Homogeneity vs. Adjacency: Generalising Some Graph Decomposition Algorithms

B.-M. Bui Xuan[1], M. Habib[2], V. Limouzy[2], and F. de Montgolfier[2]

[1] LIRMM, Université Montpellier 2, France
buixuan@lirmm.fr
[2] LIAFA, Université Paris 7, France
{habib, limouzy, fm}@liafa.jussieu.fr

Abstract. In this paper, a new general decomposition theory inspired from modular graph decomposition is presented. Our main result shows that, within this general theory, most of the nice algorithmic tools developed for modular decomposition are still efficient.

This theory not only unifies the usual modular decomposition generalisations such as modular decomposition of directed graphs and of 2-structures, but also decomposition by star cutsets.

1 Introduction

Several combinatorial algorithms are based on partition refinement techniques [16]. Graph algorithms make an intensive use of vertex splitting, the action of partitioning classes between neighbours and non-neighbours of a vertex. For instance, all known linear-time modular decomposition algorithms [4,6,8,12,14] use this technique.

In bioinformatics also, the distinction of a set by an element, so-called splitter, seems to play an important role, as for example in the nice computation of the set of common intervals of two permutations by T. Uno and M. Yagiura [17].

We investigate the abstract notion of splitters and subsequently propose a formalism based on the concept of homogeneity. Our aim is a better understanding of the existing modular decomposition algorithms by characterising the algebraic properties on which they are based. Our main result is that most of the nice algorithmic tools developed to compute a representation for modular decomposition [4,6,8,12,14] are still efficient within this general theory.

This theory not only unifies the usual modular decomposition generalisations such as modular decomposition of directed graphs [13] and of of 2-structures [9], but also allows to handle star cutsets.

The paper is structured as follows: we first detail the new combinatorial decomposition theory, then present a general algorithmic framework, and close the paper with interesting outcomes.

2 Homogeneity, a New Viewpoint

Throughout this section X is a finite set, and $\mathcal{P}(X)$ the family of all subsets of X. Two sets A and B *overlap* if $A \cap B$, $A \setminus B$ and $B \setminus A$ are all nonempty. It

F.V. Fomin (Ed.): WG 2006, LNCS 4271, pp. 278–288, 2006.

is denoted $A \between B$. A *reflectless* triple is $(x, y, z) \subseteq X^3$ with $x \neq y$ and $x \neq z$. Reflectless triples will be denoted by $(x|yz)$ instead of (x, y, z) since the first element does not play the same role. Let H be a relation over the reflectless triples of X. Given $s \in X$, we define H_s as the binary relation on $X \setminus \{s\}$ such that $H_s(x, y)$ iff $H(s|xy)$.

Definition 1. *H is a* homogeneous relation on X *if, for all $s \in X$, H_s is an equivalence relation on $X \setminus \{s\}$ (i.e. it fulfills the Symmetry and Reflexivity and Transitivity properties). The equivalence classes of H_s are called the s-*classes *and denoted $H_s^1 ... H_s^k$.*

Definition 2. *Let H be a homogeneous relation on X. $M \subseteq X$ is a* module *of H if $\forall m, m' \in M, \quad \forall s \in X \setminus M, \quad H(s|mm')$.*

If $\neg H(s|mm')$ we say that s distinguishes *m from m', or is a* splitter *of $\{m, m'\}$. A module M is trivial if $|M| \leq 1$ or $M = X$. The family of modules of H is denoted by $\mathcal{M}_H$, or $\mathcal{M}$ if not ambiguous. H is prime if $\mathcal{M}_H$ is reduced to the trivial modules.*

Remark 1. From the definition it is obvious that, given a module M, if $\neg H(s|xy)$ for some $x, y \in M$ then $s \in M$.

Homogeneity and distinction can be applied to graphs. Indeed, there is a natural homogeneous relation associated to graphs as follow:

Definition 3. *The* standard homogeneous relation $H(G)$ *of a directed graph $G = (X, E)$ is defined such that, for all $s, x, y \in X$, $H(s|xy)$ is true if and only if the following two conditions hold:*
1. either both x and y or none of them are in-neighbours of s, and
2. either both x and y or none of them are out-neighbours of s.

In other word, $H(s|xy)$ tells if s "sees" x and y in the same way. Of course this definition also holds for undirected graphs, tournaments, and can also be extended to 2-structures [9].

Proposition 1. *For a graph G, the modules of its standard homogeneous relation $H(G)$ are the modules of G in the usual sense [11,15].*

Proposition 2. *For all $A, B \in \mathcal{M}$ if $A \between B$ then $(A \cap B) \in \mathcal{M}$ and $(A \cup B) \in \mathcal{M}$.*

This property is called here *closure under intersection and union.* It is easy to check, and can be used to prove:

Proposition 3 (Lattice structure). *Let H be a homogeneous relation on X and $\mathcal{M}'_H = \mathcal{M}_H \cup \{\emptyset\}$. Then, $(\mathcal{M}'_H, \subseteq)$ is a lattice.*

This lattice is a sublattice of the boolean lattice (hypercube) on X. Moreover, if we consider $A \in \mathcal{M}$ such that $|A| \geq 1$, $\mathcal{M}(A) = \{M \in \mathcal{M}_H$ *and* $M \supseteq A\}$ then $(\mathcal{M}(A), \subseteq)$ is a distributive lattice. Let us now define some useful types of homogeneous relations.

Definition 4. *A homogeneous relation H is said to be*

- ***Graphical*** *if* $\forall\ x, y, z \in X,\ H(x|yz)\ \wedge\ H(y|xz)\ \Rightarrow\ H(z|xy)$
- ***Quotiental*** *if* $\forall\ s, t, x, y \in X,\ H(x|st)\ \wedge\ H(y|st)\ \wedge\ H(t|xy)\ \Rightarrow\ H(s|xy)$
- ***Digraphical*** *if* $\forall s, t, x, y \in X, H(x|st) \wedge H(y|st) \wedge H(t|sx) \wedge H(t|sy) \Rightarrow H(s|xy)$

Notice that if H is ***Quotiental*** then for each module M and for all $x, y \in M$ and $s, t \in X \setminus M$, $H(x|st) \Leftrightarrow H(y|st)$. Indeed, for the *Quotiental* relations, elements in a module M uniformly perceive a set A not intersecting M: if one element of M distinguishes A then so do all. This allows to shrink M into a single element, the quotient by M, or to pick a *representative element* from the module. This is here called the *Quotient* property.

Given $A \subseteq X$ one can define the induced relation $H[A]$ as H restricted to reflectless triples of A^3. If A is a module we have the following nice property:

Proposition 4 (Restriction). *Let H be a homogeneous relation, M a module and $M' \subseteq M$. Then, $M' \in \mathcal{M}_{H[M]} \Leftrightarrow M' \in \mathcal{M}_H$.*

Recursiveness can therefore be used when dealing with modules. Notice that the proposition is not always true if M is not a module. The *Quotient* and *Restriction* properties were used first with modular decomposition of graphs and are useful for algorithmics [15].

3 Submodularity of Homogeneous Relations

Definition 5. *A set function $\mu : \mathcal{P}(X) \to \mathbb{R}$ is submodular if and only if for all $A, B \subseteq X$ $\mu(A) + \mu(B) \geq \mu(A \cup B) + \mu(A \cap B)$ (see e.g. [10]).*

Theorem 1 (Submodularity). *Let H be a homogeneous relation on X. Let $s(E)$ be the function counting the number of splitters of a nonempty subset E of X, and such that $s(\emptyset) = -|X|$. s is submodular.*

Proof. It suffices to prove $s(A) + s(B) \geq s(A \cup B) + s(A \cap B)$ for all overlapping $A, B \subseteq X$. So let $A, B \subseteq X$ be two overlapping sets. For convenience $\mathcal{S}_A$ denotes the set of all splitters of A. We note $X = \{X_1, \ldots, X_k\}$ if $\{X_1, \ldots, X_k\}$ is a partition of X. Obviously, $\mathcal{S}_{A\cap B} = \{\mathcal{S}_{A\cap B} \setminus B,\ \mathcal{S}_{A\cap B} \cap B\}$.

As $\mathcal{S}_A \cap A = \emptyset$, the partition $\mathcal{S}_{A\cup B} = \{\mathcal{S}_{A\cup B} \setminus \mathcal{S}_A,\ \mathcal{S}_{A\cup B} \cap \mathcal{S}_A\}$ can be reduced to $\mathcal{S}_{A\cup B} = \{\mathcal{S}_{A\cup B} \setminus \mathcal{S}_A,\ \mathcal{S}_A \setminus (A \cup B)\}$. Similarly, $\mathcal{S}_B = \{\mathcal{S}_B \setminus \mathcal{S}_{A\cap B},\ \mathcal{S}_{A\cap B} \setminus B\}$. Finally, $\mathcal{S}_A = \{\mathcal{S}_A \setminus B,\ (\mathcal{S}_A \cap B) \setminus \mathcal{S}_{A\cap B},\ (\mathcal{S}_A \cap B) \cap \mathcal{S}_{A\cap B}\}$ can be reduced to $\mathcal{S}_A = \{\mathcal{S}_A \setminus (A \cup B),\ (\mathcal{S}_A \cap B) \setminus \mathcal{S}_{A\cap B},\ \mathcal{S}_{A\cap B} \cap B\}$.. Hence,

$|\mathcal{S}_A| + |\mathcal{S}_B| - |\mathcal{S}_{A\cup B}| - |\mathcal{S}_{A\cap B}| = |(\mathcal{S}_A \cap B) \setminus \mathcal{S}_{A\cap B}| + |\mathcal{S}_B \setminus \mathcal{S}_{A\cap B}| - |\mathcal{S}_{A\cup B} \setminus \mathcal{S}_A|$.

To achieve proving the theorem, we prove that $\mathcal{S}_{A\cup B} \setminus \mathcal{S}_A \subseteq \mathcal{S}_B \setminus \mathcal{S}_{A\cap B}$. Indeed, let $z \in \mathcal{S}_{A\cup B} \setminus \mathcal{S}_A$. Then, $z \notin A \cup B$ and $H(z|xy)$ for all $x, y \in A$. Now, suppose that $z \notin \mathcal{S}_B$. Since z is not in $A \cup B$, we have $H(z|xy)$ for all $x, y \in B$. Furthermore, as A and B overlap and thanks to the transitivity of H, we have $z \notin A \cup B$ and $H(z|xy)$ for all $x, y \in A \cup B$, which is by definition $z \notin \mathcal{S}_{A\cup B}$. Contradiction. Finally, supposing $z \in \mathcal{S}_{A\cap B}$ would imply $z \in \mathcal{S}_A$. □

Submodular functions are combinatorial objects with powerful potential (see e.g. [10]). Theorem 1 enables the application of this theory to homogeneous relations. In [17], T. Uno and M. Yagiura gave a (restricted) version of this theorem, and constructed a very nice algorithm computing the common intervals of a set of permutations. It would be interesting to generalise this approach to any homogeneous relation, as done in [3] for modular decomposition of graphs.

4 Strong Modules and Primality

In a family $\mathcal{F}$ of subsets of X, a subset is *strong* if it overlaps no other subset of $\mathcal{F}$. The other subsets are *weak*. If $\mathcal{F}$ contain X and the singletons $\{x\}$ for every element $x \in X$, then X and $\{x\}_{x \in X}$ form the *trivial* strong subsets. The set inclusion orders the strong subsets into a tree. This is a quick proof that there are at most $2|X|-1$ strong subsets (and at most $|X|-2$ nontrivial ones), as the tree has no internal node of degree 1.

The *parent* of a (possibly weak) subset M is the smallest strong subset M_P properly containing M, and M is said to be a *child* of M_P. If M is strong, M_P is by definition its parent in the inclusion tree.

An *overlap class* is an equivalence class of the transitive closure of the overlap relation $\between$ on $\mathcal{F}$. The *support* of an overlap class $\mathcal{C} = \{C_1, \ldots, C_k\}$ is $C_1 \cup \cdots \cup C_k$. A is an *atom* of the overlap class if it is included in at least one subset C_i, and it does not overlap any subset of the class, and is maximal for these properties. All the atoms of a class form a partition of its support, the coarsest partition compatible with the class. An overlap class is *trivial* if it contains only one subset; it is then clearly a strong one.

A strong subset is *prime* if all its children are strong, and *decomposable* otherwise. It is a classical result of set theory that

Lemma 1. *If $\mathcal{F}$ is a family closed under union of overlapping sets, then there is an one-to-one correspondence between the nontrivial overlap classes of $\mathcal{F}$ and the decomposable strong subsets of $\mathcal{F}$. More precisely, the overlap class $\mathcal{C}$ associated with a decomposable subset D is simply the set of weak children of D, and the support of $\mathcal{C}$ is D.*

Of course we apply all these notions to the family of modules of a homogeneous relation.

Theorem 2. *Let H be a homogeneous relation and $\mathcal{Z}$ be the family of modules containing x but not y, and maximal for this property, for all x and y. The strong modules of H are exactly the supports and atoms of all overlap classes of $\mathcal{Z}$.*

Proof. First, remark that, thanks to the closure under union of overlapping sets, the supports and atoms of every overlap class of $\mathcal{Z}$ are strong modules. Lemma 1 tells they cannot be overlapped by an element of $\mathcal{Z}$ and if one, say A, is overlapped by a module $B \notin \mathcal{Z}$ then for $x \in A \setminus B$, the maximal module containing y but not x overlaps A, a contradiction. So the family of supports and atoms is included in the family of strong modules. Conversely, let us prove

that if M is a strong module then it is the support or an atom of some overlap class. We shall distinguish four cases.
Let M_P be the strong parent of M (for $M \neq X$). 1. M is trivial (X or $\{x\}$). There is no problem.
2. M is decomposable. It has k strong children $M_1, \ldots, M_k$. Let us pick an element x_i in each M_i. Then for all i and j we consider the maximal module containing x_i but not x_j. They form an overlap class of $\mathcal{Z}$. Its support is M, thanks to Lemma 1.
3. M is prime and M_P is prime. Then for all $x \in M$ and all $y \in M_P \setminus M$, M is the maximal module containing x but not y. As it is strong, it belongs to a trivial overlap class and is equals to its support.
4. M is prime and M_P is decomposable. Then for all $x \in M_P \setminus M$, M is included in some maximal module M_x not containing x (the one that contains the vertices of M). Let us consider the intersection I of all subsets of $\{M_x \mid x \in M_P \setminus M\}$. It is an atom of the overlap class associated with M_P and thus is strong. As M is a children of M_P, $I = M$. □

5 Partitive Families of Homogeneous Sets

A generalisation of modular decomposition, known from [5], less general than homogeneous relations but more powerful, is the *partitives families.* The *symmetric difference* of two sets A and B, denoted $A \Delta B$, is $(A \setminus B) \cup (B \setminus A)$.

Definition 6. *A family $\mathcal{F} \subseteq \mathcal{P}(X)$ is* weakly partitive *if it contains X and the singletons $\{x\}$ for all $x \in X$, and is closed under union, intersection and difference of overlapping subsets, i.e.*
$A \in \mathcal{F} \;\wedge\; B \in \mathcal{F} \;\wedge\; A \between B \;\Rightarrow\; A \cap B \in \mathcal{F} \;\wedge\; A \cup B \in \mathcal{F} \;\wedge\; A \setminus B \in \mathcal{F}$
Furthermore a weakly partitive family $\mathcal{F}$ is partitive *if it is also closed under symmetric difference: $A \in \mathcal{F} \;\wedge\; B \in \mathcal{F} \;\wedge\; A \between B \;\Rightarrow\; A \Delta B \in \mathcal{F}$*

As mentionned before, strong subsets of a weakly partitive family $\mathcal{F}$ can be ordered by inclusion into a tree. Let us define three types of strong subsets, i.e. three types of nodes of the tree:

- *prime* nodes which have no weak children,
- *degenerate* nodes: any union of strong children of the node belongs to $\mathcal{F}$,
- *linear* nodes: there is an ordering of the strong children such that a union of children belongs to $\mathcal{F}$ if and only if the children follow consecutively in this ordering.

Theorem 3. [5] *In a partitive family, there are only prime and degenerate nodes. In a weakly partitive family, there are only prime and degenerate and linear nodes.*

The strong subsets are therefore an $O(|X|)$ space coding of the family: it is enough to type the nodes into complete, linear or prime, and to order the children of the linear nodes. All weak subsets can be outputted by making simple combinations of the strong children of decomposable (complete or linear) nodes. Now, the following properties state that modules of some homogeneous relations are a proper generalisation of (weakly) partitive families.

Proposition 5. *Let H be a homogeneous relation. If H is* Graphical *or is* Quotiental*, then H is* Digraphical*.*

Proposition 6. *The modules of a* Quotiental *(resp.* Digraphical*) relation form a weakly partitive family. The modules of a* Graphical *relation form a partitive family.*

Proof. Let us suppose $A \in \mathfrak{F}_H$ and $B \in \mathfrak{F}_H$ and $A \between B$. Thanks to transitivity an element not in $A \cup B$ cannot distinguish $A \cup B$ (it would distinguish A or B). As an element not in A cannot distinguish A and an element not in B cannot distinguish B, then no element can distinguish $A \cap B$. For the same reason, only an element of $A \cap B$ can distinguish $A \setminus B$ or $A \Delta B$.

If $s \in A \cap B$ distinguishes $A \setminus B$, then this set contains x and y such that $\neg H(z|xy)$. As $B \setminus A$ is nonempty it contains t and we have $H(x|st)$ and $H(y|st)$ and $H(t|sx)$ and $H(t|sy)$ and $H(t|xy)$. Then H is neither *Quotiental* nor *Digraphical*.

Let us suppose H is *Graphical*. As it is also *Digraphical*, $A \setminus B$ and $B \setminus A$ are modules. If $z \in A \cap B$ distinguishes $A \Delta B$, then there exists $x \in A$ and $y \in B$ such that $\neg H(z|xy)$. Since $H(x|yz)$ and $H(y|xz)$, H is not *Graphical*, a contradiction. □

6 Modular Algorithmics for Homogeneous Relations

In the following, we consider a given ground set X and a homogeneous relation H on X, that are the input of all algorithms described here. The input H consists in $|X|$ partitions (the equivalence classes of H_x for each x) and thus can be stored in $O(|X|^2)$ space, instead of the naive $O(|X|^3)$ space representation storing all triples.

6.1 Smallest Module Containing a Subset

Let S be a nonempty subset of X. As $\mathfrak{F}_H$ is closed under intersection, there is a unique smallest module containing S, the intersection of all modules containing S, denoted henceforth $SM(S)$.

Theorem 4. *Algorithm 1 computes $SM(S)$ in $O(|X|.|SM(S)|) = O(|X|^2)$ time.*

Proof. Time complexity is obvious as the **while** loop runs $|M|-1$ times and the **for** loop $|X|$ times. The algorithm maintains the invariant that every splitter of M is in F. When M is replaced by $M \cup \{y\}$, every element that distinguishes $M \cup \{y\}$ distinguishes x from y, or already is in F. The algorithm ends therefore on a homogeneous set that contains S, and thus we have $SM(S) \subseteq M$. If $M \neq SM(S)$ let s be the first element of $M \setminus SM(S)$ added to F (eventually added to M). It distinguished two elements x and y from $SM(S)$, contradicting its homogeneity. So $SM(S) = M$. □

Algorithm 1. Smallest Module containing S

Let x be an element of S, $M := \{x\}$ and $F := S \setminus \{x\}$
while *F is not empty* **do**
  pick an element y in F ; $F := F \setminus \{y\}$; $M := M \cup \{y\}$
  for *every element z* **do**
    if $H(z|x, y)$ **then** $F := F \cup \{z\}$
  output M (now equals to $SM(S)$)

6.2 Maximal Modules Not Containing an Element

Proposition 7. *Let x be an element of X. As $\mathcal{F}_H$ is closed under union of intersecting subsets, there is a unique partition of $X \setminus \{x\}$ into $S_1, \ldots, S_k$ such that every S_i is a module of $\mathcal{F}_H$ and is maximal w.r.t. inclusion in $\mathcal{F}_H$.*

We call $MaxM(x) \subset \mathcal{P}(V)$ this partition of maximal modules not containing x. We propose a partition refining algorithm [16]. It is obvious that

Lemma 2. *Every module (especially the maximal ones) not containing x is included in a x-class H_x^i of H.*

Therefore our algorithm starts with the partition $P = \{H_x^1, \ldots, H_x^k\}$ of the x-classes of H, each part is an x-class. Then the partition is refined (parts are splitted) using the following rule. Let y be an element, called the *pivot*, and Y the part of P containing y.

Rule 1. *split every part A of P, except for Y, into $A \cap H_y^1$,. . . ,$A \cap H_y^l$*

Notice that a part if broken in smaller ones iff it is distinguished by y.

Lemma 3. *Starting from the partition $P_0 = \{H_x^1, \ldots, H_x^k\}$, the application of Rule 1 (for any pivot in any order) until no part can be actually splitted, produces $MaxM(x)$.*

Proof. The refining process ends when no pivot can split a part, i.e when every part is a module. Let us suppose one of these modules M is not maximal w.r.t. inclusion: it is included in a module M', itself included in a x-class H_x^i. Let us consider the pivot y that first broke M'. It cannot be out of M', as M' is module, nor within M', as a pivot does not break its own part. But M' was broken, contradiction. □

Let us now implement this lemma into an efficient algorithm. Let P_i be the partition after the ith application of Rule 1, y be a given vertex used as pivot, and Y_i the part of P_i containing y. We say that a part B of P_j descends from a part A of P_i if $i < j$ and $A \subset B$. Clearly, after y is chosen as pivot at step i, y does not distinguish any part of P_i excepted Y_i. If y is chosen as pivot after, at step $j > i$, y may only split the parts of P_{j-1} that descend from Y_i. Only these parts have to be examined for implementing Rule 1. But Y_j itself has not to be examined.

Let us suppose that, for a part A, we can split it in $O(|A|)$ time when applying Rule 1 with pivot y. Then the time spent at step j is $O(|Y_i| - |Y_j|)$, the sum of the size of the parts that descend from Y_i save Y_j. The time of all splittings with y as pivot is $O(|X|)$, leading to an $O(|X|^2)$ time complexity.

Let us suppose that the parts are implemented as a linked list [12], and the new parts created after splitting an old one replace it and follow consecutively in the list. Then for each pivot y two pointers, one on the first part that descend from Y_i and the second to the last part, are enough to tell the parts to be examinated. A simple sweep between the pointers, omitting Y_j, gives them.

Now let us see how a part A can be split in $O(|A|)$ time. It is a classical trick of partition refining [16,12]. If the y-classes are numbered from 1 to k, then A can be bucket sorted in $O(|A| + k)$ time, then each bucket gives a new part that descend from A. If $|A| < k$, we have to renumber the used y-classes from 1 to $k' \leq |A|$ before bucket sorting. A first sweep on A marks the used y-class numbers. A second sweep unmarks an used number the first time it is seen, and replace it by the new number (an incremented counter) which is less than $|A|$. The vector of y-classes numbers is initialized once in $O(k)$ time.

The last point is the order in which the pivot are taken. Using all elements as pivots, and repeating this $|X|$ time, i.e. $|X|^2$ applications of Rule 1, is enough. A clever choice is to use y only if Y_i has been split, keeping a queue of "active" pivots. We thus have:

Theorem 5. *$MaxM(x)$ can be computed in $O(|X|^2)$ time.*

6.3 Testing if a Homogeneity Relation Is Trivial

A homogeneous relation H on X is *trivial* if $\mathfrak{F}_H$ contains only X and the singletons.

Theorem 6. *Let S be a nonempty subset of X. One can test in $O(|X|^2)$ time if H is trivial.*

Proof. If $|X| < 2$ the answer is yes. Otherwise let x and y be two elements of X. In $O(|X|^2)$ time, the algorithm of Section 6.2 outputs the maximal modules not containing x. If one of them is nontrivial the answer is no. Otherwise all nontrivial modules contain x. In $O(|X|^2)$ time, the algorithm of Section 6.2 outputs the maximal modules not containing y. If one of them is nontrivial the answer is no. Otherwise all nontrivial modules contain x and y. Then Algorithm 1 is used with $S = \{x, y\}$, in $O(|X|^2)$ time. The answer is yes iff $SM(\{x, y\}) = X$. □

6.4 Strong Modules of a Homogeneous Relation

Theorem 2 straightforwardly leads to an algorithm:

Theorem 7. *The strong modules of a homogeneous relation H on X can be computed in $O(|X|^3)$ time.*

Proof. First compute $MaxM(x)$ for all $x \in X$. All these sets together exactly form the family $\mathcal{Z}$ defined in Theorem 2. It can be done in $O(|X|^3)$ time using the algorithm of Section 6.2 $|X|$ times. The size of this family (sum of the cardinals of every subsets) is $O(|X|^2)$ since they form $|X|$ partitions. Using Dahlhaus algorithm [7] the overlap components can be found in time linear on the size of the family, namely $O(|X|^2)$. According to Lemma 1 there are at most $|X|$ nontrivial overlap classes.

For each class it is easy to compute its support, and in $O(|X|^2)$ time easy to compute its atoms. For instance, consider the vector of parts of the overlap class containing a given element: the atoms are the elements with the same vector. Sorting the list of elements of the supports $O(|X|)$ times, one time per part, gives the elements with the same vector, thus the atoms.

And at least the $O(|X|^2)$ supports and atoms must be sorted by inclusion order into the inclusion tree of the strong modules. It can be done in $O(|X|^3)$ time using the same sorting technique. □

7 Outcomes

Let us examine in the sequel some of the applications of this homogeneity theory to modular decomposition of graphs and 2-structures, and to other graph relations. The name of *Graphical*, *Quotiental*, and *Digraphical* relations are justified by the following proposition:

Proposition 8. *The standart homogeneous relation of a directed graph is* Quotiental *and* Digraphical. *If the graph is undirected, its standard relation also is* Graphical.

The notion of modules also extends to 2-structures [9]. A (symmetric) 2-structure is a complete edge-coloured (undirected) graph and $H(x|yz)$ is true when edges (xy) and (xz) have the same colour. We still have:

Proposition 9. *The standart homogeneous relation of a 2-structure* Quotiental *and* Digraphical. *If the 2-structure is symmetric, its standard relation also is* Graphical.

The modules of an undirected graph and of a symmetric 2-structure thus form a partitive family, while the modules of a directed graph just form a weakly partitive family. All know properties of modular decomposition [15] can be derived from this result. An $O(n^3)$ modular decomposition algorithm can also be derived from Section 6.4 algorithm, but it is less efficient than the existing algorithms [4,6,8,12,14].

In a graph we can consider different homogeneous relations, for instance the relation *"there exists a path from vertex x to vertex y avoiding the vertex s"*, or a more general relation *"there exists a path from x to y avoiding the neighbourhood of s"*. It is easy to see that these two relations fulfill the basic axioms (symmetry, reflexivity and transitivity). In the first case, the strong modules form a partition (into the 2-vertex-connected components, minus the articulation points). The second relation is related to decomposition into star cutsets.

Another interesting relation is $D_k(s|xy)$ if $d(s,x) \leq k$ and $d(s,y) \leq k$, where $d(x,y)$ denotes the distance between x and y. The case $k = 1$ corresponds to modular decomposition. It is worth investigating the general case.

8 Conclusion

We hope that this homogeneity theory will have many other applications and will be useful to decompose automata [1] and boolean functions [2]. Obviously, the algorithmic framework presented here can be optimised in each particular application, as it can be done for modular decomposition [4,6,8,12,14]. We think the homogeneity concept is a very general idea.

References

1. C. Allauzen and M. Mohri. Efficient algorithms for testing the twins property. *Journal of Automata, Languages and Combinatorics*, 8(2):117–144, 2003.
2. J. Bioch. The complexity of modular decomposition of boolean functions. *Discrete Applied Mathematics*, 149(1-3):1–13, 2005.
3. B.-M. Bui Xuan, M. Habib, and C. Paul. Revisiting T. Uno and M. Yagiura's Algorithm. In *16th International Symposium of Algorithms and Computation (ISAAC05)*, volume 3827 of *LNCS*, pages 146–155, 2006.
4. C. Capelle, M. Habib, and F. de Montgolfier. Graph decomposition and factorizing permutations. *D.M.T.C.S*, 5(1):55–70, 2002.
5. M. Chein, M. Habib, and M.C. Maurer. Partitive hypergraphs. *Discrete Mathematics*, 37(1):35–50, 1981.
6. A. Cournier and M. Habib. A new linear algorithm for modular decomposition. In *Trees in algebra and programming (CAAP 94)*, volume 787 of *LNCS*, 1994.
7. E. Dahlhaus. Parallel algorithms for hierarchical clustering, and applications to split decomposition and parity graph recognition. *Journal of Algorithms*, 36(2):205–240, 2000.
8. E. Dahlhaus, J. Gustedt, and R.M. McConnell. Efficient and practical algorithms for sequential modular decomposition. *Journal of Algorithms*, 41(2):360–387, 2001.
9. A. Ehrenfeucht and G. Rozenberg. Theory of 2-structures. *Theoretical Computer Science*, 3(70):277–342, 1990.
10. S. Fujishige. *Submodular Functions and Optimization.* North-Holland, 1991.
11. Tibor Gallai. Transitiv orientierbare Graphen. *Acta Math. Acad. Sci. Hungar.*, 18:25–66, 1967.
12. M. Habib, C. Paul, and L. Viennot. Partition refinement techniques: An interesting algorithmic tool kit. *I.J.F.C.S*, 10(2):147–170, 1999.
13. R.M. McConnell and F. de Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):189–209, 2005.
14. R.M. McConnell and J.P. Spinrad. Modular decomposition and transitive orientation. *Discrete Maths.*, 201:189–241, 1999. Extended abstract at SODA 94.

15. R.H. Möhring and F.J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. *Annals of Discrete Mathematics*, 19:257–356, 1984.
16. Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.
17. T. Uno and M. Yagiura. Fast algorithms to enumerate all common intervals of two permutations. *Algorithmica*, 26(2):290–309, 2000.