Closure of independencies under graphoid properties: some experimental results

Marco Baioletti
Dip. Matematica e Informatica,
Università di Perugia, Italy,
baioletti@dipmat.unipg.it

Giuseppe Busanello **Barbara Vantaggi**
Dip. Metodi e Modelli Matematici,
Università "La Sapienza" Roma, Italy,
e-mail: {busanello, vantaggi}@dmmm.uniroma1.it

Abstract

In this paper we describe an algorithm for computing the closure with respect to graphoid properties of a set of independencies. Since the computation of the complete closure is infeasible, we provide a procedure, called FC1, which is based on a unique inference rule and on the elimination of redundant independencies. FC1 is able to compute a reduced form of the closure, called *fast closure*, which is equivalent to the complete closure, but whose size is much smaller. Some experimental tests have been performed with an implementation of the procedure in order to show the computational behavior of the algorithm. We have also compared the computational cost and the size of the fast closure with the corresponding data for the complete closure.

Keywords. Conditional independence models, Graphoid properties, Inferential rules.

1 Introduction

Conditional independence structures arise in different frameworks, in particular, in probability and in multivariate statistics [11, 14, 15, 18, 20, 23, 31]. It is well known [14] that for any probability measure P the associated independence model $\mathcal{M}$, under the classical definition of independence, is a semi–graphoid (i.e. it satisfies symmetry, decomposition, weak union, contraction) and if P is strictly positive, then $\mathcal{M}$ is a graphoid (also intersection property holds). On the other hand, other independence notions have been introduced in a probabilistic setting [7, 8, 12, 21, 26] and under them graphoid properties have been tested. Moreover, it is well known that graphoid properties are met also by other relations (see [15]) like separation property in graph.

The significance of independence models and graphoid structures is not limited to probabilistic models: in fact many independence models arising from different uncertainty measures are tested on the basis of graphoid properties (see e.g. [1, 9, 10, 13, 15, 16, 17, 19, 23, 27, 30]) and obviously not all the properties among those of graphoid hold.

A significant problem is when a field expert provides an uncertainty measure φ (or better a partial uncertainty assessment, e.g. a coherent conditional probability assessment) and a set J of conditional independence statements, in such case it is necessary to check whether the set J is induced or compatible with φ [29] and then to find all the set of independencies deducible from J.

Then, the aim of this paper is to consider a set J of conditional independence statements, compatible with an uncertainty assessment, and to build in an efficient way the closure through graphoid properties of J.

The computation of the closure is infeasible since its size is exponentially larger than the size of the initial set J of independence statements (see [23, 24]). Then, our aim in [3, 4] (as that in [23, 24] essentially for the case of semi–graphoids) is to build a suitable reduced set of independence statements (obviously included in the closure of J with respect to graphoids), which is as small as possible and it represents the same independence structure. From this reduced set all the relations in the closure should be easily deducible.

In other words, this small set of independence statements, which is called "fast closure", can be considered a basis for the closure.

The computation of the fast closure is relevant also for the selection problem (based essentially on statistical tests) of a model on the basis of data for building, for example, the relevant Bayesian network.

In this paper we describe an algorithm to compute the reduced set. This algorithm is based on a unique inference rule introduced in [4]. In the quoted paper we have also compared this algorithm with another based on two inferential rules, which are deduced from [24] and studied in our previous paper.

An empirical evaluation of the performance of the introduced algorithm is provided by showing computation times and number of iterations, as well as a comparison between the needed time to compute the fast closure and the time for computing the complete closure (the size of both closures is compared).

The paper is organized as follows: in Section 2 some preliminaries concepts about graphoids, closure and implications for independence relations are recalled. In Section 3 we describe the generalized inference rules and the concept of fast closure; while in Section 4 a system based on a unique inference rule and its corresponding algorithm FC1 are introduced. In Section 5 we describe and comment some experimental results.

2 Graphoid structures

Throughout the paper the symbol $\tilde{S} = \{Y_1, \ldots, Y_n\}$ denotes a finite not empty set of variables. Given an uncertainty measure φ, a conditional independence statement $Y_A \perp\!\!\!\perp Y_B | Y_C$ (compatible with φ), where A, B, C are disjoint subsets of the set of indices $S = \{1, \ldots, n\}$, is denoted simply also as an ordered triple (A, B, C).

Let $S^{(3)}$ be the set of triples (A, B, C) of disjoint sets of S such that A and B are not empty, then a conditional independence model, related to an uncertainty measure φ, is a subset of $S^{(3)}$.

In particular, we deal with independence models closed under graphoid properties. We recall that a graphoid is a couple $(S, \mathcal{I})$, where $\mathcal{I}$ is a ternary relation on the set S, which satisfies the following properties:

- G1 if $(A, B, C) \in \mathcal{I}$, then $(B, A, C) \in \mathcal{I}$ (Symmetry);
- G2 if $(A, B, C) \in \mathcal{I}$, then $(A, B', C) \in \mathcal{I}$ for any nonempty subset B' of B (Decomposition);
- G3 if $(A, B_1 \cup B_2, C) \in \mathcal{I}$ with B_1 and B_2 disjoint, then $(A, B_1, C \cup B_2) \in \mathcal{I}$ (Weak Union);
- G4 if $(A, B, C \cup D) \in \mathcal{I}$ and $(A, C, D) \in \mathcal{I}$, then $(A, B \cup C, D) \in \mathcal{I}$ (Contraction);
- G5 if $(A, B, C \cup D) \in \mathcal{I}$ and $(A, C, B \cup D) \in \mathcal{I}$, then $(A, B \cup C, D) \in \mathcal{I}$ (Intersection).

$(S, \mathcal{I})$ is a semi–graphoid if it satisfies only the properties G1–G4.
The symmetric versions of rules G2 and G3 are denoted by

- G2s if $(A, B, C) \in \mathcal{I}$, then $(A', B, C) \in \mathcal{I}$ for any nonempty subset A' of A;
- G3s if $(A_1 \cup A_2, B, C) \in \mathcal{I}$, then $(A_1, B, C \cup A_2) \in \mathcal{I}$.

Let θ, $\theta' \in S^{(3)}$, we denote by

$$\theta \vdash_R \theta'$$

the fact that θ' is obtained by applying once the property R to θ, where in this context R can be G1, G2 or G3.
Moreover, let θ_1, θ_2, $\theta \in S^{(3)}$;

$$\theta_1, \theta_2 \vdash_R \theta$$

denotes that θ is obtained by applying once R to the pair θ_1, θ_2 of triples. In this case R can be either G4 or G5.

Now, we start from a set $J \subset S^{(3)}$ of triples, compatible with an uncertainty measure, and we are interested to establish whether a triple $\theta \in S^{(3)}$ can be derived from J, in symbols

$$J \vdash^* \theta\,.$$

This means that θ can be obtained by applying a finite number of times the rules G1–G5 starting from the set of triples J. This problem is called "implication problem" and has been already studied, for instance, in [32].
A strictly related problem is to compute the closure of a set J, defined as

$$\bar{J} = \{\theta \in S^{(3)} : J \vdash^* \theta\}\,.$$

It is clear that the implication problem can be easily solved once the closure of J has been computed. But the computation of the closure is infeasible because its size is exponentially larger than the size of J.

Then, in the following sections we describe how it is possible to compute a smaller set of triples having the same information as the closure.

This problem has been already faced in [24], with particular attention to semi–graphoid structures.

3 Generalized inference rules

In the following subsections we recall some notions introduced in [2, 4] useful to compute the closure in a more efficient way.

In particular, in Subsection 3.1 a notion of generalized inclusion, that is related to the notion of dominance given in [23] is studied.

In Subsection 3.2 we study some properties of intersection and contraction, which lead to suitable inferential rules. Moreover, we provide a procedure to compute a "small" set that can be considered a sort of basis for the closure, with respect to graphoid, of a given set of conditional independence statements.

3.1 Generalized inclusion

Let us focus our attention, first of all, to the first three graphoid rules. Given a triple $\theta_2 \in S^{(3)}$, it is possible to compute all the triples θ_1 which can be obtained from θ_2 with a finite number of applications of G1, G2 and G3. We say (see [2, 3, 4]) that, for any such pair of triples, θ_1 is *generalized–included* in θ_2 (briefly g–included), in symbol $\theta_1 \sqsubseteq \theta_2$.

In order to simplify the notation in the following, given a triple $\theta_i = (A_i, B_i, C_i)$, X_i stands for $(A_i \cup B_i \cup C_i)$.

Now, some properties of g–inclusion are recalled.

Proposition 1 *Given $\theta_1 = (A_1, B_1, C_1)$ and $\theta_2 = (A_2, B_2, C_2)$, then $\theta_1 \sqsubseteq \theta_2$ if and only if the following conditions hold*

(i) $C_2 \subseteq C_1 \subseteq X_2$;

(ii) either $A_1 \subseteq A_2$ and $B_1 \subseteq B_2$ or $A_1 \subseteq B_2$ and $B_1 \subseteq A_2$.

Generalized inclusion is strictly related to the *partial order relation* $\sqsubseteq_a$ on $S^{(3)}$, defined in [23] and called dominance: the triple $\theta = (A, B, C)$ is said to dominate $\theta' = (A', B', C')$ (in symbol $\theta' \sqsubseteq_a \theta$) if θ' can be derived from θ by means of decomposition, weak union and their symmetric properties (i.e. G2, G3, G2s and G3s).

The relation between $\sqsubseteq$ and $\sqsubseteq_a$ is simple: $\theta' \sqsubseteq \theta$ if and only if

$$\text{either } \theta' \sqsubseteq_a \theta \text{ or } \theta' \sqsubseteq_a \theta^T,$$

where θ^T is the transpose of θ (i.e. if $\theta = (A, B, C)$, then $\theta^T = (B, A, C)$).

The g–inclusion verifies almost all the properties of a partial order relation on $S^{(3)}$ [4], in fact it is reflexive and transitive, but it is not anti–symmetric. However, it satisfies a weak form of anti–symmetry, and denoted by $(AS)^*$:

$$\theta_1 \sqsubseteq \theta_2 \text{ and } \theta_2 \sqsubseteq \theta_1 \text{ implies either } \theta_1 = \theta_2 \text{ or } \theta_1 = \theta_2^T.$$

The definition of g–inclusion between triples can be extended as follows to the case of sets of triples.

Definition 1 *Let H, J be subsets of $S^{(3)}$. J is a covering of H (in symbol $H \sqsubseteq J$) if and only if for any triple $\theta \in H$ there exists a triple $\theta' \in J$ such that $\theta \sqsubseteq \theta'$.*

The g–inclusion between sets of triples verifies reflexivity and transitivity, while as the following example shows it does not satisfy the anti–symmetry neither in its weak form.

Example 1 *Given $S = \{1, 2, 3, 4\}$, consider the triples $\theta = (\{1\}, \{2\}, \{3\})$, $\theta' = (\{1, 4\}, \{2\}, \{3\}) \in S^{(3)}$ and the subsets $H = \{\theta, \theta'\}$ and $J = \{\theta'\}$ of $S^{(3)}$. It is easy to check that $H \sqsubseteq J$ and $J \sqsubseteq H$, but $\theta \in H$ is such that $\theta \notin J$ and $\theta^T \notin J$.*

However, in [3] we show that weak anti–symmetry holds for particular sets.

3.2 Closure through the generalization of G4 and G5

Now, we recall the two inference rules introduced in [2, 3].

Given $\theta_1, \theta_2 \in S^{(3)}$, $W_C(\theta_1, \theta_2)$ is the set

$$\{\tau : \theta'_1, \theta'_2 \vdash_{G4} \tau, \text{ with } \theta'_1 \sqsubseteq_a \theta_1, \theta'_2 \sqsubseteq_a \theta_2\}.$$

Concerning $W_C(\theta_1, \theta_2)$ the following result holds (see [3, 4]).

Proposition 2 *Let $\theta_1 = (A_1, B_1, C_1)$, $\theta_2 = (A_2, B_2, C_2)$ be a pair of triples belonging to $S^{(3)}$, then*

1. *$W_C(\theta_1, \theta_2)$ is not empty if and only if all the following five conditions hold:*
   - *(a)* $A_1 \cap A_2 \neq \emptyset$;
   - *(b)* $C_1 \subseteq X_2$ *and* $C_2 \subseteq X_1$;
   - *(c)* $B_1 \setminus C_2 \neq \emptyset$;
   - *(d)* $B_2 \cap X_1 \neq \emptyset$;
   - *(e)* $|(B_1 \setminus C_2) \cup (B_2 \cap X_1)| \geq 2$.
2. *If $W_C(\theta_1, \theta_2)$ is not empty the triple $gc(\theta_1, \theta_2) =$*

   $$(A_1 \cap A_2, (B_1 \setminus C_2) \cup (B_2 \cap X_1), C_2 \cup (A_2 \cap C_1)),$$

   is in $W_C(\theta_1, \theta_2)$ and dominates any triple belonging to $W_C(\theta_1, \theta_2)$.

When $W_C(\theta_1, \theta_2)$ is empty, we set $gc(\theta_1, \theta_2) = \perp$.

The function $gc(\cdot, \cdot)$ has already been introduced in [24] in an essentially equivalent form.

The conditions *(a)–(e)*, which assure that $W_C(\theta_1, \theta_2)$ is not empty, are however stronger than those given in [24]: in fact, we are looking for the triple dominating all the triples obtained, through G4, from θ_1 and θ_2 or from some of their dominated triples. This is clarified in the next example.

Example 2 *Consider the triples*

$$\theta_1 = (\{1,4\},\{2\},\{3\})$$

and

$$\theta_2 = (\{1,3\},\{2\},\{4\}).$$

The condition (e) fails, since $(B_1 \setminus C_2) = (B_2 \cap X_1)$ *and it contains just the element* 2.

Then, in this case $W_C(\theta_1,\theta_2) = \emptyset$*, however it could be noted that by applying G3 to one of the two triples we get* $\theta = (\{1\},\{2\},\{3,4\}) \sqsubseteq_a \theta_i$ *(for* $i = 1,2$*) and so* θ *adds no further information.*

We denote with $GC(\theta_1,\theta_2)$ the set formed by the possible (i.e. belonging to $S^{(3)}$) triples among $gc(\theta_1,\theta_2)$, $gc(\theta_1,\theta_2^T)$, $gc(\theta_1^T,\theta_2)$ and $gc(\theta_1^T,\theta_2^T)$.

Obviously, $GC(\theta_1,\theta_2)$ is in general different from $GC(\theta_2,\theta_1)$.

Note if $\theta_1,\theta_2 \vdash_{G4} \tau$, then $\tau = gc(\theta_1,\theta_2)$.

A result similar to Proposition 2, related to intersection property, holds (see [3]) by considering the set

$$W_I(\theta_1,\theta_2) = \{\tau : \theta_1',\theta_2' \vdash_{G5} \tau, \text{ with } \theta_1' \sqsubseteq_a \theta_1, \theta_2' \sqsubseteq_a \theta_2\}.$$

Proposition 3 *Let* $\theta_1 = (A_1,B_1,C_1)$*,* $\theta_2 = (A_2,B_2,C_2)$ *be a pair of triples belonging to* $S^{(3)}$*, then*

1. $W_I(\theta_1,\theta_2)$ *is not empty if and only if all the following five conditions hold:*
   - (a) $A_1 \cap A_2 \neq \emptyset$;
   - (b) $C_1 \subseteq X_2$ *and* $C_2 \subseteq X_1$;
   - (c) $B_1 \cap X_2 \neq \emptyset$;
   - (d) $B_2 \cap X_1 \neq \emptyset$;
   - (e) $|(B_1 \cap X_2) \cup (B_2 \cap X_1)| \geq 2$.
2. *If* $W_I(\theta_1,\theta_2)$ *is not empty, then the triple* $gi(\theta_1,\theta_2) = (A_{gi},B_{gi},C_{gi})$ *with*
   - $A_{gi} = A_1 \cap A_2$;
   - $B_{gi} = (B_1 \cap X_2) \cup (B_2 \cap X_1)$;
   - $C_{gi} = (C_1 \cap A_2) \cup (C_2 \cap A_1) \cup (C_2 \cap C_1)$;

   is in $W_I(\theta_1,\theta_2)$ *and dominates any triple belonging to* $W_I(\theta_1,\theta_2)$.

Given two triples θ_1, θ_2, Proposition 3 gives rise to the dominant triple generated, through G5, by θ_1, θ_2 or by some dominated triples, respectively, by θ_1 and θ_2.

The set $GI(\theta_1,\theta_2)$ is formed by the possible (i.e. belonging to $S^{(3)}$) triples among $gi(\theta_1,\theta_2)$, $gi(\theta_1,\theta_2^T)$, $gi(\theta_1^T,\theta_2)$ and $gi(\theta_1^T,\theta_2^T)$.

Then, $GI(\theta_1,\theta_2) = GI(\theta_2,\theta_1)$.

Also in this case, if $\theta_1,\theta_2 \vdash_{G5} \tau$, then $\tau = gi(\theta_1,\theta_2)$.

The previous sets GC and GI are used to introduce two new inference rules

- G4* "generalized contraction": from θ_1,θ_2 deduce any triple $\tau \in GC(\theta_1,\theta_2)$;
- G5* "generalized intersection": from θ_1,θ_2 deduce any triple $\tau \in GI(\theta_1,\theta_2)$;

which, as explained above, generalize the two classical inference rules. These rules are useful to compute the closure of a set J of triples in $S^{(3)}$, that is

$$J^* = \{\tau : J \vdash_G^* \tau\} \quad (1)$$

where $J \vdash_G^* \tau$ means that τ is obtained by applying a finite number of times the rules G4* and G5*.

In [3, 4] the relationship between the two closures J^* and $\bar{J}$ is studied, in particular, we prove that any triple obtained through G1–G5 is g–included in a triple deduced from G4* and G5*. This implies that $J^* \subseteq \bar{J}$ and moreover

$$\bar{J} \sqsubseteq J^*.$$

Note that J^* is a subset of $\bar{J}$, so even if J^* has the same information of $\bar{J}$, is smaller than $\bar{J}$. Actually, J^* contains some "redundant" triples, that means that are g–included in some of the other ones. In fact, (see (1)) each application G4* and G5* can generate a triple which is g–included in a triple of J or in an already generated triple.

3.3 Fast closure

In [2, 3, 4] we introduced the concept of "maximal"(with respect to g–inclusion) triple: given a set J of triples, a triple τ is maximal in J if there exists no $\bar{\tau} \in J$ with $\bar{\tau} \neq \tau, \tau^T$ such that $\tau \sqsubseteq \bar{\tau}$.

We denote with $J_{/\sqsubseteq}$ the subset of J composed only by its maximal triples and we call FindMaximal the function which computes $J_{/\sqsubseteq}$ from J.

There is no loss of information by using $J_{/\sqsubseteq}$ instead of J [3], in fact

$$J \sqsubseteq J_{/\sqsubseteq}.$$

Then, given a set J of triples in $S^{(3)}$, we compute J^* (see equation (1)) and then we take only its maximal triples, i.e. $J^*_{/\sqsubseteq}$.

We call the set $J^*_{/\sqsubseteq}$ "fast closure" and we denote it, for simplicity, with J_*.

Note that we have also the following relationship: $J_* \subseteq \bar{J}$ and

$$\bar{J} \sqsubseteq J_*.$$

It is interesting to observe $\bar{J}_{/\sqsubseteq}$ and J_* essentially coincide [3], in fact

$$\bar{J}_{/\sqsubseteq} \sqsubseteq J_* \qquad \text{and} \qquad J_* \sqsubseteq \bar{J}_{/\sqsubseteq}.$$

4 Unique inference rule

In [3, 4] we describe a procedure to compute efficiently the closure of a set of conditional independence statements, which is based on the two above inferential rules (generalized contraction and intersection). In order to improve such procedure, in we look for a unique inferential rule with the aim of simplifying the procedure.

In particular, by taking into account Proposition 2 and Proposition 3, which provide necessary and sufficient conditions for the application of generalized contraction and intersection, respectively, the notion of almost complete pair of triples is introduced in [4] in order to characterize the couples of triples which lead to the largest fast closure.

We recall first of all that the fast closure $\{\theta_1, \theta_2\}_*$ of a couple $\theta_1, \theta_2 \in S^{(3)}$ is composed by a maximum of nine extra triples, no matter how many variables occur in θ_1 and θ_2.

In particular, any pair of triples (θ_1, θ_2) can be re–written, in a general form, as

$$\begin{aligned} \theta_1 &= ([A_A, A_B, A_C, A_N], [B_A, B_B, B_C, B_N], \\ &\quad [C_A, C_B, C_C, C_N]) \\ \theta_2 &= ([A_A, B_A, C_A, A'_N], [A_B, B_B, C_B, B'_N], \\ &\quad [A_C, B_C, C_C, C'_N]) \end{aligned}$$

where some sets can be empty and with the notation that $[A, B, C]$ stands for $A \cup B \cup C$.

Each triple of the fast closure of (θ_1, θ_2) is g–included in the set of possible (i.e. belonging to $S^{(3)}$) triples

$$K(\theta_1, \theta_2) = \{\theta_1, \theta_2, \theta_a, \theta_b, \theta_c, \theta_d, \theta_e, \theta_f, \theta_g, \theta_h, \theta_{ad}\}$$

where

$\theta_a = (A_A, [A_B, B_A, B_B, B_C, C_B, B_N], [A_C, C_A, C_C]);$

$\theta_b = (A_B, [A_A, B_A, B_B, B_C, C_A, B_N], [A_C, C_B, C_C]);$

$\theta_c = (B_A, [A_A, A_B, A_C, B_B, C_B, A_N], [B_C, C_A, C_C]);$

$\theta_d = (B_B, [A_A, A_B, A_C, B_A, C_A, A_N], [B_C, C_B, C_C]);$

$\theta_e = (A_A, [A_B, B_A, B_B, B_C, C_B, B'_N], [A_C, C_A, C_C]);$

$\theta_f = (A_B, [A_A, B_A, B_B, B_C, C_A, A'_N], [A_C, C_B, C_C]);$

$\theta_g = (B_A, [A_A, A_B, A_C, B_B, C_B, B'_N], [B_C, C_A, C_C]);$

$\theta_h = (B_B, [A_A, A_B, A_C, B_A, C_A, A'_N], [B_C, C_B, C_C]);$

$\theta_{ad} = ([A_B, B_A], [A_A, B_B], [A_C, B_C, C_A, C_B, C_C]).$

Therefore,

$$\{\theta_1, \theta_2\}_* \sqsubseteq K(\theta_1, \theta_2).$$

Moreover, in [3, 4] it is also proved that

$$K(\theta_1, \theta_2) \sqsubseteq \{\theta_1, \theta_2\}_*.$$

Note that in general $K(\theta_1, \theta_2)$ may not coincide with $\{\theta_1, \theta_2\}_*$ because it could contain some redundant triple or the transpose triple of one belonging to $\{\theta_1, \theta_2\}_*$.

However, it is easy to see that

$$K(\theta_1, \theta_2)_{/\sqsubseteq} \sqsubseteq \{\theta_1, \theta_2\}_*$$

and

$$\{\theta_1, \theta_2\}_* \sqsubseteq K(\theta_1, \theta_2)_{/\sqsubseteq},$$

since both sets are maximal.

Therefore the set $K(\theta_1, \theta_2)$ allows to compute $\{\theta_1, \theta_2\}_*$: in fact, it is possible to build up such a set and apply the function FindMaximal to it.

All this computation requires a constant number of steps with respect to the size of θ_1, θ_2.

By using $\{\theta_1, \theta_2\}_*$, it is possible to provide a new inference rule

U : from θ_1, θ_2 deduce any triple $\tau \in \{\theta_1, \theta_2\}_*$.

4.1 Algorithm FC1

By using the unique inference rule U, we provided the Algorithm 1.

Concerning the above algorithm we have the following result:

Theorem 1 *Let J be a nonempty subset of $S^{(3)}$, then*

1. *FC1(J) $\sqsubseteq J_*$;*
2. *$J_* \sqsubseteq$ FC1(J).*

Both theoretical and empirical comparisons between FC1 and an algorithm based on two inferential rules in [4] are carried out, hereby showing the better performances of FC1.

Algorithm 1 Fast closure by U

function FC1(J)
  $J_0 \leftarrow J$
  $N_0 \leftarrow J$
  $k \leftarrow 0$
  repeat
    $k \leftarrow k + 1$
    $N_k := \bigcup_{\theta_1 \in J_{k-1}, \theta_2 \in N_{k-1}} \{\theta_1, \theta_2\}_*$
    $J_k \leftarrow$ FindMaximal$(J_{k-1} \cup N_k)$
  until $J_k = J_{k-1}$
  return J_k
end function

Note that FC1 can be optimized by observing that if θ'_1 and θ'_2 belong to $\{\theta_1, \theta_2\}_*$, then $\{\theta'_1, \theta'_2\}_*$ is g–included to $\{\theta_1, \theta_2\}_*$. The validity of this observation follows easily since

$$\{\theta'_1, \theta'_2\}_* \sqsubseteq \{\theta'_1, \theta'_2\}^* \sqsubseteq \{\theta_1, \theta_2\}^* \sqsubseteq \{\theta_1, \theta_2\}_*.$$

Therefore, it is not necessary to apply the inference rule U to a pair of triples θ'_1 and θ'_2, generated by U from the same two triples θ_1 and θ_2, since from θ'_1 and θ'_2 we would obtain only redundant triples, which would be discarded by the function FindMaximal.

Note that for the same reasons, we do not need to apply the rule U between a triple θ and another one θ' generated from θ (by combining θ with another triple θ''): in fact if $\theta' \in \{\theta, \theta''\}_*$, then $\{\theta, \theta'\} \subseteq \{\theta, \theta''\}^*$ and so

$$\{\theta, \theta'\}_* \sqsubseteq \{\theta, \theta''\}_*,$$

which implies that no maximal triple can be obtained.

Then, the use of the inference rule U in FC1 can be enhanced by keeping track of the "parents" of each triple and by neglecting the pairs which satisfies the two previously described situations ("sibling" triples and "father–child").

In our implementation, we use this optimization, but we consider $K(\theta_1, \theta_2)$ instead of $\{\theta_1, \theta_2\}_*$, because in any case in each cycle of FC1 a call to function FindMaximal is however performed.

5 Experimental results

In this section we describe some experimental results obtained with an implementation in C++ of the algorithm FC1, as well as an implementation of an algorithm to compute the complete closure (with respect to G1–G5). The main purpose of these experiments is to prove the viability of the fast closure computation.

The first aspect, that these experiments can clarify, is to show how difficult it is, from the computational point of view, to compute the fast closure. It is clear that this problem is a computationally hard problem, for which no efficient (i.e. polynomial time) solution can exist as already noted in [23, 24].

Therefore an empirical evaluation is necessary in order to establish whether the computation of the fast closure is reasonably fast and uses an acceptable amount of memory.

The other question is which is the quantitative difference in size and in computation time of the fast closure with respect to the complete closure. The fast closure is clearly smaller than the complete closure (each triple $\theta \in J_*$ corresponds to several triple in $\bar{J}$), but we have not been able to find any theoretical bounds for the size of J_* with respect to the size of $\bar{J}$.

The experiments were performed on an AMD Dual Core Opteron running at 1.8 GHz with 2 GByte main memory. We applied a cut–off of 5,000,000 triples that can be stored (to avoid problems with memory) and a time–out of 3600 seconds. Some preliminary results, with different experimental parameters, have already been given in [6, 2].

In the first set of experiments, we have generated 200 random sets of triples having nv variables and nr triples, for $nr = 10, 15, 20, 25, 30$ and $nv = \lfloor 0.5 \cdot nr \rfloor, nr, \lfloor 1.5 \cdot nr \rfloor, 2nr$. and we have computed the fast closure by means of (see Table 1).

In the Table 1, the value *perc* is the percentage of the sets for which FC1 has been able to compute the fast closure, within the limits of time and memory, *time* is the average computation times in seconds, *size* is the average size of the fast closure, *iter* is the average number of iterations needed to find the closure, and *gen* is the average number (rounded to the nearest integer) of the overall generated triples.

The behavior of FC1, as explained in the following, is influenced by many factors, which can have contradictory and not well understandable effects. However it is possible to observe that as nr grows, instances with a small value for $\frac{nv}{nr}$ become more and more difficult: with $nr = 30$ and $nv = 15$ FC1 has not been able to solve any instance. The same happens with $nr = 35$ and 40 (nv being $0.5 \cdot nr$), in experimental tests not described here.

On the other hand, when the ratio $\frac{nv}{nr}$ is large, instances get easier and easier to solve.

The first behavior can be explained with the fact that generating at random an instance with fewer variables, with respect to the number of relations, can produce many triples to which it is possible to repeatedly apply the generalized inference rules. In these

Table 1: Fast Closure FC1

nr	nv	perc	time	size	iter.	gen.
10	5	100	0	10.83	3.99	202
10	10	100	1.06	95.93	6.42	27524
10	15	99	44.43	226.08	6.263	241219
10	20	98.5	22.16	153.54	4.81	115006
15	7	100	9.11E-02	46.84	5.50	5841
15	15	63	500.42	982.68	10.03	1926990
15	22	80.5	111.49	365.29	6.63	359213
15	30	98	9.77	72.14	3.25	32615
20	10	100	79.19	433.835	7.41	652608
20	20	27.5	376.43	921.47	10.2	1105693
20	30	93.5	84.64	305.21	5.58	240052
20	40	98.5	3.64	54.95	2.20	16514
25	12	49.5	1383.23	1354.33	8.3	5231558
25	25	35	254.46	719.69	9.04	720993
25	37	97.5	14.25	124.42	3.8	62761
25	50	100	1.1E-03	29.685	1.445	84
30	15	0	–	–	–	–
30	30	51.28	118.59	514.58	7.65	3631898
30	45	100	0.03	48.38	2.41	1063
30	60	100	8.55E-05	31.06	1.12	7

cases, the computation of the fast closure requires several iterations during which a large number of triples are generated (most of them are discarded). These kinds of instances seem to be the hardest to solve, if compared to the other kinds.

At the same time, if the number of variables is too large, the chance of application of the inference rules becomes very low, as proved by the average size of the fast closure (which is roughly similar to nr) and the number of generated triples (which is rather small). In these cases, the closure often coincides or is similar to the initial set of triples and therefore can be computed with a little computational effort.

In the second set of experiments we compare the computation time needed for finding the complete closure and its size with respect to the time and size of the fast closure. The complete closure is obtained by using an algorithm similar to FC1, which uses all the inference rules G1–G5, without calling FindMaximal. Furthermore, we did not apply for it any cut–off with respect to number of triples.

Since we expect that the complete closure is much larger than its fast version, we have performed these new experiments with smaller instances, instead of using the previous one. In particular, we generate 20 sets of nr triples and nv variables, for $nr = 4, 7, 10$ and $nv = nr, \lfloor 1.5 \cdot nr \rfloor$.

In Table 2 the results for the fast closure are reported, with the average values calculated with respect to the solved instances by FC1, the average computation time is negligible, except that in the last row, where we obtain results similar in magnitude order, as those displayed in Table 1. The algorithm FC1 has been able to build the closure for each instance.

Table 2: Fast Closure with FC1

nr	nv	time	size	iter.	gen.
4	4	0	3.95	2.75	12.1
4	6	0	5.85	2.95	29.2
7	7	2E-03	18.65	4.95	559.25
7	10	1.8E-02	32.05	4.7	1756.15
10	10	0.6755	86.9	5.95	18415
10	15	42.7225	320.45	6.7	335910.5

In Table 3 we report the results obtained in the computation of the complete closure. The last column contains the number of instances for which the algorithm has been able to compute the complete closure within an hour of computation. Note that with $nr = 10$ and $nv = 15$ we could solve only one instance, which almost reached the time limit, while the fast closure of this instance has only 27 triples and has been found in a negligible amount of time. The values in the last column are used to compute the average values showed in Table 2.

The comparison of the size between fast and complete closure is impressive, as it is possible to see in the graph of Figure 1 (the last rows of both tables have been ignored).

Table 3: Complete Closure

nr	nv	time	size	iter.	gen.	res.
4	4	0	64	7	57	20
4	6	0.05	527	8.9	899	20
7	7	1.75	3282	13.15	9526	20
7	10	248	28808	13.89	147249	19
10	10	603	50760	16.67	268381	15
10	15	3513	159164	14	683991	1

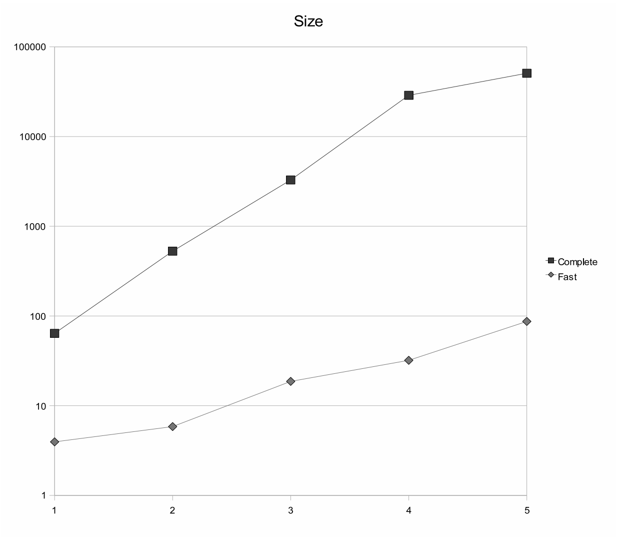

Figure 1: Sizes of the closure

Clearly also the computation times for computing the complete closure are much higher than the time needed to compute the fast closure, as displayed in the Figure 2.

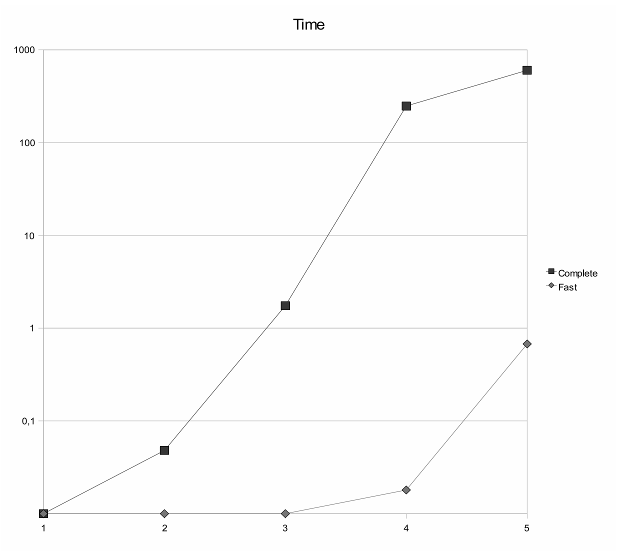

Figure 2: Computation times

6 Conclusions

We study some properties of graphoid structures with the aim to compute efficiently the closure of a set of conditional independence statements. It is well known that the size of the closure of a set is exponentially greater than the size of the given set.

In particular, we give an algorithm FC1, which is able to compute the closure of a set of triples by looking for a suitable subset of the closure, that has the same information, but it is smaller than the closure as shown by experimental results. Actually, FC1 computes just the maximal (with respect to g–inclusion) triples, then it also allows to improve the computational time.

By means of this set also the well known implication problem can be solved in an efficient way: in fact, to verify whether a triple belongs to the closure it is enough to look for a triple in the set, obtained through FC1, which g–includes the given triple. Moreover, to check the g–inclusion relation requires constant time, therefore the computational time is linear with respect to the size of the set.

A straightforward extension of this work is to adapt this framework for computing the closure by using semi-graphoid axioms and compare it with that proposed in [24].

A further open problem, partially studied in [5], consists into using this set for building in an efficient way an acyclic directed graph representing the independence statements in the closure.

References

[1] B.N. Amor, S. Benferhat. Graphoid properties of qualitative possibilistic independence relations. *Inter. Jour. Uncertainty Fuzziness Knowledge-Based Systems*, 13(1):59–96, 2005.

[2] M. Baioletti, G. Busanello, B. Vantaggi. Algorithms for the closure of graphoid structures *Proc. of 12th Inter. Conf. IPMU 2008*, Malaga, 899-906, 2008.

[3] M. Baioletti, G. Busanello, B. Vantaggi. Conditional independence structure and its closure: inferential rules and algorithms. In *Technical Report*, 5/2009 of University of Perugia.

[4] M. Baioletti, G. Busanello, B. Vantaggi. Conditional independence structure and its closure: inferential rules and algorithms. *International Journal of Approximate Reasoning*, in press doi: 10.1016/j.ijar.2009.05.002.

[5] M. Baioletti, G. Busanello, B. Vantaggi. Acyclic Directed Graphs to Represent Conditional Independence Models. *Proc. of ECSQARU 2009. Lecture Notes* LNAI 5590, 530541, 2009.

[6] G. Busanello (2008). Probabilistic conditional independence models: closure and construction of acyclic directed graphs. PhD Thesis.

[7] G. Coletti, R. Scozzafava. Zero probabilities in stochastical independence. In *Information, Uncertainty, Fusion, Kluwer Academic Publishers, Dordrecht*, B. Bouchon- Meunier, R.R. Yager, L.A. Zadeh (Eds.), 185–196, 2000.

[8] G. Coletti, R. Scozzafava. *Probabilistic logic in a coherent setting*. Dordrecht/Boston/London: Kluwer (Trends in logic n.15), 2002.

[9] G. Coletti, B. Vantaggi. Possibility theory: conditional independence. *Fuzzy Sets and Systems*, 157:1491–1513, 2006.

[10] I. Couso, S. Moral, P.Walley. A survey of concepts of independence for imprecise probabilities. *Risk, Decision and Policy*, 5:165–181, 2000.

[11] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, D.J. Spiegelhalter. Probabilistic Networks and Expert Systems, Springer-Verlag, New York, 1999.

[12] F.G. Cozman, T. Seidenfeld. Independence for full conditional measures, graphoids and Bayesian networks, *Boletim BT/PMR/0711 Escola Politecnica da Universidade de Sao Paulo*, Sao Paulo, Brazil (To appear at Benedikt Lwe, Eric Pacuit, Jan-Willem Romeijn (eds.), Foundations of the Formal Sciences VI - Reasoning about Probabilities and Probabilistic Reasoning), 2007.

[13] F. G. Cozman, P. Walley. Graphoid properties of epistemic irrelevance and independence. *Annals of Mathematics and Artificial Intelligence*, 45: 173–195, 2005.

[14] A.P. Dawid. Conditional independence in statistical theory. *J. Roy. Stat. Soc. B*, 41:15–31, 1979.

[15] A.P. Dawid. Separoids: a mathematical framework for conditional independence and irrelevance. Representations of uncertainty. *Annals of Mathematics and Artificial Intelligence*, 32:335–372, 2001.

[16] L.M. de Campos, J.F. Huete. Independence concepts in possibility theory. I. *Fuzzy Sets and Systems*, 103:127–152, 1999.

[17] L. de Campos, S. Moral. Independence Concepts for Convex Sets of Probabilities", *Proc. XI Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, 108–115, 1995.

[18] S.L. Lauritzen. *Graphical models.* Clarendon Press, Oxford, 1996.

[19] S. Moral, A. Cano. Strong conditional independence for credal sets. *Annals of Mathematics and Artificial Intelligence*, 35:295–321, 2002.

[20] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, Los Altos, CA, 1988.

[21] E. San Martìn, M. Mouchart, J. Rolin (2005). Ignorable common information, null sets and Basu's first theorem. *Sankhya*, 67:674–698, 2005.

[22] P.P. Shenoy. Conditional independence in valuation-based systems. *International Journal of Approximate Reasoning*, 10:203-234, 1994.

[23] M. Studený. Semigraphoids and structures of probabilistic conditional independence. *Ann. Math. Artif. Intell.*, 21:71–98, 1997.

[24] M. Studený. Complexity of structural models. *Proc. Prague Stochastics '98*, Prague, 521–528, 1998.

[25] M. Studený, R.R. Bouckaert (1998). On chain graph models for description of conditional independence structures. *Ann. Statist.*, 26(4):1434–1495, 1998.

[26] B. Vantaggi. Conditional independence in a coherent setting. *Ann. Math. Artif. Intell.*, 32:287–313, 2001.

[27] B. Vantaggi. Conditional independence structures and graphical models. *Int. J. Uncertain. Fuzziness Knowledge-Based Systems*, 11(5): 545–571, 2003.

[28] B. Vantaggi. Qualitative Bayesian networks with logical constraints. *Lecture Notes in Computer Sciences*, 2711, Springer, New York, 100–112, 2003.

[29] B. Vantaggi. The role of coherence for handling probabilistic evaluations and independence. *Soft Computing*, Springer, Berlin / Heidelberg, 69(8):617–628, 2005.

[30] J. Veinarova. Conditional Independence Relations in Possibility Theory. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 8(3):253–269, 2000.

[31] J. Witthaker. *Graphical models in applied multivariate statistic*, Wiley, New York, 1990.

[32] S.K.M. Wong, C.J. Butz, D. Wu. On the Implication Problem for Probabilistic Conditional Independency. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 30(6):785–805, 2000.