# Building Causal Graphs from Statistical Data in the Presence of Latent Variables

by

Peter Spirtes

November 1991

Report CMU-PHIL-23

Carnegie Mellon

Philosophy
Methodology
Logic

Pittsburgh, Pennsylvania 15213-3890

## Abstract

The problem of inferring causal relations from statistical data in the absence of experiments arises repeatedly in many scientific disciplines, including sociology, economics, epidemiology, and psychology. In addition, the building of expert systems could be expedited if background knowledge elicited from experts could be supplemented with automated techniques using relevant statistics.

Recently, efficient algorithms for determining causal relationships between random variables (in the form of Bayesian networks) from appropriate statistical data when there are no unmeasured or "latent" variables have been discovered. (See Spirtes, Glymour and Scheines 1990, Spirtes and Glymour 1991, Verma and Pearl 1990, and Pearl and Verma 1991.) Inferring causal relations when unmeasured variables are also acting is a much more difficult problem. In many cases it is impossible to infer the structure among the latent variables from statistical relations among the measured variables. But the presence of latent variables can also make it difficult to infer the causal relations among the measured variables themselves. When only two variables, A and B, have been measured, and there is a correlation between the two, this does not suffice to establish whether A causes B, B causes A, or there is a third unmeasured variable causing both A and B. Nevertheless, when other variables are measured, more knowledge about the causal relations between A and B is possible. We will prove in Theorem 2 that there are some circumstances in which it is possible to establish that A causes B, rather than that B causes A, or that a third unmeasured variable causes both A and B; and we will prove in Theorem 3 that there are other circumstances in which the possibility that A causes B can be eliminated.

# Building Causal Graphs From Statistical Data In the Presence of Latent Variables

by Peter Spirtes[1]

## 1 Introduction

The problem of inferring causal relations from statistical data in the absence of experiments arises repeatedly in many scientific disciplines, including sociology, economics, epidemiology, and psychology. In addition, the building of expert systems could be expedited if background knowledge elicited from experts could be supplemented with automated techniques using relevant statistics.

Recently, efficient algorithms for determining causal relationships between random variables (in the form of Bayesian networks) from appropriate statistical data when there are no unmeasured or "latent" variables have been discovered. (See Spirtes, Glymour and Scheines 1990, Spirtes and Glymour 1991, Verma and Pearl 1990, and Pearl and Verma 1991.) Inferring causal relations when unmeasured variables are also acting is a much more difficult problem. In many cases it is impossible to infer the structure among the latent variables from statistical relations among the measured variables. But the presence of latent variables can also make it difficult to infer the causal relations among the measured variables themselves. When only two variables, A and B, have been measured, and there is a correlation between the two, this does not suffice to establish whether A causes B, B causes A, or there is a third unmeasured variable causing both A and B. Nevertheless, when other variables are measured, more knowledge about the causal relations between A and B is possible. We will prove in Theorem 2 that there are some circumstances in which it is possible to establish that A causes B, rather than that B causes A, or that a third unmeasured variable causes both A and B; and we will prove in Theorem 3 that there are other circumstances in which the possibility that A causes B can be eliminated. The proofs are given in the Appendix.

## 2 Directed Acyclic Graphs

[1] I thank C. Glymour and R. Scheines for valuable help with the work described here. Research for this paper was supported by the Naval Personnel Research and Development Center and the Office of Naval Research under contract number N00114-89-J-1964.

Causal processes between a set of random variables **V** are represented by a directed acyclic graph over **V**, where there is an edge from A to B if and only if A is an immediate cause of B relative to **V** (i.e. there is a mechanism by which A causes B that is not blocked by holding fixed any of the other variables in **V**.) If there is a directed path from A to B in the causal graph, we will say that A is a (possibly indirect) cause of B. (In what follows, we will capitalize random variables, and boldface any sets of variables. We will use the terms "vertices in a graph" and "variables in a graph" interchangably.)

A directed acyclic graph G over a set of random variables **V** can also be used to represent the set of probability distributions over **V** that satisfy the following two conditions:

**Markov Condition:** Let **Parents**(X) be the set of parents of X in G (i.e. the set of Z such that Z -> X is in G) and **Descendants**(X) be the set of descendants of X in a graph G (i.e. the set of Z such that there is a directed path from X to Z in G.) A directed acyclic graph G and a probability distribution P on the vertices **V** of G satisfy the Markov condition if and only if for every X in **V**, X and **V**\({X} $\cup$ **Descendants**(X)) are independent conditional on **Parents**(X).

**Faithfulness Condition:** If G is a directed acyclic graph G and P is a distribution over the set of vertices **V** in G, then P is faithful to G if and only <G,P> satisfy the Markov condition and every conditional independence relation true in P is entailed by the Markov condition for G.

If a distribution is placed over the exogenous variables (variables of zero indegree) in the causal graph of a causal process, which in turn affect the values of other random variables, the result is a joint distribution over all of the random variables. In that case, we will say that the causal process generated the joint distribution. We assume that the distribution generated by a causal process satisfies the Markov and Faithfulness conditions for the causal graph of that process; we will call this the **Causal Faithfulness Assumption.** In Pearl's terminology (Pearl 1988) the causal graph is a **Bayes network** of any distribution that it generates.

In a directed graph G, we will write X -> Y if there is an edge from X to Y in G, and we will say that X is **parent** of Y. X and Y are **adjacent in a directed graph** G if and only if either X -> Y or Y -> X in G. If X and Y are adjacent in G, we will also say that **X is a neighbor of Y** and **Y is a neighbor of X.** In a directed acyclic graph G, an **undirected path U from X to Y** is a sequence of vertices starting with X and ending with Y such that for every pair of vertices A and B that are adjacent to each other in the sequence, A and B are adjacent in G, and no vertex occurs more than once in U. In a directed acyclic graph G, a **directed path P from X to Y** is a sequence of vertices starting with X and ending with Y such that for every pair of variables A and B that are adjacent to each other in the sequence in that order, the edge A -> B occurs in G, and no vertex occurs more than once in P. X and Y are **adjacent on path P** (as distinct from adjacent in the graph) if and only if X and Y are adjacent in the sequence P. An **edge between X and Y occurs in a path P** (directed or undirected) if and only if X and Y are adjacent in P. If an undirected path U contains an edge between X and Y, and an edge between Y and Z, the two edges **collide** at Z if and only if X -> Y and Z -> Y in G. On an undirected path U, Z is an **unshielded collider** if and only if there exist edges X -> Y and Z -> Y in U, and Z and X are not adjacent in G. X is an **ancestor** of **Y** and **Y** is **a descendant** of X if and only if there is a directed path from X to Y. (We count the sequence consisting of a single vertex <X> as a directed path from X to X, so X is its own ancestor and descendant, although it is not its own parent or child.) X, Y, and Z form **triangle X-Y-Z** in G if and only if X is adjacent to Y, Y is adjacent to Z, and Z is adjacent to X in G. A **trek** between X and Y is either a directed path from X to Y, a directed path from Y to X, or a pair of directed paths from some third variable Z to X and Y respectively that intersect only at Z.

Verma and Pearl (see Pearl 1988) have shown how to calculate the conditional independence relations that are entailed by distributions satisfying the Markov condition for a graph G using the d-separability relation. In graph G, a path **U d-connects variables X and Y given a set of vertices S** not containing X or Y if and only if (i) every collider on U has a descendent in S and (ii) no other vertex on U is in S. Vertices **X and Y are d-separated given a set S** not containing X and Y if and only if no path d-connects X and Y given S. Disjoint sets of vertices X and Y are d-separated given S in G if and only if every member of X is d-separated from every member of Y given S in G. If distribution P satisfies the Markov and Faithfulness Conditions, then for disjoint sets of vertices X, Y, and $Z_f$ X is independent of Y conditional on S if and only if X is d-separated from Y given S in G (Pearl 1988).

We say that **V** is **causally sufficient** if and only if every cause of any two members of V is itself in V. If the distribution P is generated by some causal process, then given the Causal Faithfulness Assumption, P is faithful to some directed acyclic graph; however, if a set of measured variables V is not causally sufficient, the marginal distribution of P over V may not be faithful to any directed acyclic graph. Our strategy for making inferences about causal relationships when latent variables may be present is to find properties held in common by all directed acyclic graphs that have faithful distributions for which P could be the marginal.

## 3. Spurious Causal Dependencies

In a directed acyclic graph G over a set of variables V, if A and B are adjacent in G, then A and B are not d-separated by any subset of V\{$A_9$B}. Hence under the assumption of causal sufficiency, either A is a direct cause of B or B is a direct cause of A relative to V if and only if A and B are independent conditional on no subset of V. (Recently more efficient and reliable algorithms for determining causal structure from statistical data when there are no latent variables have been devised. See Spirtes, Glymour and Scheines 1990, Spirtes and Glymour 1991.) At first glance, it appears that this technique can be generalized to the case where V is not causally sufficient by inferring from the dependence of A and B conditional on every subset of V\{A,B} that either A is a direct cause of B relative to V, or B is a direct cause of A relative to V, or there is some latent variable L that is a common cause of both A and B. Unfortunately, this is not the case, as the following example shows.

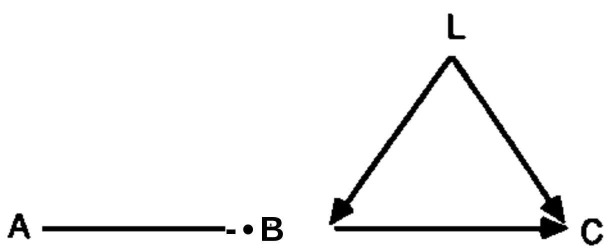



**Figure 1: Graph G**

Let **V** = {A,B,C,L} and O = {A,B,C}. O is not causally sufficient because L is a cause of both B and C which are in O, but L itself is not in O. A and C are not d-separated by any subset of 0\{$A_y$B}, so in any marginal of a distribution faithful to G, A and B are not independent conditional on any subset of 0\{A,B}. Nevertheless, A is not a direct cause of C relative to O, C is not a direct cause of A relative to O, and there is no latent common cause of A and C. If the algorithms for the causally sufficient case where applied to O, they would find, erroneously that A and C are adjacent. Our problem is to find a more reliable procedure.

## 4. Inducing Paths

Given a directed acyclic graph G over a set of variables **V**, and **O** a subset of **V**, Verma and Pearl (1990) have characterized the conditions under which two variables in **O** are not d-separated by any subset of **O**\{A,B}. In a directed acyclic graph G over a set of variables **V**, an undirected path U between A and B is an **inducing path over a subset O of V** if and only if every member of **O** on U is a collider on U, and every collider is an ancestor of either A or B. (We will sometimes refer to members of **O** as **observed** variables.)

**Theorem 1:** In a directed acyclic graph G over **V**, where **O** is a subset of **V**, A and B are not d-separated by any subset of **O**\{A,B} if and only if there is an inducing path over the subset **O** between A and B.

In Figure 1, the inducing path between A and C is <A,B,L,C>.

## 4. Inducing Path Graphs

The inducing paths relative to **O** in a graph G over **V** can be represented in the following structure described (but not named) in Pearl and Verma (1990). In an **inducing path graph G' for directed acyclic graph G over a subset of variables O** there is an edge between variables A and B with an arrowhead at A if and only if A and B are in **O**, and there is an inducing path in G between A and B relative to **O** that is into A (i.e. there is an edge in the path with an arrowhead into B.) Note that in an inducing path graph, there are two kinds of edges: A -> B means that there every inducing path over **O** between A and B is out of A and into B, and A <-> B means that there is an inducing path over **O** that is into A and into B. This latter kind of edge can only occur when there is a latent common cause of A and B.

We can extend the concept of d-separability to inducing path graphs without modification, if we interpret directed paths in inducing path graphs as paths containing only edges with one arrowhead, and undirected paths as containing edges with either single or double arrowheads. If G is a directed acyclic graph, G' is the inducing path graph for G over **O**, and **X**, **Y**, and **S** are disjoint sets of variables included in **O**, then **X** and **Y** are d-separated by S in G' if and only they are d-separated by **S** in G.

However, there is one very important difference between d-separability relations in an inducing path graph and in a directed acyclic graph due to the existence of double-headed arrows in the former. In a directed acyclic graph over **O**, if A and B are d-separated by any subset of 0\{A,B} then A and B are d-separated either by a Parents(A) or Parents(B). This is not true in inducing path graphs. However, we have shown the following. (The numbering of the lemmas and theorems is taken from the Appendix.)

Let NA(A,B) (mnemonic for non-ancestor) be A if A is not an ancestor of B, and otherwise let it be B. (In an acyclic graph either A is not an ancestor of B or B is not an ancestor of A, so the vertex that is NA(A,B) is not an ancestor of the other vertex.)

Let V E DA(NA(A,B)) (double-arrow ancestor) for inducing path graph G' if and only if

1. V is an ancestor of A or B and V <-> NA($A_f$B) in G\ or
2. V is an ancestor of A or B and V <-> W for W e DA(NA(A,B)).

Let D-SEP(A,B) = Parents(NA(A,B)) u DA(NA(A,B)) u Parents(DA(NA(A,B))).

Lemma 14: In an inducing path graph G\ if A and B are not adjacent then A and B are d-separated by D-SEP(A,B).

The importance of this fact is that we can determine whether A and B are adjacent in an inducing path graph without determining whether A and B are dependent conditional on *all* subsets of O.

If O is not a causally sufficient set of variables, then although we can infer the existence of an inducing path between A and B if A and B are dependent conditional on every subset of 0\{A,B}, we cannot infer that either A is a direct cause of B relative to O, B is a direct cause of A relative to O, or there is a latent common cause of A and B. Nevertheless, the existence of an inducing path between A and B relative to O does contain information about the causal relationships between A and B, as the following lemma shows.

Lemma 4: If G is a directed acyclic graph over V that contains an inducing path relative to O (included in V) between A and B that is into B, then there is a directed path from A to B in G.

It follows from lemma 4 that if O is a subset of V and we can determine that there is an inducing path between A and B relative to O that is into B, then we can infer that A is a (possibly indirect) cause of B. Hence, if we can infer properties of the inducing path graph over O from the distribution over O, we can draw inferences about the causal relationships between variables, regardless of what variables we have failed to measure. In the next section we describe algorithms for inferring properties of the inducing path graph over O from the distribution over O.

## 5. Partially Oriented Inducing Path Graphs

The following algorithm constructs a partially oriented inducing path graph from conditional independence relations true of a distribution over **O**, (or under the assumption of faithfulness, d-separation relations between variables in **O**.) There is an edge between A and B in the partially oriented inducing path graph if and only if there is an edge between A and B in the inducing path graph. However, there are four kinds of edges in a partially oriented inducing path graph. An edge A -> B means that in the inducing path graph, there is an edge A -> B; an edge A <-> B means that in the inducing path graph, there is an edge A <-> B; an edge A o-> B means that in the inducing path graph there is an edge between A and B that is into B, but whether it is into A is left unspecified; and an edge A o-o B means that there is an edge between A and B in the inducing path graph, but whether it is into A or into B is left unspecified.

We use "*" as a metasymbol to represent any of the three kinds of ends (nothing, ">", or "o") that an edge in a partially oriented inducing path graph can have; the "*" symbol itself does not appear in a partially oriented inducing path graph. In addition, the notation A *-*B*-* C means that either or both of the edges between A and B and B and C are out of B, although whether the edge between A and B is out of B, or whether the edge between B and C is out of B is left unspecified.

The algorithm is divided into two main parts. First, the adjacencies in the partially oriented inducing path graph are determined. Then the edges are oriented in so far as possible. In order to state the algorithm, one more definition is needed.

In an inducing path graph G, E **is a discriminating vertex for C with respect to triangle A-B-C using path P and vertex B**, if and only either the edge between A and C is into A and the edge between B and A is out of A, or the edge between A and C is out of A and the edge between B and A is into A, and E is a closest vertex to A such that

1. E is not adjacent to B, and
2. P is an undirected path from E to A not containing B or C, and
3. for every vertex V on P, if V' is adjacent to V on P and between V and A on P, then V *-> V' in G, and
4. every vertex V on P between E and A is adjacent to B in G, and
5. except for the endpoints of P, V is a collider on P if and only if V is a parent of B.

Figure 2 illustrates the concept of a discriminating vertex.

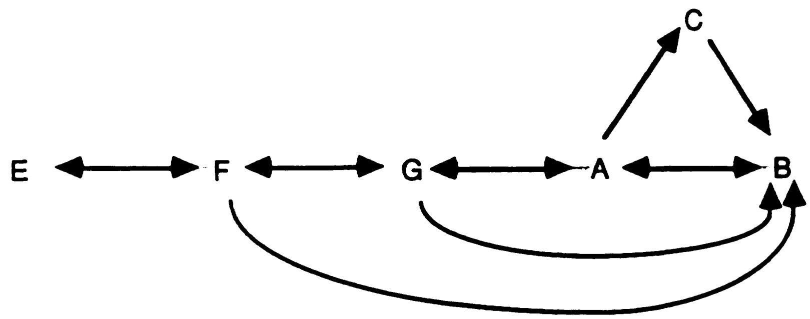



**Figure 2: E is a discriminating vertex for C with respect to triangle A-B-C using vertex B and path <E,F,G,A>**

## Causal Inference Algorithm[2]

If G is a directed acyclic graph over **V'**, and **V** is a subset of **V'**, the input to the algorithm is the set of d-separation relations involving just members of **V** that is true in G. Let $\mathbf{A}_Q(A,B)$ denote the set of vertices adjacent to A or to B in graph Q, except for A and B themselves. (Since the algorithm is continually updating Q, $\mathbf{A}_Q(A,B)$ is constantly changing as the algorithm progresses.)

A.) Form the complete undirected graph Q on the vertex set **V**.

B.) If A and B are d-separated by any subset **S** of **V**, remove the edge between A and B, and record **S** in **D**(A,B).

C.) Let F be the graph resulting from step B. Orient each edge as o-o. For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in F but the pair A, C are not adjacent in F, orient A *-* B *-* C as A *-> B <-* C if and only if B is not in **D**(A,C).

D. repeat

> If there is an edge A *-> B, and an edge B *-* C, A and C are not adjacent, and there is no arrowhead into B, then orient B *-* C as B -> C,
>
> else if there is a directed path from A to B, and an edge A *-* B, orient A *-* B as A *-> B,

---

[2]As we explain in more detail in Section 8, the Causal Inference Algorithm uses some ideas from the Inductive Causation algorithm described in Pearl and Verma (1990).

else if V is a discriminating vertex for M using R in triangle P-M-R then

if M is in D(V,R) then mark M as a non-collider on subpath P *-* M *-
* R

else orient P *-* M *-• R as P *-> M <- * R.

else if P *-> M *-« R then orient as P *-> M -> R.

until no more edges can be oriented.

Unfortunately, the Causal Inference Algorithm as stated is not practical for large numbers of variables because of the way the adjacencies are constructed. While it is theoretically correct to remove an edge between A and B from the complete graph if and only if A and B are d-separated by some subset of 0\{A,B}, this is impractical for two reasons. First, there are too many subsets of 0\{A,B} on which to test the conditional independence of A and B. Second, for discrete distributions, unless the sample sizes are enormous there are no reliable tests of independence of two variables conditional on a large set of other variables.

Remember, however, that in an inducing path graph if A and B are d-separated by any subset of O, then they are d-separated by D-SEP(A,B). Unfortunately, until we have actually constructed the inducing path graph we do not know which variables are in D-SEP(A,B). Nevertheless, as the partially oriented inducing path graph is constructed, we can determine that some variables are definitely not in D-SEP(A.B). This reduces the number and size of the subsets of O that have to be checked in order to determine whether A and B are adjacent in the inducing path graph.

We will determine which edges to remove from the complete graph in three stages. First, we will remove the edge between A and B if they are independent conditional on subsets of neighbors of A and B. This will eliminate many, but perhaps not all of the edges that are not in the inducing path graph. Second, we will orient edges by determining whether they collide or not. Third, using the partially oriented inducing path graph *n* that we have constructed thus far, we will form two sets of vertices Possible-D-SEP(A,B), and Possible-D-SEP(B,A) one of which includes every vertex that could possibly be in D-SEP(A,B). (We need two such sets because we do cannot determine from the partially oriented inducing path graph constructed thus far whether A is a descendant of B or B is a descendant of A.) Finally, we will remove the edge between A and B if A and B are independent conditional on any subset of either Possible-D-SEP(A,B,7c) or Possible-D-SEP(B,A,ic). Once we have obtained the correct set of adjacencies, we will unorient all of the edges, and then proceed to re-orient them. For a given partially constructed partially oriented inducing path graph *n,* Possible-D-SEP(A,B,ft) is defined as follows.

1. A u Neighbors(A) is in Possible-D-SEP(A,B,*).

2. If X and Z are in Possible-D-SEP(A,B,jc) and there is an edge between Y and X and between X and Z, but not between Y and Z in *, then if the edge between Y and X collides with the edge between X and Z in *n* at X, then Y is in Possible-D-SEP(A,B,7c).

3. If X and Z are in Possible-D-SEP(A,B,*) and there are edges between Y and X and Y and Z in *, then Y is in Possible-D-SEP(A,B,7c).

## Fast Causal Inference Algorithm

If G is a directed acyclic graph over V, and V is a subset of V, the input to the algorithm is the set of d-separation relations involving just members of V that is true of G. Let AQ(A,B) denote the set of vertices adjacent to A or to B in graph Q, except for A and B themselves. (Since the algorithm is continually updating Q, AQ(A,B) is constantly changing as the algorithm progresses.)

A.) Form the complete undirected graph Q on the vertex set V.

B.) n = 0.

repeat

repeat

select a pair of variables X and Y that are adjacent in Q such that AQ(X,Y) has cardinality greater than or equal to n, and a subset S(X,Y) of AQ(X,Y) of cardinality n, and if X and Y are d-separated by some subset of S(X,Y) delete the edge between X and Y from Q, and record the subset in D(X,Y)

until all variable pairs X and Y such that AQ(X,Y)has cardinality greater than n and all subsets S(X,Y) of AQ(X,Y) of cardinality n are exhausted,

n = n + 1.

until for each pair of adjacent vertices X, Y, AQ(X,Y) is of cardinality less than n.

C. Let P be the graph resulting from step B. Orient each edge as o-o. For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in P but the pair A, C are not adjacent in P, orient A *-* B *-* C as A *-> B <-* C if and only if B is not in D(A,C).

D. For each pair of variables A and B connected by an edge in F', if A and B are d-separated by any subset of **Possible-D-SEP**(A,B,F') or any subset of **Possible-D-SEP**(B,A,F') remove the edge between A and B.

The algorithm then orients an edge between any pair of variables X and Y as X o-o Y, and proceeds to re-orient the edges in the same way as steps C and D of the Causal Inference algorithm. The correctness of the algorithm is proved in the Appendix.

## 6. Preservation Theorems

Using the partially oriented inducing path graph output by the Fast Causal Inference Algorithm, and the inferences about graphs that can be drawn from inducing path graphs, we have the following two results.

**Theorem 2:** If $\pi$ is the partially oriented inducing path graph of directed acyclic graph G over **O**, and there is a directed path U from A to B in $\pi$, then there is a directed path from A to B in G.

A **semi-directed path from A to B** in partially oriented inducing path graph $\pi$ is an undirected path from A to B in which no edge contains an arrowhead pointing towards A (i.e. if X and Y are adjacent on the path, and X is between A and Y on the path, then there is no arrowhead at the X end of the edge between X and Y.)

**Theorem 3:** If $\pi$ is the partially oriented inducing path graph of directed acyclic graph G over **O**, and there is no semi-directed path from A to B in $\pi$, then there is no directed path from A to B in G.

As an example of the application of the Fast Causal Inference Algorithm, suppose that the causal structure depicted in Figure 3 is the true causal structure among a set of variables related to breathing dysfunction, and that all of the variables except those in boxes, (Environmental Pollution and Genotype) are measured. (We are not proposing this graph as a model of breathing dysfunction; we constructed it merely to illustrate the application of Theorems 2 and 3.) The partially oriented inducing graph over the measured variables constructed by the Fast Causal Inference Algorithm is depicted in Figure 4.

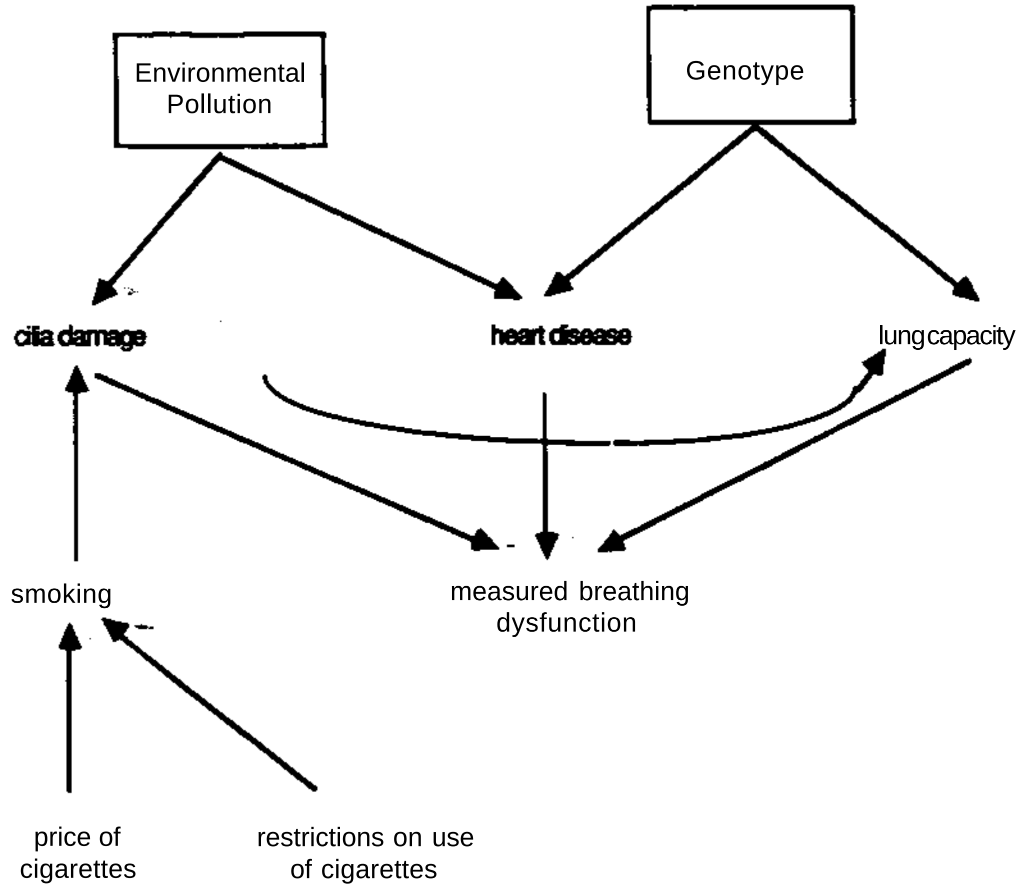



**Figure 3: Causal Graph of Breathing Dysfunction**

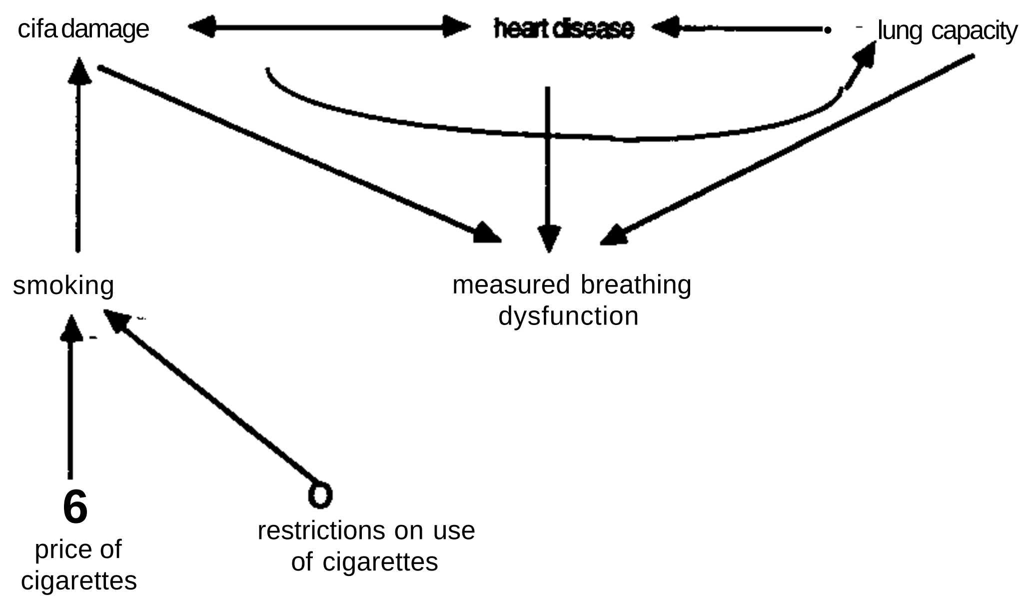



**Figure 4: Partially Oriented Inducing Graph of Breathing Dysfunction Over Measured Variables**

By applying Theorem 2, we infer that smoking does cause breathing dysfunction. By applying Theorem 3, we infer that smoking does not cause heart disease.

Note that in order to infer that smoking causes breathing dysfunction, it is necessary to measure two causes of smoking (whose collision at smoking orients the edge from smoking to cilia damage.) In general, this suggests that in the design of studies intended to determine if there is a causal path from variable A to variable B, it is useful to measure not only variables that might mediate the connection between A and B, but also to measure possible causes of A.

## 7. Open Questions

The following interesting open questions about partially oriented inducing path graphs remain.

Question 1: Let us say that the vertex **B is a collider on an undirected path U in partially oriented inducing path graph** $\pi$ if and only if there are edges C *-> B <-* A on U; otherwise B is a non-collider on U. **X and Y are p-separated by Z** in partially oriented inducing path graph $\pi$ if and only if there is an undirected path U between X and Y such that every non-collider on U is not in **Z** and every collider on U has a descendant in **Z**. If G is a directed acyclic graph with partially oriented inducing path graph $\pi$, are X and Y d-separated by **Z** in G if and only if X and Y are p-separated by **Z** in $\pi$?

Question 2: If $\pi$ is the partially oriented inducing path graph of some directed acyclic graph G, is there more information about the orientation of inducing paths in every directed acyclic graph with partially oriented inducing path graph $\pi$ than is represented in $\pi$?

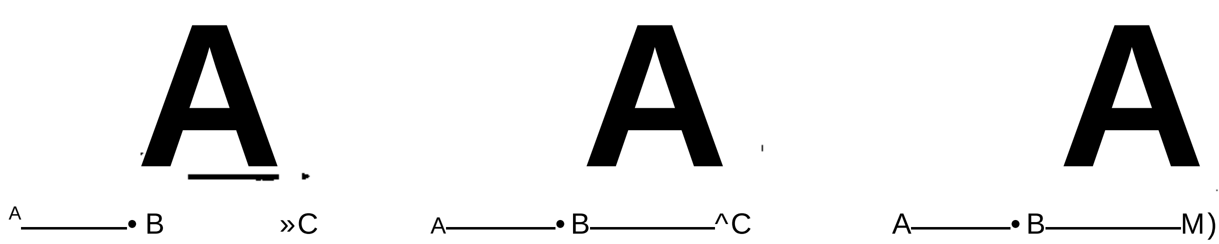
Question 3: If some distribution with the same conditional independencies as P(**O**) is the marginal of a distribution faithful to a directed acyclic graph, is P(**O**) the marginal of a distribution faithful to a directed acyclic graph?

## 8. Historical Note

In a series of papers (Pearl and Verma 1990, Pearl and Verma 1991, Verma and Pearl 1990, and Verma and Pearl 1991) Verma and Pearl describe an "Inductive Causation" algorithm that outputs a structure that they call a **pattern** (or sometimes a **completed hybrid graph**) of a directed acyclic graph G over a set of variables **O**. Their algorithm differs from the Causal Inference Algorithm in two main respects. First, early versions of the algorithm did not distinguish between A -> B and A o-> B; this distinction was introduced (in a different notation) in Spirtes and Glymour (1990a). Second it does not use discriminating vertices to orient any edges. (And unlike the Fast Causal Inference Algorithm it cannot be applied to large numbers of variables because it required testing the independence of some pairs of variables conditional on every subset of 0\{A,B}.) The most complete description of their theory appears in Pearl and Verma (1990). The key ideas of an inducing path, an inducing path graph, and the proof of (what we call) Theorem 1 all appear in this paper. Unfortunately, the two main claims that they make about patterns in this paper are both false.

In order to state their claims we need the following definitions. A pattern over O contains three kinds of edges: directed edges (e.g. A -> B), undirected edges (e.g. A - B), and bi-directed edges (e.g. A <-> B.) Directed paths and descendants are defined in a pattern the same way they are defined in acyclic directed graphs; however, an undirected path in a pattern can contain bidirected edges and undirected edges as well as directed edges. Edges between A and B, and B and C, collide at B on an undirected path in a pattern if both edges have arrowheads at B. A and B are **h-separated** by S in a pattern *n* if and only if there is no undirected path between A and B in which every collider has a descendant in S, and no non-collider is in S.

Verma and Pearl claimed first (lemma A.2 in their paper) that if *n* is the pattern of a directed acyclic graph G over O, and A and B are in 0, for all S included in O, A and B are h-separated by S in *n* if and only if A and B are d-separated by S in G. Their second claim (Theorem 2 in their paper) was that any two directed acyclic graphs with the same pattern over O were "equivalent", i.e. they entailed the same d-separation relations involving just variables in O. The following example shows that both of these claims are false.

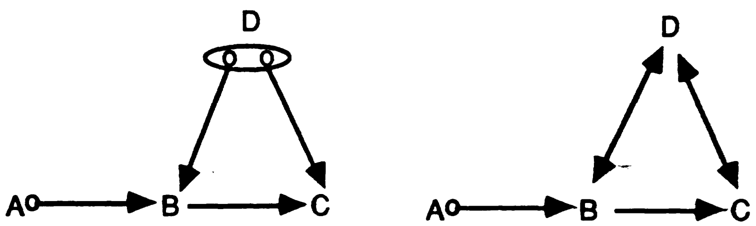



Graph G — Graph G' — Pattern of G and G' Over O - {A,B,C,D}

**Figure 5**

According to the Verma-Pearl algorithm both G and G' in Figure 5 have the pattern $\pi$ over **O** = {A,B,C,D} depicted in Figure 5. (The edge between C and D in the pattern is oriented from C to D in order to avoid a cycle involving B, C, and D.) However in G A and C are d-separated by {B,D} but not by {B}, whereas in G' A and C are d-separated by {B} but not by {B,D}. Hence G and G' have different d-separation relations among variables in **O** even though they have the same pattern. Moreover, A and C are h-separated by {B,D} in the pattern of G', even though they are not d-separated by {B,D} in G'.

Even though the patterns over **O** generated by the Verma-Pearl algorithm for G and G' are identical, the partially oriented inducing path graphs over **O** generated by the Causal Inference Algorithm for G and G' are different. This is because in both cases A is a discriminating vertex, and hence the edge between C and D is oriented differently in the partially oriented inducing path graph of G and the partially oriented inducing path graph of G'. The output of the Causal Inference Algorithm for G and G' over O is depicted in Figure 6.




Partially Oriented Inducing Path Graph of G over O = {A,B,C,D}

Partially Oriented Inducing Path Graph of G' over O = {A,B,C,D}

**Figure 6**

While the proofs of Verma and Pearl's two main claims about patterns contained fallacies, we have used several of their proof techniques in our proofs.

## References


J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988.

J. Pearl and T. Verma, "A Formal Theory of Inductive Causation" Technical Report, R-155, Department of Computer Science, University of California at Los Angeles, October, 1990.

J. Pearl and T. Verma, "A Theory of Inferred Causation", in *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, ed. by J. Allen, R. Fikes, and E. Sandewall, Morgan Kaufmann, San Mateo CA, 1991.

P. Spirtes, C. Glymour and R. Scheines, "Causality from Probability" Proceedings of the Conference on Advanced Computing for the Social Sciences, Williamsburg, Va. 1990.

P. Spirtes, C. Glymour and R. Scheines, "Causality from Probability" in G. McKee, ed. *Evolving Knowledge in Natural and Artificial Intelligence*, Pitman, 1990.

P. Spirtes and C. Glymour, "Causal Structure among Measured Variables Preserved with Unmeasured Variables", Laboratory for Computational Linguistics Technical Report No. CMU-LCL-90-5, August, 1990a.

P. Spirtes and C. Glymour, "An Algorithm for Fast Recovery of Sparse Causal Graphs", *Social Science Computer Review*, 9, 1991.

P. Spirtes, C. Glymour and R. Scheines, *Causality, Statistics and Search*, forthcoming.

T. Verma and J. Pearl, "On Equivalence of Causal Models" Technical Report, R-150, Department of Computer Science, University of California at Los Angeles, April 1990.

T. Verma and J. Pearl, "Equivalence and Synthesis of Causal Models" Technical Report, R-150, Department of Computer Science, University of California at Los Angeles, March, 1991.

# Appendix

## A.1 Inducing Paths

In a directed acyclic graph G over a set of variables V, an undirected path U between A and B is an **inducing path over a subset O** of **V** if and only if every member of O on U is a collider on U, and every collider is an ancestor of either A or B. We will sometimes refer to members of **O** as **observed** variables. (The lemmas and theorem in section A.1 are minor variations on lemmas and theorems from Verma and Peart 19?.)

We will use lemmas 1 through 3 to demonstrate that A and B are not d-separated by any subset of 0\{A,B} if and only if there is an inducing path over the subset O between A and B. If U is an undirected path from A to $B_v$ and X and Y occur on U, then we will denote the subpath of U from X to Y as U(X,Y).

**Lemma 1:** In a directed acyclic graph G, if there is an inducing path relative to O between A and B that is not into A then for any subset Z of 0\{A,B} there is a d-connecting path between A and B given Z that is not into A.

Proof. Let U be an inducing path between A and B that is not into A. Every observed vertex on U is a collider, and every collider is an ancestor of either A or B. If there are no observed variables on U then there are no colliders on U and U d-connects A and B given Z. Suppose then that there are colliders on U, and that W is the first collider on U after A. W is an ancestor of B, because if it is an ancestor of A there is a cycle in G. U(A,W) is a directed path from A to W. Let D be a directed path from W to B. Because G is acyclic, D does not intersect U(A,W) except at W. If D does not contain any variable in Z, then the concatenation of U(A,W) with D is an path that d-connects A and B given Z that is not into A.

Suppose then that D does contain some member of Z. Because U is out of A, W is a descendant of A. W has a descendant in Z because W is the source of D, and D contains a member of Z by hypothesis. It follows that every collider on U that is an ancestor of A has a descendant in Z. If every collider on U after A has a member of Z as a descendant then U d-connects A and B given Z, and is out of A. Otherwise, let R be the first collider after A that is an ancestor of B that does not have a descendant in Z. Let T be the point of intersection of U(A,R) and a directed path D from R to B closest to A on U. The concatenation of U(A,T) and D(T,B) is an undirected path out of A. T is not a collider on the concatenation of U(A,T) and D(T,B) because D(T,B) contains an edge out of T. T is not in Z because R does not have a descendant in Z. Hence the concatenation of U(A,T) and D(T,B) d-connects A and B given Z.

*QED.*

**Lemma 2: In a directed acyclic graph G, if there is an inducing path U relative to 0 between A and B that is into A then for every subset Z of 0\{A,B} there is a d-connecting path between A and B given Z that is into A.**

**Proof. If every collider on U has a descendant in Z, then U is a d-connecting path between A and B given Z that is into A. Suppose then that there is a collider that does not have a descendant in Z. If the first collider W on U after A that does not have a descendant in Z is an ancestor of B, let D be a directed path from W to B. Let T be the vertex on U(A,W) that is the point of intersection of U(A,W) and D closest to A. The concatenation of U(A,T) and D(T,B) is into A. T is not a collider on the concatenation of U(A,T) and D(T,B) because D(T,B) contains an edge out of T. T is not in Z because W does not have a descendant in Z. Hence the concatenation of U(A,T) and D(T,B) d-connects A and B given Z.**

**If all colliders that do not have a descendant in Z are ancestors of A, let R be the collider closest to B on U that does not have a descendant in Z. There is a directed path D from R to A. Let T be the vertex on U(B,R) closest to B that intersects D. The concatenation of U(B,T) and D(T,A) is an undirected path from B to A. T is not a collider on the concatenation of U(B,T) and D(T,A) because D(T,A) contains an edge out of T. T is not in Z because R does not have a descendant in Z. Hence, the concatenation of U(B,T) and D(T,A) d-connects A and B given Z.**

**Otherwise let R be the first collider after A on U that is an ancestor of B that does not have a descendant in Z and let S be the first collider before R on U that is an an ancestor of A that does not have a descendant in Z. There is a directed path D1 from R to B and a directed path D2 from S to A. Let T1 be the vertex on U(S,R) closest to R on U that intersects D2. The concatenation of U(R,T1) and D2(T1,A) is an undirected path $U^1$ from R to A. T1 is not a collider on LP because D2(T1,A) contains an edge out of T1. T1 is not in Z because S does not have a descendant in Z. Every vertex on U' is either a collider with a descendant in Z, or a non-collider not in Z. Let T2 be the point of intersection of D1 and $U^1$ closest to R on U\ The concatenation of D1(T2,B) and $U^f$(A,T2) is a path into A. T2 is not a collider on the concatenation of D1(T2,B) and U'(A,T2) because D2(T2,B) contains an edge out of T2. T2 is not in Z because R does not have a descendant in Z. Hence every vertex on the concatenation of D1(T2,B) and U'(A,T2) is either a collider with a descendant in Z or a non-collider not in Z. So the concatenation of D1(T2,B) and $U^f$(A,T2) d-connects A and B given Z. Q.E.D.**

Lemma 3: In a directed acyclic graph G, if A and B are not d-separated by any subset Z of 0\{A,B} then there is an inducing path over the subset O between A and B.

Proof. Suppose that A and B are not d-separated by any subset Z of 0\{A,B}. If there is an edge between A and B in G, then that edge is an inducing path. Suppose then that there is no edge between A and B. Let A(A,B) be the union of the ancestors of A or B. If there is a path U that d-connects A and B given A(A,B) n O then every collider on U is an ancestor of a member of A(A,B) n O, and hence an ancestor of A or B. Every vertex on U is an ancestor of either A or B or a collider on U, and hence every vertex on U is an ancestor of A or B. If U d-connects A and B given A(A,B) n O, then every member of A(A,B) n O that is on U is a collider. Since every vertex on U is in A(A,B), every member of of O that is on U is a collider. Q.E.D.

Theorem 1: In a directed acyclic graph G over $V_f$ A and B are not d-separated by any subset Z of 0\{A,B} if and only if there is an inducing path over the subset O between A andB.

Proof. This follows from lemmas 1, 2, and 3. Q.E.D.

The following pair of lemmas state some basic properties of inducing paths.

Lemma 4: If G is a directed acyclic graph over V that contains an inducing path relative to 0 (included in V) between A and B that is not into A, then there is a directed path from A to B in G.

Proof. Let U be an inducing path between A and B relative to O that is not into A. If U does not contain a collider, then U is a directed path from A to B. If U does contain a collider, let C be the first collider after A. By definition of inducing path, there is a directed path from C to B or C to A. There is no path from C to A because there is no cycle in G; hence there is a directed path from C to B. Because U is out of A, and C is the first collider after A, there is a directed path from A to C. Hence there is a directed path from AtoB. Q.E.D.

Lemma 5: If G is a directed acyclic graph over V that contains an inducing path relative to 0 (included in V) between A and B that is not into A, then every inducing path relative to O between A and B is into B.

Proof. By lemma 4, if there an inducing path not into A, and an inducing path not into B, there is a cycle in G. Q.E.D.

## A.2 Inducing Path Graphs

**In** an **inducing path graph G' for directed acyclic graph G over a subset of variables O** there is an edge between variables A and B with an arrowhead at A if and only if A and B are in O, and there is an inducing path in G between A and B relative to O that is into A.

Unlike a directed acyclic graph, an inducing path graph can contain double-headed arrows. By lemma 5, however, it does not contain any edges with no arrowheads. If there is an inducing path between A and B in G that is into A, then the edge between A and B in G' is into A. However, if there is an inducing path between A and B in G that is out of A, it does not follow that the edge in G* between A and B is out of A. Only if no inducing path between A and B in G is into A is the edge between A and B in G' out of A. The definitions of directed path, d-separability, inducing path, collider, ancestor, and descendant are the same as those for directed graphs, i.e. a directed path in an inducing path graph, as in an acyclic directed graph, contains only directed edges (e.g. A -> B). However, an undirected path in an inducing path graph can contain either directed edges, or bi-directed edges (e.g. C <-> D.) Also, if A <-> B in an inducing path graph, A is not a parent of B. In an inducing path graph G we will say that an edge A is **into** B if A <->B or A <- B, and that it is **out of** A if and only if A->B. Note that if G is a directed acyclic graph, and G' the inducing path graph for G over 0, then there are no directed cycles in G\

The next series of lemmas show that two variables A and B in O are d-separated in G by a subset of O if and only if they are d-separated in the inducing path graph $G^v$ by a subset ofO.

**Lemma 6:** If $G^1$ is the inducing path graph for G over O and there is a directed path from A to B in G\ then there is a directed path from A to B in G.

Proof. Suppose there is a directed path D from A to B in G. Let X and Y be any two vertices along the directed path. There is a directed edge from X to Y in G\ By the definition of inducing path graph, there is an inducing path between X and Y in G that is not into X. Hence by lemma 4, there is a directed path from X to Y in G.

In G, the concatenation of the directed paths between vertices that are adjacent on D contains a directed path from A to B. Q.E.D.

**Lemma 7:** In a directed acyclic graph G, if there is a sequence S of vertices from X to Y such that,

1. for each pair of adjacent vertices V and W in the sequence there is a path that d-connects V and W given Z\{V,W}, and

2. if a vertex Q in the sequence is in Z, then the d-connecting paths collide at Q, and

3. if for three vertices V, W, Q in the sequence the d-connecting paths between V and W, and W and Q collide at W then W has a descendant in Z,

then there is a path U in G that d-connects X and Y given Z. In addition, if all of the edges in all of the d-connecting paths that contain X are into X, then U is into X.

Proof. Let U' be the concatenation of all of the d-connecting paths in the order of the sequence S. U' may not be an undirected path, because it could contain some vertices more than once. Let U be the result of removing all of the cycles from U'. If each edge on U' that contains X is into X, then U is into X, because each edge in U is an edge in U'. We will prove that U d-connects X and Y given Z.

We will call an edge in U containing a given vertex V an **endpoint edge** if V is in the sequence S, and the edge containing V occurs on the d-connecting path between V and its predecessor or successor in S; otherwise the edge is an **internal edge.**

Every member R of Z that is on U is a collider on U. Suppose first that an edge on U is an endpoint edge. If there is an endpoint edge containing R then it is into R because by assumption the d-connecting paths collide at R. If an edge on U is an internal edge then it is into R because it is an edge on a path that d-connects two variables A and B not equal to R given Z\{A,B}, and R is in Z. All of the edges on the original d-connecting paths are into R, and hence the subset of those edges that occur on U are into R.

We will now show every collider R on U has a descendant in Z. R is not equal to X or Y, because the endpoints of a path are not colliders along the path. If R is a collider on any of the original d-connecting paths then R has a descendant in Z because it is an edge on a path that d-connects two variables A and B not equal to R given Z\{A,B}. If R is a collider on two endpoint edges then it has a descendant in Z by hypothesis. Suppose then that R is not a collider on a d-connecting path from A to B, and not a collider on a d-connecting path from C to D, but after cycles have been removed from the concatenation of the d-connecting paths, R is a collider on U. In that case U' contains an undirected cycle containing R. Because G is acyclic, the undirected cycle contains a collider. Hence R has a descendant that is a collider on U'. Each collider on U' has a descendant in Z. Hence R has a descendant in Z. Q.E.D.

**Lemma 8:** If G' is the inducing path graph for G over **O**, and there is a path d-connecting A and B given **Z** in G', then there is a path d-connecting A and B given **Z** in G.

Proof. Suppose that U d-connects A and B in G'. For every edge in U between R and S, there is an inducing path in G between R and S. By lemmas 1 and 2, there is a path that d-connects R and S given **Z**\{R,S} in G.

If there are vertices R, S, and T on U such that R and S are adjacent on U, and S and T are adjacent on U, and S is in **Z**, then S is a collider on U. By the definition of inducing path graph, in G there are inducing paths between R and S, and S and T, that are both into S. By lemmas 1 and 2, in G there are d-connecting paths given **Z** between R and S, and S and T, that are both into S.

If there are vertices R, S, and T on U such that R and S are adjacent on U, and S and T are adjacent on U, and S is a collider on U, then S has a descendant in **Z**. By the definition of inducing path graph, in G there are inducing paths between R and S, and S and T, that are both into S. By lemmas 1 and 2, in G there are d-connecting paths given **Z** between R and S, and S and T, that are both into S. If S has a descendant in **Z** in G' then by lemma 6 it has a descendant in Z in G.

By lemma 7, there is a path in G that d-connects X and Y given **Z**. Q.E.D.

**Lemma 9:** If G is a directed acyclic graph, and in G there is a sequence of vertices M starting with A and ending with C such that for every pair of vertices I and J adjacent in M there is an inducing path W relative to **O** between I and J, and if J ≠ C then W is into J, and if I ≠ A then W is into I, and I and J are ancestors of either A or C, then there is an inducing path between A and C. Furthermore, if the inducing path between A and its successor in M is into A, then the inducing path between A and C is into A.

Proof. Suppose that in G there is a sequence M of vertices starting with A and ending with C such that for every pair of vertices I and J adjacent in M there is an inducing path W relative to **O** between I and J, and if J ≠ C then W is into J, and if I ≠ A then W is into A, and I and J are ancestors of either A or C. Let T' be the concatenation of these inducing paths in G. T' may not be an undirected path because it might contain undirected cycles. Let T be an undirected subpath of T' between A and C. We will now show that except for the endpoints, every vertex in **O** on T is a collider, and every collider on T is an ancestor of A or C.

Suppose V is a vertex in **O** that is on T but that is not equal to A or C. Let W be one of the inducing paths between adjacent vertices in M that is a subpath of T'. We will show every edge on W that contains V is into V. If V is not an endpoint of W, then every edge on W is into V because V is a collider on W. If V is an endpoint of W, then the edge containing V on W is into V by hypothesis. It follows that V is a collider on T, because the edges on T are a subset of the edges on inducing paths between variables adjacent in M.

Let R and S be the endpoints of W. We will now show that every vertex on W is either an ancestor of A or an ancestor of C. By hypothesis, R is an ancestor of either A or C, and S is an ancestor of either A or C. Because W is an inducing path, every collider on W is an ancestor of either R or S, and hence an ancestor of either A or C. Every non-collider on W is either an ancestor of R or S, or an ancestor of a collider on W. Hence every vertex on W is an ancestor of either A or C. It follows that every collider on T is an ancestor of A or C, because the vertices on T are a subset of the vertices on the sequence of inducing paths.

By definition, T is an inducing path between A and C relative to **O**. Suppose the inducing path between A and its successor in the sequence is into A. If the edge on T with endpoint A comes from the inducing path on which A is an endpoint, then T is into A because by hypothesis that inducing path is into A. If the edge on T with endpoint A comes from an inducing path in which A is not an endpoint of the path, then T is into A because A is in **O**, and hence a collider on every inducing path for which it is not an endpoint. Q.E.D.

**Lemma 10:** If G' is the inducing path graph for G over **O**, **Z** is included in **O**, and there is a path U d-connecting X and Y given **Z** in G, then there is a path V d-connecting X and Y given **Z** in G'. Furthermore, if U is into X in G, then V is into X in G'.

Proof. Suppose that in G with inducing path graph G' that U is a path d-connecting X and Y given Z. We will use the following algorithm to construct two sequences of vertices, Ancestor, and D-Path. The vertices in D-Path are always observed (i.e. vertices in **O**), but might not be on U; vertices in Ancestor are always on the path U, but might not be observed. For any sequence of vertices, Sequence(n) refers to the nth vertex in Sequence.

**Algorithm D-Path**

Ancestor(0) = <X>.

D-connect(0) = <X>.

n = 0.

repeat

```
if Ancestor(n) * D-path(n) then
    if on U there is no collider C after Ancestor(n) that has D-path(n) as the first observed variable on a directed path from C to a member of Z, then Ancestor(n+1) = D-path(n+1) • first observed variable on U after Ancestor (n);
    else
        let C2 be the collider closest to Y that has D-path(n) as the first observed variable on a directed path from C2 to a member of Z;
        if there is no collider between C2 and the first observed variable after C2 then Ancestor(n+1) = D-path(n+1) = first observed variable after C2;
        else let C1 be the first collider after C2, let Ancestor(n+1) = C1 and D-path(n+1) = the first observed variable on a directed path from C1 to a member ofZ;
else if Ancestor(n) = D-path(n) then
    if there is no collider between Ancestor(n) and the next observed variable V on U, Ancestor(n+1) = D-path(n+1) = V;
    else Ancestor(n+1) = first collider after Ancestor(n) and D-path(n+1) = first observed variable on a path from Ancestor(n+1) to a member of Z;
n = n + 1.
until Y is the last member of D-path(n).
```

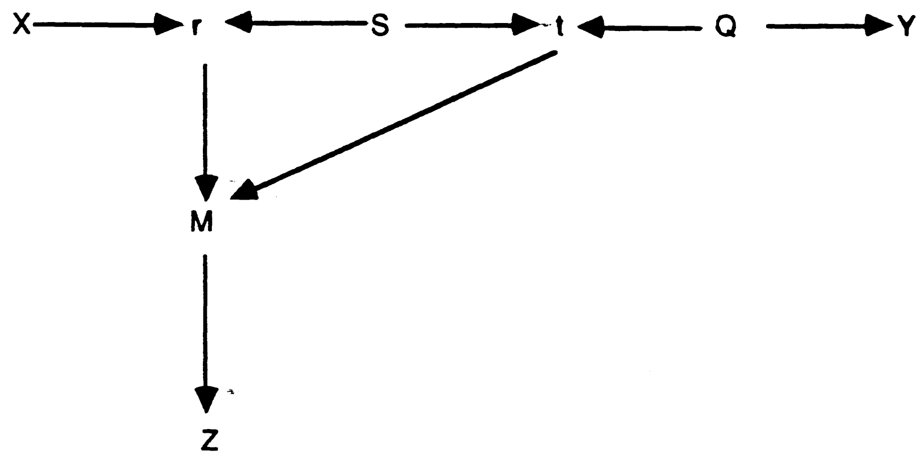



**Figure 7**

For example, when the algorithm is applied to the graph in Figure 7 (where the lower case vertices are not observed), U = <X,r,S,t,Q,Y>, and the result is Ancestor = <X,r,Q,Y> and D-path = <X,M,Q,Y>.

We will now show that either D-path d-connects X and Y given Z in G', or some other path in G' d-connects X and Y given Z.

All of the vertices in D-path are observed variables, and hence in G'. By the way that D-path is constructed, each adjacent pair of vertices A and B in D-path is connected in G by a trek T(A,B) that contains no observed variables, except for the endpoints. T(A,B) is constructed out of subpaths of U, and subpaths of directed paths from colliders on U to vertices in Z. T(A,B) is an inducing path in G, and hence each adjacent pair of vertices in D-path is adjacent in G'. The method of construction of D-path makes D-path acyclic. It follows that D-path is an undirected path from X to Y in G'.

If W is not a collider on D-path, then W is on U in G, and is not a collider on U. It follows that W is not in Z.

We will now show that we can transform D-path into a path D-path' in G' in which every collider B on D-path' is induced by two treks that collide at B in G. We will say that a vertex B that is a collider on D-path, but is not induced by two treks in G that collide at B is a collider on D-path but not in G. Let B be the first vertex on D-Path after X that is a collider on D-path but not in G. At least one of the two inducing treks T(A,B) or T(B,C) is out of B. Suppose without loss of generality that T(B,C) is out of B. It follows that B is an ancestor of C. In addition since there is an arrowhead at B in G', there is an inducing path between B and C that is into B and C. For any vertex W, we will call the vertex adjacent to W on D-path and between W and Y (if it exists) the successor of W on D-path.

Let E be the vertex on D-path cbsest to C and between X and C such that either

a. the predecessor of E is not into E on D-path, or

b. there is no edge from E to C in G\ or

c. the edge from E to C is not into C in G\ or

d. E is a collider along D-path

if such an edge exists, otherwise let E = X. For every vertex M between E and B, the predecessor of M is into M on D-path and there is an edge from M to C that is into C and M is not a collider along D-path. See Figure 8.

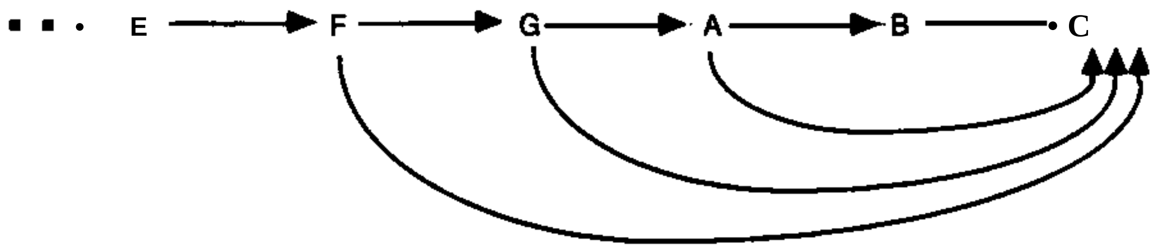



Figure 8

If any of the M-C edges is not into M in G\ then since M is not a collider along D-path, when the M-C edge is substituted for the subpath of D-path from M to C, every vertex on the new path up to C is a collider on D-path if and only if it is a collider in G.

Suppose then that for each M between E and B, M is a not collider on D-path but the M-C edge is into M. See Figure 9.

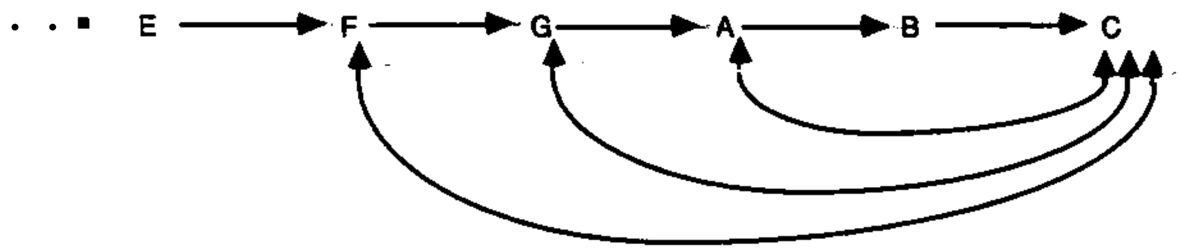



**Figure 9**

By hypothesis, the edge from E to its successor is into E's successor on D-path. It follows that E is an ancestor of B. By hypothesis, B is an ancestor of C, so E is an ancestor of C. If E = A, then by hypothesis there is a pair of inducing paths in G from E to its successor (B), and from its successor to C that are into E. Let F be the successor of E on D-path. If E * A, then by the definition of E there is an edge from E to F that is into F, and hence an inducing path in G between E and F that is into F. By assumption there is an edge between F and C that is into F, and hence an inducing path in G between F and C that is into F. Hence by lemma 9 there is an edge from E to C that is into C. Since there is an edge from E to C that is into C, by hypothesis either E is a collider on D-path, or the edge from the predecessor of E to E is not into E, or E = X. If the edge between E and the predecessor of E on D-path is not into E, then when the edge between E and C is substituted for the subpath from E to C, every vertex on the new path up to C is a collider on D-path if and only if it is a collider in G. If E is a collider on D-path then the edge between E and C is into E, and when the edge between E and C is substituted for the subpath from E to C, every vertex on the new path up to C is a collider on D-path if and only if it is a collider in G. If E = X, when the edge between E and C edge is substituted for the subpath from E to C, every vertex on the new path up to C is a collider on D-path if and only if it is a collider in G. Furthermore, since the edge from E to C is into C, and the subpath of D-path from E to C is into C, C is a collider along D-path if and only if E is a collider along the path that results from substituting the edge between E and C for the subpath from E to C.

By repeating this process we can transform D-path into D-path', so that every collider along D-path[1] is a collider in G. Every collider on D-path is a descendant of a collider on U, and hence the ancestor of a member of Z in G. Suppose however that some collider B on D-path is an ancestor of some vertex in Z in G but not in G\ We will show how to transform D-path[1] into a path in which every collider has a descendant in Z in G\ Let P be a directed path in G from B to a member Z of Z. In G\ let P be the path from B to Z that consists of the observed variables on P in the order in which they occur. P exists in G' because in G the directed path between any two observed variables on P is an inducing path. The only way that P is not a directed path in G' from B to Z is if some edge between on P is a double-headed arrow. Let S be the vertex on P* closest to B for which S is not a descendant of B in G'. Let R be the vertex on P adjacent to S between S and B if there is one, and otherwise let R = B. R and S are the endpoints of a double-headed arrow on P. In that case there are two inducing paths between R and S in G; one is a directed path from R to S, and the other is an inducing path that is into R and into S. Consider the undirected path V that is contained in the concatenation of D-path' from X to B and the subpath of P from B to S. By the same argument that showed the existence of D-path\ there is a vertex E on V such that there is an edge from E to S that is into S, and is into E if and only if the edge between E and the successor of E on V is into E. E is not on P because there is an edge from E to S, and hence a directed path from B to S. It follows that E is on D-path$^1$. Similarly, there is a vertex F between B and Y such that there is an edge from F to S that is into S, and is into F if and only if the edge between F and the predecessor F is into F. By substituting the edges from E to S and F to S for the subpath of D-path' between E and F we have constructed a path in G* from X to Y in which every collider up to S has a descendant in Z. If S does not have Z as a descendant in G' repeat this process until a vertex T along P is reached which does have Z as a descendant in G. We have thus removed a collider that did not have a descendant in Z from D-path. By repeating the process we can construct a path in G$^1$ in which every collider has a descendant in Z. Q.E.D.

The following lemmas state some key properties of d-separability in inducing path graphs.

**Lemma 11:** If G* is the inducing path graph for directed acyclic graph G over O, A and B are d-connected by every subset of 0\{A,B} in G' if and only if there is an inducing path over O between A and B in G\

Proof. The proofs of each of the lemmas used to prove Theorem 1 are also proofs of the corresponding lemmas for inducing paths in inducing path graphs. Q.E.D.

**Lemma 12:** If G' is the inducing path graph for directed acyclic graph G over O and there is an inducing path U over O between A and C in G\ then there is an edge between A and C in G$^f$. Furthermore if U is into A in G\ then the edge between A and C is into A in G\

Proof. If there is an inducing path between A and C in G\ then by lemma 11 in G' A and C are d-connected by every subset of 0\{A,C}. Hence by lemma 8, A and C are d-separated by every subset of 0\{A,C} in G. By Theorem 1, there is an inducing path between A and C in G. It follows by definition that there is an edge between A and C in G'. Furthermore, if any inducing path between A and C over O in G is into A, then the edge between A and C is into A in $G^V$ by definition. Q.E.D.

Let a total order Totalfa) of variables in an inducing path graph *n* be **acceptable** if and only if whenever there is a directed path from A to B in *n,* A precedes B in Total(ic).

Let W E **Double-arrow(V,Total(7c))** in *n* if and only if either

1. there is an edge V <-> W in $K_9$ and W precedes V in Total(ic) or

2. there is a variable U in **Double-arrow(V,Total(ic))$_f$** U <-> W in *, and U precedes V in Total(rc).

For example in Figure 10, if Total(ic) = <$X_f$S,$t_f$r,$M_f$$Z_f$$Q_f$Y>, then **Double-**arrow(Y,Total(K)) = {Q.t.S}, and if Total(w) = <$X_f$$S_f$t,$r_f$$M_f$$Z_f$$Y_f$Q>$_f$ then Double-arrow(YJotal(*)) = 0.

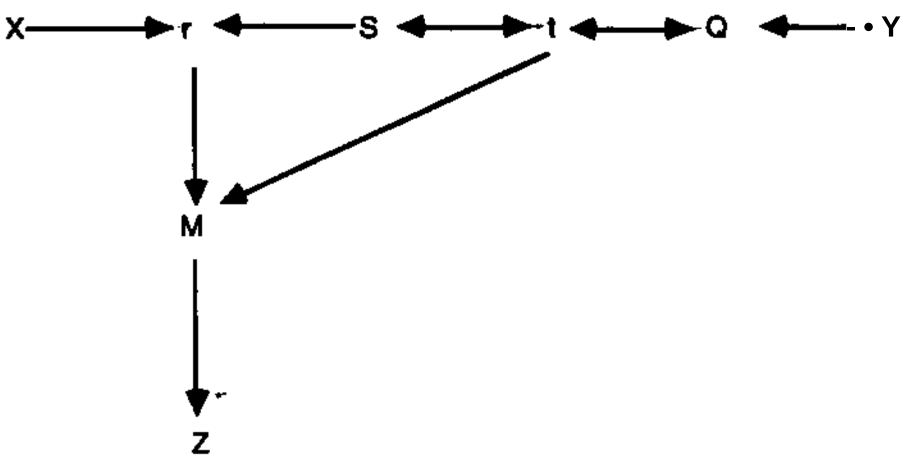



**Figure 10**

For inducing path graph *n* and acceptable total ordering Total(71), let Predecessors(Total(ft),V) equal the set of all variables that precede V (not including V) according to Total(Tc).

For inducing path graph $\pi$ and acceptable total ordering Total($\pi$), let **SP**(Total($\pi$),$\pi$,V) (separating predecessors of V in $\pi$ for ordering Total($\pi$)) = **Parents**(V) $\cup$ **Double-Arrow**(V) $\cup$ **Parents**(**Double-Arrow**(V)).

**Lemma 13:** For induced path graph G and acceptable total ordering Total($\pi$), **SP**(Total($\pi$),$\pi$,X) d-separates X from **Predecessors**(Total($\pi$),X)**SP**(Total($\pi$),$\pi$,X).

Proof. Suppose on the contrary that there is a path U that d-connects some V in **Predecessors**(Total($\pi$),X)**SP**(Total($\pi$),$\pi$,X) to X given **SP**(Total($\pi$),$\pi$,X). There are three cases.

First suppose U has an edge into X that is not a double-headed arrow. Then some parent R of X is on U, and is not a collider on U. R is in **SP**(Total($\pi$),$\pi$,X) and hence is not equal to V. Because R is not a collider on U, U does not d-connect V to X given **SP**(Total($\pi$),$\pi$,X), contrary to our assumption.

Next suppose U has an edge out of X. Since V is in **Predecessors**(Total($\pi$),X)**SP**(Total($\pi$),$\pi$,X) it precedes X in Total($\pi$); hence there is no directed path from X to V. It follows that U contains a collider. Let the first collider after X be R. R is a descendant of X, and the descendants of R are descendants of X. It follows that no descendant of R (including R itself) is in **SP**(Total($\pi$),$\pi$,X), and hence U does not d-connect V and X, contrary to our assumption. See Figure 11.

$$
\begin{tikzcd}
X \arrow[r] & R \arrow[d] & S \arrow[l] \arrow[r] & V \\
& T & &
\end{tikzcd}
$$

**Figure 11**

Finally, suppose that U contains a double-arrow into X. Because U d-connects X and V, each collider along U has a descendant in **SP**(Total($\pi$),$\pi$,X). There is a subpath of double-headed arrows with endpoint X on U. If some variable R in the subpath does not precede X in Total($\pi$), neither does any of its descendants. In that case, neither R nor its descendants is in **SP**(Total($\pi$),$\pi$,X), and R is a collider along U. Hence U does not d-connect V and X contrary to our assumption. If on the other hand each variable on the subpath of double-headed arrows with endpoint X on U precedes X in Total($\pi$) let W be the last variable in the subpath. W $\neq$ V because W is in **SP**(Total($\pi$),$\pi$,X). There is either an edge into or out of W. If there is an edge into W, let Y be the tail of the edge into W. Y is a parent of W, and hence also in **SP**(Total($\pi$),$\pi$,X) and not equal to V. But then Y is not a collider on U and is in **SP**(Total($\pi$),$\pi$,X), so U does not d-connect X and V given **SP**(Total($\pi$),$\pi$,X), contrary to our assumption. If there is an edge out of W, W is in **SP**(Total($\pi$),$\pi$,X) and is not a collider along U, so U does not d-connect X and V given **SP**(Total($\pi$),$\pi$,X), contrary to our assumption. Q.E.D.

Let **NA**(A,B) (mnemonic for non-ancestor) be A if A is not an ancestor of B, and otherwise let it be B. (In an acyclic graph either A is not an ancestor of B or B is not an ancestor of A, so the vertex that is NA(A,B) is not an ancestor of the other vertex.)

Let V $\in$ **DA**(NA(A,B)) (double-arrow ancestor) for inducing path graph G if and only if

1. V is an ancestor of A or B and V <-> NA(A,B) in G, or
2. V is an ancestor of A or B and V <-> W for W $\in$ **DA**(NA(A,B)).

Let **D-SEP**(A,B) = **Parents**(NA(A,B)) $\cup$ **DA**(NA(A,B)) $\cup$ **Parents**(**DA**(NA(A,B))).

**Lemma 14:** In an inducing path graph G, if A and B are not adjacent then A and B are d-separated by **D-SEP**(A,B).

Proof. Suppose that A and B are not adjacent, and suppose without loss of generality that NA(A,B) = A. Let the total order Total(G) on the variables in G be such that all ancestors of A and B are prior to A, B is prior to A, and all vertices that are not ancestors of A and not ancestors of B are after A. Then **SP**(Total(G),G,A) = **D-SEP**(A,B). Hence by lemma 25, if B is not in **D-SEP**(A,B) then **D-SEP**(A,B) d-separates A from B in G. If A and B are not adjacent then B is in **D-SEP**(A,B) if and only if there is a path P of double-headed arrows from A to B such that each vertex on the path is an ancestor of A or B, or it is a parent of the endpoint of such a path P. In either case there is a path from A to B in which each vertex except the endpoints is a collider on the path, and each collider is an ancestor of A or B. But then there is an inducing path between A and B, and because G is an inducing path graph, by lemma 12 A and B are adjacent, contrary to our assumption. Q.E.D.

In an inducing path graph G, E **is a discriminating vertex for C with respect to triangle A-B-C using path P and vertex B**, if and only either the edge between A and C is into A and the edge between B and A is out of A, or the edge between A and C is out of A and the edge between B and A is into A, and E is a closest vertex to A such that

1. E is not adjacent to B, and
2. P is an undirected path from E to A not containing B or C, and

3. for every vertex V on P, if V is adjacent to V on P and between V and A on P, then V *-> V in G, and

4. every vertex V on P between E and A is adjacent to B in G, and

5. except for the endpoints of P, V is a collider on P if and only if V is a parent of B.

**Lemma 15:** In an inducing path graph G\ if E is a discriminating vertex for C with respect to triangle A-B-C using path P and vertex B then for any set S such that E and B are d-separated given S, C is a collider in A-B-C if and only if C is not in S.

Proof. Suppose E is a discriminating vertex for C with respect to triangle A-B-C using path P and vertex B. By definition E is not adjacent to B. It follows that there is some set S such that E and B are d-separated by S. First we will prove by induction that for any set S such that E and B are d-separated by S, P d-connects E and A given S\A by showing that S\A contains every collider along P and no non-collider along P.

**Base Case:** Let F be the first vertex after E on P. If F = A then the condition is trivially satisfied because there are no colliders or non-collider on P. Suppose then that F * A. If F is a collider on P then F is a parent of B. It follows that the path concatenating the edges between E and F and between F and B, d-connects E and B given any set that does not contain F. Hence F is in S\A if F is a collider on P. If F is not a collider on P then there is an edge from B to F that is into F. The path concatenating the edges between E and F and between F and B d-connects E and B given any set containing F. Hence F is not in S\A if F is not a collider on P.

**Induction Case:** Suppose that for the first n vertices on P after E, each vertex is in S\A if and only if it is a collider on P. If P has only n vertices we are done. Otherwise let V be the $n+1^{st}$ vertex on P. If V = A then it immediately follows that E and A are d-connected by S\A. If V * A then suppose first that V is a collider on P. The subpath of P from E to V d-connects E and V given S\V because each vertex preceding V on P is in S\V if and only it is a collider. Because V is a collider on P it is a parent of B. If V is not in S then the concatenation of the subpath of P from E to V and the edge between V and B forms a path that d-connects E and B given S. Hence V is in S\A if V is a collider on P. Suppose that V is not a collider on P. Then V is adjacent to B, and the edge between V and B is into V. If V is in S then the concatenation of the subpath of P from E to V and the edge between V and B forms a path that d-connects E and B given S. Hence V is not in S\A if V is not a collider on P.

It follows that P d-connects E and A given S\A. Because P d-connects E and A given S\A and E and B are d-separated given S, A is in S if and only if the edge between A and B does not collide with P at A.

Let P' be the concatenation of P and the A-C edge. P' is an undirected path because P does not contain B or C. If A is in S, then the edge between A and B does not collide with P at A. By hypothesis, the predecessor of A on P is into A. It follows that the edge between A and B is out of A, and hence the edge between A and C is into A. In that case, the edge between A and C collides with P at A, and P' d-connects E and C given S\C.

If A is not in S, then the the edge between A and B is into A, and hence the edge between A and C is out of A. In that case, the edge between A and C does not collide with P at A, and P' d-connects E and C given S\C.

In all cases, E and C are d-connected by P' given S\C. Let P" be the concatenation of P' with the edge between B and C. P" is an undirected path because P' does not contain B. Because P' d-connects E and C given S\C, C is a collider along P" if and only if C is not in S. Q.E.D.

### A.3 Partially Oriented Inducing Path Graphs

The partially oriented inducing path graph for a directed acyclic graph G is given by the output of the Causal Inference Algorithm.

There is an edge between A and B in the partially oriented inducing path graph if and only if there is an edge between A and B in the inducing path graph. However, there are four kinds of edges in a partially oriented inducing path graph. An edge A -> B means that in the inducing path graph, there is an edge A -> B; an edge A <-> B means that in the inducing path graph, there is an edge A <-> B; an edge A o-> B means that in the inducing path graph there is an edge between A and B that is into B, but we do not know whether it is into A; and an edge A o-o B means that there is an edge between A and B in the inducing path graph, but we do not know whether it is into A or into B.

We use "*" as a metasymbol to represent any of the three kinds of ends (nothing, ">", or "o") that an edge in a partially oriented inducing path graph can have; the "*" symbol itself does not appear in a partially oriented inducing path graph. In addition, A *-o B o-* C means that either or both of the edges between A and B and B and C are out of B, although we do not know whether the edge between A and B is out of B, or whether the edge between B and C is out of B.

The definitions of directed path, d-separability, inducing path, collider, ancestor, and descendant are the same as those for directed graphs, i.e. a directed path in a partially oriented inducing path graph, as in an acyclic directed graph, contains only directed edges (e.g. A-> B). However, an undirected path in a partially oriented inducing path graph can contain either directed edges (e.g. A -> B), bi-directed edges (e.g. A <-> B), undirected edges (e.g. A o-o B), or partially directed edges (e.g. A o-> B). Also, A is a parent of B only if there is a directed edge from A to B. In a partially oriented inducing path graph G we will say that an edge A is **into** B if A *-> B, and that it is **out of** A if and only if A -> B.

### Causal Inference Algorithm

If G is a directed acyclic graph over **V'**, and **V** is a subset of **V'**, the input to the algorithm is the set of d-separation relations involving just members of **V** that is true in G. Let $A_Q(A,B)$ denote the set of vertices adjacent to A or to B in graph Q, except for A and B themselves. (Since the algorithm is continually updating Q, $\mathbf{A_Q}(A,B)$ is constantly changing as the algorithm progresses.)

A.) Form the complete undirected graph Q on the vertex set **V**.

B.) If A and B are d-separated by any subset **S** of **V**, remove the edge between A and B, and record **S** in **D**(A,B).

C.) Let F be the graph resulting from step B. Orient each edge as o-o. For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in F but the pair A, C are not adjacent in F, orient A *-* B *-* C as A *-> B <-* C if and only if B is not in **D**(A,C).

D. repeat

If there is an edge A *-> B, and an edge B *-* C, A and C are not adjacent, and there is no arrowhead into B, then orient B *-* C as B -> C,

else if there is a directed path from A to B, and an edge A *-* B, orient A *-* B as A *-> B,

else if V is a discriminating vertex for M using R in triangle P-M-R then

if M is in **D**(V,R) then mark M as a non-collider on subpath P *-* M *-* R

else orient P *-* M *-* R as P *-> M <- * R.

else if P *-> M *-* R then orient as P *-> M -> R.

until no more edges can be oriented.

**Because the d-separation relations in a directed acyclic graph G are the same as the d-separation relations in the inducing path graph G' of G, we will also speak of the partially oriented inducing path graph of an inducing path graph G'. Note that there is an edge between A and B in the partially oriented inducing path graph of graph G if and only if there is an inducing path between A and B in G. Hence two vertices are adjacent in the partially oriented inducing path graph of G if and only if they are adjacent in the inducing path graph of G.**

**Lemma 16: If *n* is the partially oriented inducing path graph of inducing path graph G\ then if B *-> C in *n* then B *-> C in G, and if B -> C in TC, then B -> C in G**

**Proof. The proof is by induction on the number of applications of orientation rules in the repeat loop of the Causal Inference Algorithm.**

**Base Case: Suppose that the only orientation rule that has been applied is that if A *-* B *-* C in F, but A and C are not adjacent in F, A *-* B *-* C is oriented as A *-> B <-* C if and only if B is not a member of D(A,C). If A *-> B <-* C in TC, then A and C are not d-separated by any subset containing B. If B is a parent of either A or C in G, then there is an undirected path between A and C that does not collide at B, and except for the endpoints contains only B. If that path does not d-connect A and C given any subset S, then S contains B. Since A and C are d-separated given D(A,C), and D(A,C) does not contain B, B is not a parent of either A or C, and the edges between A and C, and B and C collide at B.**

**Induction Case: Suppose that after the $n^{th}$ iteration of the repeat loop that if B *-> C in *n* then B *-> C in G\ and if B -> C in *n,* then B -> C in G\ and if A *-* B •-* C that the edges between A and B, and B and C do not collide at B in G\ We will now show that after the $n+1^{st}$ iteration of the repeat loop that if B *-> C in *n* then B *-> C in G\ and if B -> C in 7c, then B -> C in G**

**Case 1: If there is an edge A *-> B, and an edge B *-* C, and A and C are not adjacent, and there is no arrowhead into B, then B *-* C is oriented as B -> C. If A *-> B in *K,* then by the induction hypothesis, A *-> B in $G^f$ . If A *-> B <-* C in $G_f$ then A and C are d-connected given every subset containing B. Hence, every unshielded collider in G' has been oriented as an unshielded collider in *n* before the repeat loop is entered. Because A *-> B *-* C is not a collider at B in G\ and A *-> B in G\ it follows that B -> C in G**

Case 2: If there is a directed path from A to B, and an edge A *-* B, orient A *-* B as A *-> B. By the induction hypothesis if there is an edge R -> S in *n,* then there is an edge R -> S in G'. It follows that if there is a directed path from A to B in *n,* then there is a directed path from A to B in G'. Because G' is acyclic, A *-> B in G\

Case 3: If V is a discriminating vertex for Q in triangle P-Q-R then

if Q is in D($V_f$R) then mark Q as a non-collider on subpath P *-* Q *-* R

else orient P *-* Q *-* R as P *-> Q <- * R.

By the induction hypothesis, if V is a discriminating vertex for Q in triangle P-Q-R in a, then it is a discriminating vertex for Q in triangle P-Q-R in G*. By lemma 15, in G\ if V is a discriminating vertex for Q in triangle P-Q-R, then Q is a collider on P *-1JQ. 1-* R if and only if Q is not in D(V,R).

Case 4: If P *-> Q »-» R then orient as P *-> Q -> R. By the induction hypothesis, if P *-> Q *•* R in 7c, then in $G^f$ the edge from P to Q is into Q, but Q is not a collider on P *-> Q *-* R. It follows that P *-> Q -> R in G\ Q.E.D.

**Lemma 17: If TC** is the partially oriented inducing path graph of directed acyclic graph G, and A -> B in *, then in G there is no inducing path between A and B that is into A.

Proof. Let G* be the inducing path graph of G. By lemma 16, if A -> B in a, then A -> B in G\ By the method of construction of G* there is no inducing path between A and B in G that is into both A and B, and some inducing path between A and B in G is into B. By lemma 4 there is a directed path from A to B in G. If there is an inducing path in G between A and B that is into A but not into B then by lemma 4 there is a directed path from B to A. Because G is acyclic, there is no inducing path in G between A and B that is into A but not into B. Hence in G no inducing path between A and B is into A. Q.E.D.

**Lemma** 18: If TC is the partially oriented inducing path graph of directed acyclic graph G, and A -> B in *n,* then there is a directed path from A to B in G.

Proof. Let G' be the inducing path graph of G. If A -> B in *n,* then A -> B in G\ If A -> B in G\ then in G there is an inducing path from A to B that is not into A. Hence by lemma 4 there is a directed path from **A** to **B** in G. Q.E.D.

**Theorem 2:** If *K* is the partially oriented inducing path graph of directed acyclic graph G, and there is a directed path U from A to B in *n,* then there is a directed path from A to BinG.

Proof. By lemma 18, for each edge between R and S in U there is a directed path from R to S in G. The concatenation of the directed paths in G contains a subpath that is a directed path from A to B in G. Q.E.D.

**Lemma 19: If** TC is the partially oriented inducing path graph of graph G over O, E is a discriminating vertex for B in triangle A-B-C using C and path P, V is on P, V is on P and between V and C, and V is not a collider on P, then V is not a collider on P.

Proof. Let G' be the inducing path graph of G over 0. Suppose that in *n* is the partially oriented inducing path graph of graph G, E is a discriminating vertex for B in triangle A-B-C using C and path P in *n,* V is on P, V is on P and between V and C, and V is not a collider on P, but V is a collider on P. By lemma 16, in G\ E is a discriminating vertex for B in triangle A-B-C using C and path P in TC, V is on P, V is on P and between V and C, and V is not a collider on P, but V is a collider on P. In G\ E and B are not d-connected given every subset of O, because by definition of discriminating vertex E and B are not adjacent in G\ Suppose S is a subset of O that d-separates E and B in G\ We have already shown in the proof of lemma 15 that every collider on P is in S and no non-collider on P is in S. Let V be the collider closest to C on P such that there is a non-collider on P between V and A, and let V be the collider adjacent to V on P and between V and C. Hence V has a descendant (V) that is a collider on P. There is an edge between V and B that is into V, because V is not a collider on P. The concatenation of P(E,V) with the edge from V to B is a path that d-connects E and B given S because every collider along $P(E_V V)$ is in S, no non-collider along $P(E_V V)$ is in S, and V is a collider on the concatenated path that has a descendant V that is in S. This contradicts the assumption that S d-separates E and B. Q.E.D.

**Lemma** 20: If *n* is the partially oriented inducing path graph of graph G, and A *-> B in ic, then every inducing path in G between A and B is into B.

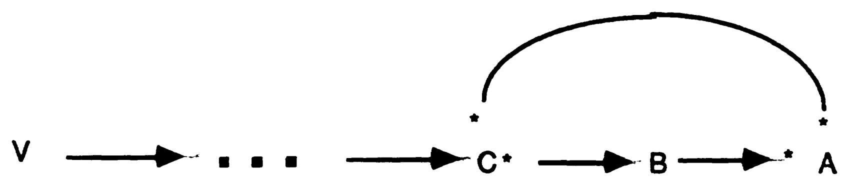
Proof. We will prove that each orientation rule in the Causal Inference Algorithm is such that if the rule orients the edge between A and B as A *-> B, then every inducing path between A and B in G is into B. Let G' be the inducing path graph of G.

Case 1: By lemma 18 any of the rules that orients the edge between A and B as A -> B in 7c entails that there is a path from A to B in G. By lemma 5 every inducing path between A and B that is into A or B. If there is an inducing path between A and B in G that is into A and not into B then by lemma 4 there is a directed path from B to A in G. But G is not cyclic, so there is no inducing path between A and B in G that is into A and not into B.

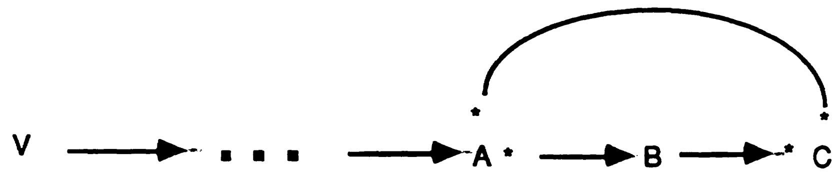
**Case 2: Suppose the edge between A and B is oriented as A *-> B in order to avoid a cycle in *n* because there is a directed path from A to B in *n.* By lemma 20 there is a directed path from A to B in G. By lemma 5 every inducing path between A and B that is into A or B. If there is an inducing path between A and B in G that is into A and not into B then by lemma 4 there is a directed path from B to A in G. But G is not cyclic, so there is no inducing path between A and B in G that is into A and not into B.**

**Case 3: The edge between A and B is oriented as A *-> B because there is a vertex C such that A and B are adjacent in *n,* B and C are adjacent in *, A and C are not adjacent in *n,* and B is not in D(A,C). It follows by lemma 16 that A *-> B <-* C in G\ By the construction of G' it follows that in G there is an inducing path between A and B into $B_f$ and an inducing path between B and C into B. Suppose contrary to the theorem that there is another inducing path between A and B in G that is not into B. By lemma 4 A is a descendant of B in G. By lemma 9 there is an inducing path between A and C. But if there is an inducing path between A and C in G, then A and C are adjacent in TC, contrary to our assumption.**

**Case 4: Suppose the edge between A and B in *K* is oriented as A *-> B because V is a discriminating vertex for B using C in triangle A-B-C and B is not in D(V,C). Suppose contrary to the theorem that in G there is also an inducing path between B and A that is not into B. It follows from lemma 4 there is a directed path from B to A in G. There are two cases: either V is a discriminating vertex for B using A and some path U, or V is a discriminating vertex for B using C and some path U. In either case we will refer to the vertex that is the endpoint of U on triangle A-B-C as NS (near side) and the vertex that is in ABC and not equal to B or on U as FS (far side.) See Figure 12.**




V is discriminating vertex for B using A. NS = C, FS = A.




V is discriminating vertex for B using A. NS = A, FS = C.

**Figure 12**

We will first prove that in $\pi$ there is an undirected path W from V to FS in which all vertices between V and FS are colliders, and each collider is an ancestor of FS. There are three cases.

First, if every vertex on U is a collider and the edge from NS to B does collide with the last edge in U, let W be the concatenation of U and the edges from NS to B and B to FS. It follows trivially that every vertex on W between V and FS is a collider.

Second, if every vertex on U is a collider but the edge from NS to B does not collide with the last edge in U, let W be the concatenation of U and the edge from NS to FS. Because V is a discriminating vertex for B in the NS-B-FS triangle using U, the last edge on U is into NS. If the edge from NS to B does not collide with the last edge on U, the edge from NS to B is out of NS. It follows that the edge from FS to NS is into NS, and hence does collide with the last edge on U. It follows that every vertex on W between V and FS is a collider.

Third, if there is a non-collider on U, let Q be the first non-collider after V. Because Q is not a collider on U, and V is a discriminating vertex using U, there is an edge between Q and FS that is into Q. Let W be the concatenation of the subpath of U from V to Q, with the edge from Q to FS. Suppose that there is a non-collider on U, and Q is the first non-collider after V. Because Q is not a collider on U, and V is a discriminating vertex using U, there is an edge between Q and FS that is into Q, and the last edge on the subpath of U from V to Q is into Q. Hence every vertex on W between V and FS is a collider.

We will now show that every collider on W is an ancestor of FS.

First, every collider on W that is also on U is a parent of FS because V is a discriminating vertex for B using FS and U.

Second, both B and NS are ancestors of FS. First consider the case where NS = C and FS = A. Suppose that C -> B <-* A in $\pi$. It follows that in G, C is an ancestor of B and by hypothesis B is an ancestor of A. Hence both B and NS are ancestors of FS. Suppose that C <-> B in p. Because V is a discriminating vertex for ABC, it follows that C -> A in p, and hence C is an ancestor of A in G. It follows that both B and NS are ancestors of FS. Next consider the case where NS = A and FS = C. If A -> B in $\pi$, then A is an ancestor of B in G, and hence there is a cycle. Suppose then that A <-> B in p. Because V is a discriminating vertex for B in ABC, it follows that the edge from A to C is out of A and A is an ancestor of C. In that case, both B and NS are ancestors of FS.

Finally, suppose Q which is a non-collider on U is on W. If Q is a non-collider on U, then by lemma 19 every vertex after Q on U is a non-collider on U, but is into its successor on U. Hence Q is an ancestor of NS, which is an ancestor of FS.

In $\pi$, there is an undirected path W from V to FS in which every vertex between V and FS is a collider, and in G every collider is an ancestor of V or FS. It follows by lemma 16 that there is an undirected path W' from V to FS in G' in which every vertex between V and FS is a collider. From the construction of G', it follows that there is a sequence of vertices in **O**, such that in G there is a sequence of vertices starting with V and ending with FS such that for every pair of vertices I and J adjacent in the sequence there is an inducing path W relative to **O** between I and J, and if J $\neq$ V then W is into V, and if I $\neq$ FS then W is into I, and I and J are ancestors of either V or FS. By lemma 9 there is an inducing path between V and FS in G. Hence V and FS are adjacent in $\pi$. But this contradicts the assumption that V is a discriminating vertex for B using FS. Q.E.D.

A **semi-directed path from A to B** in partially oriented inducing path graph $\pi$ is an undirected path from A to B in which no edge contains an arrowhead pointing towards A (i.e. if X and Y are adjacent on the path, and X is between A and Y on the path, then there is no arrowhead at the X end of the edge between X and Y.)

**Theorem 3:** If $\pi$ is the partially oriented inducing path graph of directed acyclic graph G over **O**, and there is no semi-directed path from A to B in $\pi$, then there is no directed path from A to B in G.

Proof. Suppose there is a directed path P from A to B in G. Let P' in $\pi$ be the sequence of vertices in **O** along P in the order in which they occur. P' is an undirected path in $\pi$ because for each pair of vertices X and Y adjacent in P' for which X is between A and Y there is an inducing path in G that is out of X. P' is a semi-directed path from X to Y in $\pi$ because by lemma 20, there is no arrowhead into X on P'. Q.E.D.

Unfortunately, the Causal Inference Algorithm as stated is not practical for large numbers of variables because of the way the adjacencies are constructed. While it is theoretically correct to remove an edge between A and B from the complete graph if and only if A and B are d-separated by some subset of **O**\{A,B}, this is impractical for two reasons. First, there are too many subsets of **O**\{A,B} to test the conditional independence of A and B on. Second, for discrete distributions, unless the sample sizes are enormous there are no reliable tests of independence of two variables conditional on a large set of other variables.

Remember, however, that in an inducing path graph if A and B are d-separated by any subset of **O**, then they are d-separated by **D-SEP**(A,B). Unfortunately, until we have actually constructed the inducing path graph we do not know which variables are in **D-SEP**(A,B). Nevertheless, as the partially oriented inducing path graph is constructed, we can determine that some variables are definitely not in **D-SEP**(A,B). This reduces the number and size of the subsets of **O** that have to be checked in order to determine whether A and B are adjacent in the inducing path graph.

We will determine which edges to remove from the complete graph in three stages. First, we will remove the edge between A and B if they are independent conditional on subsets of neighbors of A and B. This will eliminate many, but perhaps not all of the edges that are not in the inducing path graph. Second, we will orient edges by determining whether they collide or not. Third, using the partially oriented inducing path graph $\pi$ that we have constructed thus far, we will form two sets of vertices **Possible-D-SEP**(A,B), and **Possible-D-SEP**(B,A) one of which includes every vertex that could possibly be in **D-SEP**(A,B). (We need two such sets because we cannot determine from the partially oriented inducing path graph constructed thus far whether A is a descendant of B or B is a descendant of A.) Finally, we will remove the edge between A and B if A and B are independent conditional on any subset of either **Possible-D-SEP**(A,B,$\pi$) or **Possible-D-SEP**(B,A,$\pi$). Once we have obtained the correct set of adjacencies, we will unorient all of the edges, and then proceed to re-orient them. For a given partially constructed partially oriented inducing path graph $\pi$, **Possible-D-SEP**(A,B,$\pi$) is defined as follows.

1. A ∪ **Neighbors**(A) is in **Possible-D-SEP**(A,B,π).

2. If X and Z are in **Possible-D-SEP**(A,B,π) and there is an edge between Y and X and between X and Z, but not between Y and Z in π, then if the edge between Y and X collides with the edge between X and Z in π at X, then Y is in **Possible-D-SEP**(A,B,π).

3. If X and Z are in **Possible-D-SEP**(A,B,π) and there are edges between Y and X and Y and Z in π, then Y is in **Possible-D-SEP**(A,B,π).

### Fast Causal Inference Algorithm

If G is a directed acyclic graph over **V'**, and **V** is a subset of **V'**, the input to the algorithm is the set of d-separation relations involving just members of **V** that is true of G. Let **$A_Q$**(A,B) denote the set of vertices adjacent to A or to B in graph Q, except for A and B themselves. (Since the algorithm is continually updating Q, **$A_Q$**(A,B) is constantly changing as the algorithm progresses.)

A.) Form the complete undirected graph Q on the vertex set **V**.

B.) n = 0.

repeat

    repeat

        select a pair of variables X and Y that are adjacent in Q such that **$A_Q$**(X,Y) has cardinality greater than or equal to n, and a subset **S**(X,Y) of **$A_Q$**(X,Y) of cardinality n, and if X and Y are d-separated by some subset of **S**(X,Y) delete the edge between X and Y from Q, and record the subset in **D**(X,Y)

    until all variable pairs X and Y such that **$A_Q$**(X,Y)has cardinality greater than n and all subsets **S**(X,Y) of **$A_Q$**(X,Y) of cardinality n are exhausted.

    n = n + 1.

until for each pair of adjacent vertices X, Y, **$A_Q$**(X,Y) is of cardinality less than n.

C. Let F' be the graph resulting from step B. Orient each edge as o-o. For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in F' but the pair A, C are not adjacent in F', orient A *-* B *-* C as A *-> B <-* C if and only if B is not in **D**(A,C).

D. For each pair of variables A and B connected by an edge in F', if A and B are d-separated by any subset of **Possible-D-SEP**(A,B,F') or any subset of **Possible-D-SEP**(B,A,F') remove the edge between A and B.

E.) Let F be the graph resulting from step D. Orient each edge as o-o. For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in F but the pair A, C are not adjacent in F, orient A *-* B *-* C as A *-> B <-* C if and only if A and C are not d-separated by any subset of $\mathbf{A}_F$(A,C) containing B.

F. repeat

If there is an edge A *-> B, and an edge B *-* C, A and C are not adjacent, and there is no arrowhead into B, then orient B *-* C as B -> C,

else if there is a directed path from A to B, and an edge A *-* B, orient A *-* B as A *-> B,

else if V is a discriminating vertex for Q in triangle P-Q-R then

if Q is an in **D**(V,R) then mark Q as a non-collider on subpath P *-* Q *-* R

else orient P *-* Q *-* R as P *-> Q <- * R.

else if P *-> Q *-* R then orient as P *-> Q -> R.

until no more edges can be oriented.

**Lemma 21:** If G' is the inducing path graph of directed acyclic graph G, and F' is the partially oriented inducing path graph constructed in step C of Fast Causal Inference Algorithm from the d-separability relations true of G and involving only the variables in **O**, then either every vertex in **D-SEP**(A,B) in G' is in **Possible-D-SEP**(A,B,F') or every vertex in **D-SEP**(A,B) in G' is in **Possible-D-SEP**(B,A,F').

Proof. Suppose without loss of generality that A is not a descendant of B. If V is in **D-SEP**(A,B) in G', then there is a path from A to V in which every vertex except the endpoints is a collider; we will call such a path a **collider path** from A to V. We will show that every vertex in **D-SEP**(A,B) is in **Possible-D-SEP**(A,B,F') by induction over the lengths of the shortest collider paths from A in G'.

Base Case: Suppose that V is a neighbor of A in F'. The neighbors of A in F' are a superset of the neighbors of A in G'. Every neighbor of A in F' is in **Possible-D-SEP**(A,B,F'), so every neighbor of A in G' is in **Possible-D-SEP**(A,B,F').

Induction Case: Suppose that in G' every vertex V in **D-SEP**(A,B) for which the shortest collider path from A to V is length n or less is in **Possible-D-SEP**(A,B,F'). Let Z be a vertex in **D-SEP**(A,B) in G' for which the shortest collider path P from A to Z is length n+1. In G' there are vertices X and Y that are also on the collider path P such that there are edges from Z to Y and from X to Y that collide at Y. By the induction hypothesis, X and Y are in **Possible-D-SEP**(A,B,F'). If there is no edge between X and

Z in G' the edge between X and Y and the edge between Y and Z also collide in P. By the second clause of the definition of **Possible-D-SEP(A,B,F), Z** is in **Possible-D-**SEP(A,B,F). Suppose then that that there is an edge between X and Z in G\ In that case, Z is in **Possible-D-SEP(A,B,F)** by the third clause of the definition. Q.E.D.

**Theorem 4:** If G is a directed acyclic graph, the partially oriented inducing graph constructed by the Fast Causal Inference Algorithm from the d-separability relations true of G and involving only the variables in O is the same as the partially oriented inducing graph constructed by the Causal Inference Algorithm from the d-separability relations true of G and involving only the variables in O.

Proof. This follows immediately from lemmas 13 and 21. Q.E.D.