A new tractable combinatorial decomposition

Binh-Minh Bui-Xuan [a,*] Michel Habib [b] Vincent Limouzy [b]
Fabien de Montgolfier [b]

[a] *CNRS - LIRMM - Univ. Montpellier II, 161 rue Ada, 34392 Montpellier Cedex 5, France.*
`buixuan@lirmm.fr`

[b] *CNRS - LIAFA - Univ. Paris Diderot - INRIA Team-Project GANG, Case 7014, 75205 Paris Cedex 13 , France.*
{`habib,limouzy,fm`}`@liafa.jussieu.fr`

Abstract

We introduces the *umodules*, a generalisation of the notion of a graph module. The theory we develop captures among other undirected graphs, tournaments, digraphs, and 2−structures. We show that, under some axioms, a unique decomposition tree exists for umodules. Polynomial-time algorithms are provided for: non-trivial umodule test, maximal umodules enumeration, and decomposition tree computation when the tree exists. Our results unify many known decompositions, such as modular decomposition and bi-join decomposition of graphs, and a new decomposition of tournaments.

Key words: modular decomposition, graph theory, 2−structure, homogeneous relation

1 Introduction

In graph theory, modular decomposition is now a well-studied notion [24,10,33,17,16], as well as some of its generalisations [15,29,35]. As having been rediscovered in other fields, the notion also appears under various names, including intervals, externally related sets, autonomous sets, partitive sets, and clans. Direct

* Corresponding author. Tel: +33 4 67 41 86 05. Fax: +33 4 67 41 85 00.
1 Research supported by the ANR project *Graph Decompositions and Algorithms*

applications of modular decomposition include tractable constraint satisfaction problems [12], computational biology [23], graph clustering for network analysis, and graph drawing.

Besides, in the area of social networks, several vertex partitioning have been introduced in order to catch the idea of putting in the same part vertices acknowledging similar behaviour, in other words finding regularities [43]. Modular decomposition provides such a partitioning, yet seemingly too restrictive for real life applications. The concept of a role [19] on the other hand seems promising, however its computation is unfortunately $NP-$hard [20]. As a natural consequence, there is need for the search of *relaxed*, but *tractable*, variations of the modular decomposition scheme. A step following this direction has generalised graph modules to those of larger combinatorial structures, so-called homogeneous relations [8,9]. This paper follows the same research stream, and weakens the definition of module in order to further decompose. Fortunately we obtain a new tractable variation of modular decomposition, which follows.

Modular decomposition is based on *modules*, vertex subsets with no *splitter*. In graphs, a splitter of a vertex subset is linked with some, but not all, vertices of this subset. Hence, all vertices of the module have the same neighbourhood. Thus, one can say that sorting the "outside" of a module by their behaviour with respect to any vertex of the module results in the same ordered partition. We shall see how this can be extended to homogeneous relations. Then, we shall address unordered-modules, so-called *umodules* for short: for any vertex v of a umodule, sorting the outside of the umodule by their behaviour with respect to vertex v has to result in a same *unordered* partition. For graphs, the umodules are exactly the *bijoins* (see Fig. 1(a) and Section 6). As there are clearly more umodules than modules, this allows deeper decomposition. We shall see that this decomposition is tractable.

After comparing umodule to previous notions in the topic, we display its tractability by giving an $O(|X|^5)$ time enumeration of the maximal umodules of a given homogeneous relation over a finite set X, and show how this can also be used as a non-trivial umodule existence test. The structure of the family of umodules is then investigated under different scenarios. We focus on a particular case, and provide a potent tractability theorem which makes use of the so-called *Seidel-switching* graph operation [40]. Fortunately enough, undirected graphs and tournaments fit into the latter formalism. We then deepen the study and address total decomposability issues, namely when any "large enough" sub-structure is decomposable. Surprisingly enough, this shows how our theory provides a very natural manner to obtain several results on *round tournaments* (also known as *locally transitive tournaments*), including characterisation, recognition, and isomorphism testing (see e.g. [5] for more detailed information), as well as further computational results, such as the *feedback vertex set* computation.

2 Umodule, an enlarged notion of module

The following definitions and properties are from previous works of the same authors [8,9], which generalise modular decomposition from graphs to homogeneous relations.

2.1 Homogeneous Relation & Modules

Let X be a finite set. The family of all subsets of X is denoted by $\mathcal{P}(X)$. A *diverse* triple is $(x, y, z) \subseteq X^3$ with $x \neq y$ and $x \neq z$, which will be denoted by $(x|yz)$ instead of (x, y, z) since the first element plays a particular role. Let H be a boolean relation over the diverse triples of X. Then, H_x denotes the binary relation on $X \setminus \{x\}$ such that $H_x(y, z) \Leftrightarrow H(x|yz)$.

Definition 1 (Homogeneous Relation) *H is a* homogeneous relation on *X if, for all $x \in X$, H_x is an equivalence relation on $X \setminus \{x\}$ (i.e. it fulfils the symmetry, reflexivity and transitivity properties).*

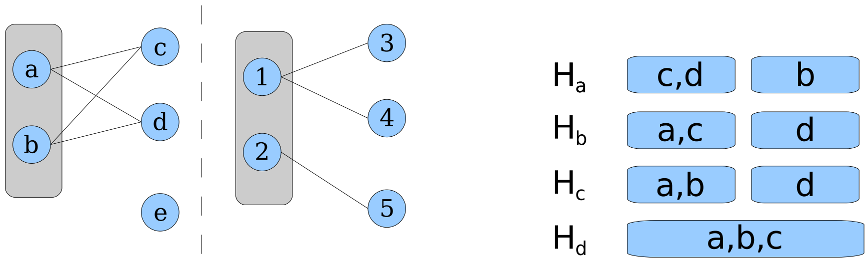
If $\neg H(x|yz)$ we say that x *splits* y from z. Equivalently, a homogeneous relation H can be seen as a mapping from each $x \in X$ to a partition of $X \setminus \{x\}$, namely the equivalence classes of H_x.

Definition 2 (Module) *Let H be a homogeneous relation on X. A subset $M \subseteq X$ is a* module *of H if $H(x|mm')$ for all $m, m' \in M$ and $x \in X \setminus M$.*

Homogeneous relations generalise not only graphs but also *2-structures*, where modular decomposition still applies under the different but equivalent name of *clan decomposition* [16,17]. Roughly, a $2-$structure $G = (X, C)$ is a ground set X and an edge colouration $C : X^2 \to \mathbb{N}$ [16,17]. Thus, a digraph is a $2-$structure using two colours, denoting the existing (when $C(x, y) = 1$) and absent arcs (when $C(x, y) = 0$). Notice that there is no need for the concept of *adjacency* nor *neighbourhood* nor *incidence* in a homogeneous relation. However, a homogeneous relation is canonically derived from graphs and $2-$structures as follows.

Definition 3 (Standard Homogeneous Relation) *The* standard homogeneous relation *$H(G)$ of a 2-structure $G = (X, C)$ is*

$$H(G)(x|uv) \iff C(x, u) = C(x, v) \text{ and } C(u, x) = C(v, x)$$



(a) Undirected graphs (b) Homogeneous relation

Fig. 1. **(a)** Modules and umodules in a graph: $\{a, b\}$ is a module and also a umodule, $\{1, 2\}$ is a umodule but not a module. **(b)** A homogeneous relation with a module which is not a umodule. $\{a, b\}$ is a module: they belong to the same equivalence class in both H_c and H_d. $\{a, b\}$ is not a umodule: c and d belong to the same class in H_a, and to different classes in H_b.

Proposition 1 *Let G be a graph, or a tournament, or an oriented graph, or a directed graph, or a $2-$structure. The modules of $H(G)$ are exactly the modules, or clans, of G in the usual sense (see definitions in [24,33,16]).*

2.2 Umodules

We now introduce the central notion of this paper which, thanks to Proposition 3 (below), can be seen at the same time as a proper generalisation of the classical modules/clans (in the sense of [24,33,16]), and a dual notion to the generalised modules (in the sense of [8,9]).

Definition 4 (Umodules) *A subset U of X is a* umodule *of H if*

$$\forall u, u' \in U, \quad \forall x, x' \in X \setminus U, \quad H(u|xx') \Longleftrightarrow H(u'|xx').$$

Roughly, elements of a umodule come from the same "school of thinking": if one element of a umodule differentiates, resp. mixes together, some exterior elements, so does every element of the umodule (Fig. 1). A umodule U is *trivial* if $|U| \leq 1$ or if $|U| \geq |X| - 1$. The family of umodules of H is denoted by $\mathcal{U}_H$, and $\mathcal{U}$ when no confusion occurs. H is *umodular prime* if all umodules of H are trivial. The following proposition links umodules to the 1-intersecting families framework as defined in [26]. The subsequent one tells how far umodules may generalise modules.

Proposition 2 *For any two umodules U, U' of a homogeneous relation H, if $U \cap U' \neq \emptyset$ then $U \cup U'$ is also a umodule of H.*

Proof: Let v belong to (the non-empty set) $U \cap U'$. For all $u \in U$, for all

$u' \in U'$, and for all $x, x' \notin U \cup U'$,

$$H(u|xx') \iff H(v|xx') \iff H(u'|xx').$$

□

Proposition 3

(1) If H is a standard homogeneous relation (see Definition 3), then any module of H is a umodule of H.
(2) If H is an arbitrary homogeneous relation over a finite set X, then any module M of H is such that $X \setminus M$ is a umodule of H.

Proof: Point 1 immediately follows from Definitions 2 and 4. For Point 2, one just has to notice that if $X \setminus M$ is not a umodule, there exists $m, m' \in M$ and $x, x' \in X \setminus M$ such that $H(x|mm')$ and $H(x'|mm')$. But then M is not a module because of x'. □

Remark 1 *The umodules do not inherit all modules properties. For instance, if M is a module, then $M' \subset M$ is a module of H if and only if M' is a module of $H[M]$ (the restriction of H to M). For umodules this is no longer true.*

2.3 Characterisation of graphs where umodules and modules coincide

In case of graphs, a natural question arises [14]: which are the graphs where the notions of module and umodule coincide? The following result, which can also be seen as a relaxed converse of Proposition 3, solves this problem. As with modules, let the umodules of a graph refer to those of its standard homogeneous relation. Notice here in a graph that the complementary of a umodule is also a umodule. A *threshold graph* is one that can be constructed from the single vertex by repeated additions of a single isolated or dominating vertex. Equivalently, a threshold graph is a graph with no induced P_4, C_4, nor co-C_4, where P_4 denotes the four vertex path, C_4 the four vertex cycle, and co-C_4 the dual graph of C_4.

Proposition 4 *G is a threshold graph if and only if in all induced subgraph of G, every umodule is either a module or the complementary of a module (or both).*

Proof: If G is not threshold, then it contains a subgraph G' isomorphic to either a P_4, a C_4, or a co-C_4. In any case G' contains a two-vertices umodule (two of the latter four) which is neither a module nor the complementary of a module.

Conversely, let $U \subseteq V'$ be a umodule of some induced subgraph $G' = (V', E')$ of G such that U is neither a module nor the complementary of a module of G'. Let $W = V' \setminus U$. That U is not the complementary of a module implies the existence of $a \in U$ and $b, c \in W$ such that a is a splitter of $\{b, c\}$, i.e. $\neg H(a|bc)$ with H being the standard homogeneous relation of G. W.l.o.g. we suppose that ab is an edge, and ac is a non-edge. Let A be the set containing all neighbours of b that belong to U, and $D = U \setminus A$. Let B be the set containing all neighbours of a that belong to W, and $C = W \setminus B$. Using the fact that U is a umodule, and that $a \in A$, $b \in B$, and $c \in C$, one can deduce for all $x \in A, y \in B, z \in C, t \in D$ that xy and zt are edges while xz and yt are non-edges (this corresponds to bi-joins, which are detailed in see Section 6.1). Moreover, U is not a module, and we can deduce that there is a vertex d belonging to D. Finally, one can check that $G[\{a, b, c, d\}]$ is either a P_4, a C_4, or a co-C_4. $\square$

Threshold graphs are known to be one of the smallest graph classes (see e.g. [6]). Therefore for most graphs umodules and modules differ, and Section 6 is devoted to the umodular graph decomposition. However, before deepening decomposition issues, let us first display umodule tractability.

3 Algorithmic Tractability for the general case

As far as we are aware, there is no evidence of a decomposition scheme for arbitrary umodules. The first valuable objects to compute seem to be the maximal umodules with respect to some cut. Using this, we also provide a polynomial time algorithm computing the *strong* umodules (see definition afterwards).

3.1 Maximal Umodules with respect to a cut

Partitions will be ordered with respect to the usual partition lattice: $\mathcal{P} = \{P_1, \ldots, P_p\}$ is *coarser* than $\mathcal{Q} = \{Q_1, \ldots, Q_q\}$, and $\mathcal{Q}$ is *thinner* than $\mathcal{P}$, if every part Q_i is contained in some P_j. It is noted $\mathcal{Q} \leq \mathcal{P}$ and $\mathcal{Q} < \mathcal{P}$ if the partitions are different. Let S be a subset of X. As the umodule family $\mathcal{U}$ is closed under union of intersecting members (Proposition 2), the inclusionwise maximal umodules included in either S or $X \setminus S$ form a partition of X, denoted by $MU(S) = MU(X \setminus S)$. In other words, this is the coarsest partition of X into umodules of H, which is thinner than $\{S, X \setminus S\}$. Roughly, it gives an indication on how the umodules are structured with respect to S: a umodule is either included in a umodule of $MU(S)$, or properly intersects S, or properly intersects $X \setminus S$, or is trivial.

Definition 5 *Let H be a homogeneous relation over X. Let $C \subseteq X$. The relation R_C on C is defined as:*

$$\forall x, y \in C,\ R_C(x, y)\ if\ \forall a, b \in (X \setminus C),\quad H(x|ab) \Longleftrightarrow H(y|ab).$$

This clearly is an equivalence relation on C. Furthermore, C is a umodule if and only if R_C has only one equivalence class. Let us define a *refinement* operation, the main algorithmic tool for constructing $MU(S)$.

Definition 6 *Let $\mathcal{P}$ be a partition of X and C a part of $\mathcal{P}$. Let $C_1, \ldots, C_k$ be the equivalence classes of R_C. $RefineSelf(\mathcal{P}, C)$ is the partition obtained from $\mathcal{P}$, by replacing part C by the parts $C_1, \ldots, C_k$.*

Lemma 1 *Let H be a homogeneous relation over X, U a umodule of H, and $\mathcal{P}$ a partition of X.*

(1) If U is included in a part of $\mathcal{P}$, then for any part C of $\mathcal{P}$, U is included in a part of $RefineSelf(\mathcal{P}, C)$.
(2) A part C of $\mathcal{P}$ is a umodule if and only if $\mathcal{P} = RefineSelf(\mathcal{P}, C)$

Proof: Let U be a umodule and P be the part containing U. For the first statement let us consider $\mathcal{Q} = RefineSelf(\mathcal{P}, C)$ where C is a part of $\mathcal{P}$. If $P \neq C$, then P remains a part of $\mathcal{Q}$ and still contains U. Else $P = C$. Then, as U is a umodule, the vertices of U cannot be separated using refinement. The proof for the second statement is immediate since C is a umodule if and only if $\mathcal{P} = RefineSelf(\mathcal{P}, C)$. □

We can then propose a refinement algorithm computing $MU(S)$ by iterated uses of $RefineSelf$:

$\mathcal{P} \leftarrow \{S, X \setminus S\}$
while *there exists an unmarked part C in $\mathcal{P}$* **do**
  if $\mathcal{P} = RefineSelf(\mathcal{P}, C)$ **then** mark C
  else $\mathcal{P} \leftarrow RefineSelf(\mathcal{P}, C)$

Algorithm 1: Refinement algorithm computing $MU(S)$ given homogeneous relation H over X and $S \subseteq X$

Theorem 1 *Algorithm 1 computes $MU(S)$ in $O(|X|^3)$ time.*

Proof: Thanks to Lemma 1 Point 1, at any step we have $MU(S) \leq \mathcal{P} \leq \{S, X \setminus S\}$. Thanks to Point 2, the process stops when $\mathcal{P} = MU(S)$. The marked parts are the umodules.

For achieving the time complexity, each part of the current partition can be *marked*. A part gets marked if it does not refine the partition. When $\mathcal{P}$ is replaced by $Refine(\mathcal{P}, C)$, the marked parts of $Refine(\mathcal{P}, C)$ are exactly the marked parts of $\mathcal{P}$. At any step either a part is broken, or marked, and marked parts never gets unmarked. So there are less that $2|X|-1$ calls to $RefineSelf$. Complexity follows from the lemma below. $\square$

Lemma 2 *It is possible to compute $RefineSelf(\mathcal{P}, C)$ in $O(|X|^2)$ time.*

Proof: We first show how to test for $R_C(x, y)$. Compute, for every element x of C, a partition $\mathcal{H}(x, C) = \{P_x^1, \ldots, P_x^{k(x)}\}$ of $X \setminus C$. It is the restriction of H_x to $X \setminus C$, i.e $P_x^i = H_x^i \setminus C$. It is easy to build in $O(|X|)$ time for each element of C. Then we have $R_C(x, y)$ if and only if $H(x, C)$ is exactly the same partition as $H(y, C)$. It can be tested in $O(|X|)$ time, but performing this for each couple of elements of C would lead to an $O(|X|^3)$ global time. Let us instead consider $\mathcal{H}(x, C)$ as a b bit vectors (with $b = |X \setminus C| = O(|X|)$). Looking for duplicates among these vectors can be performed easily, by bucket sorting them on their first bit, then the second, and so on. A scan of all vectors (i.e. of all elements of C) computes the pairwise equal vectors, i.e the R_C equivalent elements of C. It is then easy to split C and to update $\mathcal{P}$, in $O(|X|^2)$ time. $\square$

3.2 Strong Umodules: Maximal Umodules Computation and Primality Test

A umodule is *strong* if it does not overlap any other umodule, where two subsets overlap if none of the intersection and differences are empty. As two strong umodules are either disjoint, or one contains another, they can be ordered by inclusion into a tree (see e.g. laminar families in [39]).

Theorem 2 *There exists an $O(|X|^5)$ algorithm to compute the inclusion tree of strong umodules.*

Proof: Consider a non-trivial strong umodule M. For each pairwise distinct $x, y \notin M$ (at least two of them exists since M is not trivial), M is contained in exactly one set of $MU(\{x, y\})$. The intersection of all these sets is exactly M. Indeed if it were M' such that $M \subsetneq M'$ then there would exist $x \in M' \setminus M$. For $y \notin M$, $MU(\{x, y\})$ contains a umodule M'' smaller than M' but containing M, a contradiction. Then the algorithm is as follow. For every pair $\{x, y\}$ compute $MU(\{x, y\})$ in $O(|X|^3)$ time (Theorem 1). That gives a family of at most $|X|^3$ umodules. Add the trivial modules to the family. Greedily compute the intersection of overlapping umodules of the family. It is possible in $O(|X|^4)$ time: for each triple (a, b, c) look for the umodules containing exactly two of the three elements, they overlap. Then we have all strong umodules. We finally just have to order them into a tree. $\square$

This answers both maximal umodules enumeration and primality test since a non-trivial umodule exists if and only if a non-trivial strong umodule exists.

4 Two Decomposition Scenarios

Of course, the number of umodules may be as large as $2^{|X|}$. But we shall now focus on certain umodule families having a compact (polynomial-size) representation. Umodules of local congruence 2 relations, on the first hand, and self-complemented umodules families, on the second hand, have such properties. They can be stored in $O(|X|^2)$ and $O(|X|)$ space, respectively.

4.1 Local Congruence and Crossing Families

Definition 7 (Local congruence) *Let H be a homogeneous relation on X. For $x \in X$, the* congruence *of x is the maximal number of elements that x pairwise splits. In other words, it is the number of equivalence classes of H_x. The* local congruence *of H is the maximum congruence of the elements of X.*

Remark 2 *The standard homogeneous relation of an undirected graph or a tournament has local congruence 2. This value is 3 for an antisymmetric directed graph or a directed acyclic graph. The value is 4 for digraphs.*

When the local congruence of H is 2, so-call *LC2 condition* for short, we obtain the following structural property on its umodule family.

Definition 8 (Crossing family) *$\mathcal{F} \subseteq 2^X$ is a crossing family if, for any $A, B \in \mathcal{F}$, that $A \cap B \neq \emptyset$ and $A \cup B \neq X$ implies $A \cap B \in \mathcal{F}$ and $A \cup B \in \mathcal{F}$ (see e.g. [39] for further details).*

Crossing families commonly arise as the minimisers of a submodular function. For instance, the minimum $s,t-$cuts of a network form a crossing family. Gabow proved that a crossing family admits a compact representation in $O(|X|^2)$ space using a tree representation [21,22].

Proposition 5 *The umodules of a homogeneous relation with local congruence* 2 *form a crossing family, and can thus be stored in $O(|X|^2)$ space.*

Proof: Without any assumption on the relation, the union of two overlapping umodules is also a umodule. Now let us consider two overlapping sets $A, B \in \mathcal{F}$, with $A \cap B \neq \emptyset$ and $A \cup B \neq X$, by hypothesis $A \setminus B$ and $B \setminus A$ are non-empty, $a \in A \setminus B$ and $b \in B \setminus B$. Moreover to be relevant $A \cap B$ must contain at least two elements otherwise the intersection is obviously a umodules. So $y, z \in A \cap B$. And finally $x \in X \setminus A \cup B$. By hypothesis we have $H(a|xb) \Leftrightarrow H(y|xb) \Leftrightarrow H(z|xb)$ and $H(b|xa) \Leftrightarrow H(y|xa) \Leftrightarrow H(z|xa)$ and as there are only two possible classes we have $H(y|ab) \Leftrightarrow H(z|ab)$. □

4.2 Self-complementarity and Bipartitive Families

A consequence of previous proposition is that standard homogeneous relations of graphs and tournaments have crossing umodules families. But they have stronger properties, which we will use to show a linear-space structure coding the umodule family.

Definition 9 (Four elements condition) *H fulfils the* four elements condition *if* $\forall\ m, m', x, x' \in X$,

$$\begin{cases} H(m|xx') \wedge H(m'|xx') \wedge H(x|mm') \Rightarrow H(x'|mm') \\ \neg H(m|xx') \wedge \neg H(m'|xx') \wedge \neg H(x|mm') \Rightarrow \neg H(x'|mm') \end{cases}.$$

Proposition 6 *Standard homogeneous relations of undirected graphs and tournaments satisfy the four elements condition.*

The proof is not hard. This is a light regularity condition, allowing to avoid examples similar to that of Fig. 1(b). Surprisingly enough, it suffices to make the umodule family behave in a very tractable manner (Proposition 7 and Corollary 1 below).

Definition 10 (Self-complementary condition) *A family $\mathfrak{F}$ of subsets of X is* self-complemented *if for every subset A, $A \in \mathfrak{F}$ implies $X \setminus A \in \mathfrak{F}$.*

Proposition 7 *If a homogeneous relation H fulfils the four elements condition then the family $\mathfrak{U}$ of umodules of H is self-complemented.*

Proof: Let us assume that U is a umodule and $X \setminus U$ is not, i.e. there are two elements x and x' of $X \setminus U$, and two elements m and m' of X such that $H(x|mm')$ but $\neg H(x'|mm')$. As U is a umodule, either both m and m' split x from x' (i.e. $\neg H(m|xx')$ and $\neg H(m'|xx')$) or none of m and m' split x from x' (i.e. $H(m|xx')$ and $H(m'|xx')$). The first case is forbidden by the first implication of the four elements condition, and the second case is forbidden by the second implication. □

The *four elements condition* is quite convenient since it allows to shrink a umodule, hence apply the divide and conquer paradigm to solve optimisation problems. However, as far as umodules are concerned, the *self-complementary* relaxation is sufficient to describe a tree-decomposition theorem as can be seen in the following section. Finally, notice that the converse of Proposition 7 does not necessarily hold. The characterisation of relations having a self-complemented umodule family by a local axiom, such as the four elements condition, actually appears to be more difficult.

4.2.1 Tree Decomposition Theorem

Let us now present known properties of certain families of bipartitions. The following results can be found in [15] under the name of "decomposition frame with the intersection and transitivity properties", in [34] under the name of "bipartitive families" (the formalism used in this paper), and in [29] under the name of "unrooted set families".

We call $\{X_i^1, X_i^2\}$ a *bipartition* of X if $X_i^1 \cup X_i^2 = X$ and $X_i^1 \cap X_i^2 = \emptyset$. Two bipartitions $\{X_i^1, X_i^2\}$ and $\{X_j^1, X_j^2\}$ *overlap* if for all $a, b = 1, 2$ the four intersections $X_i^a \cap X_j^b$ are not empty. A bipartition is *trivial* if one of the two parts is of size 1. Let $\mathcal{B} = \{\{X_i^1, X_i^2\}_{i \in 1,\dots,k}\}$ be a family of k bipartitions of X. The *strong* bipartitions of $\mathcal{B}$ are those that do not overlap any other bipartition of $\mathcal{B}$. For instance, the trivial bipartitions of $\mathcal{B}$ are strong bipartitions of $\mathcal{B}$.

Proposition 8 *If $\mathcal{B}$ contains all trivial bipartitions of X, then there exists a unique tree $T(\mathcal{B})$*

- *with $|X|$ leaves, each leaf being labelled by an element of X.*
- *such that each edge e of $T(\mathcal{B})$ correspond to a strong bipartition of $\mathcal{B}$: the leaf labels of the two connected components of $T - e$ are exactly the two parts of a strong bipartition, and the converse also holds.*

Let N be a node of $T(\mathcal{B})$ of degree k. The labels of the leaves of the connected components of $T - N$ form a partition $X_1, \dots, X_k$ of X. For $I \subseteq \{1, \dots, k\}$ with $1 < |I| < k$, the bipartition $B(I)$ is $\{\cup_{i \in I} X_i, X \setminus \cup_{i \in I} X_i\}$.

Definition 11 (Bipartitive Family) *A family of bipartitions is a* bipartitive family *if it contains all the trivial bipartitions and if, for two overlapping bipartitions $\{X_i^1, X_i^2\}$ and $\{X_j^1, X_j^2\}$, the four bipartitions $\{X_i^a \cup X_j^b, X \setminus (X_i^a \cup X_j^b)\}$ (for all $a, b = 1, 2$) belong to $\mathcal{B}$.*

Theorem 3 *If $\mathcal{B}$ is a bipartitive family, the nodes of $T(\mathcal{B})$ can be labelled* complete, circular *or* prime, *and the children of the* circular *nodes can be ordered in such a way that:*

- *If N is a complete node, for any $I \subseteq \{1, \dots, k\}$ such that $1 < |I| < k$,*

$B(I) \in \mathcal{B}$.

- *If N is a circular node, for any interval $I = [a, \ldots, b]$ of $\{1, \ldots, k\}$ such that $1 < |b - a| < k$, $B(I) \in \mathcal{B}$.*
- *If N is a prime node, for any element $I = \{a\}$ of $\{1, \ldots, k\}$ $B(I) \in \mathcal{B}$.*
- *There are no more bipartitions in $\mathcal{B}$ than the ones described above.*

For a bipartitive family $\mathcal{B}$, the labelled tree $T(\mathcal{B})$ is an $O(|X|)$-sized representation of $\mathcal{B}$, while the family can have up to $2^{|X|-1} - 1$ bipartitions of $|X|$ elements each. This allows to efficiently perform algorithmic operations on $\mathcal{B}$. Notice that any self-complemented subset family can be seen as a family of bipartitions.

Proposition 9 *The members of a self-complemented umodule family form a bipartitive family.*

Proof: As $\mathcal{U}'(H)$ is self-complemented each part of a bipartition belonging to $\mathcal{U}'(H)$ is a umodule. Furthermore if the bipartitions $\{U, X \setminus U\}$ and $\{V, X \setminus V\}$ overlap (in the bipartition sense) then U and V overlap (in the set sense). According to Proposition 2 if U and V overlap $U \cup V$ is a umodule, and therefore $\{U \cup V, X \setminus (U \cup V) \in \mathcal{U}'(H)\}$. The self-complementary condition gives the results needed for the three other bipartitions. $\square$

Corollary 1 (Umodular Decomposition Theorem) *There is a unique $O(|X|)$-sized tree that gives a description of all possible umodules of a homogeneous relation H fulfilling the self-complementary condition. This tree is henceforth called* umodular decomposition tree. *Notice that it is an unrooted tree, unlike the modular decomposition tree.*

4.2.2 Tree Decomposition Algorithm

Let H be a self-complemented homogeneous relation, $T(H)$ its umodular decomposition tree, and U a nontrivial strong umodule (if any). Let us examine some consequences of Theorem 3. Notice that two umodules overlap if and only if they are incident to the same node of $T(H)$. As H is self-complemented the union of two overlapping umodules is a umodule (Proposition 2) but also their intersection. The strong umodule U is an edge in $T(H)$ incident with two nodes A and B.

- If one of them, say A, is labelled prime then for any $x, y \notin U$ such that the least common ancestor of them in $T(H)$ is A, then $U \in MU(\{x, y\})$.
- If one of them, say A, is labelled circular then for any x belonging to the subtree rooted in the successor of U in the ordered circular list of A, and for any y belonging to the subtree rooted in the predecessor of U, then $U \in MU(\{x, y\})$.

- If one of them, say A, is labelled complete then the intersection, for all $x, y \notin U$ whose least common ancestor is A, the intersection of all parts of $MU(\{x, y\})$ containing U is exactly U.

Theorem 2 then can be used to compute the strong umodule inclusion tree. After this, typing the nodes and ordering their sons according to the above definition is straightforward. Hence,

Theorem 4 *There exists an $O(|X|^5)$ algorithm to compute the unique decomposition tree for a self complemented umodule family.*

Proof: First compute the inclusion tree of strong module using Theorem 2. Then type the node and order their sons according to definition above, an easy task. □

5 A potent Tractability Theorem: Seidel-switching

Standard homogeneous relations of graphs and tournaments are of local congruence 2, and their umodule families are self-complemented. Firstly this means we can either decompose those families using the crossing families decomposition or using the bipartitive decomposition. Moreover, relations that satisfy both the self-complementary and *LC2* properties seem to own stronger potential. In particular, let us show a nice local transformation from the umodules of such a relation to the modules of another relation. This operation was first introduced in J. Seidel in [40] on undirected graphs. It was later studied by several authors interested in some computational aspects [13,30] and structural properties [27,28] and recently in [35]. The operation is referred to as *Seidel switch* in [28], and we will adopt this terminology. We generalise it to homogeneous relations but take a restricted case of switch, with the slight difference that we remove from the transformation an element (see Fig. 2). For convenience, if H is a homogeneous relation on X and $s \in X$, we also refer to the equivalence classes of H_s as $H_s^1, \ldots, H_s^k$.

5.1 Seidel-switching Theorem

Definition 12 (Seidel switch) *Let H be a homogeneous relation of local congruence* 2 *on X, and s an element of X. The* Seidel switch *at s transforms H into the homogeneous relation $H(s)$ on $X \setminus \{s\}$ defined as follows.*

$$\forall x \in X \setminus \{s\}, H(s)_x^1 = (H_x^1 \Delta H_s^j) \setminus \{s\} \text{ and } H(s)_x^2 = (H_x^2 \Delta H_s^j) \setminus \{s\}$$

with j such that $x \notin H_s^j$, where $A \Delta B$ denotes the symmetric difference of A and B.

When applied to graphs, the Seidel switch simply exchanges edges and nonedges going between $N(x)$ and $\overline{N(x)}$.

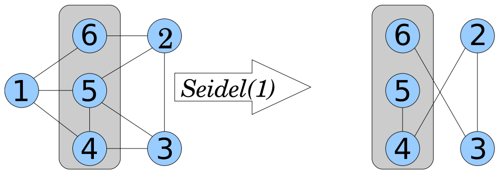


Fig. 2. An example of a Seidel switch on an undirected graph

Theorem 5 (Seidel-switching Theorem) *Let H be a homogeneous relation of local congruence 2 on X such that $\mathcal{U}_H$ is self-complemented. Let s be a member of X, and $U \subseteq X$ a subset containing s. Then, U is a umodule of H if and only if $M = X \setminus U$ is a module of the Seidel switch $H(s)$.*

Proof: Let $C = H_s^1 \cap M$ and $D = H_s^2 \cap M$. Since H is of local congruence 2, $\{C, D\}$ is a partition of M. Let $a \in U \setminus \{s\}$. Suppose that U is a umodule of H. Then, for all $y, z \notin U$, $H(a|yz)$ if and only if $H(s|yz)$. In other words, C is included in one class among H_a^1 and H_a^2 while D is included in the other class. As $C \subseteq H_s^1$ and $D \cap H_s^1 = \emptyset$, $C \cup D$ is included in one among the two classes $H(s)_a^i = H_a^i \Delta H_s^j$ ($i \in \{1, 2\}$ and j as in Definition 12). Hence, $M = C \cup D$ is a module of $H(s)$.

Conversely, if M is a module of $H(s)$, then $C \cup D$ is included in either $H(s)_a^1$ or $H(s)_a^2$. Moreover, the definition of the Seidel switch can also be written as $H_a^i = H(s)_a^i \Delta H_s^j$ for $i \in \{1, 2\}$ and j as in Definition 12. Therefore, C is included in one class among H_a^1 and H_a^2 while D is included in the other class. In other words, for all $a \in U \setminus \{s\}$, and $y, z \notin U$, $H(a|yz)$ if and only if $H(s|yz)$. This implies for all $a, b \in U$, and $y, z \notin U$, $H(a|yz)$ if and only if $H(b|yz)$ and U is therefore a umodule. □

Notice that the modular decomposition tree of H may be trivial, while the one of its Seidel switch at s may be not.

5.2 Links between umodules and modules of the Seidel switch

Modular decomposition trees have well-known properties [10,33]. They are rooted trees whose leaves are in one-to-one correspondence with elements of X. A node of the modular decomposition tree is exactly a strong module, a module that overlap (in the set sense) no other modules. For a node N let $F_1, \ldots, F_k$ be the leaf-sets of its k children in the tree. When the family of umodules of H is bipartitive as it is the case in Theorem 5, the family of modules of any Seidel switch of H is a partitive set family [10], also known as *rooted set family* [29]. The following theorem from [10] describes the structure of partitive set families.

Theorem 6 *[10] The nodes of a modular decomposition tree T can be labelled* complete*,* linear *or* prime*, and the children of the* linear *nodes can be ordered in such a way that:*

- *If N is a complete node, for any $I \subset \{1, \ldots, k\}$ such that $1 < |I| < k$, $\bigcup_{i \in I} F_i$ is a module.*
- *If N is a linear node, for any interval $I = [a, \ldots, b]$ of $\{1, \ldots, k\}$ such that $1 < |b - a| < k$, $\bigcup_{i \in I} F_i$ is a module.*
- *If N is a prime node, for any element $I = \{a\}$ of $\{1, \ldots, k\}$ $\bigcup_{i \in I} F_i$ is a module.*
- *There are no more modules than the ones described above.*

The relationships between the umodular decomposition tree of H and the modular decomposition tree of $H(v)$ are very tight:

Proposition 10 *Let H be a homogeneous relation of local congruence 2 on X such that $\mathcal{U}_H$ is self-complemented. Let s be an element of X. The umodular decomposition tree $\mathcal{T}_H$ of H and the modular decomposition tree $\mathcal{T}_H(s)$ of the Seidel switch $H(s)$ of H at s have the following properties:*

- *the two trees are exactly the same (same nodes and edges) excepted that the leaf with label s is missing in $\mathcal{T}_H(s)$ but present in $\mathcal{T}_H$.*
- *The node of $\mathcal{T}_H$ that is adjacent to the leaf s corresponds to the root of $\mathcal{T}_H(s)$ (while $\mathcal{T}_H$ is unrooted).*
- *A circular node of $\mathcal{T}_U$ corresponds to a linear node of $\mathcal{T}_M(s)$. The orderings of the children are the same. The prime and complete nodes are the same in both trees.*

Proof: This is a consequence of Theorem 5. Every strong module of $H(s)$ gives a strong bipartition of $\mathcal{T}_H$, and the converse is true. Then for a node N of the modular decomposition tree, for any union $\bigcup_{i \in I} F_i$ of leaf-sets of children there is a bipartition $\{\bigcup_{i \in I} F_i, X \backslash (\bigcup_{i \in I} F_i) = B(I)$ using the notations defined above. For each bipartition of umodules of H, the part that contain s is dropped and the other part is put is the family of modules of $H(s)$. $\square$

A similar result (with more details) can be found in [29]. That article indeed describes the relationship between the consecutive-ones ordering and the circular-ones ordering of a boolean matrix, but the transformations (described in [29] as the casting of a PQ-tree into a PC-tree) are the same.

5.3 Linear-time umodular decomposition algorithm

Theorem 7 *The umodular decomposition tree of a self-complemented homogeneous relation of local congruence* 2 *on* X *can be computed in* $O(|X|^2)$ *time.*

Proof: Using a Seidel switch on any element will result in a homogeneous relation having the so-called *modular quotient* property [8]: every module of the relation is also a umodule. Then, the $O(|X|^2)$-time modular decomposition algorithm for modular quotient relations depicted in [8] and Proposition 10 allow to conclude. The algorithm is basically an adaptation of an algorithm from Ehrenfreucht *et alii* [18] to homogeneous relations. □

6 Umodular Decomposition of Graphs and Tournaments

Let us now apply umodular decomposition to two well-known combinatorial objects: undirected graphs and tournaments. In this section we always implicitly refer to their standard homogeneous relations, for instance "the umodules of the graph G" stands for "the umodules of the standard homogeneous relation $H(G)$ of the graph G" and so on. And "graph" stands for "undirected graph". As we have seen, graphs and tournaments fulfil the four elements conditions, are of local congruence two, and their umodule family is self-complemented.

6.1 Bi-join decomposition of undirected graphs

Let us now apply the umodular decomposition framework to graphs, or more exactly to the standard homogeneous relation of a graph. The resulting decomposition was already published in [35,36]. We summarise here the main results of that paper and establish the link with umodules.

Definition 13 (bi-join) *A* bi-join *of a graph* $G = (X, E)$ *is a bipartition* $\{X^1, X^2\}$ *of the vertex-set such that the edges between* X^1 *and* X^2 *form at most two disjoint complete bipartite graphs, and that for each* $i, j = 1, 2$ *every vertex of* X^i *is adjacent to a vertex of* X^j*.*

Proposition 11 *If* $\{X^1, X^2\}$ *is a bi-join of a graph then both* X^1 *and* X^2 *are umodules of* G*.*

Proof: Let A (resp. C) be the vertices of X^1 (resp. X^2) incident with the first complete bipartite graph, and B (resp. D) be the other vertices of X^1 (resp.

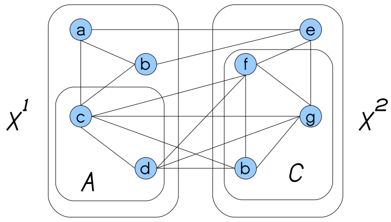


Fig. 3. Example of a bi-join of a graph

X^2). Any vertex of X^1 splits a vertex of C from a vertex of D, but can not split two vertices from C, nor two vertices from D. X^1 is thus a umodule, and a similar proof holds for X^2. □

In [35,36] the Seidel switch was used to derive most of the properties:

Proposition 12 *Let G be a graph. $\{X^1, X^2\}$ is a bi-join of G if and only if for every $v \in X^1$ (resp. X^2) X^2 (resp. X^1) is a module of the Seidel switch $G(v)$.*

It may be used to prove the converse of Proposition 11:

Proposition 13 *If U is a umodule of a graph $G = (X, E)$ then $\{M, X \setminus U\}$ is a bi-join of G.*

This is because the homogeneous relation of a tournament has local congruence 2 and is self-complemented (see Section 4.1).

Corollary 2 *The umodular decomposition of a graph equipped with its standard homogeneous relation is exactly its bi-join decomposition.*

Among the consequences exposed in [35,36], bi-join (thus umodular) decomposition trees have no *circular* nodes.

Theorem 8 *[36] There is a unique unrooted decomposition tree T associated to an undirected graph G. All the nodes are labelled degenerate or prime. There is exactly two kind of degenerate nodes: The clique nodes K_n and the complete bipartite node $K_{n,m}$.*

Isomorphism of (C_5,bull,gem,co-gem)-free graphs

In this section, we prove that the isomorphism testing testing between two graphs totally decomposable w.r.t bi-join decomposition can be tested in linear time. This class of graph is studied by [36] but without considerations about isomorphism. They are exactly the (C_5,bull,gem,co-gem)-free graphs (see Figure 4), and also exactly the graphs that can be obtained from a single vertex by a sequence of (twin,antitwin)-extensions.

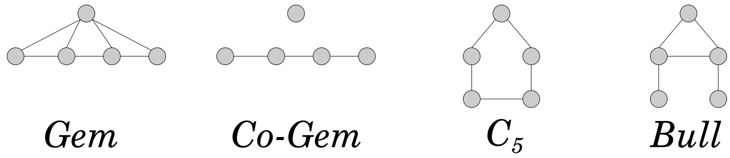


Fig. 4. Forbidden induced subgraphs for Completely Bi-join Decomposable Graphs

It follows from definition that the decomposition tree has no *prime* nodes; furthermore, the decomposition tree alone is an $O(n)$-sized encoding of the graph (like the cotree is an $O(n)$-sized encoding of a cograph). We are then reduced to a tree isomorphism problem, as proven below.

Theorem 9 *Let G_1 and G_2 be graphs totally decomposable w.r.t. bi-join decomposition. Isomorphism between G_1 and G_2 can be tested in linear time.*

Proof: From Theorem 8 and [36], the decomposition tree of a graph is uniquely defined, and a decomposition tree with no prime nodes corresponds to exactly one graph. It is then sufficient to test for decomposition trees isomorphism.

It is possible to compute the decomposition trees of G_1 and G_2 in $O(n + m)$ (see [36]). Then the tree isomorphism is achieved in linear time [1]. Notice that decomposition trees are unrooted, and that the internal node labelling with a K or S is already known. □

6.2 A New Tournament Decomposition

We have investigated in Section 6.1 the umodules of undirected graphs, and noticed that they lead to a nice decomposition. Similarly for tournaments our theory applies and we present a new tournament decomposition: the umodular decomposition. It is indeed the umodular decomposition of the standard homogeneous relation of the tournament. Actually this decomposition is more powerful than the modular decomposition, because every module of a tournament is a umodule, but the converse is not true. Let us say a tournament is M-prime (resp. U-prime) if it has no non-trivial module (resp. umodule). Then we may decompose M-prime tournaments (Figure 5).

We can deduce from Proposition 10 some very interesting properties of the umodular decomposition of tournaments.

Corollary 3 *The umodular decomposition tree of a tournament has no* complete node. *And there exists a circular ordering of the vertices of the tournament such that every umodule of the tournament is a factor (interval) of this circular ordering.*

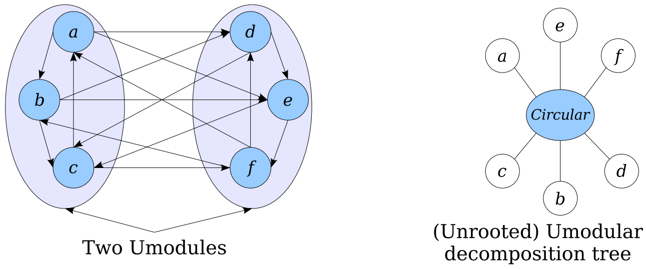


Fig. 5. An example of a M-prime tournament which is not U-prime. The umodular decomposition tree is drawn on the right.

Proof: The first observation follows from Theorem 5: after a Seidel switch at any vertex, we get a tournament $H(s)$. It is well-known that the modular decomposition tree of a tournament has no *complete* node. Then apply Proposition 10 to any vertex s: umodular decomposition tree of T has no *complete* node because modular decomposition tree of $T(s)$ has no *complete* node. The second sentence is straight from Theorem 3. □

This result was already known for modular decomposition [34]: there exists a (not circular) permutation of the vertices whose every module of the tournament is a factor. It is called *factorising permutation*.

Proposition 14 *The umodular decomposition tree of a tournament can be computed in $O(|X|^2)$ time.*

Proof: Again Theorem 5 says that one just has to perform a Seidel switch on a arbitrarily chosen vertex, then to compute the modular decomposition of the tournament. This can be done in linear (in fact $O(|X|^2)$) time using the algorithm from [32]. Proposition 10 tells how to cast the modular decomposition tree into the umodular one. □

Given a graph decomposition scheme, is often worth to consider the totally decomposable graphs with respect to that scheme, namely the graphs in which every "large enough" subgraph admits a non trivial decomposition. In general this leads to the definition of very interesting class of graphs, such as cographs with modular decomposition or distance hereditary graphs with split decomposition. Totally umodular decomposable homogeneous relations may also be defined. Let us deepen the special case of standard homogeneous relations of tournaments.

6.3 Locally transitive tournaments

In this section, we focus on totally umodular decomposable tournaments. We first obtain strong structural relationship between the important graph class of cographs (see e.g. [6]) and the tournament class of locally transitive tournaments, which is also known as round tournaments, a sub-class of locally semicomplete digraphs (refer to [5] for more details). We then show how our theory provides a very natural manner to obtain several results on round tournaments, including characterisation by forbidding induced subgraphs, recognition, isomorphism testing, and feedback vertex set computation. It is well-known that:

Proposition 15

- *If T is an M-prime tournament then T contains an induced cycle with 3 vertices.*
- *T is totally decomposable w.r.t. modular decomposition if and only if it contains no induced cycle with 3 vertices (it is a transitive tournament).*

6.3.1 Characterisation theorems

We have:

Theorem 10 *If T is an U-prime tournament then T contains a* diamond *(one of the induced subgraph described in Figure 6). T is totally decomposable w.r.t. umodular decomposition if and only if it is diamond-free.*

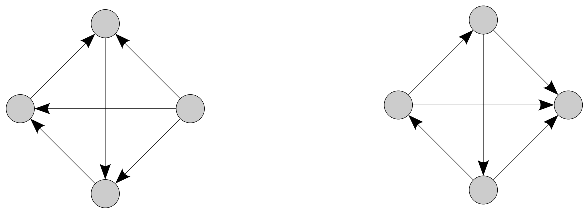


Fig. 6. Diamonds = minimal U-prime configurations in tournaments = forbidden subgraphs of a tournament totally decomposable w.r.t. umodular decomposition

Proof: Thanks to Theorem 5, T is U-prime if and only if for any vertex v a Seidel switch at v gives an M-prime tournament. Thanks to Proposition 15, one just has to check all the four-vertices tournaments where a Seidel switch on a vertex produces the cycle with 3 vertices. It is tedious but no hard. □

Another characterisation is possible:

Definition 14 *A tournament T is* locally transitive *if for each vertex $x \in V(T)$, $T_{[N^+(x)]}$ and $T_{[N^-(x)]}$ are transitive tournaments.*

It is not hard to see the equivalence between the two classes, a classical result:

Proposition 16 *A tournament T is diamond-free if and only if it is locally transitive*

6.3.2 Recognition algorithm

Thanks to Theorem 10 the class membership can be checked in $O(|X|^4)$ time, and thanks to Proposition 16 in $O(|X|^3)$ time. The following condition provides however a faster test by checking only one vertex of the graph.

Another linear-time recognition algorithm was given by [11]. As far as we know, this french thesis was never published in English. We present here another linear-time recognition algorithm, based on the factorising permutation instead of so-called *circular ordering* (see below). Our algorithm is furthermore *certifying*: it outputs a diamond if the graph is not diamond-free, i.e. not locally transitive.

Proposition 17 *Let T be a tournament and x an arbitrary vertex. T is locally transitive if and only if*

(1) $T_{[N^+(x)]}$ and $T_{[N^-(x)]}$ are transitive tournaments, and
(2) if a vertex $a \in N^+(x)$ has an out-neighbour $b \in N^-(x)$ and an in-neighbour $c \in N^-(x)$ then $(b, c) \in T$.
(3) if a vertex $a \in N^-(x)$ has an out-neighbour $b \in N^+(x)$ and an in-neighbour $c \in N^+(x)$ then $(b, c) \in T$.

Proof: Let us suppose T is totally decomposable. According to Proposition 16, (i) holds. If (ii) does not hold for some vertices a, b and c, i.e. if there is an arc (c, b) instead of (b, c), then $\{a, b, c, x\}$ induce a forbidden configuration of Figure 6. Same if (iii) does not hold.

Conversely let us suppose that the three conditions hold. We shall prove that for every vertex, its in- and out-neighbourhoods are transitive. Then Proposition 16 tells T is totally decomposable w.r.t. umodular decomposition. For x, this is true thanks to (i) . Let t be a vertex of $N^+(x)$. If $T_{[N^+(t)]}$ is not transitive then it contains a circuit (u, v, w) (with arcs, w.l.o.g., (u, v) and (v, w) and (w, u).) As both $T_{[N^+(x)]}$ and $T_{[N^-(x)]}$ are transitive, the circuit overlaps them. Suppose w.l.o.g $u \in N^+(x)$ and $w \in N^-(x)$. Then (iii) is not true: take $a = w$ and $b = u$ and $c = t$.
If we suppose $T_{[N^+(t)]}$ is not transitive, it contains a circuit (u, v, w) with an arc (u, w), $u \in N^+(x)$ and $w \in N^-(x)$. (iii) is also violated: take $a = w$ and $b = t$ and $c = u$.

Now let t be a vertex of $N^-(x)$. If $T_{[N^+(t)]}$ is not transitive then it contains a circuit (u, v, w) with an arc (u, w), $u \in N^+(x)$ and $w \in N^-(x)$. (ii) is violated with $a = u$ and $b = w$ and $c = t$. And if $T_{[N^-(t)]}$ is not transitive then it contains a circuit (u, v, w) with an arc (u, w), $u \in N^-(x)$ and $w \in N^+(x)$. (ii) is violated with $a = u$ and $b = t$ and $c = w$. □

Theorem 11 *There exists an $O(|X|^2)$-time certifying algorithm to recognize if a tournament is locally transitive.*

Proof: Condition (i) of the Proposition 17 can be tested in $O(|X|^2)$ time. Number $a_0 \dots a_k$ the vertices of $N^+(x)$ in increasing order along the transitive tournament $T_{[N^+(x)]}$, and $b_0 \dots b_l$ the vertices of $N^-(x)$ in increasing order. If a_i fulfills (ii), then its out-neighbourhood contains $b_0 \dots b_{f(i)}$ and its in-neighbourhood $b_{f(i)+1} \dots b_l$. This can be tested in $O(|X|)$ time. A similar test in $O(|X|)$ time is performed for each a_i and b_j, leading to an $O(|X|^2)$-time algorithm. $\square$

Algorithm 2 presents a certifying implementation of this proof.

6.3.3 Umodular decomposition tree of a locally transitive tournament

Theorem 12 (Umodules of a locally transitive tournament) *The umodular decomposition tree of a locally transitive has only one single node. Moreover this node is a circular node.*

Proof: According to Theorem 5 for any x the Seidel switch at vertex x of a tournament T totally decomposable w.r.t. umodular decomposition gives a tournament $T(x)$ totally decomposable w.r.t. modular decomposition. According to Proposition 15 $T(x)$ is transitive: its modular decomposition tree has a single linear node. According to Proposition 10 the umodular decomposition tree of T only has a circular node. $\square$

It is well known that for encoding a cograph, it is enough to store its modular decomposition tree. Unfortunately, for a locally transitive tournament, the decomposition tree is not enough since it does not encode adjacencies between vertices.

The circular ordering of the vertices along this unique *circular* node is called a *circular factorising permutation*, since every umodule of G is an interval of this circular permutation, and the converse also holds, by definition of a circular node.

Further results on locally transitive tournaments are known:

6.3.4 Circular structure of locally transitive tournaments

A circular structure result of Locally transitive tournament is known from Lopez and Rauzy, we recall it here.

Definition 15 *A tournament $G = (V, E)$ is a* complete circuit *if the vertices*

Data: A Tournament $T = (V, A)$
Result:
  Yes: A circular factorising permutation σ of T.
  No: An obstruction.

begin
  Pick a vertex $x \in V$
  $\mathbf{A} \leftarrow N^+(x)$
  $\mathbf{B} \leftarrow N^-(x)$
  if $T_{[\mathbf{A}]}$ *is not a transitive tournament* **then**
    Failure: certificate is *a 3-circuit contained in A dominated by x*
  Order $a_1..a_k$ the vertices of A according to the total ordering (a_1 is the source and a_k the sink)
  if $T_{[\mathbf{B}]}$ *is not a transitive tournament* **then**
    Failure: certificate is *a 3-circuit contained in B anti-dominated by x*
  Order $b_1..b_l$ the vertices of B according to the total ordering (b_1 is the source and b_l the sink)
  for $i \leftarrow 1$ **to** k **do**
    $j \leftarrow 1$ **while** $j \leq l$ *and* $b_j \in N^+(a_i)$ **do**
      j++
    while $j \leq l$ *and* $b_j \in N^-(a_i)$ **do**
      j++
    if $j \neq l + 1$ **then**
      Failure: certificate is $\{x, a_i, b_{j-1}, b_j\}$*; a diamond where b_{j-1} is the source*
      Comment: indeed then $b_j \in N^+(a_i)$ but $b_{j-1} \in N^-(a_i)$ and $b_j \in N^+(b_{j-1})$
  for $j \leftarrow 1$ **to** l **do**
    $i \leftarrow 1$ **while** $i \leq k$ *and* $a_i \in N^+(b_j)$ **do**
      i++
    while $i \leq k$ *and* $a_i \in N^-(b_j)$ **do**
      i++
    if $i \neq k + 1$ **then**
      Failure: certificate is $\{x, b_j, a_{i-1}, a_i\}$*; a diamond where a_i is the sink*
      Comment: indeed then $a_i \in N^+(b_j)$ but $a_{i-1} \in N^-(b_j)$ and $a_i \in N^+(a_{i-1})$
  Pick a vertex x
  Compute a Seidel switch on x
  $\sigma \leftarrow Seidel(x) \cup x$
  return $\sigma(V)$ *a circular factorising permutation.*
end

Algorithm 2: Certifying recognition of a totally U-decomposable tournament. The certificate output on failure is a diamond; on success is a circular factorising permutation (the set of intervals of this permutation is exactly the set of umodules).

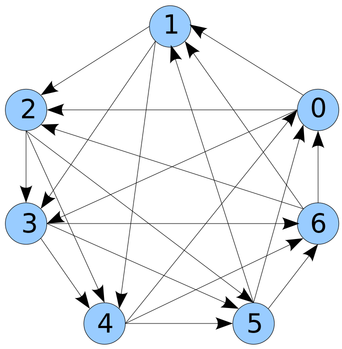


Fig. 7. The complete circuit of 7 vertices ($k = 3$).

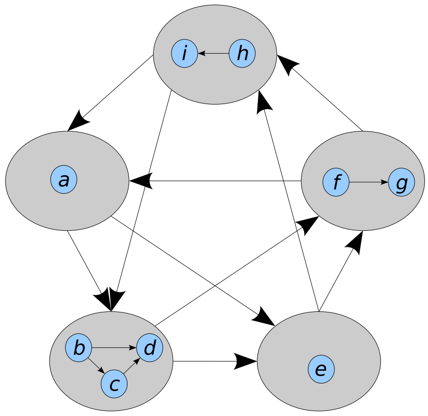


Fig. 8. Example of locally transitive graph of 9 vertices: the complete circuit of 5 vertices where each vertex is substituted with a transitive tournament. A circular ordering is $(\{a\}, \{b, c, d\}, \{e\}, \{f, g\}, \{h, i\})$ while a factorizing permutation is $(a, e, i, h, d, c, b, g, f)$ (see Section 6.3.4)

can be numbered from 0 to $2k$ *and if for every vertex numbered* i*, its out-neighbourhood is the vertices numbered from* $i + 1$ *to* $i + k$ *inclusively (modulo* $2k + 1$*).*

Theorem 13 *[31] Let* $G = (V, E)$ *be a locally transitive tournament.* V *can be partitioned into* $V_0 \dots V_{2k}$*,* $k \geq 0$*, and*

- *for each* $0 \leq i \leq 2k$ $G[V_i]$ *is a transitive tournament*
- *for* $x \in V_i$ *and* $y \in V_j$*, if there exists* $a \leq k$ *such that* $i = a + j$ *modulo* $2k + 1$ *then* (x, y) *is an arc of* G*, otherwise* (y, x) *is an arc of* G

Notice than every V_i is a module of the graph, furthermore these modules are maximal with respect to inclusion (the only module containing V_i is V).

Corollary 4 *A nontrivial module* M *of a locally transitive tournament induces a transitive tournament.*

The circular ordering of the $2k + 1$ strong modules, i.e. the circular partition $V_0 \dots V_{2k}$ as defined in Theorem 13. is henceforth called *circular ordering*

We have seen another "circular structure" exists: the circular factorising permutation of the n vertices. These two circular orderings are not isomorphic however.

Let $G = ([0 \dots 2k], E)$ be the unique (up to isomorphism) complete circuits of $2k + 1$ vertices. Then let τ be the bijection

$$\tau(i) = ki \ modulo \ 2k + 1$$

Let τ' be τ seen as a circular list

Proposition 18 *The intervals of τ' are exactly the umodules of G*

Proof: Thanks to the property of closure under union of overlapping umodules (Proposition 2) we just have to check that the umodules of tow vertices are exactly the pairs $\{i, i + k\}$ (additions are performed modulo $2k + 1$). This is easy to check. □

This proposition can be generalised if G is locally transitive, but not a complete circuit.

Proposition 19 *Let $G = (V, E)$ be a locally transitive tournament and $V_0 \dots V_{2k}$ be its circular ordering. Each V_i induce a transitive tournament, i.e its vertices form a chain $v_i^1 ... v_i^{f(i)}$.*

Let σ be the circular permutation such that

- *Each V_i is a factor (interval) of σ*
- *The V_i follow consecutively following τ', ie V_0 then V_k then V_{2k} then V_{3k}... (subscripts modulo $2k + 1$)*
- *Within each V_i the ordering of vertices is the reverse of the ordering of the chain: $v_i^{f(i)} ... v_i^1$.*

The umodules of G are exactly the intervals of σ (i.e. σ is a circular factorising permutation of G).

Furthermore, σ is the unique circular factorising permutation of G.

Figure 8 gives a example of the relationship between the circular ordering and the circular factorising permutation.

This proposition allows to construct the circular ordering, given the circular factorizing permutation computed by Algorithm 2.

A first step should identify the $2k+1$ induced tournaments. Two vertices u and v are *twins* if $N^+(u) \setminus \{v\} = N^+(v) \setminus \{u\}$. They are *consecutive twins* for a circular factorising permutation σ if they follow consecutively in σ. Let R be the transitive closure of the *consecutive twins* relation

Proposition 20 *The equivalence classes of R are exactly the induced transitive tournaments $V_0 \dots V_{2k}$ of the circular ordering of a locally transitive tournament.*

Proof: It is not hard to check that, in a tournament, two twins form a module of two vertices, and that the classes of the transitive closure R are thus modules. Then just apply Corollary 4. We just have to check that each class M of R is a *maximal* module: if not then there exists x such that $M \cup \{x\}$ is a module, but then either x and M sink, or x and M source, are twins, contradiction. $\square$

Then we can give another quadratic-time algorithm than the one of [11]

Theorem 14 *The circular ordering of a tournament can be computed in $O(n^2)$*

Proof: First Algorithm 2 computes the circular factorising permutation. Then the relation R of Proposition 20 can be computed by checking if the n pairs of consecutive vertices are twins or not. Then the sets $V_0 \dots V_{2k}$ are re-ordered using the inverse of τ as in Proposition 19. $\square$

6.3.5 Efficient storage of locally transitive tournaments

Definition 16 *A* composition *of n is a list of k integer terms such that the sum of the terms is n. The composition is* odd *if k is odd.*

An circular odd composition *of n is a circular list of $2k+1$ integer terms such that the sum of the terms is n. Notice that a circular list has reading direction: $\{1,2,3\}$ differs from $\{3,2,1\}$ but is same than $\{2,3,1\}$.*

Brouwer [7] computed the number of locally transitive tournaments by establishing a bijection between them and "*shift registers where the complement of the bit shifted out of the last position is shifted into the first position*". These results can be rephrased as:

Theorem 15 *[7] There is a bijection between the totally decomposable tournaments of n vertices and the circular odd compositions of n elements.*

Brouwer gave the first terms of the sequence, i.e. the number of locally transitive tournaments on n vertices, referred in Sloane encyclopedia [41] as A000016:

1, 1, 1, 2, 2, 4, 6, 10, 16, 30, 52, 94, 172, 316, 586, 1096, 2048, 3856, 7286, 13798, 26216, 49940, 95326, 182362, 349536, 671092, 1290556, 2485534, 4793492, 9256396, 17895736. He also gave the exact value:

$$\sum_{d|n} \frac{2^{d-1}}{d}\, odd(\frac{d}{n}) \sum_{e|\frac{n}{d}} \frac{\mu(e)}{e}$$

where μ is the Möbius function and $odd(x)$ is 1 if x odd, 0 otherwise.

Remark that the number of tournaments totally decomposable w.r.t. umodular decomposition is strictly larger than the number of tournaments totally decomposable w.r.t. to modular decomposition. For modular decomposition there exists indeed only *one* tournament totally decomposable with n vertices!

Theorem 16 *A locally transitive unlabelled (resp. labelled) tournament of n vertices can be stored in $O(n)$ (resp. $O(n \log n)$) bits.*

Proof: If the tournament is unlabelled, one just has to store the corresponding circular odd integer composition. This can be done using a standard encoding of compositions: a vector of $n-1$ bits. If the kth term of the composition is x, it is stored by $x-1$ ones followed by a zero. The last bit is always zero and thus can be omitted. For instance the composition $\{2, 3, 1, 1, 3\}$ of 10 is stored as $[1, 0, 1, 1, 0, 0, 0, 1, 1]$. This is a classical canonical encoding of compositions [2].

If the graph is labelled, the permutation of vertices is must also be stored, in $O(n \log n)$ bits. $\square$

6.3.6 Minimum Feedback Vertex Set

Definition 17 *A* feedback vertex set *of directed graph $G = (V, A)$ is a subset $V' \subseteq V$ such that each element of V' belongs to at least one circuit of G.*

The Minimum Feedback Vertex Set problem *consists in finding a feedback vertex set of minimum cardinality.*

The Minimum Feedback Vertex Set problem is NP-Hard on directed graphs [25, GT7], and remains NP-Hard on tournaments [42]. In this section we show that the Minimum Feedback Vertex Set is polynomial on tournaments totally decomposable w.r.t. umodular decomposition.

Another way of considering this problem is to find a minimum set whose removal will result in an acyclic graph. Consequently in tournaments, the problem is equivalent to find the maximum sub-tournament induced, which is transitive. Let us call *minimum degree* $m(T)$ of a tournament T the minimum of minimum in-degree and minimum out-degree.

Lemma 3 *Let x be a vertex such that either $|N^-(x)| = m(T)$ or $|N^+(x)| = m(T)$. If $|N^-(x)| = m(T)$ then $N^-(x)$ is a minimum Feedback Vertex Set of T. And if $|N^+(x)| = m(T)$ then $N^+(x)$ is a minimum Feedback Vertex Set of T.*

Proof: Let us suppose $|N^-(x)| = m(T)$; the other case is the same up to complement. As T is locally transitive, $T[\{x\} \cup N^+(x)]$ is a transitive tournament, so $N^-(x)$ is a feedback vertex set. Conversely let us suppose that T contains a feedback vertex set F of size strictly less than $|N^-(x)|$. Then since we took a minimum degree vertex, for all vertices y, $T[X \setminus F]$ contains vertices from both $N^+(y)$ and $N^-(y)$. So it can not be a transitive tournament (it has no source). □

A direct consequence of this lemma is

Theorem 17 *the Minimum Feedback Vertex Set of a locally transitive tournament can be found in $O(n^2)$-time.*

The algorithm (finding a minimum degree vertex and removing its neighbors) can not be more simple !

6.3.7 Isomorphism

As far as we know, the status of the isomorphism problem is still unknown for tournaments. [4,3]. [11] gave a linear-time algorithm for locally transitive tournaments isomorphism. It is not hard to see that, given the compact encoding given in Section 6.3.5, isomorphism can be tested in $O(n)$ time.

7 Extensions and further developments

We have presented the umodules and homogeneous relations focusing on graph theory field. But umodules may be found in many other objects. Let us briefly present an example.

7.1 Homogeneous relation based on a binary function

Let f be a binary function $X \times X \to Y$. The homogeneous relation *based on* f, written H_f, is defined as $H_f(s|ab)$ if and only if $f(s,a) = f(s,b)$ and

$f(a, s) = f(b, s)$.

For instance on graphs f is the existence of an edge. On directed graph is the existence of an arc. And on a 2-structures $f(x, y)$ is the number of equivalence class of the couple (x, y). It can also be seen as a colouring of the edge (x, y).

Notice that weaker homogeneous relations can be defined from a binary function: the *left* homogeneous relation based on f, H^l_f, is defined as $H^l_f(s|ab)$ if and only if $f(s, a) = f(s, b)$. And the *right* homogeneous relation based on f, H^r_f ,is defined as $H^r_f(s|ab)$ if and only if $f(a, s) = f(b, s)$. But these relation do not have the quotient properties, and have not the same umodules. We have:

Proposition 21 *If M is a umodule for H^r_f and for H^l_f then is a umodule for H_f.*

The proof is immediate from definition. Notice that the converse is not true. For instance for $X = \{a, b, c, d\}$ if $f(a, c) = f(a, d)$, $f(b, c) = f(b, d)$ and all other couples have pairwise different values, then $\{a, b\}$ is a umodule for H_f but neither for H^l_f nor for H^r_f. If f is a symmetric function, then the three homogeneous relations of course are the same. This is true for graphs and for symmetric 2-structures, for instance.

Proposition 22 *The principal ideals of a ring are umodules (w.r.t. its multiplication homogeneous relation).*

7.2 Open problems

In this paper we study umodular decomposition applied to graphs, when the local congruence is 2, the next challenge is now to understand umodular decomposition of directed graphs or directed acyclic graphs, starting with the self-complemented case first.

Our computation of strong umodules is polynomial, but its asymptotic complexity of $O(|X|^5)$ can surely be reduced, especially when applied to particular combinatorial objects.

We have noticed here the great importance of the Seidel switch operation, and following the notion of vertex minor as defined in [37,38], let us called H a *Seidel minor* of a graph G, if H can be obtained from G by the two following operations:

- delete a vertex,
- choose a vertex and do a Seidel switch on this vertex

It could be of interest to study such Seidel minors.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Analysis of Computer algorithms*. Assison-Wesley, 1974.

[2] G. E. Andrews. *The theory of partitions*. Addison-Wesley, 1976. Chapter 4: Compositions and Simon Newcomb's problem.

[3] V. Arvind, B. Das, and P. Mukhopadhyay. On isomorphism and canonization of tournaments and hypertournaments. In *17th International Symposium of Algorithms and Computation (ISAAC)*, volume 4288 of *LNCS*, pages 449–459, 2006.

[4] L. Babai and E. M. Luks. Canonical labeling of graphs. In *15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 171–183, 1983.

[5] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer-Verlag, 2001.

[6] A. Brandstadt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, 1999.

[7] A. E. Brouwer. The enumeration of locally transitive tournaments, 1980. Technical report ZW138 of stichting mathematisch centrum, Amsterdam.

[8] B.-M. Bui-Xuan, M. Habib, V. Limouzy, and F. de Montgolfier. Algorithmic aspects of a general modular decomposition theory. *Special issue of Discrete Applied Mathematics for the 3rd conference on Optimal Discrete Structures and Algorithms (ODSA'06)*, 2006. *to appear*.

[9] B.-M. Bui Xuan, M. Habib, V. Limouzy, and F. de Montgolfier. Homogeneity vs. adjacency: generalising some graph decomposition algorithms. In *32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 4271 of *LNCS*, June 2006.

[10] M. Chein, M. Habib, and M. C. Maurer. Partitive hypergraphs. *Discrete Mathematics*, 37(1):35–50, 1981.

[11] E. Clarou. *Une hiérarchie de forçage pour les tournois indécomposables*. PhD thesis, Université Claude Bernard Lyon I, 1996.

[12] D. A. Cohen, M. C. Cooper, and P. G. Jeavons. Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. Technical Report CS-RR-06-06, Oxford University, 2006.

[13] C. J. Colbourn and D. G. Corneil. On deciding switching equivalence of graphs. *Discrete Applied Mathematics*, 2(3):181–184, 1980.

[14] D. G. Corneil. Private communication. Dagstuhl, 2007.

[15] W. H. Cunningham. *A combinatorial decomposition theory.* PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 1973.

[16] A. Ehrenfeucht, T. Harju, and G. Rozenberg. *The Theory of 2-Structures- A Framework for Decomposition and Transformation of Graphs.* World Scientific, 1999.

[17] A. Ehrenfeucht and G. Rozenberg. Theory of 2-structures. *Theoretical Computer Science*, 3(70):277–342, 1990.

[18] A. Ehrenfreucht, H. N. Gabow, R. M. McConnell, and S. J. Sullivan. An $O(n^2)$ Divide-and-conquer algorithm for the Prime Tree decomposition of 2-structures and the Modular Decomposition of graphs. *Journal of Algorithms*, 16(2):283–294, 1994.

[19] M. G. Everett and S. P. Borgatti. Role colouring a graph. *Mathematical Social Sciences*, 21:183–188, 1991.

[20] J. Fiala and D. Paulusma. The computational complexity of the role assignment problem. In *30th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 817–828, 2003.

[21] H. N. Gabow. A representation for crossing set families with applications to submodular flow problems. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 202–211. ACM/SIAM, 1993.

[22] H. N. Gabow. Centroids, representations, and submodular flows. *Journal of Algorithms*, 18(3):586–628, 1995.

[23] J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari. Modular decomposition of protein-protein interaction networks. *Genome Biology*, 5(8), 2004.

[24] T. Gallai. Transitiv orientierbare Graphen. *Acta Math. Acad. Sci. Hungar.*, 18:25–66, 1967.

[25] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.

[26] M. Habib and M. C. Maurer. 1-intersecting families. *Discrete Mathematics*, 53:91–101, 1985.

[27] R. B. Hayward. Recognizing 3-structure: A switching approach. *Journal of Combinatorial Theory, Serie B*, 66(2):247–262, 1996.

[28] A. Hertz. On perfect switching classes. *Discrete Applied Mathematics*, 94(1-3):3–7, 1999.

[29] W.-L. Hsu and R. M. McConnell. PC-trees and circular-ones arrangements. *Theoretical Computer Science*, 296:99–116, 2003.

[30] J. Kratochvíl, J. Nešetřil, and O. Zýka. On the computational complexity of Seidel's switching. In *Fourth Czechoslovakian Symposium on Combinatorics, Graphs and Complexity (Prachatice, 1990)*, volume 51 of *Ann. Discrete Math.*, pages 161–166. North-Holland, Amsterdam, 1992.

[31] G. Lopez and C. Rauzy. Reconstruction of binary relations from their restrictions of cardinality 2, 3, 4 and (n-1). *Z. Math. Logik Grundlag. Math.*, 38(1):27–37 and 157–168, 1992. in two parts.

[32] R. M. McConnell and F. de Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):189–209, 2005.

[33] 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.

[34] F. de Montgolfier. *Décomposition modulaire des graphes. Théorie, extensions et algorithmes*. PhD thesis, Université Montpellier II, 2003.

[35] F. de Montgolfier and M. Rao. The bi-join decomposition. In *ICGT '05, 7th International Colloquium on Graph Theory*, 2005.

[36] F. de Montgolfier and M. Rao. Bipartitive families and the bi-join decomposition. Technical report, 2005. Submitted.`http://hal.archives-ouvertes.fr/hal-00132862`.

[37] S.-I. Oum. *Graphs Of Bounded Rank Width*. PhD thesis, Princeton University, 2005.

[38] S.-I. Oum. Rank-width and vertex-minors. *Journal of Combinatorial Theory, Series B*, 95(1):79–100, 2005.

[39] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer-Verlag, 2003.

[40] J. J. Seidel. A survey of two-graphs. In *Colloquio Internazionale sulle Teorie Combinatorie (Rome, 1973), Tomo I*, pages 481–511. Atti dei Convegni Lincei, No. 17. Accad. Naz. Lincei, Rome, 1976.

[41] N. J. A. Sloane. The on-line encyclopedia of integer sequences . `http://www.research.att.com/~njas/sequences/`.

[42] E. Speckenmeyer. On feedback problems in digraphs. In *15th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 411 of *LNCS*, pages 218–231, 1989.

[43] D. R. White and K. P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5:193–234, 1983.