Efficiently recognizing, decomposing and triangulating hole- and diamond-free graphs

Anne Berry* Andreas Brandstädt† Vassilis Giakoumakis‡ Frédéric Maffray§

Abstract

We show that the atoms by clique separator decomposition of hole- and diamond-free graphs are of three simple types: clique, matched co-bipartite or chordal bipartite.

We present an $O(n^2)$ time algorithm to recognize these graphs and decompose them into atoms, whereas this decomposition requires $O(nm)$ time for almost all graph classes. To ensure this good time bound, we use algorithm LexBFS in a novel fashion, highlighting its behavior in the vicinity of clique separators in a general graph.

Given the atoms, we show how to compute a minimal triangulation in $O(n^2)$ time, thereby introducing algorithms for the minimal triangulation of both matched co-bipartite graphs and chordal bipartite graphs.

Our results enable us to provide efficient solutions to some enumeration and optimization problems.

Keywords: Algorithms, Graph classes, LexBFS, Clique separator decomposition.

1 Introduction

Cycle properties of graphs and their algorithmic aspects play a fundamental role in combinatorial optimization, discrete mathematics and computer science. Chordal graphs, weakly chordal graphs and perfect graphs are characterized in terms of cycle properties; these classes are of fundamental importance for algorithmic graph theory and various applications.

A *diamond* is a 4-clique minus one edge, a C_4 is a chordless cycle with 4 vertices, and a *hole* is a chordless cycle of length at least 5. A graph is *chordal* (also called *triangulated*) if it is hole- and C_4-free. See e.g. [11, 14, 27] for the many facets of chordal graphs. A graph is *weakly chordal* (also called *weakly triangulated*) if it is hole- and antihole-free. These graphs have been extensively studied in [15, 16, 17, 34]; they are perfect. Hole- and diamond-free graphs (*HD-free graphs* for short) generalize in a natural way the important class of chordal bipartite graphs (which are exactly the bipartite weakly chordal graphs), and diamond-free chordal graphs are the well-known block graphs - see [11] for various characterizations and the importance of chordal bipartite graphs as well as of block graphs.

Recently there has been much work on related classes such as even-hole- and diamond-free graphs (forbidding also C_4) [23] (see also [36]) and [13] dealing with the structure and recognition of C_4- and diamond-free graphs.

Work has also appeared describing properties of the atoms of a graph. Atoms are obtained by clique separator decomposition: a *clique separator* (or clique cutset) is a clique whose removal increases the number of connected components of the graph. Given a graph G with a clique

*LIMOS UMR CNRS 6158, Université Blaise Pascal, F-63 173 Aubière, berry@isima.fr

†Institut für Informatik, Universität Rostock, D-18051 Rostock, Germany, ab@informatik.uni-rostock.de

‡MIS, Université de Picardie Jules Verne, Amiens, France, vassilis.giakoumakis@u-picardie.fr

§CNRS, G-SCOP, Grenoble-INP, Université Joseph Fourier, Grenoble, France, frederic.maffray@inpg.fr

separator S, where C is a component of $G - S$, graph G is decomposed into subgraphs $C \cup S$ and $G - C$, which can in turn be decomposed. The final set of subgraphs obtained, which are then devoid of clique separators, are called *atoms*. This process was introduced by Tarjan [35]. It is well known that a graph is decomposable into cliques if and only if it is chordal. Though a graph may not have any clique separator, the process is of high interest since it is hole- and C_4- preserving (see [6] for a complete survey on clique separator decomposition).

Structural properties of atoms in some graph classes were used recently to give new efficient algorithms for the Maximum Weight Independent Set problem on these classes [9, 10]. These properties of atoms improve polynomial time solutions of this problem published previously in various papers.

Another problem related to the presence of holes is Minimal Triangulation, which consists in adding an inclusion-minimal set of edges to obtain a chordal graph. This problem has also given rise to many recent papers. The time bound for this problem, originally $O(nm)$ [32], was recently lowered to $O(n^{2.69})$ time [24] and even $O(n^{\alpha}logn)$ [19]. A special issue of Discrete Mathematics [1] is published on the subject, containing a survey on minimal triangulation [18].

Recently, [8] characterized the class of hole- and paraglider-free graphs via clique separator decomposition (a paraglider is a graph with vertices a, b, c, d, e and edges $ab, ac, bc, bd, cd, ae, de$). ¿From this, the structure of HD-free atomic graphs with no induced $\overline{C_6}$ can be deduced. [8] also gave as algorithmic consequences an $O(n^3)$ process for recognizing hole- and paraglider-free graphs, as well as some polynomial time algorithms for solving several optimization problems such as Maximum Clique and Coloring.

In this paper, we focus our attention on HD-free graphs. Our results are threefold:

We characterize the atoms.

We give an efficient recognition algorithm.

We give an efficient minimal triangulation algorithm.

First, we show that in an HD-free graph, an atom which is not a clique can be of only two 'opposite' kinds: a *chordal bipartite graph* (*i.e.* a bipartite graph with no hole) or a *matched co-bipartite graph* (*i.e.* a graph consisting of two disjoint cliques of the same size and a perfect matching between them); in the rest, we will call an edge between X and Y a *matching edge*.

This property is not sufficient to characterize the class of HD-free graphs, as a graph whose atoms are HD-free may contain diamonds which belong to two different atoms. This leads to the definition of two graph classes: HD-free graphs, and the super-class of graphs whose atoms are HD-free, which we call *HD-free decomposable* graphs. HD-free graphs generalize chordal bipartite graphs, whereas HD-free decomposable graphs also generalize chordal graphs.

We then go on to the algorithmic aspects of recognition and decomposition. We show that we can decompose and recognize HD-free graphs in $O(n^2)$ time. A straightforward approach would cost $O(nm)$ time to compute a decomposition into atoms [32]. Note that testing for a hole costs $O(m^2)$ time [29], which is roughly also the cost of testing for the presence of a diamond, so that a brute-force approach to recognizing HD-free graphs would cost $O(m^2)$ time.

This is an interesting result in its own right, as there are few other known classes where clique separator decomposition can be done faster than in the general case. AT-free claw-free graphs can be decomposed in linear time by first computing a minimal triangulation into a proper interval graph, which is linear [28].

To recognize HD-free graphs efficiently, we first decompose the graph using the clique separators of size 1 or 2. Since in a chordal bipartite graph the clique separators are of size at most 2, the chordal bipartite atoms are thereby separated from the rest. We are left with some possibly non-atomic subgraphs which can contain only matched co-bipartite atoms (with at least 6 vertices) and clique atoms; we call those graphs *multimat graphs*.

We then apply algorithm LexBFS to these multimat subgraphs. *LexBFS*, originally tailored to define a simplicial elimination scheme on a chordal graph [32], is a linear-time breadth-first

search which numbers the vertices from n to 1 and chooses at each step a vertex whose label (which is the list of its numbered neighbors) is lexicographically highest.

In this paper, we show two new properties for LexBFS, which are both true on general graphs. The first property is that the vertex which is numbered as 1 by LexBFS belongs to no clique separator. In a multimat graph, this will enable us to find a maximal set of vertices which belong to only one atom, and also to determine the corresponding clique separator which separates this atom from the rest of the graph. We will thus define for multimat graphs an elimination scheme which is very similar to the simplicial elimination scheme for chordal graphs.

The second property is a new invariant for LexBFS *w.r.t.* clique separators: given a LexBFS ordering α, we show that we can remove any component of any clique separator: the sub-ordering induced by α on the remaining subgraph is a LexBFS ordering in its own right. The benefit of this in a multimat graph is that when we find a set C of vertices which belong to only one atom, this defines a component of the corresponding clique separator. We can remove C and avoid running LexBFS on the resulting graph. Thus instead of running $O(n)$ times LexBFS, we use a single LexBFS ordering, thus gaining a factor of n.

This leads us to another aspect of our work: in the general case, computing a minimal triangulation is a mandatory pre-processing step to ensure an efficient time bound for clique decomposition [35]. Here, we will do the exact opposite: we use the atoms to ensure a good time bound for minimal triangulation.

We show how to compute a minimal triangulation of an HD-free atom in $O(n^2)$ time. For matched co-bipartite graphs, we define a standard minimal triangulation: computing each added edge will cost constant time. Triangulation of chordal bipartite graphs was investigated by Kloks and Kratsch [22] in the context of computing their treewidth, but they did not propose an efficient minimal triangulation algorithm for this class; we introduce an $O(n^2)$ minimal triangulation process for chordal bipartite graphs using the associated Γ-free matrix. Again, few graph classes are known to have a better bound than $O(nm)$ for this problem. This result contributes to illustrate how clique separator decomposition can be useful for developing more efficient algorithms.

The paper is organized as follows: in Section 2, we characterize the atoms of HD-free graphs. In Section 3, we present our $O(n^2)$ time algorithm to recognize and decompose HD-free graphs. In Section 4, we address the problem of computing a minimal triangulation for HD-free atoms in $O(n^2)$ time. After a conclusion and bibliography, we add an Appendix containing the proofs not given in the previous sections.

Throughout the paper, we consider finite undirected graphs; $+$ stands for disjoint union; for a graph $G = (V, E)$, $n = |V|$ and $m = |E|$, $d(x)$ denotes the degree of vertex x, $N[x] = N(x) \cup \{x\}$, and for $X \subset V$, $G(X)$ denotes the subgraph induced by X. A *separator* is a set of vertices whose removal increases the number of connected components. In a connected graph, a separator S is said to be *minimal* if $G(V - S)$ has at least two *full components*, *i.e.* components C_1 and C_2 such that $N(C_1) = N(C_2) = S$. For other classical graph definitions, see [14] and [11].

2 Characterizing HD-free atoms

In this section, we will show the following structural result:

Theorem 2.1. *Let G be an HD-free graph with no clique separator. Then G is either:*
- *a clique, or*
- *a matched co-bipartite graph, or*
- *a chordal bipartite graph.*

The following theorem describes the structure of HD-free atoms that contain a $\overline{C_6}$. This result also appears as a corollary in [8], but to make the paper self-contained we give a simpler direct proof below.

Theorem 2.2. *Let G be an HD-free atom that contains an induced $\overline{C_6}$. Then G is a matched co-bipartite graph.*

Proof. Let H be a maximal matched co-bipartite graph that extends a $\overline{C_6}$ in G. Let $V(H)$ be partitioned into two cliques $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k\}$, with $k \geq 3$, where H has matching edges $a_1b_1, \ldots, a_kb_k$.
Claim 1: For every vertex x of $G - H$, $N_H(x)$ is a clique.
Proof of Claim 1: Suppose that $N_H(x)$ is not a clique, so, up to relabeling, x is adjacent to a_1 and b_2. Then x is adjacent to one of a_3 and b_3, for otherwise $\{x, a_1, a_3, b_3, b_2\}$ induces a C_5. Assume, up to symmetry, that x is adjacent to a_3. If x is not adjacent to a_2, then $\{x, a_1, a_2, a_3\}$ induces a diamond; if x is adjacent to a_2, then $\{x, a_1, a_2, b_2\}$ induces a diamond, a contradiction. So Claim 1 holds. $\diamond$
Claim 2: Let F be any component of $G - H$. Then $N_H(F)$ is a clique.
Proof of Claim 2: Suppose that there are non-adjacent vertices x and y in $N_H(F)$. Let u be a neighbor of x in F and v be a neighbor of y in F. Note that $u \neq v$ by Claim 1. There is a chordless path P between u and v in F. We choose x, y, u, v and P such that P is as short as possible. Up to relabeling, let $x = a_1$ and $y = b_2$. Then Claim 1 implies that ub_3 and va_3 are not edges. Any interior vertex w of P is not adjacent to a_1 or a_3, for otherwise the subpath of P between w and v contradicts the choice of P; and similarly, w is not adjacent to b_2 or b_3. Since $V(P) \cup \{a_1, a_3, b_2, b_3\}$ cannot contain a hole, it must be that ua_3 and vb_3 are edges and $P = uv$. Now, since u is adjacent to a_1 and a_3, by Claim 1, we have $N_H(u) \subseteq A$ and similarly, $N_H(v) \subseteq B$. Then u is adjacent to every vertex a_i in A, for otherwise $\{u, a_1, a_3, a_i\}$ induces a diamond. So $N_H(u) = A$, and similarly, $N_H(v) = B$. But then $V(H) \cup \{u, v\}$ induces a matched co-bipartite graph, which contradicts the maximality of H. $\diamond$
In conclusion, if $G - H$ has a component F, then, by Claim 2, $N_H(F)$ is a clique separator (separating F from $H - N_H(F)$), a contradiction to the assumption that G is an atom. Therefore we have $G = H$, and so G is a matched co-bipartite graph. This finishes the proof of Theorem 2.2. $\square$

The following theorem describes the structure of HD-free atoms that do not contain a $\overline{C_6}$.

Theorem 2.3. *Let G be an HD-free atom that contains no induced $\overline{C_6}$. Then G is either a clique or a chordal bipartite graph.*

Proof. The fact that G is diamond-free means that (i) the neighborhood of every vertex is P_3-free, *i.e.*, it is a disjoint union of cliques, and (ii) every edge lies in exactly one maximal clique of G.

Suppose that G is not bipartite. So G contains an odd cycle, and since G is hole-free, G contains a triangle. Let a, x, y be three pairwise adjacent vertices in G. Let K be the maximal clique that contains $\{a, x, y\}$ (K is unique by (ii)). Suppose that $N[a] = K$. If $V(G) - K \neq \emptyset$, then $K - \{a\}$ is a clique separator, a contradiction. So $V(G) = K$, thus G is a clique. Now we may assume that a belongs to another maximal clique L. By (i), there is no edge between $K - \{a\}$ and $L - \{a\}$. Since G has no clique separator, $K - \{x\}$ and $K - \{y\}$ are not separators, so there is a chordless path P_x between x and L in $G - (K \setminus \{x\})$ and there is a chordless path P_y between y and L in $G - (K - \{y\})$. We choose P_x and P_y such that $V(P_x) \cup V(P_y)$ is minimal.

Let u_x be the neighbor of x in P_x and u_y be the neighbor of y in P_y. Clearly, $u_x, u_y \notin N[a]$. We have $yu_x \notin E$, for otherwise $\{a, x, y, u_x\}$ induces a diamond. Likewise, $xu_y \notin E$. Thus $u_x \neq u_y$. Vertex x has no neighbor in $P_y - \{y\}$ (else there is a path P'_x from x to L in $P_y \cup \{x\}$,

and we have $u_x \notin V(P'_x) \cup V(P_y)$, so the minimality of $V(P_x) \cup V(P_y)$ is contradicted). Similarly y has no neighbor in $P_x - \{x\}$. It follows that $u_x u_y \in E$, for otherwise u_x, x, y, u_y lie on a hole in $P_x \cup P_y$. Let v_x be the first vertex in $P_x - \{x\}$ (starting from u_x) that belongs to $N(a)$; then $u_x v_x$ is an edge, for otherwise $P[x, u_x, \ldots, v_x] \cup \{a\}$ induces a hole. If $u_y v_x \in E$, then $\{a, x, y, u_x, u_y, v_x\}$ induces a $\overline{C_6}$. If $u_y v_x \notin E$, then $\{a, y, u_y, u_x, v_x\}$ induces a C_5, a contradiction. In conclusion, G is bipartite, and since it is hole-free it is chordal bipartite. □

Theorem 2.1 now follows from Theorems 2.2 and 2.3. As discussed in the introduction, this atomic structure is not characterizing, as it is shared by the classes of HD-free graphs and of HD-free decomposable graphs.

3 Decomposing and recognizing HD-free graphs

We will now address the recognition problem for HD-free graphs. To achieve this, we will find the atoms of the graph, so the decomposition is obtained at the same time.

We will apply five successive steps. The first four are straightforward; as we will see, after completing these steps, we will be left with a last, but more difficult step, which is the recognition (and decomposition into atoms) of multimat graphs:

Definition 3.1. *A* multimat graph *is a diamond-free graph whose atoms are either matched co-bipartite graphs with at least six vertices, or cliques, and whose clique separators are of size at least* 3*.*

In a multimat graph, a matched co-bipartite atom $X + Y$ is separated from the other atoms by a clique separator of size at least 3, so atom $X + Y$ is separated from the other atoms by vertices of X or by vertices of Y, but never by a separator containing a matching edge.

We will describe the first four steps in Section 3.1, and then go on to give the more difficult details of handling multimat graphs, our final step, in Section 3.2.

3.1 Obtaining the multimat subgraphs

1. For recognizing HD-free graphs, the first step is to decompose G into its biconnected components [21], since G is HD-free if and only if its biconnected components are HD-free.

2. The second step is to decompose the graph using its clique separators of size 2. This can easily be extracted from the decomposition of the graph into its triconnected components [20]. The subgraphs obtained will be the maximal subgraphs containing no clique separator of size 2.

After these two steps, all the chordal bipartite atoms are defined. Note that if a matched co-bipartite atom has only two or four vertices, it is considered in all the rest to be a chordal bipartite atom, so what we call matched co-bipartite atoms have at least six vertices. At the end of Step 2, we obtain subgraphs which can be of four types: chordal bipartite atoms, clique atoms, matched co-bipartite atoms, or multimat graphs.

3. Our third step will be to check the subgraphs obtained after Step 2. Cliques and matched co-bipartite graphs can be recognized in linear time. Chordal bipartite graphs can be recognized in $O(\min(n^2, mlogn))$ time [30],[33]. Any other subgraph should be a multimat subgraph, and still needs to be decomposed and recognized. The cost of this third step is globally $O(n^2)$: the overlap between the various subgraphs is a clique minimal separator of size at most 2, there are less than n clique minimal separators, and each clique minimal separator yields at most n components [6], so the overall extra cost generated by the overlap is $O(n^2)$.

4. The fourth step is to check for the presence of diamonds which are separated by a clique separator of size 2: this can occur when two subgraphs which are not both chordal bipartite share such a separator. We will thus check, for each size 2 clique separator, which atoms or multimat subgraphs it belongs to. Since there are less than n atoms and less than n clique minimal separators in a graph [6], this can be done in $O(n^2)$ time.

After these four steps (which can be considered as a pre-processing), we will check and decompose each multimat subgraph separately in $O(n^2)$ time. The global cost of our decomposition and algorithm is thus $O(n^2)$, since the decompositions into biconnected and triconnected components both require only linear time [21], [20].

3.2 Decomposing a multimat graph

On each multimat subgraph obtained, we will apply the following decomposition step, introduced in [35] to produce atoms: *Let $G = (V, E)$ be a graph, S a clique separator of G, C a connected component of $G(V - S)$; decompose the graph into subgraphs $G(C \cup S)$ and $G(V - C)$.*

To obtain the atoms, we will apply a special instance of this decomposition step: we will use a component C consisting of vertices which belong to only one atom. Thus the decomposition step becomes:

Decomposition Step 3.2. *Let $G = (V, E)$ be a graph, S a clique separator of G, C a maximal connected set of vertices which belong to only one atom; produce atom $G(C \cup S)$ and remove C from the graph.*

The desired component and separator will be found in the vicinity of a vertex x_1 which is numbered as 1 by some LexBFS execution, by virtue of the properties of x_1 regarding separators which were introduced in [3]:

Theorem 3.3. *[3] The vertex numbered 1 by LexBFS in a non-clique graph belongs to a maximal clique module X_1 (i.e. $\forall u, v \in X_1, N[u] = N[v]$) whose neighborhood is a minimal separator (X_1 is called a* moplex*).*

This means that it is easy to find a minimal separator S_1 in $N(x_1)$ and that moreover all the vertices of $N(x_1)$ which are not in S_1 are 'true twins' with x_1, forming the moplex X_1. The particular moplex found by LexBFS has additional local connectivity properties which will be useful in our proofs

We will now present our first new result on LexBFS:

Theorem 3.4. *The vertex x_1 numbered 1 by an execution of LexBFS cannot belong to a clique minimal separator. Moreover, if X_1 is the maximal clique module containing x_1, no vertex of X_1 can belong to a clique minimal separator, i.e., all the vertices of X_1 belong to only one atom.*

Theorem 3.4 will enable us to define the atom A which vertex x_1 belongs to, and to find the corresponding clique separator. We will examine separately the case where this atom A is a clique and the case where A is matched co-bipartite.

If x_1 belongs to a clique atom, then $S_1 = N(X_1)$ is a clique minimal separator, and X_1 is a maximal connected set of vertices belonging to only one atom: applying Decomposition Step 3.2, we produce atom $X_1 \cup S_1$ and remove X_1.

When the atom containing x_1 is matched co-bipartite, things are more complicated. Let X be the clique of the matched co-bipartite which x_1 belongs to, let Y be the second clique. We will first remark that $X_1 = \{x_1\}$, as in a matched co-bipartite graph, no vertex can have a 'true twin'. $N(x_1)$ is made of two parts: the rest of X, and y_1, the neighbor of x_1 in Y. Since in a multimat graph the matched co-bipartite atoms have at least six vertices, X must be of size at

least 3, and thus $X - \{x_1\}$ must be of size at least 2. Thus vertex y_1 is isolated in S_1, and is easily distinguished from the vertices of $X - \{x_1\}$, which define a clique of size at least 2. The vertices of X all belong to only one atom, so $Y = N(X)$. This is summarized in the following Lemma:

Lemma 3.5. *Let G be a multimat graph containing at least two atoms, let x_1 be a vertex numbered as 1 by a LexBFS execution, let X_1 will be the maximal clique module containing x_1, let $S_1 = N(X_1)$; if x_1 belongs to a matched co-bipartite atom, then $X_1 = \{x_1\}$, S_1 consists of an isolated vertex y and a clique X' of size at least 2, x_1 belongs to atom $(X + Y)$, with $X = X' + \{x_1\}$, $Y = N(X)$, for every vertex x in X, $d(x) = d(x_1)$; let $Y' = \{y \in Y | N(y) \subset X + Y\}$, $Y - Y'$ is a clique separator, no vertex of $X + Y'$ belongs to more than one atom, and the atoms of G are the atoms of $G - (X + Y')$ plus atom $(X + Y)$.*

Note that if only clique minimal separators are used, the decomposition is unique and optimal in the sense that each atom obtained is a maximal connected subgraph containing no clique separator [25]. $Y - Y'$ is a clique minimal separator if and only if $G(V - (Y - Y'))$ has another full component besides $X + Y'$; if this is not the case, our process will generate an extra clique atom $Y - Y'$, which clearly is not maximal since it is included in atom $X + Y$.

Now that we have seen how to extract an atom with the help of a vertex numbered 1 by LexBFS, let us go on with considerations about complexity. We will need our second new result on LexBFS:

Theorem 3.6. *Let G be a graph, α a LexBFS ordering of G, S a clique separator of G, C a connected component of $G(V - S)$. Then the sub-order β defined by α on $V - C$ is a LexBFS ordering of $G(V - C)$.*

By virtue of Theorem 3.6, when we have defined a clique separator S and the corresponding connected component C of $G(V - S)$ whose vertices belong to only one atom, we can simply remove C and use the same LexBFS ordering on the resulting graph.

Another consideration regarding complexity issues is that in order to reach the time bound of $O(n^2)$, we cannot afford to search each subgraph Y of a matched co-bipartite atom as defined by Lemma 3.5 to check whether it is a clique: we could add an $O(n)$ cost factor since the same clique could be traversed $O(n)$ times if Y belongs to $O(n)$ different atoms. To solve this problem, we will use the following structural theorem:

Theorem 3.7. *Let G be a multimat graph, and let G'' be the graph obtained from G by removing all the matching edges of all the matched co-bipartite atoms of G. Then G'' is a disjoint union of cliques.*

Proof. Suppose there is a connected component C of G'' which is not a clique: C must contain vertices of the cliques of several atoms. There must be in C the cliques of two atoms A and A' which have a proper intersection of size at least 3, else there is a clique separator of size 1 or 2 in G, which is impossible since G is a multimat graph. Let v be in $A - A'$, w in $A' - A$: v, w, together with $A \cap A'$, contain a diamond. □

From Theorem 3.7, we can deduce the following inductive definition of a multimat graph:

Characterization 3.8. *A graph is a multimat graph if and only if it can be constructed from the inductive definition:*
Basis: Each clique with at least three vertices is a multimat graph, and each matched co-bipartite graph with at least six vertices is a multimat graph.
Rules: Let $G = (V_G, E_G)$ be any multimat graph, let K be a clique in G with at least three

vertices and let R be a new clique with $V_G \cap R = \emptyset$.
a) If K is a maximal clique, then $G' = (V_G \cup R, E_G \cup E(R) \cup \{xy \mid x \in K, y \in R\})$ is a multimat graph.
b) If $|R| = |K|$ and D is a perfect matching between R and K then $G' = (V_G \cup R, E_G \cup E(R) \cup D)$ is a multimat graph.

Corollary 3.9. *Let Q be the quotient graph obtained from a multimat graph G by contracting each maximal clique of size at least 3 of G into a vertex: Q is a tree.*

Our algorithm for recognizing multimat graphs applies the 'leaf'-elimination scheme associated with Characterization 3.8 and Corollary 3.9.

ALGORITHM Multimat graph recognition
Input : A graph G whose clique minimal separators are of size at least 3, a LexBFS ordering α of G.
Question: Is G a multimat graph? If yes, a decomposition of G into set $\mathscr{A}$ of atoms.

Initialize: $\mathscr{A} \leftarrow \emptyset$; $G' \leftarrow G$; $D \leftarrow \emptyset$;*//D will store the matching edges*
while *G' is not empty* **do**
  $x_1 \leftarrow \alpha(1)$; $X_1 \leftarrow$ maximal clique module containing x_1; $S_1 \leftarrow N_{G'}(X_1)$;
  if *S_1 is a clique//(x_1 belongs to a clique atom)* **then**
    $\mathscr{A} \leftarrow \mathscr{A} + \{N[x_1]\}$; $G' \leftarrow G' - X_1$;
    if $G' = S_1$ **then** $G' \leftarrow \emptyset$;
  else
    //(x_1 belongs to a matched co-bipartite atom);
    if *{(S_1 is not an isolated vertex plus a clique) or ($X_1 \neq \{x_1\}$)}* **then**
      Stop with output No;
    $y_1 \leftarrow$ isolated vertex of S_1; $Y \leftarrow \{y_1\}$; $X' \leftarrow S_1 - \{y_1\}$; $X \leftarrow X' + \{x_1\}$;
    $Y' \leftarrow \emptyset$;
    foreach $x_i \in X'$ **do**
      if $d(x_i) \neq d(x_1)$ **then** Stop with output No;
      else
        $y_i \leftarrow$ unique vertex of $(N(x_i) - X)$; $D \leftarrow D + \{x_i y_i\}$; $Y \leftarrow Y + \{y_i\}$;
        if $d(y_i) = |S_1|$ **then** $Y' \leftarrow Y' + \{y_i\}$;
    $\mathscr{A} \leftarrow \mathscr{A} + \{X + Y\}$; $G' \leftarrow G' - (X \cup Y')$;
  $\alpha \leftarrow$ restriction of α to G';
$G'' \leftarrow G - D$;
if *G'' is not the union of disjoint cliques* **then** Stop with output No;
$Q \leftarrow$ quotient graph obtained from G by contracting the cliques of G'';
if *Q is not a tree* **then** Stop with output No;
Return(Yes, $\mathscr{A}$).

Theorem 3.10. *Algorithm Multimat graph recognition correctly recognizes and decomposes a multimat graph in $O(n^2)$ time.*

4 Triangulating HD-free graphs in $O(n^2)$ time

A *triangulation* of a non-chordal graph $G = (V, E)$ is a chordal supergraph $H = (V, E + F)$. A triangulation H of G is *minimal* if $\forall F' \subsetneq F$, $H = (V, E + F')$ fails to be chordal, and a minimal triangulation is *minimum* if F is the smallest over all possible minimal triangulations. F is the

set of *fill edges*. We will say that we *saturate* a set X of vertices if we add all edges necessary to make X a clique.

In this section, we present efficient algorithms for triangulating matched co-bipartite graphs and chordal bipartite graphs. This leads to the triangulation of HD-free graphs and of HD-free decomposable graphs.

4.1 Triangulating matched co-bipartite graphs

A co-bipartite graph is AT-free and claw-free, so a minimal triangulation can be computed in linear time [28]; however, the overlap generated by the large clique separators may cause the cost to be more than $O(nm)$ time globally, so there may be no gain on the general complexity for computing a minimal triangulation. Fortunately, a matched co-bipartite graph is easy to triangulate without searching the graph.

Theorem 4.1. *Let $G = (X + Y, E)$ be a matched co-bipartite graph, with $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$, $|X| = |Y| = k$, and with matching edges $\{x_1y_1, x_2y_2, \ldots, x_ky_k\}$. A minimal (and minimum) triangulation of G is obtained by adding the following sets of edges: $\{x_1\} \times \{y_2, y_3, \ldots, y_k\}$, $\{x_2\} \times \{y_3, \ldots, y_k\}$,$\ldots$,$\{x_i\} \times \{y_{i+1}, \ldots, y_k\}$,$\ldots$, $\{x_{k-1}\} \times \{y_k\}$.*

Proof. Observe first that every C_4 in G is formed from the vertices of two matching edges, so to break a C_4 formed by the vertices x_iy_i and x_jy_j it is necessary and sufficient to add one of the edges x_iy_j or x_jy_i. Assume now that graph G' resulting from these edge additions contains a C_4 $\{x_i, x_j, y_k, y_l\}$, with $i < j$ and $k < l$; $k, l < i$ else edge x_iy_k has been added; but then edge x_iy_l does not exist, a contradiction. All added edges are necessary, so the triangulation is minimal. □

As each added edge costs $O(1)$ time to compute, the following results from Theorem 4.1:

Theorem 4.2. *A minimal triangulation of a matched co-bipartite graph can be computed in $O(|F|)$ time, where F is the set of fill edges.*

4.2 Triangulating chordal bipartite graphs

We will need a few preliminary results and definitions.

Definition 4.3. *[31] Two minimal separators S and T are said to be* crossing *if T separates a from b for some non-edge ab of S.*

Note that if S and T are two crossing separators, and if some vertex y sees all the vertices of S, then y must be in T.

Theorem 4.4. *[31] For a non-chordal graph $G = (V, E)$, a minimal triangulation is obtained by saturating a maximal set of pairwise non-crossing minimal separators.*

Property 4.5. *For a bipartite graph $G = (X + Y, E)$, saturating X yields a triangulation.*

Corollary 4.6. *For a bipartite graph $G = (X + Y, E)$, there is a minimal triangulation which is obtained by adding only edges with both endpoints in X.*

Property 4.7. *For a chordal bipartite graph $G = (X + Y, E)$, the minimal triangulation which is obtained by adding only edges with both endpoints in X is unique. Moreover, x_ix_j is an added edge of this minimal triangulation if and only if x_i and x_j have at least two common neighbors in G.*

Proof. Let $G = (X+Y, E)$ be a chordal bipartite graph, let S and S' be two minimal separators of G which are subsets of X. We claim that S and S' are non-crossing separators: since G is a weakly chordal graph, each co-connected component has a confluence point in every full component of S [5]; since S is a stable set, S is co-connected, so there are two vertices y_1 and y_2 which see all the vertices of S; clearly, y_1 and y_2 are in Y. But any minimal separator which crosses S must contain both y_1 and y_2, so this crossing separator cannot be in X. Since a minimal triangulation needs to saturate a maximal set of pairwise non-crossing minimal separators, we must use all the minimal separators which are included in X. A pair $\{x_i, x_j\}$ of X will belong to some such a minimal separator *iff* there are two vertices of Y which see both x_i and x_j, since failing to add such an x_ix_j edge would leave a C_4 in the resulting graph. □

We will use the following remarkable property of chordal bipartite graphs (a Γ is a 2×2 submatrix with the unique 0 entry at the lower right-hand corner):

Property 4.8. *[26]Let M be the $0-1$ neighborhood matrix of a chordal bipartite graph; there is a reordering of M which is Γ-free.*

Such a reordering can be computed in $O(\min(mlogn, n^2))$ time [30],[33]. We will use this Γ-free matrix M of a chordal bipartite graph $G = (X+Y, E)$ to find all the $\{x_i, x_j\}$ pairs of X which have two common neighbors in Y. In M, let the elements of X be rows and the elements of Y be columns. We will check each pair $\{x_i, x_j\}$ of X; let x_i be before x_j in M; let x_iy_k be the last non-zero entry on row x_i; we will check entry x_jy_k: if it is 0, then any common neighbor y_l would induce a $\Gamma\{x_i, x_j, y_l, y_k\}$, so x_i and x_j can have no common neighbor in Y; if entry x_jy_k is 1, then there is a common neighbor y_k; to check for the presence of a second neighbor, we will use the next-to-last non-zero entry on row x_i, call it y_l, and then likewise test entry x_iy_l. If it is 0, then there is only one common neighbor, if it is 1 then $\{x_i, x_j\}$ have two common neighbors y_k and y_l. This costs constant time.

Doing this for each pair of vertices of X will cost $O(n^2)$ time if M is pre-processed to store the last and next-to-last entry for each row. The following theorem results:

Theorem 4.9. *A minimal triangulation of a chordal bipartite graph can be computed in $O(n^2)$ time.*

4.3 Triangulating HD-free graphs and HD-free decomposable graphs

Since clique separator decomposition is hole- and C_4- preserving, a minimal triangulation can be computed separately for each non-clique atom, and the resulting set of added edges over the whole graph will define a minimal triangulation. The following theorem thus results:

Theorem 4.10. *A minimal triangulation of an HD-free graph and of an HD-free decomposable graph can be computed in $O(n^2)$ time.*

Proof. From Theorems 4.2 and 4.9, the atoms of an HD-free graph and of an HD-free decomposable graph can be computed in $O(n^2)$ time. The chordal bipartite atoms have a small overlap, so no extra cost arises from triangulating these atoms separately. The other non-clique atoms are matched co-bipartite, and each fill edge can be computed in constant time, so the overall complexity is in $O(n^2)$ time. □

5 Conclusion

We have proved that the atoms of HD-free graphs are matched co-bipartite graphs, cliques or chordal bipartite graphs. We have shown how to recognize a graph of this class, obtain the

decomposition, and triangulate it in $O(n^2)$ time. We have thereby introduced a new class of graphs, multimat graphs, for which we have uncovered strong structural properties.

Our algorithm for multimat graphs uses a novel approach with LexBFS, illustrating how there are other classes than chordal graphs in which LexBFS can be

Clique separator decomposition was refined by [25] to a unique decomposition using clique *minimal* separators; the atoms obtained are then characterized as the maximal connected subgraphs with no clique separator. With an appropriate data structure, we can refine our decomposition into the unique one, by removing the extra cliques produced when encountering a matched co-bipartite atom, without losing our $O(n^2)$ time bound.

Our algorithm can also easily be adapted to recognizing the superclass of HD-free decomposable graphs (the class of graphs whose atoms are HD-free); these graphs are H-free but not necessarily D-free, as there can be a diamond between two atoms.

A decomposition into HD-free atoms enables us to solve some enumeration problems: we can enumerate all the $\overline{C_6}$s, which are found within the matched co-bipartite atoms; we can also enumerate the minimal separators of the graph: aside from the less than n clique minimal separators, the other minimal separators (of which there can be an exponential number) are partitioned into the atoms; minimal separators of a matched co-bipartite graph are easily defined in constant time per separator, and in a chordal bipartite graph, which is weakly chordal, there are at most m minimal separators and these can be found in $O(m)$ time each [5]. Globally, this complexity is better than the current $O(nm)$ time per minimal separator in a general graph [4]. We can also enumerate the maximal cliques, which we can easily find in the multimat subgraphs as in Theorem 3.7; we can thus also solve the MAXIMUM CLIQUE problem. Another hard problem which can be solved is that of MINIMUM FILL-IN: all minimal triangulations of a matched co-bipartite graph are minimum, so a minimum triangulation is easy to find given a minimum triangulation of each chordal bipartite atom, which is polynomial since the graph is weakly chordal.

References

[1] A. Berry, J. R. S. Blair, G. Simonet. Preface to Special issue on Minimal Separation and Minimal Triangulation. *Discrete Math.*, 306(3):293, 2006.

[2] A. Berry, J. R. Blair, J.-P. Bordat, G. Simonet. Graph extremities defined by search algorithms. *Algorithms*, 3(2):100–124, 2010.

[3] A. Berry, J. P. Bordat. Separability generalizes Dirac's theorem. *Discrete Appl. Math.*, 84(1-3):43–53, 1998.

[4] A. Berry, J. P. Bordat, O. Cogis. Generating all the minimal separators of a graph. *Int. J. Found. Comput. Sci.*, 11(3):397–403, 2000.

[5] A. Berry, J.-P. Bordat, P. Heggernes. Recognizing weakly triangulated graphs by edge separability. *Nordic J. of Computing*, 7:164–177, September 2000.

[6] A. Berry, R. Pogorelcnik, G. Simonet. An introduction to clique minimal separator decomposition. *Algorithms*, 3(2):197–215, 2010.

[7] A. Brandstädt, F.F. Dragan, F. Nicolai. LexBFS-orderings and powers of chordal graphs, *Discrete Math.* 171 (1997) 27-42.

[8] A. Brandstädt, V. Giakoumakis, F. Maffray. Clique separator decomposition of hole-free and diamond-free graphs and algorithmic consequences. *Discrete Appl. Math.*, 160:471-478, 2012.

[9] A. Brandstädt, C.T. Hoàng, On clique separators, nearly chordal graphs, and the Maximum Weight Stable Set Problem, IPCO 2005, Berlin, LNCS 3509, 265-275; *Theoret. Comput. Sci.*, 389:295-306, 2007.

[10] A. Brandstädt, V.B. Le, S. Mahfud, New applications of clique separator decomposition for the Maximum Weight Stable Set Problem, *Theoret. Comput. Sci.*, 370:229-239 2007.

[11] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, *SIAM Monographs on Discrete Math. Appl., Vol.* 3, Philadelphia, 1999.
[12] D. G. Corneil, R. Krueger. A Unified View of Graph Searching. *SIAM J. Discrete Math.* 22(4): 1259-1276 (2008).
[13] E. M. Eschen, C. T. Hoàng, J. Spinrad, R. Sritharan. On graphs without a C_4 or a diamond. *Discrete Appl. Math.*, 159(7):581–587, 2011.
[14] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs. *Academic Press*, 1980.
[15] R.B. Hayward, Weakly triangulated graphs, *J. Combin. Theory Ser. B*, 39:200-208, 1985.
[16] R.B. Hayward, C.T. Hoàng, F. Maffray, Optimizing weakly triangulated graphs, *Graphs Combin.*, 5:339-349, 1989; erratum in 6:33-35,1990.
[17] R.B. Hayward, J.P. Spinrad, R. Sritharan, Improved algorithms for weakly chordal graphs, *Graphs Combin.* 3(2), Art. 14, 2007.
[18] P. Heggernes, Minimal triangulations of graphs: A survey, *Discrete Math.*, 306(3):297-317, 2006.
[19] P. Heggernes, J. A. Telle, Y. Villanger, Computing Minimal Triangulations in Time $O(n^{\alpha}logn) = o(n^{2.376})$, *SIAM J.Discrete Math.*, 19(4):900-913, 2005.
[20] J. E. Hopcroft, R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973.
[21] J. E. Hopcroft, R. E. Tarjan. Efficient algorithms for graph manipulation [h] (algorithm 447). *Commun. ACM*, 16(6):372–378, 1973.
[22] T. Kloks, D. Kratsch. Treewidth of Chordal Bipartite Graphs. *J. Algorithms* 19(2): 266-281, 1995.
[23] T. Kloks, H. Müller, K. Vušković, Even-hole-free graphs that do not contain diamonds: A structure theorem and its consequences. *J. Combin. Theory Ser. B*, 99(5):733–800, 2009.
[24] D. Kratsch, J. Spinrad, Minimal fill in $O(n^{2.69})$ time, *Discrete Math.* 306(3): 366-371, 2006.
[25] H.-G. Leimer. Optimal decomposition by clique separators. *Discrete Math.*, 113(1-3):99–123, 1993.
[26] A. Lubiw. Doubly lexical orderings of matrices. *SIAM J. Comput.*, 16(5):854–879, 1987.
[27] T.A. McKee, F.R. McMorris, Topics in Intersection Graph Theory, *SIAM Monographs on Discrete Math. and Appl.* Vol. 2, Philadelphia, 1999.
[28] D. Meister. Recognition and computation of minimal triangulations for AT-free claw-free and co-comparability graphs. *Discrete Appl. Math.*, 146(3):193–218, 2005.
[29] S. D. Nikolopoulos, L. Palios. Detecting holes and antiholes in graphs. *Algorithmica*, 47(2):119–138, 2007.
[30] R. Paige, R. E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.* 16 (6): 973-989.
[31] A. Parra, P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Appl. Math.*, 79(1-3):171–188, 1997.
[32] D. J. Rose, R. E. Tarjan, G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.
[33] J. Spinrad. Doubly lexical ordering of dense 0 - 1 matrices. *Inform. Process. Lett.*, 45(5):229–235, 1993.
[34] J.P. Spinrad, R. Sritharan, Algorithms for weakly triangulated graphs, *Discrete Appl. Math.*, 59:181-191, 1995.
[35] R. E. Tarjan. Decomposition by clique separators. *Discrete Math.*, 55(2):221–232, 1985.
[36] K. Vušković, Even-hole-free graphs: a survey, electronically available in *Appl. Anal. and Discrete Math.*, 2010.

Appendix: proofs

Theorem. 3.4. *The vertex x_1 numbered 1 by an execution of LexBFS cannot belong to a clique minimal separator. Moreover, if X_1 is the maximal clique module containing x_1, no vertex of X_1 can belong to a clique minimal separator.*

Proof. Let x_1 be the vertex numbered 1 by some LexBFS execution, and suppose to the contrary that $x_1 \in S$ for some clique minimal separator S. Then $G(V - S)$ has at least two connected components, say C_1 and C_2 such that x_1 sees a vertex $y_1 \in C_1$ and $y_2 \in C_2$, respectively. By Theorem 3.3, x_1 belongs to a maximal clique module, X_1, whose neighborhood $N(X_1)$ is a minimal separator. Note that $y_1, y_2 \notin X_1$ since they are not twins of x_1, but $y_1, y_2 \in N(X_1)$. Thus, taking $N(X_1)$ as a minimal separator, there is a full component C different from X_1 seeing y_1 and y_2, and now there is a path between y_1 and y_2 in $G(V - S)$ (note that $S \subseteq X_1 \cup N(X_1)$) contradicting the assumption that S separates y_1 and y_2. □

Lemma 5.1. *[2] Let X_1 be the maximal clique module containing the vertex numbered 1 by a LexBFS execution, let $S_1 = N(X_1)$ be the corresponding minimal separator. If $G(V - S_1)$ has more than two connected components, then LexBFS numbers each component completely before starting on a new component; let $C_1 \dots C_k$ be the connected components of $G(V - S_1)$, where C_1 is the component numbered first, let $i < j$; then all the vertices of C_j are adjacent to all the vertices of $N(C_i)$.*

Lemma. 3.5. *Let G be a multimat graph containing at least two atoms, and such that x_1 belongs to a matched co-bipartite atom. Then $X_1 = \{x_1\}$, S_1 consists of an isolated vertex y and a clique X', x_1 belongs to atom $(X + Y)$, with $X = X' + \{x_1\}$, $Y = N(X)$, for every vertex x in X, $d(x) = d(x_1)$; let $Y' = \{y \in Y | N(y) \subset X + Y\}$, $Y - Y'$ is a clique separator, no vertex of $X + Y'$ belongs to more than one atom, and the atoms of G are the atoms of $G - (X + Y')$ plus atom $(X + Y)$.*

Figure 1 illustrates this case.

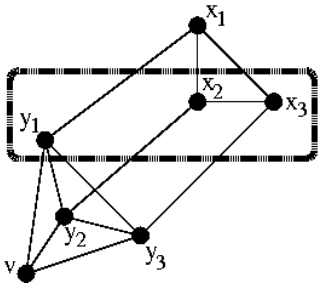

Figure 1: *x_1 is number 1 LexBFS vertex in a matched co-bipartite atom. $X_1 = \{x1\}$; $S1 = N(x1) = \{y_1, x_2, x_3\}$; X together with $Y = \{y_1, y_2, y_3\}$ form the atom.*

Proof. We claim that $X_1 = \{x_1\}$: let $\{a, b\}$ be a non-edge of S_1: if x_1 and x_1' both belong to the same clique module, $\{a, b, x_1, x_1'\}$ induce a diamond.

By Theorem 3.4, x_1 belongs to only one atom, thus x_1 sees only the vertices of its own atom, $X + Y$, with $x_1 \in X$, therefore $N(x_1) = S_1$ consists of the rest of X, which is X', a clique, and of the unique vertex y_1 of Y which sees x_1.

We claim that no vertex of X sees another atom: suppose X' sees another atom A'; clearly, X' separates A from A'; but then S_1 induces more than two components; let C_i be the component containing $Y - \{y_1\}$, let C_j be the component containing $A' - X$. $N(C_i) \subset S$, so by Lemma

5.1, if $i < j$, all vertices of $A' - X$ must see y_1, which is impossible as X separates A' from Y and $y_1 \in Y$; if $j < i$, y_2 must see x_3, which is also impossible.

Since X sees no other atom, $Y = N(X)$ and $d(x) = d(x_1)$.

Y is a clique separator since we assume that G has at least two atoms. Clearly, the vertices of Y' belong to only one atom, so $Y - Y'$ is a clique separator, $C = X + Y'$ is a component of $Y - Y'$, and C consists of vertices which belong to only one atom. We can thus apply Decomposition Step 3.2: decompose the graph into $G(V - C)$ and atom $G(C \cup Y)$. □

Theorem. 3.6. *Let G be a graph, α a LexBFS ordering of G, S a clique separator of G, and D a connected component of $G(V - S)$. Then the sub-order β defined by α on $V - D$ is a LexBFS ordering of $G(V - D)$.*

Proof. The reader is referred to [32] and [7] for a description of LexBFS. We will use the following characterization of LexBFS orderings from [7] (see also [12]): α is a LexBFS ordering if and only if the following 4-point condition holds:

$$\text{For all } a < b < c \text{ with } ac \in E, bc \notin E \text{ there is } d > c \text{ with } bd \in E, ad \notin E. \tag{1}$$

We also say that vertex d *results* from $a < b < c$.

Now suppose that for a connected component D of $G(V - S)$, sub-order β of α on $G(V - D)$ fails to be a LexBFS ordering of $G(V - D)$; there are vertices $a < b < c$ in β but no $d > c$ fulfills condition (1) in β; since α is a LexBFS ordering of G, such a d must exist in α by condition (1).

Thus, d must be in the component D which was removed. Since $bd \in E$, $b \in S$ follows; likewise, since $bc \notin E$, $c \notin S$. Therefore, in G, c and d lie in two different connected components C and D of $G(V - S)$, say $c \in C$ and $d \in D$.

We claim that then the repeated application of condition (1) leads to infinitely many resulting new vertices (which is a contradiction since G is finite):

Since $b \in S$, $c \in C - S$, and $d \in D - S$, and $bd \in E$, we can apply condition (1) to $b < c < d$; let e be the resulting vertex; $ce \in E$ and $be \notin E$. Thus, $e \notin S$ and thus $e \in C - S$. Moreover, $de \notin E$. Thus, we can apply condition (1) to $c < d < e$; let f be the resulting vertex and so on. Now, as long as the new vertices are not in S, the procedure continues until the resulting vertex, say w, is in S. Then $bw \in E$, and since w resulted from, say $u < v < x$, $uw \notin E$ and (1) can be applied to $b < u < w$; let the resulting vertex be y. Note that u and v are in different components $u \in C$ and $v \in D$ or vice versa. Since $by \notin E$, y is in $C - S$ if $u \in C - S$ or in $D - S$ if $u \in D - S$, respectively. Now the process continues with $u < v < y$ since v and y are in different components until again a resulting vertex is in S but then the same argument as above can be applied. This shows the theorem. □

Characterization. 3.8. *A graph is a multimat graph if and only if it can be constructed from the inductive definition:*
Basis: Each clique with at least three vertices is a multimat graph, and each matched co-bipartite graph with at least six vertices is a multimat graph.
Rules: Let $G = (V_G, E_G)$ be any multimat graph, let K be a clique in G with at least three vertices and let R be a new clique with $V_G \cap R = \emptyset$.
a) If K is a maximal clique, then $G' = (V_G \cup R, E_G \cup E(R) \cup \{xy \mid x \in K, y \in R\})$ is a multimat graph.
b) If $|R| = |K|$ and D is a perfect matching between R and K then $G' = (V_G \cup R, E_G \cup E(R) \cup D)$ is a multimat graph.

Proof. Clearly, only multimat graphs are constructed, as well as all multimat graphs with only one atom. Suppose a multimat graph G cannot be constructed: G has at least two atoms, let S and C be the clique separator and component containing a number 1 LexBFS vertex which either define a clique atom, or belong to a matched co-bipartite atom as described by Lemma 3.5:we can remove C, and an induction on the number of atoms leads to a contradiction. □

Theorem. 3.10. *Algorithm Multimat graph recognition correctly recognizes and decomposes a multimat graph in $O(n^2)$ time.*

Proof. If G is a multimat graph, G is decomposed into atoms, by Theorem 2.1 and by Lemma 3.5. At the end, the graph is emptied.

Suppose G is not a multimat graph: either one of its atoms is not a clique or a matched co-bipartite graph, or there is a diamond between two atoms.

Any matched co-bipartite atom encountered has the correct degree for vertices of X, and if Y is not a clique, by Theorem 3.7, either the algorithm will fail when finding that G'' is not a set of disjoint cliques, or when finding that the quotient graph Q is not a tree. Also by Theorem 3.7, there cannot be a diamond between two atoms since G'' is a set of disjoint cliques.

LexBFS is run only once and requires linear time. It is easy to determine X_1 rapidly at each step, as the vertices of X_1 are numbered consecutively [3].

Computing $S_1 = N[X_1]$ costs $O(m)$ globally since X_1 is removed at the end of the step. Testing whether S_1 is a clique or an isolated vertex plus a clique costs $O(m)$ time, since if x_1 is in a matched co-bipartite atom, all edges of S_1 are in X, which is removed at the end of the step; if x_1 is in a clique atom, S_1 belongs to a unique other atom which is a matched co-bipartite atom Z: S_1 cannot belong to another clique atom, or this together with X_1 and S_1 would induce diamonds; and because of Lemma 5.1, S_1 cannot belong to two different matched co-bipartite atoms. Thus S_1 will be traversed exactly once more when Z is processed. Computing the neighborhoods of the form $N(x_i)$ costs $O(m)$ time, as $x_i \in X$ and X is removed at the end of the step. Checking G'' costs $O(m)$ time, since the tested cliques are pairwise disjoint. Constructing quotient graph Q and checking that it is a tree costs $O(m)$ time. □