Computer Science Department Technical Report
Cognitive Systems Laboratory
University of California
Los Angeles, CA 90024-1596

CLOSURE ALGORITHMS AND DECISION PROBLEMS FOR GRAPHOIDS GENERATED BY TWO UNDIRECTED GRAPHS - ABRIDGED VERSION

Azaria Paz
Robert Schulhoff

November 1988
CSD-880096

TECHNICAL REPORT
R-118
September 1988

Closure Algorithms and Decision Problems for Graphoids Generated by Two Undirected Graphs - Abridged Version

Azaria Paz* & Robert Schulhoff†
Cognitive Systems Laboratory
UCLA Computer Science Department
Los Angeles, CA 90024-1596
and
Technion IIT - Haifa Israel

Abstract: Let R and B be two given undirected graphs and let M be the set of all independencies which are implied by R and B under closure by the 5 graphoid axioms (defined in the text). The following problems are solved, in polynomial time and constructively, for M.

1. Are these elements in M which are implied from R and B by contraction and not by the other axioms?

2. Given an independency t is $t \subset M$?

It is also shown by an infinite set of counterexamples that in some cases the representation of M by a set of graphs may require, necessarily, an exponential number of graphs.

* Contribution of first author was supported in part by the National Science Foundation, Grant #DCR 85-01234.

† The results shown in this report are part of the MSC Thesis of the second author done under the supervision of the first author - to be presented to Graduate School of the Technion, Israel Institute of Technology.

1. Preliminaries & Motivation

Graphoids have been introduced in the literature as a means for representing probabilistic knowledge in graphical form [4]. Let $U = \{a, b, c, ...\}$ be a finite universe of random variables, each variable with its own domain. Consider a (polynomial) set of triplets $M = \{(x, z, y)\}$ where x, z, y are mutually disjoint sets from U. A triplet (x, z, y) as above can be given the interpretation:

> "the random variables in x are independent on the random variables in y given any instantiation of the random variables in z".

If all the information we want to extract from a probabilistic distribution is concerned with the interdependence of its random variables we might be able to **represent** that information in the form of a relation M as above and then **record** that information in some compact graphical form.

With regard to the **representation** part, consider the axioms below, defined over any set M of triplets.

(1)	$(xx, z, y) \rightarrow (y, z, x)$:	Symmetry
(2)	$(x, z, yw) \rightarrow (x, z, y) \wedge (x, z, w)$:	Decomposition
(3)	$(x, z, yw) \rightarrow (x, zy, w)$:	Weak Union
(4)	$(x, z, y) \wedge (x, zy, w) \rightarrow (x, z, yw)$:	Contraction
(5)	$(x, zy, w) \wedge (x, zw, y) \rightarrow (x, z, yw)$:	Intersection

It has been shown that any M relation induced by a PD (probabilistic distribution) with strictly positive entries satisfies all the above 5 axioms, while if not all the entries of the PD are strictly positive then the first 4 axioms are always satisfied [4].

Ideally we would be able to represent every independency relation, induced by a PD, as the closure under the above set of axioms of some set of triplets M of polynomial size.

Unfortunately, this is not possible, and for 2 reasons: First, it has been shown lately that PD's do satisfy additional axioms independent on the 5 axioms specified above so that the closure of M under the 5 mentioned axioms might not be enough [2]. Second, it has been shown by combinational arguments that it is impossible to fully specify all the possible PD induced relations in polynomial space (in the size of the universe U) [6].

Still, it is reasonable to assume that many interesting PD induced relations can be fully or approximately represented in the above form (the larger the 'basis' M, the better the approximation) where the approximation is defined in terms of I-mapness: $(x, z, y) \subset$ closure M implies that the independency (x, z, y) holds in the represented PD, but not necessarily the other way around.

As far as the **recording** of a relation M in graphical form is concerned we can use the following graphical model.

Let $G = \{G_1, G_2, \ldots G_K\}$ be a set of graphs whose vertices are elements from the set U of random variables. Let $\{<x \mid z \mid y>\}$ be the set of all triplets such that there is a graph G_i in G whose vertex set contains the vertices in x, y, z and such that the set of vertices z is a cutset between the vertices in x and the vertices in y in G_i. If there is a $1-1$ correspondence between G and M then G perfectly represents M. We will say that G is closed under a given set of axioms if the relation represented by G is closed under that set of axioms.

The use of graphs for recording purposes is motivated by the fact that graphs are easy to record in computer memory and are very compact, in the sense that a single graph with n vertices can represent exponentially many triplets of the form $<x \mid z \mid y>$.

Consider the axioms below

6. $(x, z, y) \rightarrow (x, zw, y),\ w \subset U$: Strong Union

7. $(x, z, y) \rightarrow (x, z, \gamma) \vee (\gamma, z, y),\ \gamma \in U$: Transitivity

the following properties have been shown:

> A relation M is closed under the first 3 axioms iff it can be perfectly represented by a set of graphs [1].
>
> A relation M is closed under axioms 1, 2, 3, 6, 7 iff it can be perfectly represented by a single graph [3].
>
> If the relation M above is specified as the closure of a polynomial set M' under the corresponding set of axioms then polynomial decision algorithms, for ascertaining whether a given triplet is in the relation, are easy to construct for the above 2 cases.

The first of the above mentioned results is a particular case of a result proved in [2]:

A relation M is closed under the axioms 1, 2, 3 and 5 iff it can be represented in a set of graphs which is closed under the '$\otimes$' graph operations (the $\otimes$ operation will be defined in the sequel).

It follows from [2] that even thought the closure of a polynomial set M under the axioms 1, 2, 3 and 5 may require exponentially many graphs for its full representation, still, a simple polynomial algorithm for deciding whether a given triplet (x, z, y) is in the closure can be easily constructed.

A relation M which is closed under the first 5 axioms was denoted in [4] by the term Graphoid. The purpose of this paper is to extend the results in [2] to Graphoids. It will become clear in the sequel that such an extension is far from trivial. We have therefore restricted our goal to the case where the Graphoid is given as the closure under the first 5 axioms of a relation M which is perfectly represented by 2 graphs only, to be called R (red) and B (blue).

2. Notations and Definitions

$G_i(V_i)$ is a graph over V_i.

If α, β, are elements of V_i then "(α, β) is a *nonedge* of G_i" means that (α, β) is not in the edge set of G_i.

For $V_j \subsetneq V_i$

define (α, β) as a *nonedge* of G_i mod V_j if α is not connected to β in $G_i(V_i/V_j)$ i.e., removing the vertices V_j and the incident edges from $G_i(V_i)$ will render α and β disconnected.

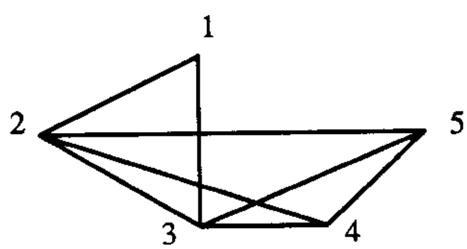
Given 2 graphs $G_i(V_i)$ and $G_j(V_j)$ let $V_k = V_i \cap V_j$ and assume $V_k \neq \phi$. Define the graph $G_k(V_k) = G_i \otimes G_j$ as follows:

1. Every edge of G_i and G_j over V_k is an edge of $G_k(V_k)$

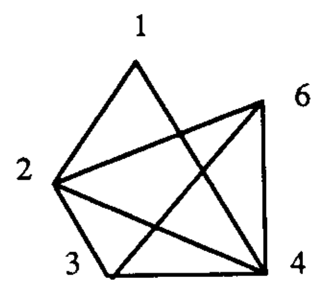
2. Every pair (α, β) over V_k which is a nonedge of G_i mod $(V_k - \alpha - \beta)$ *or* is a nonedge of G_j mod $(V_k - \alpha - \beta)$ is a nonedge of G_k

3. Every pair (α, β) to which 1 or 2 above does not apply is an edge of G_k

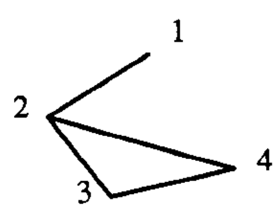
Example 1.



G_i



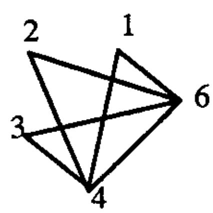
G_j



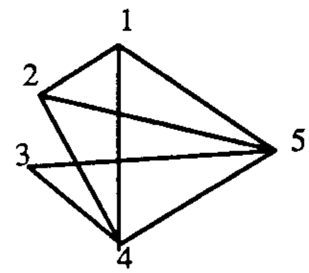
$G_k = G_i \otimes G_j$

G_k is not implied by either G_i nor G_j e.g. (1,2,3,4), (1,2,3) and (1,2,4) represented in G_k are not represented in G_i nor in G_j.

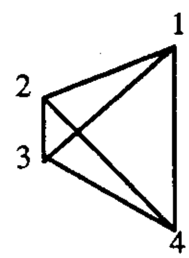
Example 2.



G_i



G_j



$G_k = G_i \otimes G_j$

In example 2, (1,4), (2,4), (3,4) are edges of G_k by the first rule. (1,2) is an edge of G_k since (1,2) is not a nonedge of G_i mod $\{3,4\}$ and a similar situation exists for (2,3) and (3,1). Here the G_k graph is complete and is therefore superfluous. The relations discussed in this paper are relations which can be perfectly represented by graphs. We will use, therefore, in the sequel, the notation $<x \mid z \mid y>$ for both the independency statements in the relation (the (x, z, y) triplets) and their corresponding graph cutset separation triplets.

It has been shown in [2] that the set of graphs $\{R, B, R \otimes B\}$ is closed under the axioms 1, 2, 3, 5. The vertices common to two graphs R and B will be denoted by V_{BR}. In the left hand side of the contraction axiom there are two triplets: one of the form $<X \mid YZ \mid W>$ and the other $<X \mid Z \mid Y>$. The first one will sometimes be referred to as the 'big' one, and the second one as the 'small' one.

Let S be a set of graphs. We define

1. $[S]$ is the set of triplets represented by the graphs in the set S.

2. $CL_G(S)$ is the set of all triplets in the closure under the 5 graphoid axioms of the triplets represented by the graphs in S.

3. $cont\ (R, B)$ is defined as $cont\ (R, B) = CL_G(\{R, B, R, \otimes B\}) - [\{R, B, R, \otimes B\}]$.

Remark: We will say that t is in S when the triplet t is represented in one of the graphs in the set of graphs S.

Preliminary Lemmas

Lemma 3.1: Let R and B be two graphs, such that $\langle X \mid ZY \mid W \rangle$ is in R, $\langle X \mid Z \mid Y \rangle$ is in B, and in addition the set W is in V_B. Then the triplet $t = \langle X \mid Z \mid YW \rangle$, which is implied by the previous two triplets by contraction, is in $R \otimes B$.

Proof: Since W is in V_B, it follows by strong union that $\langle X \mid ZW \mid Y \rangle$ is in B. t is implied by this triplet and $\langle X \mid YZ \mid W \rangle$, which is in R, by intersection. Therefore t is represented in $R \otimes B = [\{R, B, R \otimes B\}]$. □

Lemma 3.2: Let R and B be two graphs, such that their vertex sets are equal. Then $cont\ (R, B) = \varnothing$.

Proof: For any two triplets, one in R and one in B, which imply a third triplet by contraction, Lemma 3.1 holds. Therefore $CL_G(\{R, B, R \otimes B\}] = [\{R, B, R \otimes B\}]$. □

Define the symmetric contraction (notation: SC) axiom as below:

SC: $<Q \mid XZ \mid YW> \;\&\; <QX \mid ZY \mid W> \;\&\; <X \mid Z \mid Y> \rightarrow <QX \mid Z \mid YW>$.

Lemma 3.3: The contraction axiom and SC are equivalent modulo the decomposition axiom.

Proof: (I) Symmetric contraction implies contraction: assume that SC holds. Let Q be the empty set. $<\varnothing \mid XZ \mid WY>$ always holds. Thus: $<X \mid ZY \mid W>$ and $<X \mid Z \mid Y>$ imply $<X \mid Z \mid YW>$. (II) contraction implies SC: assume that contraction holds. Given that the left hand side of SC holds we have the following:

1. $<Q \mid XZ \mid YW>$

2. $<QX \mid ZY \mid W>$

3. $<X \mid Z \mid Y>$

4. $<X \mid ZY \mid W>$ follows from 2 by decomposition.

5. $<X \mid Z \mid YW>$ follows from 3 and 4 by contraction.

6. $<QX \mid Z \mid YW>$ follows from 1 and 5 by contraction. □

Note: in order to get the right has side of the SC axiom one has to apply twice the contraction axiom. This observation is valuable since in the development of a contraction algorithm (see section 3 in the sequel) the use of the SC axiom instead of the contraction axiom may save time.

Similar to our previous notations, in SC the triplets $<QX \mid ZY \mid W>$ and $<Q \mid XZ \mid YW>$ will be referred to as the big triplets, and the triplet $<X \mid Z \mid Y>$ will be referred to as the small triplet.

Let $G = (V, E)$ be a graph, and V' a subset of V. Let $C(V')$ be the complete graph over the vertices V'. $\oslash\ G(V')$ is defined to be the graph resulting from G and $C(V')$ by the $\oslash$ operation. We can prove now the following:

Lemma 3.4: (The restriction lemma): Let $G = (V, E)$ be any graph, V' is a subset of V. Let $\overline{G}_{V'} = \oslash\ G(V')$. Then $<X \mid Z \mid Y>$ is represented in $\overline{G}_{V'}$ if and only if $<X \mid Z \mid Y>$ is represented in G and $X \cup Y \cup Z \subset V'$. (We call this lemma the restriction lemma because it shows that if we have a graph G, representing a set of triplets, and we want to restrict our attention to a subset of its vertices, namely to represent the subset of triplets over a certain subset of vertices, we can do that by considering the restricted graph $\overline{G}_{V'}$.)

Proof: The set of triplets over V' represented in G satisfy axioms 1, 2, 3, 6 and 7. Therefore they can be perfectly represented in a graph over V' (Section 1). It was shown in [3] that such a set of triplets (and its corresponding graph) is completely determined by its subset of triplets of the form $<a \mid V' - a - b \mid b>$. It follows from the definitions that the nonedges of $\overline{G}_{V'}$ are in a 1 - 1 correspondence with the triplets of the form $<a \mid V' - a - b \mid b>$ represented in G (a, b are vertices in V'). $\overline{G}_{V'}$ is therefore the graph, perfectly representing the set of triplets represented in G over V' □.

4. The Emptiness Problem

Given two graphs, R and B, it is sometimes desirable to know whether any triplets not represented in the given graphs but implied by contraction (and not by intersection) from the represented triplets exist. The algorithm provided in this section solves this problem polynomially. We will use this algorithm in the sequel as a stopping rule for a more complex algorithm. Formally, the emptiness problem for two given graphs R and B is to decide whether $cont\ (R, B) \neq \emptyset$.

We need first a few lemmas. If $cont\ (R, B) \neq \emptyset$ then there exist 3 triplets t_1, t_2 and t_3 such that t_1 and t_2 are in different graphs of the set $S = \{R, B, R \otimes B\}$ and t_3, which is implied by contraction from T_1 and T_2, is not in S.

Lemma 4.1: If $cont\ (R, B) \neq \emptyset$ then we may assume the following configuration w.l.o.g.

$t_1 = \langle X \mid YZ \mid W \rangle, t_1 \in R, t_1 \notin B, t_1 \notin R \otimes B$

$t_2 = \langle X \mid Z \mid Y \rangle, t_2 \in B, t_2 \notin R$

$t_3 = \langle X \mid Z \mid YW \rangle, t_3 \notin S, S = \{R, B, R \otimes B\}$

$X_1 Y_1 Z \subset V_{BR}, W \not\subset V_B, W \neq \emptyset.$

Proof: $cont\ (R, B) \neq \emptyset$ implies the existence of a triplet $t_3 \notin S$ as above. $t_3 \notin s$ implies that $t_1 \notin R \otimes B$. (Otherwise t_2, whose vertex set is a subset of the vertex set of t_1, must also be represented in $R \otimes B$ which would imply that t_3 is in $R \otimes B$). We will assume therefore that $t_1 \in R$. (Otherwise exchange the names of R and B). If t_2 is in B then t_2 is in $R \otimes B$.

If t_2 is in $R \otimes B$ and is not in R then t_2 is in B. Therefore we can assume that t_2 is in B.

W is not a subset of V_B since otherwise the set of vertices of t_3 would be a subset of V_{BR} implying that $t_3 \in S$.

Lemma 4.2: Given that triplets t_1, t_2 and t_3 as in lemma 4.1 exist, another set of triplets t_1', t_2', t_3' can be found such that

$t'_1 = \langle a \mid V_{BR} - a \mid W' \rangle$

$t'_2 = \langle a \mid V_{BR} - a - b \mid b \rangle$

$t'_3 = \langle a \mid V_{BR} - a - \mid bW' \rangle$

a, b are vertices; $W' = W - V_{BR} \neq \emptyset$

$t'_1 \in R, t'_1 \notin B, t'_1 \notin R \otimes B; t'_2 \in B, t'_2 \notin R; t'_3 \notin S.$

The proof of this lemma is lengthy and is omitted. The proof is based on the following argument:

Given that $t_2 = \langle X \mid Z?Y \rangle \notin R$ there must be two vertices $a \in X$ and $b \in Y$ which are connected in R according to one of the following 4 alternatives

1. (a, b) is an edge in R

2. a is connected to b in R via a path inside $V_{BR} - Z$

3. a is connected to b in R via a path outside V_{BR}

4. a is connected to b in R via an alternating path, inside $V_{BR} - Z$ and outside V_{BR}

For any of the above alternatives, triplets t'_1, t'_2, t'_3 as required can be found - based on the given triplets t_1, t_2, t_3. Notice that since $R \otimes B \in S$, $t'_3 \notin S$ implies that t'_3 is not implied by the intersection of any two triplets in S.

We are now ready to present the algorithm.

Emptiness Algorithm, Empty (B, R).

Procedure check $(B\ R)$

1. Construct the two graphs: $\bar{B} = B \otimes comp\ (V_{BR})$ and $\bar{R} = R \otimes comp\ (V_{BR})$ where $comp\ (V)$ is the complete graph over V.

2. Check if $\bar{R}$ implies $\bar{B}$ (i.e., check whether all the independencies represented inn $\bar{B}$ are represented in $\bar{R}$, due to graph properties it is enough to check the independencies represented by nonedges in $\bar{B}$, $<a \mid V_{\bar{B}} - a - b \mid b>$). If $\bar{R}$ implies $\bar{B}$ return 'false'

3. Else -- collect all witnesses to that fact, namely all edges (a, b) in $\bar{R}$ which are nonedges in $\bar{B}$.

4. For each witness (a, b) check if there exists a nonempty set $U \subset V_R - V_{BR}$ such that either $<a \mid V_{BR} - a \mid U>$ or $<b \mid V_{BR} - b \mid U>$ holds in R. If this test succeeds for at least one of the witnesses -- return true, otherwise return false.

End of procedure.

Set Empty $(B, R) = no$ iff Check (B, R) or Check $(R, B) = true$.

Lemma 4.3: The emptiness algorithm is polynomial (in the maximal number of vertices in R or B).

Proof: The complexity of the algorithm equals the complexity of the procedure involved in it. As to the procedure:

Step 1 is polynomial by definition of the $\otimes$ operation.

Step 2 is polynomial (implication is determined by edges comparison, since both graphs have the same set of vertices).

Step 3 is also polynomial since it depends on the number of edges and the procedure for each edge is trivially polynomial.

Step 4 is polynomial as is easily seen.

Lemma 4.4: The emptiness algorithm is correct.

If $cont(B, R) \neq \emptyset$ then, by lemma 4.2 a triplet of the form $<a \mid V_{BR} - e - b \mid bW'>$ is implied, and all such triplets will be found in step 4 of the procedure. Therefore if the algorithm succeeds then $cont(R, B) \neq \emptyset$, while if the algorithm fails (in step 2 implying that all small triplets in B are in R or in step 4 implying that the small triplets in B have no corresponding contraction - candidate triplets in R) then $cont(R, B) \neq \emptyset$. □

5. Contraction Algorithm

Unlike the intersection algorithm [2], the contraction algorithm involves some intrinsic difficulties which must be taken into account. Specifically, consider the following example: Let R and B be the graphs shown in Figure 5.1. Then $<x \mid yz \mid w>$ is represented in R and $<x \mid z \mid y>$ is represented in B. By contraction we get $<x \mid z \mid yw>$ which is not represented in R nor in B.

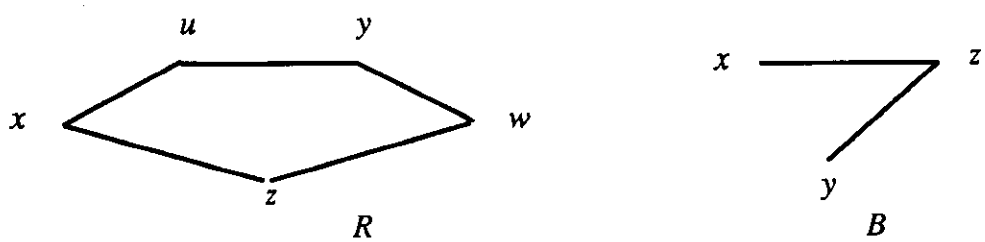


Figure 5.1

If we disconnect u from y or u from x in R (which are the only possibilities for representing the new triplet in R) we will get a new graph, say the one in Figure 5.2, replacing R, in which $<x \mid z \mid yw>$ is represented. But also the triplet $<ux \mid z \mid yw>$ is represented, and this triplet is not implied (by the graphoid axioms) by the triplets represented in R and B, as one can check explicitly.

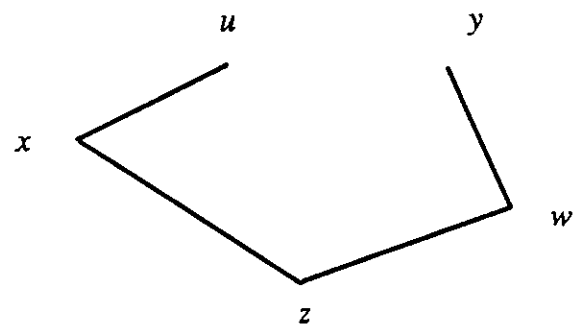


Figure 5.2

Therefore the only way out is to represent the new triplet $<x \mid z \mid yw>$ in a new graph, the one in Figure 5.3, and add this new graph to the set of graphs B and R.

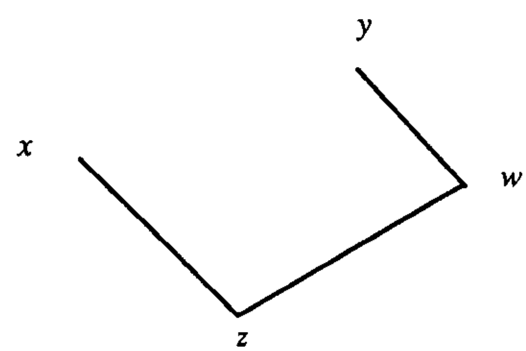


Figure 5.3

To overcome the above mentioned difficulties, we will use the following construction. We will split the algorithm into cycles and each cycle will have two stages: In the first stage we will check (using the emptyness algorithm) for any pair of graphs in the set of graphs constructed up to the previous stage whether there exists a **new** triplet implied by contraction from the triplets represented in the two graphs in the pair. If the answer is yes then we will either modify one of the graphs to account for the new triplet or construct a new graph which will represent that triplet (given that the first alternative is not feasible).

It will follow from the construction that the algorithm is exponential in it's input, but we will show that this might be necessary in some cases. On the other hand we will show that the algorithm is polynomial in its' output.

The Algorithm:

Procedure Confront (current, done):

/* done and current are sets of graphs */

1. Initialize: **next** = ∅;

/* next is a set of graphs */

2. Loop: while **current** ≠ ∅ do

   Let G be a graph in current

   2.1 Collect edges: insert to a set S all $(a, b) \in V_P \times V_P$ such that

   i) $a \neq b$.

   ii) (a, b) is a nonedge of P

   iii) (a, b) is not a nonedge of G mod $V_P - a - b$.

   2.2 Constructions: For each $(a, b) \in S$ find $V_G(a, b)$ - the set of all vertices in G such that discarding $V_P - a$ from G will render them connected to a, and discarding $V_P - b$ from G will render them connect to b.

2.2.1 If $V_P \cup V_{G(a,b)} = V_G$ - ignore this edge and proceed to the next one.

2.2.2 If $V_G(a, b) = \varnothing$ disconnect a from b in G.

2.2.3 else - construct $G_{a,b} = \oslash G(V_G - V_G(a, b))$ and disconnect a from b in that graph. Add the new graph to the set **next**.

2.3 **done** = **done** $\cup \{G\}$; **current** = **current** $-\{G\}$;

3. **current** = **next**;

End.

Contraction Algorithm

1. Construct $P = R \oslash B$

2. Initializations: Set **current** = $\{R, B\}$; **done** = $\varnothing$;

3. For each graph G in **current** if $cont(G, P) = \varnothing$ then set **done** = **done** $\cup \{G\}$; **current** = **current** $-\{G\}$.

4. If **current** = $\varnothing$ then **done** = **done** $\cup \{P\}$; Output (**done**); halt.

5. Confront (current, done);

6. Go to 3.

Lemma 5.1: The algorithm halts.

Claim: The Confront procedure ends.

Proof: The number of operations involved in steps 1 and 3 is finite. Step 2 is a loop. The loop is carried out once for each graph in **current**. For every graph in **current** the number of operations involved in step 2 is polynomial in the number of vertices in P. Therefore the confront procedure halts.

Proof (of the lemma): The number of operations involved in steps 1 and 2 of the algorithm is polynomial in the size of V_P.

Step 3 has a loop, which is carried out once for each graph in **current**. The number of operations performed by step 3 for any particular graph is polynomial in the size of V_P.

The number of operations involved in step 4 is finite.

Step 5 is finite by the above claim.

The main loop of the algorithm is defined in steps 3 to 6. Let $n = \max\{|V_R|-|V_P|, |V_B|-|V_P|\}$. As shown by the argument below, this loop is carried out at most n times and therefore this loop is also finite.

Let C' be the set of graphs resulting from the set C after an execution of the confront procedure, where C is the set **current** input to the procedure. The vertex set of a graph in C' is a proper subset of the vertex set of the graph in C from which the graph in C' originated. It follows from the definitions that V_P is a subset of the vertex set of any graph produced all through the algorithm. Thus after at most n loops all the graphs in **current** have the property that their vertex set coincides with V_P. By lemma 3.2, when this happens all the graphs in **current** will be moved to **done** in step 3 of the algorithm and the algorithm will halt.

Lemma 5.2: done $\subset CL_G(R, B)$.

Proof: By induction on the loops of the main part of the algorithm.

Basis: **done** is initiated to the empty set. After the first loop **done** $=\{R, B\}$ and the triplets in R and B are in the closure by definition.

Assume all triplets represented in the set of graphs **current** input to the confront procedure are in $CL_G(R, B)$. New triplets will perhaps be created by deleting edges from graphs. If an edge (a, b) was deleted from a graph H then the condition enabling this operation held, namely $V_H(a, b) = \emptyset$. That is to say there is no path in H between a and b outside V_P. The set of vertices in H can be partitioned mod $V_P - a - b$ into two subsets U_1 and U_2 such that U_1 remains connected to a but not to b, and U_2 remains connected to b and not to a when $V_P - a - b$ is removed. In other words, after deleting $V_P - a - b$ from H, the remaining graph consists of two components: one of U_1 and a, and the other of U_2 and b, which means that $t_1 = <U_1 \mid V_P - b \mid bU_2>$ and $t_2 = <U_1 a \mid V_P - a \mid U_2>$ were represented in H and $t_3 <a \mid V_P - a - b \mid b>$ was in B. After the deletion of the edge (a, b) in H we will have $t_4 = <U_1 \cup \{a\} \mid V_P - a - b \mid \{b\} \cup U_2>$ represented in the resulting graph H', and this triplet is implied from t_1, t_2 and t_3 by symmetric contraction. This implies, by weak union which holds in graphs, that $<a \mid V_H - a - b \mid b>$ is represented in H'. Now $CL_G(R, B)$ is closed under weak union which implies that this new triplet is also in $CL_G(R, B)$. But $<a \mid V_H - a - b \mid b>$ is the only triplet, involving **all** the vertices of H, which is in H' and not in H. Therefore, by the induction hypothesis, all the triplets of the form $<x \mid V_H - x - y \mid y>$; x, y vertices in V_H, which are in H' are in the closure. We can use now an inductive argument, similar to the argument used in the proof of Theorem 1 in [3] to show that all the triplets in H' are in the closure. If step 2.3 of the confront procedure is applied then, by the restriction lemma (section 3) $G_{a,b}$ is in $CL_G(R, B)$ given that R and B are in $CL_G(R, B)$ and the same argument as above applies for the removal of an edge from $G_{a,b}$. The above argument holds for every edge deletion and therefore the set of triplets in **current** after the execution of the confront procedure is also in $CL_G(R, B)$. □

Lemma 5.3: After the algorithm halts, the set of triplets in **done** in closed under the 4 pseudographoid axioms.

Proof: Closure under symmetry, decomposition and weak union is trivial, since for every triplet represented in a graph G, all triplets resulting by these axioms are in the same graph. To show closure under intersection we prove that after the algorithm halts, for every two graphs G_1, G_2 in **done** there is a graph G_{12} is **done** such that G_{12} *implies* $G_1 \otimes G_2$. The details of this part of the proof are omitted.

Lemma 5.4: When the contraction algorithm halts, $CL_G(R, B) \subset$ **done**.

Proof: Assume b.w.o.c. that the algorithm halted and $CL_G(R, B) \not\subset$ **done**. Then there exists a triplet t which is implied from triplets in R and B by a chain of derivations, and t is not represented in **done**. Let $t_1, t_2 ... t$ be this chain of derivations. Let t_k be the first triplet in the chain which is not represented in **done**. By lemma 5.3, since the algorithm halted, **done** is closed under the 4 pseudographoid axioms. Therefore t_k is implied from two triplets t_i and t_j by contraction, where $i, j < k$. To complete the proof we show that if t_k, implied by contraction from t_i and t_j, is not in **done** the algorithm will run forever contrary to lemma 5.1. The details of this part of the proof are omitted.

Corollary 5.5: The algorithm is correct

Proof: Follows from lemmas 5.2 and 5.4.

Lemma 5.6: The time complexity of the algorithm is polynomial in its output (but may be exponential in its input).

Proof: A close examination of the algorithm shows that every processed graph will eventually end up in the set **done**. Therefore if the output is polynomial (in the number of variables-vertices over which the graphs are defined) then the set of graphs carried in memory by the algorithm is polynomial all through its execution.

It is enough to show therefore that every iteration of the algorithm is polynomial and involves an addition of at least one graph to the set **done**. Now, every iteration (except the last one) involves one call of the procedure Confront, and, if no graph is added to the set done in step 3 then at least one graph is added to the set **done** in step 2.3 of the procedure. Straightforward examination of the steps involved in one iteration of the algorithm will show that every step (including the procedure confront) is polynomial in the number of graphs in the set **current** at the time of its processing. The number of graphs in the set current, all through the execution of the algorithm, is bounded by the number of graphs in the set **done** at output. □

6. The Space Complexity of *cont* (R, B)

The algorithm presented in the previous section constructs a set of graphs representing *cont* (R, B) for 2 given graphs R and B. Unfortunately, the number of graphs at output may be exponential. The following set of counterexamples shows that this might be necessary.

Theorem 6.1: For any integer $n \geq 2$ there are two graphs R and B over $2n+1$ vertices such that *cont* (R, B) requires exponentially many graphs for its graphical representation.

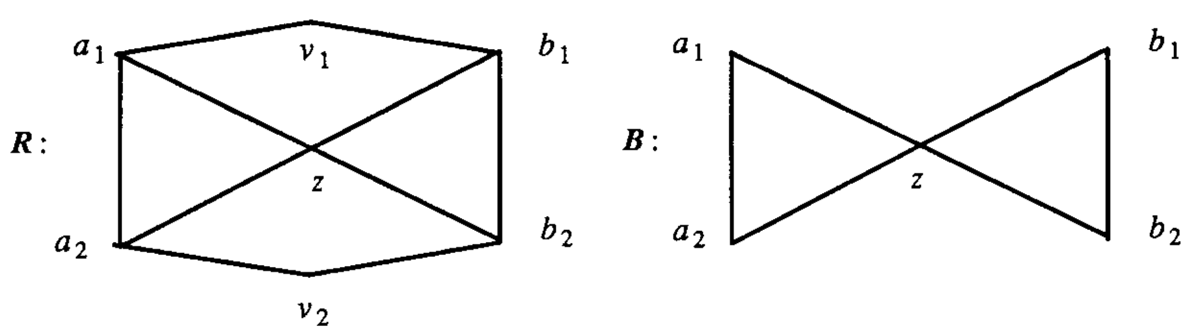
Proof: Let $X = \{a_1, ..., a_n\}$, $Y = \{b_1, ..., b_n\}$ and $V = \{V_1, ..., V_n\}$. The graph B is defined as follows: $V_B = X \cup Y \cup \{z\}$;

$$E_B = \{(a_i, z), (b_i, z), (a_i, a_j)(b_i, b_j) : i, j \in \{1, ..., n\} i \neq j\} .$$

The graph R is defined as follows: $V_R = X \cup Y \cup \{z\} \cup \{v_i : 1 \leq i \leq n\}$;

$$E_R = \{(a_i, v_i), (v_i, b_i), (b_i, z), (a_i, a_j), (b_i, b_j) : 1 \leq i, j \leq n, i \neq j\} .$$

e.g., for the case $n = 2$ the graphs are:



We proceed by proving two lemmas:

Lemma 6.2: All the $2^n - 2$ triplets of the form $< a_{i1}, ..., a_{il} \mid a_{il+1}, ..., a_{in} z \mid Y v_{il+1}, ..., v_{in} >$ where

$\{i_1,..., i_l\}$ and $i_{l+1},..., i_n\}$:$0 < l < n$ is a partition of the set $\{1,..., n\}$ - are in $CL_G(R, B)$.

Proof: Let $\{i_1,..., i_l\}$ and $\{i_{l+1},..., i_n\}$ be a partition of $\{1,..., n\}$. Then we claim that $<a_{i1},..., a_{il} \mid a_{il+1},..., a_{in} Yz \mid v_{il+1},..., v_{in}>$ is represented in R. For every i the vertex v_i has rank 2 being connected to the vertices a_i and b_i. If we discard a_i, b_i and z from R, then v_i is isolated. Therefore we have that $<a_{i1},..., a_{il} \mid a_{il+1},..., a_{in} Yz \mid v_{il+1},..., v_{in}>$ is represented in R - as claimed.

Next we claim that $<a_{i1},..., a_{il} \mid a_{il+1},..., a_{in} z \mid Y>$ is represented in B. This follows by weak union from $<X \mid z \mid Y>$, which is in B, as is easily seen. The resulting triplet by contraction is $<a_{i1},..., a_{il} \mid a_{il+1},..., a_{in} z \mid Y v_{il+1},..., v_{in}>$. There are $2^n - 2$ partitions of $\{1,..., n\}$ where neither $\{i_1,..., i_l\}$ nor $\{i_{l+1},..., i_n\}$ is empty. For each such partition our construction holds, therefore we have $2^n - 2$ different resulting triplets.

None of these resulting triplets is already represented in B ,in $R \circledcirc B$ or in R. Since no 'v' labeled vertex is in B, and in every resulting triplet there are 'v' labeled nodes, none of the triplets is in B. $B = B \circledcirc R$ therefore the triplets are not in $B \circledcirc R$ either. In all resulting triplets there are no 'v' labeled nodes in the middle part, so that if the set of vertices in the middle part of any such triplet is discarded from R then all paths between a_j and b_j:$j \in \{i_1,..., i_l\}$ are still in R, therefore by definition none of the triplets is represented in R. This completes the proof of the lemma. □

For the next lemma we need the following definition:

A graph is *two m-connected cliques of order n* (mcc in short) if it consists of two cliques, with vertices indexed from 1 to n, it has one additional vertex connected to all $2n$ vertices in the two cliques, and it has a set V of m vertices $0 \le m \le n$ $v_{i1}, \ldots, v_{im}$ and $\{i_1, \ldots, i_m\} \subset \{1, \ldots, n\}$, such that v_j is connected to the two vertices indexed j in the two cliques, and to no other vertex in the graph.

Notice that the graph R is 'two n-connected cliques' of order n, and the graph B is 'two 0-connected cliques' of order n.

Lemma 6.3: Let D be the set of graphs output by the contraction algorithm when applied to the graphs R and B. Then every graph R' in D is mcc for some m.

Proof: By induction on the order in which graphs enter the set **done** in the execution of the algorithm.

Basis: The first two graphs to enter **done** are R which is ncc and B which is $0cc$.

Step: Let R' be an mcc graph, confronted in the confront procedure with P, and as mentioned in lemma 6.2, in this case $P = B$. The set of pairs of edges collected in step 2.1 in the procedure is $S = \{(a_i, b_i): v_i \in V_{R'}\}$. Every pair of vertices (a_i, b_i) is a nonedge in B, and the pair is not a nonedge in R' modulo $V_B - a_i - b_i$ if and only if the vertex v_i is in the vertex set of the graph R'. All the pairs of vertices of the form (a_i, b_j) where $i \neq j$ are nonedges in both B and R, and clearly the vertex z cannot appear in any collected pair.

Let (a_i, b_i) be a pair in S. The only vertex connected both to a_i and to b_i modulo V_B in R' is v_i, therefore in step 2.2 of the procedure, $V_{R'}(a_i, b_i) = \{v_i\}$. If R' is 1cc, then there is only one pair in the set S, and it will be ignored. Thus the graph R' is ignored in this case, and no non *mcc* graph is built. Otherwise, a subgraph R'' of R' will be built, over the set of vertices $V_{R'} - v_i$ and the edge (a_i, b_i) will be disconnected in R''. R'' is (m-1)cc. This completes the proof of this lemma.

We can now prove the theorem:

Assume b.w.o.c. that there exist a set D' of graphs representing $CL_G(R, B)$ and there are less then $2^n - 2$ graphs in D'. Then there is a graph $H \in D'$ representing at least two triplets from the set of triplets described in lemma 6.2. Let $\langle a_{i1}, \ldots, a_{il} \mid a_{il+1}, \ldots, a_{in} z \mid Y v_{il+1}, \ldots, v_{in} \rangle$ and $\langle a_{j1}, \ldots, a_{jk} \mid a_{jk+1}, \ldots, a_{jn} z \mid Y v_{jk+1}, \ldots, v_{jn} \rangle$ be two different triplets represented in H, and let $r \in \{i_{l+1}, \ldots, i_n\} - \{j_{k+1}, \ldots, j_n\}$. (If the difference is empty then reverse the order of the subtraction). Then, by transitivity, either $t_1 = \langle a_{j1}, \ldots, a_{jk} \mid a_{jk+1}, \ldots, a_{jn} z \mid v_r \rangle$ is in H, or $t_2 = \langle v_r \mid a_{jk+1}, \ldots, a_{jn} z \mid Y v_{jk+1}, \ldots, v_{jn} \rangle$ is in H. If t_1 is in H then by decomposition the triplet $t'_1 = \langle a_r \mid a_{jk+1}, \ldots, a_{jn} z \mid v_r \rangle$ is in H, and therefore by the assumption that H is a part of a representation of $CL_G(R, B)$, $t'_1 \in CL_G(R, B)$. If t_2 is in H then again by decomposition the triplet $t'_2 = \langle b_r \mid a_{jk+1}, \ldots, a_{jn} z \mid v_r \rangle$ is in H and therefore in $CL_G(R, B)$.

But, none of the two triplets t'_1 and t'_2 can be represented in an *mcc* graph, since t'_1 requires that no edge exists between a_r and v_r, and t'_2 requires that no edge exists between b_r and v_r. Therefore no one of these two triplets is in the set of graphs D produced by the contraction algorithm. By corollary 5.5 $D = CL_G(R, B)$ and therefore no one of the triplets t'_1, t'_2 is in

$CL_G(R, B)$ and that is a contradiction. Therefore there can be no representation of $CL_G(R, B)$ for the graphs R and B defined here with less then $2^n - 2$ graph. The proof is now complete. □

7. Membership Algorithm

In this section we show that the question whether a given triplet <x | z | y> is in $cont(R, B)$ for 2 given graphs R and B is answerable in polynomial time. This result is somewhat surprising, given the fact that the graphical representation of $cont(R, B)$ may necessarily require exponential space.

Definition: The membership problem for 2-graph contraction is the following problem: Given 2 graphs R and B and a triplet t, decide whether $t \in cont(R, B)$.

Membership Algorithm for 2-graph Contraction

Let V_t be the set of nodes represented in the triplet t.

1. If $V_t \not\subset V_R$ and $V_t \not\subset V_B$ - return no.

2. If $V_t \subset V_R \cap V_B$ then construct $P = R \oslash B$. $t \in CL_G(R, B)$ if and only if t is in P.

3. Else - assume $V_t \subset V_R$ (otherwise exchange the names of the graphs).

3.1 Construct $P = R \oslash B$; $V_{tP} = V_t \cup V_P \subset V_R$ then construct $R_t = \oslash R(V_{tP})$ (see lemma 3.4).

3.2 Collect all pairs (a, b) of vertices in P which are nonedges in P and edges in R'_t.

3.3 For each pair (a, b) collected in step 3.2, find the set of vertices $V(a, b)$ such that discarding $V_P - a$ from R'_t will render them connected to a, *and* discarding $V_P - b$ from R'_t will render them connected to b. If $V(a, b) = \emptyset$ then remove the edge (a, b) from R'_t.

4. Let R_t be the graph obtained from R'_t in step 3.3. $t \in CL_G(R, B)$ if and only if t is in R_t.

Theorem 7.1: The membership algorithm is correct.

Proof: If $V_t \not\subset V_R$ and $V_t \not\subset V_B$ then t cannot be in $CL_G(R, B)$. This follows from the fact that in all the graphoid axioms, all the nodes that appear on the right hand side of an axiom appear on the left hand side of that axiom.

If $V_t \subset V_R \cap V_B$ then it follows from lemma 3.2 that $t \in CL_G(R, B)$ iff t is in P. Otherwise assume $V_t \subset V_R$ and $V_t \not\subset V_B$.

The theorem follows, for this case, from the two lemmas below.

Lemma 7.2: If the triplet t is in $CL_G(R, B)$ then t is in the graph R_t.

Proof: By corollary 5.5, $t \in CL_G(R, B)$ if and only if t is represented in a graph R' in the set of graphs **done** output by the contraction algorithm, when applied to the pair of graphs R and B. By the restriction lemma, t is in R' if and only if t is in $\oslash R(V_{tP})$. To complete the proof of the lemma we show that if t is in the graph $\oslash R'(V_{tP})$ then t is in the graph R_t by showing that the graph R_t implies the graph $\oslash R'(V_{tP})$. The details are omitted.

Lemma 7.3: If the triplet t is in the graph R_t then t is in $CL_G(R, B)$.

Proof: By the restriction lemma, the set of triplets in the graph R'_t is exactly the set of triplets represented in the graph R over the set of vertices V_{tP}. Therefore all the triplets in R'_t are in $CL_G(R, B)$.

To complete the proof we show that the transformation of R'_t into R_t involving the removal of some edges (but no vertices) preserves the above property namely, the triplets in R_t are in $CL_G(R, B)$. Details are omitted.

The algorithm has no iterations and, as is easy to see, all its steps are polynomial, which proves the following theorem.

Theorem 7.4: The membership problem for 2-graph contraction has polynomial time complexity.

Bibliography

[1] Geiger, D., "Towards the Formalization of Informational Dependencies," UCLA Cognitive Systems Laboratory, *Technical Report 880053 (R-102),* (Based on the author's MS thesis), December 1987.

[2] Paz, A., "A Full Characterization of Pseudographoids in Terms of Families of Undirected Graphs," UCLA Cognitive Systems Laboratory, *Technical Report 870055 (R-95),* September 1987.

[3] Pearl, J., and Paz, A., "GRAPHOIDS: A Graph-Based Logic for Reasoning about Relevance Relations," UCLA Computer Science Department, *Technical Report 850038 (R-53); In Advances in Artificial Intelligence-II,* Edited by B. Du Boulay et al. North-Holland Publishing Co. 1987.

[4] Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan-Kaufmann, San Mateo, CA., 1988.

[5] Studeny, M., "Attempts at Axiomatic Description of Conditional Independence," *Workshop on Uncertainty in Expert Systems,* Alsovice, Chechoslovakia, June 20-23, 1988.

[6] Verma, T., "Some Mathematical Properties of Dependency Models," UCLA Cognitive Systems Laboratory, University of California, Los Angeles, in preparation.