[15] B. Jamison and S. Olariu, Recognizing P_4-sparse graphs in linear time, *SIAM J. Comput.* **21** (1992) 381–406.

[16] B. Jamison and S. Olariu, A tree representation for P_4-sparse graphs, *Discrete Appl. Math.* **35** (1992) 115–129.

[17] R.M. McConnell and J. Spinrad, Modular decomposition and transitive orientation, *Discrete Math.* **201** (1999) 189–241.

[18] J.H. Muller and J. Spinrad, Incremental modular decomposition, *J. ACM* **36** (1989) 1–19.

[19] R. Shamir and R. Sharan, A fully dynamic algorithm for modular decomposition and recognition of cographs, *Discrete Appl. Math.* **136** (2004) 329–340.

[20] J. Spinrad, P_4-trees and substitution decomposition, *Discrete Appl. Math.* **39** (1992) 263–291.

Appendix: Proofs of Lemmas

(To assist the reviewers)

Lemma 3.2. *Let $|M(t_u)| = 1$ (i.e., $M(t_u) = \{u\}$) and suppose that t_{uv} is a P-node. Then G' is a P_4-sparse graph if and only if exactly one of the following statements is satisfied:*

(i) *vertex v sees all the vertices in $M(t_v)$.*

(ii) *vertex v misses exactly one vertex $y \in M(t_v) - \{v\}$ where y sees only one vertex $x \in M(t_v)$, and only the vertex x sees every vertex in $M(t_v)$.*

(iii) *vertex v misses $\ell > 1$ vertices in $M(t_v) - \{v\}$ where $G(t_v)$ is a thin spider (S, K, R) with $|S| = |K| = \ell$, $R = \{r\}$ and the vertex v belongs to the set $M(r)$ and sees all the vertices of $M(r)$.*

Proof. The "if"-part of the lemma follows from Figures A1, and 4 and 5, since the resulting graph G' is indeed P_4-sparse. For the "only if"-part, we have that G' is P_4-sparse; the proof relies on the following facts.

Fact A.1. *Let $M(t_u) = \{u\}$, t_{uv} be a P-node, and suppose that the path from t_v to $p(v)$ in the md-tree $T(G)$ does not contain any N-node. If there is no vertex $x \in M(t_v)$ such that x sees every vertex in $M(t_v)$ then G' is not a P_4-sparse graph.*

Proof of Fact A.1. Since t_{uv} is a P-node, the graph $G[M(t_v)]$ is connected. Moreover, since there is no N-node in the path from t_v to $p(v)$ in $T(G)$, the vertex v neither participates in any P_4 in $G[M(t_v)]$ nor is adjacent to some but not all the vertices of such a P_4 (note that v may very well be adjacent to all the vertices of such a P_4). This implies that the vertex set $M(t_v)$ can be partitioned into three sets A, B, and C, where $A = \{v\}$, B contains the neighbors of v, and C contains the non-neighbors of v. Since no vertex in $M(t_v)$ sees the entire $M(t_v)$, the set C contains at least one vertex; let y be such a vertex. Since $G[M(t_v)]$ is connected and v does not participate in a P_4, there exists a vertex, say, z, in B such that z, y are adjacent. Then, y misses all other vertices in B; otherwise, G' would contain either an $\overline{F}_1$ or an $\overline{F}_2$ as an induced subgraph. Since v does not participate in any P_4, z sees every vertex in B. But because no vertex in $M(t_v)$ sees all vertices in $M(t_v)$, z misses a vertex, say, $y' \in C$. The vertex y' misses y, for otherwise, v would participate in the P_4 $vzyy'$. Again, as with y, y' sees a vertex $z' \in B$. But then, v sees some but not all the vertices of the P_4 $yzz'y'$, which is impossible. Thus, in the only possible cases, the graph G' is not P_4-sparse. $\square$

Fact A.2. *Let $M(t_u) = \{u\}$, t_{uv} be a P-node, and suppose that the path from t_v to $p(v)$ in the md-tree $T(G)$ does not contain any N-node; suppose further that vertex v misses at least one vertex in $M(t_v)$. If G' is a P_4-sparse graph, then vertex v misses exactly one vertex, say, y, in $M(t_v)$, and y only sees the unique vertex x in $M(t_v)$ that sees all other vertices in $M(t_v)$.*

Proof of Fact A.2. We first show that there is no vertex $x' \in M(t_v) - \{x\}$ that sees all other vertices in $M(t_v)$. If there were such a vertex x', then, the five vertices u, v, x, y, and x' induce in G' the graph $\overline{F}_1$, a contradiction. Thus, only vertex x in $M(t_v)$ sees all other vertices in $M(t_v)$.

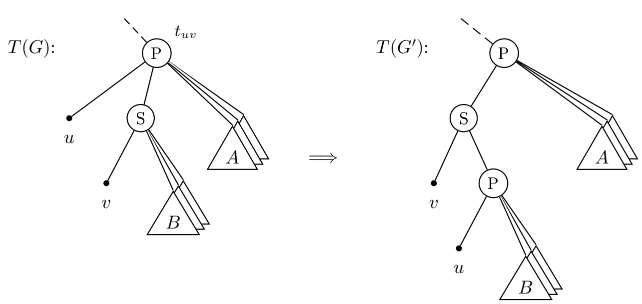

Figure A1: Illustrating case (i) of Lemma 3.2 and the corresponding updates of the md-tree.

Next, suppose that there exists another vertex $y' \in M(t_v) - \{y\}$ such that v misses y'. Then, the five vertices u, v, x, y, and y' induce in G' the graphs F_1 or F_2, a contradiction. Therefore, v misses exactly one vertex in $M(t_v)$, the vertex y.

Moreover, if y saw a vertex, say, $z \in M(t_v)$, other than x, then the five vertices u, v, x, y, and z would induce in G' the graphs F_2 or $\overline{F}_1$, a contradiction again. $\square$

Suppose now that the path from t_v to $p(v)$ contains at least one N-node t of $T(G)$. Recall that the representative graph $G(t)$ is a prime spider and let (S, K, R) be its spider partition. Note that $v \in M(t)$ and thus v belongs to the set S, or to the set K, or if $R = \{r\}$ to the set $M(r)$.

Fact A.3. *Let $M(t_u) = \{u\}$, t_{uv} be a P-node, and suppose that the path from t_v to $p(v)$ in the md-tree $T(G)$ contains at least one N-node t. Let (S, K, R) be the spider partition of $G(t)$. If either $v \in S \cup K$ or $v \in M(r)$ and v misses a vertex in $M(r)$, then G' is not a P_4-sparse graph.*

Proof of Fact A.3. Let t be the first N-node in the path from $p(v)$ to t_v. Suppose that the vertex v belongs to the set S of the prime spider $G(t)$. Since $|S| = |K| \geq 2$, there exists a vertex $v' \in S - \{v\}$. If the P_4 of $G(t)$ to which v, v' belong is $vyy'v'$, then the addition of the edge uv implies that G' contains the P_5 $uvyy'v'$. Now consider that $v \in K$; let $v' \in K - \{v\}$ and let $zvv'z'$ be the P_4 of $G(t)$ to which v, v' belong. Then, the five vertices z, v, v', z', u induce in G' the graph F_1. Thus, in the case where $v \in S \cup K$, the graph G' is not P_4-sparse.

Suppose now that $v \in M(r)$ and let $z \in M(r)$ be such that v misses z. If $x \in S$ and $y \in K$ such that x, y are adjacent in G, then by adding the edge uv, the five vertices u, v, x, y, and z induce the graph F_1 in G'. $\square$

Fact A.4. *Let $M(t_u) = \{u\}$, t_{uv} be a P-node, and suppose that the path from t_v to $p(v)$ in the md-tree $T(G)$ contains at least one N-node t. Let (S, K, R) be the spider partition of $G(t)$ with $R = \{r\}$. If either the path from t_v to $p(v)$ contains more than one N-node or $t \neq t_v$, then G' is not a P_4-sparse graph.*

Proof of Fact A.4. Suppose that there exists another N-node, say, t', in the path and let (S', K', R') be the spider partition of $G(t')$ with $R' = \{r'\}$. Then, v misses at least one vertex of $M(r')$ if $M(r) \subset M(r')$ or one vertex of $M(r)$ if $M(r') \subset M(r)$, and thus G' is not a P_4-sparse graph (see Fact A.3). Suppose now that $t \neq t_v$. Since t_{uv} is a P-node and $t \neq t_v$, it follows that t_v is an S-node (if t_v were an N-node, then the path would contain two N-nodes). Then, at least one vertex $z \in M(t_v)$ sees all the vertices of $M(t)$. Let $x, y \in S$ be two vertices of the spader $G(t)$. By adding the edge uv in G, the five vertices u, v, x, y, and z of G' induce the graph F_1 in G'. $\square$

Fact A.5. *Let $M(t_u) = \{u\}$, t_{uv} be a P-node, and suppose that t_v is the N-node in the path from t_v to $p(v)$ in the md-tree $T(G)$. Let (S, K, R) be the spider partition of $G(t_v)$ with $R = \{r\}$. If $G(t_v)$ is not a thin spider, then G' is not a P_4-sparse graph.*

Proof of Fact A.5. Since $G(t_v)$ is not a thin spider, then $|S| = |K| \geq 3$. Let $x_1, x_2 \in S$ and let $y \in K$ such that y sees both x_1 and x_2. By Fact A.3, $v \notin S \cup K$, and thus $v \in M(r)$. Then, the addition of the edge uv implies that the vertices x_1, x_2, y, v, u induce the graph F_1 in G'. $\square$

Then, from Facts A.1–A.5, we have: If the path from t_v to $p(v)$ in the md-tree $T(G)$ contains no N-nodes, then by Fact A.1, there exists a vertex in $M(t_v)$ which sees all other vertices in $M(t_v)$. If v is such a vertex, then we get Case (i). If v is not so, i.e., v misses at least one vertex in $M(t_v) - \{v\}$, then Fact A.2 implies that Case (ii) holds. If now the path from t_v to $p(v)$ in the md-tree $T(G)$ contains N-nodes, then by Fact A.3, we have that $v \in M(r)$ and v sees all other vertices in $M(r)$. Moreover, Fact A.4 implies that there exists exactly one N-node in the path from t_v to $p(v)$ and this is in fact t_v. Then, Fact A.5 implies that $G(t_v)$ is a thin spider. Facts A.3–A.5 imply that Case (iii) holds. ∎

Theorem 3.2. *For any two x-partly-adjacent nodes of $T(G)$, the graph G' is P_4-sparse only if one of them is an ancestor of the other.*

Proof. Suppose that $T(G)$ contains two x-partly-adjacent nodes t, t' such that none is an ancestor of the other. Then, t, t' are internal nodes of $T(G)$ and let t_i be the least common ancestor of t, t', and t_j and t_k be the children of t_i which are ancestors of t and t' respectively. Clearly, by Property P1, t_j and t_k are x-partly-adjacent nodes. Additionally, the node t_i is either a P-node or an S-node (recall that at most one child of an N-node is an internal node). Thus, we distinguish the following two cases:

- *the node t_i is a P-node*: Then, t_j, t_k are either S- or N-nodes; in either case, there are vertices $a_j, b_j \in M(t_j)$ and $a_k, b_k \in M(t_k)$ such that in G, a_j, b_j are adjacent, a_k, b_k are also adjacent, and x is adjacent to a_j, a_k but not to b_j, b_k (see Properties P3, P4). But then, G' would contain the P_5 $b_j a_j x a_k b_k$, and thus would not be P_4-sparse.
- *the node t_i is an S-node*: This case is the complement version of the previous case. The nodes t_j, t_k are either P- or N-nodes; in either case, there are vertices $a_j, b_j \in M(t_j)$ and $a_k, b_k \in M(t_k)$ such that in G, a_j, b_j are non-adjacent, a_k, b_k are also non-adjacent, and x sees a_j, a_k and misses b_j, b_k (see Properties P2, P4). But then, G' would not be P_4-sparse as it would contain the $\overline{P}_5$ induced by a_j, b_j, x, a_k, b_k. ∎

Lemma 3.4. *Let t be an x-partly-adjacent N-node of $T(G)$ whose corresponding spider partition of $M(t)$ is (S, K, R), and suppose that the vertex x is adjacent to a vertex in $S \cup K$. Then, the graph G' is P_4-sparse only if x is adjacent to $S \cup K$, or is adjacent to K and is not adjacent to S.*

Proof. First, suppose that x does not see S but sees $k_i \in K$. Then, x sees every $k_j \in K$, otherwise the vertices x, k_i, k_j, s, s' induce an F_1, where sk_ik_js' is the (unique) P_4 of the spider which has k_ik_j as an edge.

Suppose now that x sees $s_i \in S$. Then, x sees every $s_j \in S$, otherwise the vertices x, k, k', s_i, s_j induce an $\overline{F}_1$ if x is adjacent to both k, k', or a P_5 if x is adjacent neither to k nor to k' (note that in light of our result for K, we do not need to consider the case where x is adjacent to exactly one of k, k'), where $s_ikk's_j$ is the (unique) P_4 of the spider with s_i, s_j in its vertex set. Thus, x sees S. Then, x sees K as well. If x missed $k_i \in K$ then it would miss K altogether; then, by our result for K, the vertices x, k_i, k_j, s, s' would induce a C_5, where sk_ik_js' is the (unique) P_4 of the spider which has k_ik_j as an edge. ∎

Lemma 3.5. *Suppose that the x-partly-adjacent nodes of the md-tree $T(G)$ lie on a path $t_0t_1\cdots t_k$, where t_k is the x-partly-adjacent node farthest away from the root t_0 of $T(G)$. If t_k is a P-node then G' is P_4-sparse if and only if one of the following four (mutually exclusive) cases holds:*

(i) Vertex x sees V_S and V_{N_K}, and misses V_P and V_{N_S}.

(ii) Vertex x sees V_S, V_{N_K}, and exactly one vertex, say, y, in V_P, and misses V_{N_S} where
- *(ii.1) vertex y is a child of node t_{k-2} (which is a P-node),*
- *(ii.2) node t_{k-1} is an S-node with two children, the node t_k and one vertex, say, u (which is adjacent to x), and*
- *(ii.3) vertex x sees all the vertices in $M(t_k)$ except for a single vertex, say, b, which is a child of t_k.*

(iii) Vertex x sees V_{N_K}, all but one vertex, say, z, in V_S, and misses V_P and V_{N_S} where
- *(iii.1) vertex z is a child of node t_{k-1} (which is an S-node), and*
- *(iii.2) node t_k has two children a, b, which are leaf-nodes such that a is adjacent and b is non-adjacent to x.*

(iv) The node t_{k-1} is an N-node corresponding to a thick spider with independent set $S(t_{k-1})$, vertex x sees V_S, V_{N_K}, $S(t_{k-1})$, and all but one vertex, say, b, in $M(t_k)$, and misses V_P and $V_{N_S} - S(t_{k-1})$.

Proof. It is not difficult to see that the graph G' is P_4-sparse if Case (i) of the lemma holds: in the md-tree of G', x and the x-fully-adjacent children of t_k in $T(G)$ are children of an S-node which is a child of t_k. For the remaining cases, Figure 7 gives the md-tree $T(G')$ (it is easy to check the adjacencies), which establishes that the graph G' is P_4-sparse in these cases as well. Thus, we need to show that if G' is P_4-sparse exactly one of Cases (i)–(iv) holds.

Since the node t_k is an x-partly-adjacent P-node then by Property P2 there exist two vertices $a, b \in M(t_k)$ which are non-adjacent in G and such that x is adjacent to a and non-adjacent to b (in Figure 7, A denotes all the subtrees rooted at children of t_k which are x-fully-adjacent).

First, we note that, for G' to be P_4-sparse:

A1: *x must be adjacent to all but at most one vertex in $V_S \cup V_{N_K}$*: if x were not adjacent to vertices $y, y' \in V_S \cup V_{N_K}$, then the vertices x, a, b, y, y' would induce in G either a $\overline{F}_1$ or a $\overline{F}_2$ (see Figure 1) depending on whether y, y' are adjacent or not, and thus G' would not be P_4-sparse. Additionally, Lemma 3.4 and the fact that the clique of a spider is of size at least 2 imply that x sees all of V_{N_K}; thus, if x does not see a vertex $y \in V_S \cup V_{N_K}$, then $y \in V_S$.

A2: *x must be adjacent to at most one vertex in V_P*: suppose that x were adjacent to vertices $z, z' \in V_P$; since the father-node of t_k is either an S- or an N-node, there exists a vertex y which is adjacent to a, b and is non-adjacent to z, z'; then, if y is adjacent to x, the vertices x, y, b, z, z' would induce an F_1, whereas if y is non-adjacent to x, the vertices x, y, a, b, z would induce a P_5.

A3: *x must miss the independent sets of all the N-nodes in the subpath $t_0 t_1 \cdots t_{k-2}$*: suppose that x were adjacent to a vertex z belonging to the independent set $S(t_i)$ of the spider associated with t_i $(0 \leq i \leq k-2)$; then, there exists $k \in K(t_i)$ such that k is adjacent to z, and since x sees V_{N_K}, k is adjacent to x as well; moreover, no matter whether t_{k-1} is an S- or an N-node, there exists $u \in M(t_{k-1}) - M(t_k)$ such that u is adjacent to both a, b; then, if x is adjacent to u, the vertices x, z, k, u, b would induce an $\overline{F}_1$, otherwise, the vertices x, z, k, u, a would induce a $\overline{P}_5$.

From Properties A1–A3, it follows that if t_k is the root of the tree $T(G)$, or if x sees V_S and misses V_P and t_{k-1} is not an N-node, then Case (i) applies. Suppose next that t_k is not the root and that x sees $y \in V_P$, or misses $z \in V_S$, or t_{k-1} is an N-node, and that G' is P_4-sparse; since t_k is a P-node, we distinguish the following cases:

(a) t_{k-1} *is an S-node*: then, there exists $u \in M(t_{k-1}) - M(t_k)$ such that u is adjacent to both a, b.

- Suppose that x sees $y \in V_P$. Then, x sees u, otherwise x, y, u, a, b would induce a P_5. Moreover, y is a child of t_{k-2}: if y were a child of t_i, where $i < k-2$, then t_{i+1} would be an S- or an N-node and thus there would exist a vertex v such that v would see u, a, b; then, the vertices x, y, u, v, b or x, y, u, v, a would induce an $\overline{F}_1$ depending on whether x is adjacent to v or not. Next, u is t_{k-1}'s only child other than t_k; if there existed $u, u' \in M(t_{k-1}) - M(t_k)$, then the vertices x, y, u, u', b would induce an $\overline{F}_1$ or an $\overline{F}_2$ depending on whether u, u' are adjacent or not. Finally, x cannot miss two vertices $b, b' \in M(t_k)$ since then the vertices x, y, u, b, b' would induce either an F_1 or an F_2. This is precisely Case (ii).

- Suppose that x misses $z \in V_S$. Because $z \in V_S$, z is adjacent to u, a, b. If z is not a child of t_{k-1}, then x is adjacent to u (note that $u \in V_S$), and because t_{k-2} is a P- or N-node, there exists $v \in M(t_{k-2})$ such that v misses both a, b, whereas v is adjacent to z; then, if x is adjacent to v, the vertices x, z, u, v, b induce a $\overline{P}_5$ otherwise the vertices x, z, u, v, a induce an $\overline{F}_1$. Thus, z is a child of t_{k-1}. Furthermore, x misses exactly one vertex in $M(t_k)$; if it missed $b, b' \in M(t_k)$, then the vertices x, z, a, b, b' would induce an F_1 or an F_2. Finally, if x saw $a, a' \in M(t_k)$, then the vertices x, z, a, a', b would induce an $\overline{F}_1$ or an $\overline{F}_2$. This is precisely Case (iii).

(b) t_{k-1} *is an N-node*: then, by Property A1, x sees the clique $K(t_{k-1})$ of the spider $G(t_{k-1})$. Let $k \in K(t_{k-1})$; clearly, k is adjacent to x, a, b.

- Suppose that x does not see the independent set $S(t_{k-1})$ of $G(t_{k-1})$. Then x misses V_{N_S}. Vertex x misses V_P as well: if it saw $y \in V_P$, then the vertices x, y, k, s, b would induce an F_1, where $s \in S(t_{k-1})$ is a neighbor of k. Additionally, x sees V_S: if it missed $z \in V_S$, then the vertices x, y, k, s', b would induce an $\overline{F}_1$, where $s' \in S(t_{k-1})$ is a non-neighbor of k. This is covered by Case (i).

- Suppose that x sees $S(t_{k-1})$. Then, x sees V_S: if it missed $z \in V_S$, then the vertices x, z, s, a, b would induce an $\overline{F}_2$, where $s \in S(t_{k-1})$. Additionally, x misses V_P: if it saw $y \in V_P$, then the vertices x, y, s, k, b induce an F_1, where the vertices $s \in S(t_{k-1})$ and $k \in K(t_{k-1})$ are non-adjacent. This is precisely Case (iv). ∎

Lemma 3.6. *Suppose that the x-partly-adjacent nodes of the md-tree $T(G)$ lie on a path $t_0 t_1 \cdots t_k$, where t_k is the x-partly-adjacent node farthest away from the root t_0 of $T(G)$. If t_k is an S-node then G' is P_4-sparse if and only if one of the following four (mutually exclusive) cases holds:*

(i) *Vertex x sees V_S and V_{N_K}, and misses V_P and V_{N_S}.*

(ii) *Vertex x sees V_{N_K}, all but one vertex, say, y, in V_S, and misses V_P and V_{N_S} where*
- *(ii.1) vertex y is a child of node t_{k-2} (which is an S-node),*
- *(ii.2) node t_{k-1} is a P-node with two children, the node t_k and one vertex, say, u (which is non-adjacent to x), and*
- *(ii.3) vertex x sees only a single vertex of $M(t_k)$, which is a child of t_k.*

(iii) *Vertex x sees V_S, V_{N_K}, and exactly one vertex, say, z, in V_P, and misses V_{N_S} where*
- *(iii.1) vertex z is a child of node t_{k-1} (which is a P-node), and*
- *(iii.2) node t_k has two children a, b, which are leaf-nodes such that a is adjacent and b is non-adjacent to x.*

(iv) *The node t_{k-1} is an N-node corresponding to a thin spider with clique $K(t_{k-1})$, vertex x misses V_P, V_{N_S}, $K(t_{k-1})$, and all but one vertex, say, b, in $M(t_k)$, and sees V_S and $V_{N_K} - K(t_{k-1})$.*

Proof. Since the P_4-sparse graphs are complement-invariant (Lemma 2.1), we consider the graph $\overline{G}$: its md-tree $T(\overline{G})$ is identical in structure to $T(G)$ except that P-nodes have become S-nodes and vice versa, thin spiders have become thick and vice versa, and their cliques and independent sets have been swapped. Since a node in $T(\overline{G})$ is x-partly-adjacent iff its corresponding node in $T(G)$ is x-partly-adjacent, Lemma 3.5 applies and gives us necessary and sufficient conditions for $\overline{G}'$ to be P_4-sparse. By exchanging P- and S-nodes, thin and thick spiders, their cliques and independent sets, and what x sees/misses in these conditions, we obtain the conditions of the lemma, which are the necessary and sufficient conditions for G' to be P_4-sparse. ∎