

# Homogeneity vs. Adjacency: generalising some graph decomposition algorithms

B.-M. Bui Xuan<sup>1</sup>, M. Habib<sup>2</sup>, V. Limouzy<sup>2</sup>, and F. de Montgolfier<sup>2</sup>

<sup>1</sup> LIRMM, Université Montpellier 2, France. buixuan@lirmm.fr

<sup>2</sup> 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 or decomposition of 2-structures, but also star cutsets and bimodular decomposition. Our general framework provides a decomposition algorithm which improves the best known algorithms for bimodular decomposition.

## 1 Introduction

Several combinatorial algorithms are based on partition refinement techniques [20]. 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 [14,18,8,16,6,4,13,9] use this technique.

In bioinformatics also, the distinction of a set by a element, called a splitter, seems to play an important role, as for example in the nice algorithm of [21], which computes the set of common intervals of two permutations.

In this paper we investigate an abstract notion of splitters and propose a dual 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, within this general theory, most of the nice algorithmic tools developed for modular decomposition [14,18,8,16,6,4,13,9] are still efficient.

This theory not only unifies the usual modular decomposition generalisations such as modular decomposition of directed graphs [17] or decomposition of 2-structures [10], but also allows to handle star cutsets, and the bimodular decomposition [11]. Notice that our general framework provides a decomposition algorithm which improves the best known algorithm for bimodular decomposition.

The paper is structured as follows: first we detail this new combinatorial decomposition theory, then we present a general algorithmic framework, and we finish by listing some interesting applications.

## 2 Homogeneity, a new viewpoint

Throughout this section  $V$  is a finite set. The family of all subsets of  $V$  is denoted  $\mathcal{P}(V)$ . An *reflectless* triple is  $(x, y, z) \subseteq V^3$  with  $x \neq y$  and  $x \neq z$ . Reflectless triples will be denoted  $(x|yz)$  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  $V$ . Given  $s \in V$ , the relation  $H_s$  is a binary relation on  $V$  defined as  $H_s(x, y)$  if and only if  $H(s|xy)$ .

**Definition 1 (Homogeneous relation).**  $H$  is a homogeneous relation on  $V$  if, for all  $s \in V$ ,  $H_s$  is an equivalence relation on  $V \setminus \{s\}$ :

- (Symmetry):  $\forall s, x, y \in V, H(s|xy) \Leftrightarrow H(s|yx)$ .
- (Reflexivity):  $\forall s, x \in V, s \neq x \Rightarrow H(s|xx)$ .
- (Transitivity):  $\forall s, x, y, z \in V, H(s|xy) \wedge H(s|yz) \Rightarrow H(s|xz)$ .

**Definition 2 (Homogeneous sets).** Let  $H$  be a homogeneous relation.  $X \subseteq V$  is homogeneous with respect to an element  $s \notin X$  if  $H(s|xy)$  for all  $x, y \in X$ . If  $X$  is not homogeneous w.r.t.  $s$  then  $s$  distinguishes  $X$ , or is a splitter of  $X$ . Let  $\mathcal{S}_X$  be the set of all elements distinguishing  $X$  and  $s(X) = |\mathcal{S}_X|$ .

$M \subseteq V$  is a homogeneous set if  $M \neq \emptyset$  and for all  $x$  not in  $M$ ,  $M$  is homogeneous w.r.t.  $x$ . In other words  $s(M) = 0$ . The family of homogeneous sets for a homogeneous relation  $H$  on  $V$  is denoted  $\mathcal{F}_H$  or  $\mathcal{F}$  if not ambiguous.

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

Two sets  $A$  and  $B$  overlap if  $A \cap B$ ,  $A \setminus B$  and  $B \setminus A$  are all nonempty. It is denoted  $A \otimes B$ . The symmetric difference of two sets  $A$  and  $B$ , denoted  $A \Delta B$ , is  $(A \setminus B) \cup (B \setminus A)$ . Let us now enumerate some properties of the homogeneous relations and sets.

**Proposition 1.** If  $s$  distinguishes  $X$  then  $\neg H(s|xy)$  for some  $x, y \in X$ .

Thanks to this proposition, it is exactly equivalent to define the homogeneity relations as ternary relations on reflectless triples – as presented here – or as relations between elements and subsets of  $V$ . A homogeneous relation  $\tilde{H}(s|X)$  for  $X \subseteq V$  and  $s \in V \setminus X$  simply is a transitive relation: if  $A \otimes B$  and  $\tilde{H}(s|A)$  and  $\tilde{H}(s|B)$  then  $\tilde{H}(s|A \cup B)$ . The relation  $\tilde{H}$  has the same properties than  $H$  and the homogeneous sets are the same. We find the ternary relation much simpler.

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

This property is called *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  $V$  and  $\mathcal{F}'_H = \mathcal{F}_H \cup \{\emptyset\}$ .  $(\mathcal{F}'_H, \subseteq)$  is a lattice.

*Proof.* Since  $\emptyset \in \mathcal{F}'_H$ , and thanks to Proposition 2, the intersection of any two elements of  $\mathcal{F}'_H$  belong to  $\mathcal{F}'_H$ . It is the infimum of two sets, since any member of  $\mathcal{F}'_H$  contained in both  $A$  and  $B$  is contained in  $A \cap B$ . Let us consider the family  $\mathcal{G}$  of all set of  $\mathcal{F}'_H$  containing both  $A$  and  $B$ . It is nonempty ( $V$  is a member). Since  $\mathcal{F}'_H$  is closed under intersection,  $\mathcal{G}$  admits a unique smallest (w.r.t. inclusion) element, the intersection of all its members, that is the supremum of  $A$  and  $B$ .  $\square$

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

**Definition 3.** A homogeneous relation  $H$  fulfills

- **A1** if  $\forall x, y, z \in V, H(x|yz) \wedge H(y|xz) \Rightarrow H(z|xy)$ .
- **A2** if  $\forall s, t, x, y \in V, H(x|st) \wedge H(y|st) \wedge H(t|xy) \Rightarrow H(s|xy)$ .
- **A3** if  $\forall s, t, x, y \in V, H(x|st) \wedge H(y|st) \wedge H(t|sx) \wedge H(t|sy) \Rightarrow H(s|xy)$ .
- **A4** if  $\forall x, y, z \in V, \neg H(x|yz) \wedge \neg H(y|xz) \Rightarrow H(z|xy)$ .

**Proposition 4 (Quotient).** *Let  $H$  be a homogeneous relation. Then,  $H$  satisfies **A2** if and only if for all homogeneous set  $M$ , for all  $x, y \in M$  and  $s, t \in V \setminus M$ ,  $H(x|st) \Leftrightarrow H(y|st)$ .*

It is a simple rewriting of **A2** but it enlightens that, from a homogeneous set, one can pick a *representative element*. Indeed, elements in a homogeneous set  $M$  uniformly perceive a set  $X$  not intersecting  $M$ : if one element of  $M$  distinguishes  $X$  then so do all. This allows to shrink a homogeneous set  $M$  into a single element, the quotient by  $M$ .

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

**Proposition 5 (Restriction).** *Let  $H$  be a homogeneous relation,  $M$  a homogeneous set and  $M' \subseteq M$ .  $M' \in \mathcal{F}_{H[M]} \Leftrightarrow M' \in \mathcal{F}_H$ .*

Recursivity can therefore be used when dealing with homogeneous sets. Notice that the proposition is not always true if  $M$  is not a homogeneous set. The Quotient and Restriction properties were used first with modular decomposition and are useful for algorithmic [19].

### 3 Submodularity of homogeneous relations

**Definition 4.** *A set function  $\mu : \mathcal{P}(V) \rightarrow \mathbb{R}$  is submodular if and only if for all  $X, Y \subseteq V$   $\mu(X) + \mu(Y) \geq \mu(X \cup Y) + \mu(X \cap Y)$  (see e.g. [12]).*

**Theorem 1 (Submodularity).** *Let  $H$  be a homogeneous relation. The function  $s$  counting the number of splitters (defined in Definition 2) is submodular.*

*Proof.* Let us for convenience define  $s(\emptyset) = -|V|$ . It suffices to prove  $s(X) + s(Y) \geq s(X \cup Y) + s(X \cap Y)$  for all overlapping  $X, Y \subseteq V$ . So let  $X, Y \subseteq V$  be two overlapping sets. Obviously,  $\mathcal{S}_{X \cap Y} = (\mathcal{S}_{X \cap Y} \setminus Y, \mathcal{S}_{X \cap Y} \cap Y)$ .

As an element distinguishing  $X$  does not belong to  $X$ , the partition  $\mathcal{S}_{X \cup Y} = (\mathcal{S}_{X \cup Y} \setminus \mathcal{S}_X, \mathcal{S}_{X \cup Y} \cap \mathcal{S}_X)$  can be reduced to  $\mathcal{S}_{X \cup Y} = (\mathcal{S}_{X \cup Y} \setminus \mathcal{S}_X, \mathcal{S}_X \setminus (X \cup Y))$ . Similarly,  $\mathcal{S}_Y = (\mathcal{S}_Y \setminus \mathcal{S}_{X \cap Y}, \mathcal{S}_{X \cap Y} \setminus Y)$ . Finally,  $\mathcal{S}_X = (\mathcal{S}_X \setminus Y, (\mathcal{S}_X \cap Y) \setminus \mathcal{S}_{X \cap Y}, (\mathcal{S}_X \cap Y) \cap \mathcal{S}_{X \cap Y})$  can be reduced to  $\mathcal{S}_X = (\mathcal{S}_X \setminus (X \cup Y), (\mathcal{S}_X \cap Y) \setminus \mathcal{S}_{X \cap Y}, \mathcal{S}_{X \cap Y} \cap Y)$ . Hence,  $|\mathcal{S}_X| + |\mathcal{S}_Y| - |\mathcal{S}_{X \cup Y}| - |\mathcal{S}_{X \cap Y}| = |(\mathcal{S}_X \cap Y) \setminus \mathcal{S}_{X \cap Y}| + |\mathcal{S}_Y \setminus \mathcal{S}_{X \cap Y}| - |\mathcal{S}_{X \cup Y} \setminus \mathcal{S}_X|$ .

To achieve proving the theorem, we prove that  $\mathcal{S}_{X \cup Y} \setminus \mathcal{S}_X \subseteq \mathcal{S}_Y \setminus \mathcal{S}_{X \cap Y}$ . Indeed, let  $z \in \mathcal{S}_{X \cup Y} \setminus \mathcal{S}_X$ . Then,  $z \notin X \cup Y$  and for all  $x, y \in X$ , we have  $H(z|xy)$ .

Now, suppose that  $z \notin \mathcal{S}_Y$ . Since  $z$  is not in  $X \cup Y$ , we have  $H(z|xy)$  for all  $x, y \in Y$ . Furthermore, as  $X$  and  $Y$  overlap and thanks to the transitivity of  $H$ , we have  $z \notin \mathcal{S}_{X \cup Y}$ , which is a contradiction.

Finally, supposing  $z \in \mathcal{S}_{X \cap Y}$  would imply  $z \in \mathcal{S}_X$ .  $\square$

Submodular functions are combinatorial objects with powerful potential (see e.g. [12]). Theorem 1 enables the application of this theory to homogeneous relations. In [21], 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.

## 4 Strong homogeneous sets and Primality

In a family  $\mathcal{F}$  of subsets of  $V$ , a subset is *strong* if it overlaps no other subset of  $\mathcal{F}$ . The other subsets are *weak*. Let us suppose the family  $\mathcal{F}$  contains  $V$  and the singletons  $\{v\}$  for every element. Then  $V$  and  $\{v\}_{v \in V}$  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  $2|V| - 1$  strong subsets (and at most  $|V| - 2$  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  $\odot$  on  $\mathcal{F}$ . The *support* of an overlap class  $\mathcal{C} = \{C_1, \dots, C_k\}$  is  $C_1 \cup \dots \cup C_k$ .  $A$  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  $D$ .*

The overlap class associated with a decomposable node is simply the set of weak children of this node. Of course we apply all these notions on homogeneous set families. On partitive families, the strong homogeneous sets plays a very important role since they are exactly a coding, in  $O(|V|)$  space, of the possibly  $2^{|V|}$  subsets of the family (see the upcoming section).

**Theorem 2.** *Let  $H$  be a homogeneous relation and  $\mathcal{Z}$  be the family of homogeneous sets containing  $x$  but not  $y$ , and maximal for this property, for all  $x$  and  $y$ . The strong homogeneous sets 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 homogeneous sets. Lemma 1 tells they can not be overlapped by an element of  $\mathcal{Z}$  and if one,  $A$  is overlapped by an homogeneous set  $B \notin \mathcal{Z}$  then for  $x \in A \setminus B$ , the maximal homogeneous set containing  $y$  but not  $x$  overlaps  $A$ , a contradiction. So the family of supports and atoms is included in the family of strong homogeneous sets. Conversely, let us prove that if  $M$  is a strong homogeneous set 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  $M \neq V$ ).

1.  $M$  is trivial ( $V$  or  $\{v\}$ ). There is no problem.
2.  $M$  is decomposable. It has  $k$  strong children  $M_1..M_k$ . Let us pick an element  $x_i$  in each  $M_i$ . Then for all  $i$  and  $j$  we consider the maximal homogeneous set 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 homogeneous set 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  $x \in M_P \setminus M$  then  $M$  is included in some maximal homogeneous set  $M_x$  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  $M_P$ ,  $I = M$ .  $\square$

This theorem leads to a  $O(|V|^3)$ -time algorithm in Section 6.4.

## 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*.

**Definition 5.** A family  $\mathcal{F} \subseteq \mathcal{P}(V)$  is weakly partitive if it contains  $V$  and the singletons  $\{v\}$  for all  $v \in V$ , and is closed under union, intersection and difference of overlapping subsets, i. e.

$$A \in \mathcal{F}_H \wedge B \in \mathcal{F}_H \wedge A \odot B \Rightarrow A \cap B \in \mathcal{F}_H \wedge A \cup B \in \mathcal{F}_H \wedge A \setminus B \in \mathcal{F}_H$$

Furthermore a weakly partitive family  $\mathcal{F}$  is partitive if it is also closed under symmetric difference:

$$A \in \mathcal{F}_H \wedge B \in \mathcal{F}_H \wedge A \odot B \Rightarrow A \Delta B \in \mathcal{F}_H$$

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

- *prime* nodes who have no weak children,
- *degenerate* nodes such that all union of strong children of the node belongs to  $\mathcal{F}$ ,
- *linear* nodes such that there is an ordering of the strong children such that a union of children belongs to  $\mathcal{F}$  if and only if they are consecutive in this ordering.

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

The strong subsets are therefore an  $O(|V|)$  space encoding 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 output just by making simple combinations of the strong children of decomposable (complete or linear) nodes. Now, the following properties state that the homogeneous relations are a proper generalisation of (weakly) partitives families.

**Proposition 6.** Let  $H$  be a homogeneous relation. If  $H$  fulfills **A1**, or if  $H$  fulfills **A2**, then  $H$  fulfills **A3**.

*Remark 2.* This proposition allows to classify the homogeneous relations. There exists homogeneous relations fulfilling **Ai** but not **Aj** for all  $i$  and  $j$  excepted the two implications of Proposition 6.

**Proposition 7.** If a homogeneous relation  $H$  fulfills **A2** or **A3**, then  $\mathcal{F}_H$  is a weakly partitive family.

If  $H$  fulfills **A1**, then  $\mathcal{F}_H$  is a partitive family.

---

**Algorithm 1:** Smallest Homogeneous set 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  $SHS(S)$ )
```

---

*Proof.* Let us suppose  $A \in \mathcal{F}_H$  and  $B \in \mathcal{F}_H$  and  $A \odot B$ . Thanks to transitivity an element not in  $A \cup B$  can not distinguish  $A \cup B$  (it would distinguish  $A$  or  $B$ ). As an element not in  $A$  can not distinguish  $A$  and an element not in  $B$  can not 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  $\neg H(z|xy)$ . But 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 both **A2** and **A3** are violated.

Let us suppose **A1** holds. As **A3** also holds  $A \setminus B$  and  $B \setminus A$  are homogeneous sets. If  $z \in A \cap B$  distinguishes  $A \Delta B$ , then there exists  $x \in A$  and  $y \in B$  such that  $\neg H(z|xy)$ . But since  $H(x|yz)$  and  $H(y|xz)$  **A1** is contradicted.  $\square$

## 6 Homogeneous Set Algorithms

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

### 6.1 Smallest Homogeneous Set containing a subset

Let  $H$  be a homogeneous relation on  $V$  and  $S$  be a nonempty subset of  $V$ . As  $\mathcal{F}_H$  is closed under intersection, there is an unique smallest homogeneous set containing  $S$ , the intersection of all homogeneous set containing  $S$ , denoted henceforth  $SHS(S)$ .

**Theorem 4.** *Algorithm 1 computes  $SHS(S)$  in  $O(|V| \cdot |SHS(S)|) = O(|V|^2)$  time.*

*Proof.* Time complexity is obvious as the **while** loop runs  $|M| - 1$  times and the **for** loop  $|V|$  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  $SHS(S) \subseteq M$ . If  $M \neq SHS(S)$  let  $v$  be the first element of  $M \setminus SHS(S)$  added to  $F$  (eventually added to  $M$ ). It distinguished two elements  $x$  and  $y$  from  $SHS(S)$ , contradicting its homogeneity. So  $SHS(S) = M$ .  $\square$

### 6.2 Maximal Homogeneous Sets not containing an element

**Proposition 8.** *Let  $H$  be a homogeneous relation on  $V$  and  $x \in V$  an element. As  $\mathcal{F}_H$  is closed under union of intersecting subsets, there is an unique partition of  $V \setminus \{x\}$  into  $S_1 \dots S_k$  such that every  $S_i$  is a homogeneous set of  $\mathcal{F}_H$  and is maximal w.r.t. inclusion in  $\mathcal{F}_H$ .*

We call  $MHS(x) \subset \mathcal{P}(V)$  this partition of Maximal Homogeneous Sets not containing  $x$ . We propose a partition refining algorithm [20]. It is obvious that

**Lemma 2.** *Every homogeneous set (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  $P = \{H_x^1..H_x^k\}$  of the  $x$ -classes of  $H$ . Then the partition is refined (classes are splitted) using the following rule. Let  $y$  be an element, called the *pivot*, and  $C(y)$  the class of Partition  $P$  containing  $y$ .

**Rule.** *For a given pivot  $y$ , split every class of  $P$ , excepted  $C(y)$ , into  $C \cap H_y^1, \dots, C \cap H_y^l$*

Notice that a class is actually splitted in many new classes iff it is distinguished by  $y$ .

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

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

Let us now implement this lemma into an efficient algorithm. The idea for saving time is that, after a class is splitted by  $y$ , it has not to be examined one more time but, if the former class  $C$  containing  $y$  is later splitted into new classes  $C_1..C_a$ . W.l.o.g, suppose  $y \in C_1$ . Then only the new classes  $C_2..C_a$  must be examined when  $y$  is the pivot again. Every element is thus examined once for every pivot, leading to an  $O(|V|^2)$  time complexity. This is implemented in Algorithm 2.

We implement this idea using *groups*. The partition is refined from  $P_0$  into  $P_1$  then  $P_2$  and so on. The *group* of an element of  $P_i$  is its class in  $P_{i-1}$ . If the classes are implemented using a linked list, and if the classes are only splitted into new classes that follow consecutively in the linked list, then the group boundaries are simply markers in the linked list.

A classical trick of partition refining [20,14] is using a *refining set*  $R \subseteq V$ . Every class  $C$  of  $P$  can be splitted into  $C \cap R$  and  $C \setminus R$  in  $O(|R|)$  time only: every element of  $R$  is moved from its old class  $C_i$  to its new class  $C'_i$ , the successor of  $C_i$  in the linked list. A flag in the data structure of  $C_i$  indicates whether its successor is  $C'_i$  or not. This allows to create  $C'_i$  if it does not already exist.  $C'_i$  is  $C_i \cap R$  while the remaining elements of  $C_i$  are  $C_i \setminus R$ . All flags are reset by a second scan of  $R$ , that also allows to remove empty classes from the partition linked list (classes that were included in  $R$ ).

And at least, for a set  $Z$  and  $y \notin Z$ ,  $Z$  can be partitioned according to the  $y$ -classes in  $O(|Z|)$  time. If there are  $k$   $y$ -classes, an array of  $k$  linked list is used and each element of  $Z$  is appended to the proper list. A stack of nonempty list allow to collect and reset them in  $O(|Z|)$  time.

**Theorem 5.** *Algorithm 2 computes  $MHS(x)$  in  $O(|V|^2)$  time.*

*Proof.* For the correctness proof, one just has to check that the algorithm implements correctly Lemma 3. For the time complexity issues, notice that, for each pivot  $y$ , an element  $z$  is placed in  $Z$  only once. As partitioning  $Z$  into  $y$ -classes, and then refining using all refining sets generated by  $y$ , takes  $O(|Z|)$  time. Hence the algorithm takes  $O(|V|^2)$  time.  $\square$

---

**Algorithm 2:** Maximal Homogeneous Sets not containing  $x$ 

---

```
for every group  $G$  do
  for every class  $C$  of  $G$  do
    Compute the set  $Z$  of elements of in  $G$  but not in  $C$ 
    for every element  $y$  of  $C$  do
      Partition  $Z$  according to the  $y$ -classes
      Add each partition set to the refining sets pool
Set the group boundaries to the classes boundaries (from  $P_{i-1}$  to  $P_i$ )
for each refining set  $R$  of the pool do
  Remove  $R$  from the pool and then refine  $P_i$  using  $R$ 
```

---

### 6.3 Testing if a homogeneity relation is trivial

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

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

*Proof.* If  $|V| < 2$  the answer is yes. Otherwise let  $x$  and  $y$  be two elements of  $V$ . In  $O(|V|^2)$  time, Algorithm 2 outputs the maximal homogeneous sets not containing  $x$ . If one of them is nontrivial the answer is no. Otherwise all nontrivial homogeneous sets contain  $x$ . In  $O(|V|^2)$  time, Algorithm 2 outputs the maximal homogeneous sets not containing  $y$ . If one of them is nontrivial the answer is no. Otherwise all nontrivial homogeneous sets contain  $x$  and  $y$ . Then Algorithm 1 is used with  $S = \{x, y\}$ , in  $O(|V|^2)$  time. The answer is yes iff  $SHS(\{x, y\}) = V$ .  $\square$

### 6.4 Strong modules of a homogeneous relation

Theorem 2 straightforwardly leads to an algorithm:

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

*Proof.* First compute  $MHS(x)$  for all  $x \in V$ . All these sets together exactly form the family  $\mathcal{Z}$  defined in Theorem 2. It can be done in  $O(|V|^3)$  time using Algorithm 2  $|V|$  times. The size of this family (sum of the cardinals of every subsets) is  $O(|V|^2)$  since they form  $|V|$  partitions. Using Dahlhaus algorithm [7] the overlap components can be found in time linear in the size of the family, thus  $O(|V|^2)$ . According to Lemma 1 there are at most  $|V|$  nontrivial overlap classes. For each class it is easy to compute its support, and in  $O(|V|^2)$  time easy to compute its atoms (each subset of the class is used as pivot, in a partition refinement of the support). And after all the  $O(|V|^2)$  supports and atoms output must be sorted by inclusion order into the inclusion tree of the strong homogeneous sets (removing many duplicates), an easy task in  $O(|V|^3)$  time.  $\square$

Notice that, if the homogeneous relation defines a weakly partitive family, then the quotient property applies and helps a lot. The algorithm scheme of [8], that can be implemented in  $O(n+m)$  for graphs, could be implemented in  $O(|V|^2)$  time for an homogeneous relation. The approach is to compute  $MHS(x)$  then, using the quotient relation, to compute all strong homogeneous sets containing  $x$ . That gives the left branch of the decomposition tree. Then the algorithm is recursively launched. The amortised complexity analysis of Section 6.2 can be used: as a class is not splitted when the recursive process begins, the whole algorithm takes  $O(|V|^2)$ . But our proof needs Axiom **A4**. It holds for graphs, but not for directed graphs nor 2-structures, and seems very specific.

## 7 Applications

Let us examine in the sequel some of the applications of this Homogeneity theory.

### 7.1 Modular decomposition

In a graph, the homogeneous relation  $H(x|yz)$  is true when  $x$  “sees”  $y$  and  $y$  in the same way. In undirected graphs, this means that either there are two edges  $(xy)$  and  $(xz)$ , or no edge between  $x$  and the two other vertices. In directed graphs, this means that there are zero or two incoming arcs between  $x$  and the two other vertices, and zero or two out-coming arcs between  $x$  and the two other vertices. The homogeneous sets are then called *modules*. The notion of modules also extends to 2-structures [10]. A 2-structure is a complete edge-coloured graph and  $H(x|yz)$  is true when edges  $(xy)$  and  $(xz)$  have the same colour.

**Proposition 9.** – *The homogeneous relation of a undirected graph fulfills **A1**, **A2**, **A3** and **A4***  
– *The homogeneous relation of a directed graph fulfills **A2** and **A3***  
– *The homogeneous relation of a 2-structure fulfills **A1** and **A2** and **A3** and **A4***

The modules of a undirected graph and of a 2-structure thus form a partitive family, while the modules of a directed graph just form a weakly partitive family. All known properties of modular decomposition [19] 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 [14,18,8,16,6,4,13,9].

### 7.2 Other graph relations

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 homogeneous sets 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 problem.

### 7.3 Bimodular decomposition

Let  $G = (B, W, E)$  be a bipartite graph where  $B$  contains the black vertices and  $W$  contains the white vertices. A *bimodule* is a subset of vertices  $M \subset (B \cup W)$  such that no black vertex  $b$  not in  $M$  distinguishes two white vertices of  $M$  (there must be either no or all possible edges between  $b$  and the white vertices of  $M$ ) and conversely no white vertex not in  $M$  distinguishes two black vertices of  $M$ . In [11] is defined the *bimodular decomposition* of a bipartite graph. It is stated that, although the family of bimodules is not even partitive, then the strong bimodules are an optimal encoding of the family. Indeed, the inclusion tree of strong bimodules, plus some  $O(n^2)$  pointers (easy to add, given the graph and the inclusion tree), are enough to store and output the (potentially exponential) family of bimodules, and to test if a set is a bimodule, and allow to

solve in polynomial time some NP-complete problems, when the degree of the nodes of the tree is bounded [15]. [11] give an  $O(mn^3)$  time algorithm that compute the strong bimodules given the graph. But given the graph the homogeneous relation can be computed in  $O(n^3)$  time by testing all reflectless triples, and then using the algorithm of Section 6.4 the strong bimodules can be output in  $O(n^3)$  time, improving the previous time bound.

## 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 [14,18,8,16,6,4,13,9]. 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. *Discrete Mathematics and Theoretical Computer Science*, 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*, pages 68–84, 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. F. de Montgolfier. *Décomposition modulaire des graphes. Théorie, extensions et algorithmes*. PhD thesis, Université Montpellier II, 2003.
10. A. Ehrenfeucht and G. Rozenberg. Theory of 2-structures. *Theoretical Computer Science*, 3(70):277–342, 1990.
11. J.-L. Fouquet, M. Habib, F. de Montgolfier, and J.-M. Vanherpe. Bimodular decomposition of bipartite graphs. In *WG’04, 30th International Workshop on Graph-Theoretic Concepts in Computer Science*, 2004.
12. S. Fujishige. *Submodular Functions and Optimization*. North-Holland, 1991.
13. M. Habib, F. de Montgolfier, and C. Paul. A simple linear-time modular decomposition algorithm. In *9th Scandinavian Workshop on Algorithm Theory (SWAT04)*, volume 3111 of *LNCS*, pages 187–198, 2004.
14. M. Habib, C. Paul, and L. Viennot. Partition refinement techniques: An interesting algorithmic tool kit. *International Journal of Foundations on Computer Science*, 10(2):147–170, 1999.
15. V. Lozin. On maximum induced matchings in bipartite graphs. *Information Processing Letters*, 81:7–11, 2002.
16. R. McConnell and J. Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA)*, pages 536–545, New York, 1994. ACM.
17. R.M. McConnell and F. de Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):189–209, 2005.
18. R.M. McConnell and J.P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201:189–241, 1999.
19. 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.
20. Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.
21. T. Uno and M. Yagiura. Fast algorithms to enumerate all common intervals of two permutations. *Algorithmica*, 26(2):290–309, 2000.