Causal Inference in the Presence of Latent Variables and Selection Bias[1]

by

Peter Spirtes, Christopher Meek, and Thomas Richardson

[1] We wish to thank Clark Glymour and Greg Cooper for many helpful conversations. This research was supported in part by ONR contract Grant #: N00014-93-1-0568

I. Introduction

Whenever the use of non-experimental data for discovering causal relations or predicting the outcomes of experiments or interventions is contemplated, two difficulties are routinely faced. One is the problem of latent variables, or confounders: factors influencing two or more measured variables may not themselves have been measured or recorded. The other is the problem of sample selection bias: values of the variables or features under study may themselves influence whether a unit is included in the data sample.

Latent variables produce an association between measured variables that is not due to the influence of any measured variable on any other. It is well known that where unrecognized latent common causes occur, regression methods, for example, no matter whether linear or nonlinear, give incorrect estimates of influence. When two or more variables under study both influence membership in a sample or inclusion in a database, an association between the variables occurs in the sample that is not due to any influence of a measured variable on other measured variables, nor to an unmeasured common cause.

Difficult as these problems are separately, they can both occur in the same sample or database, as in the following example. Suppose a survey of college students is done to determine whether there is a link between *intelligence* and *sex drive*. Let *student status* be a binary variable that takes on the value 1 when someone is a college student. Suppose *age* causes *sex drive*, and *age* and *intelligence* also causes *student status* . Hence whether or not one is in the sample is influenced by the two variables in the study, and there may be a statistical dependency between *intelligence* and *sex drive* in the sample even when no such dependency exists in the population. (This example will be discussed in more detail in section II. Here *age* is obviously a proxy for a combination of physical and mental states associated with age.) The combination of a latent variable (*age*) and a selection variable (*student status*) would produce a dependency which (naively interpreted) would make it appear as if there is a causal connection of some kind between *intelligence* and *sex drive*, even though none exists.

A reasonable attitude toward most uncontrolled convenience samples (and a lot of "experimental" samples as well) is that they may be liable to both difficulties. For that reason many statistical writers have implicitly or explicitly concluded that reliable causal and predictive inference is impossible in such cases, no matter whether by human or by machine. We think it is more fruitful to consider whether, under conditions only slightly stronger than those used for causal inference from experimentally controlled data, causal inferences can sometimes reliably be made. This paper uses Bayesian network models for that investigation. Bayesian networks, or directed acyclic graph (DAG) models have proved very useful in representing both causal and statistical hypotheses. The nodes of the graph represent vertices, directed edges represent direct influences, and the topology of the graph encodes statistical constraints. We will consider features of such models that can be determined from data under assumptions that are related to those routinely applied in experimental situations:

- The Markov condition for DAGs interpreted as causal hypotheses. An instance of the Causal Markov Assumption is the foundation of the theory of randomized experiments. It is also the foundation for the practice of constructing Bayesian networks to be used for diagnosis or classification by eliciting *causal* relations from experts.
- An assumption that the population selected by sampling criteria has the same causal structure (although because of sample selection bias not necessarily the same statistical properties) as the population about which causal inferences are to be made. This assumption, which we call the Population Inference Assumption is of course essential whenever experimental results on a sample are used to guide policy on a larger population.
- A version of the Causal Faithfulness Assumption, which says essentially that observed independence and conditional independence relations are due to the topology of the causal graph rather than to special parameter values. The assumption is used, implicitly or in other terms through the behavioral sciences, for example in econometrics to test "exogeneity". See Epstein (1987).

In order to deal with the problems raised by latent variables and selection bias, we will use, *partial ancestor graphs* (or PAGs), to represent a class of DAGs. For a DAG G which may have both latent and selection variables, the PAG that represents G contains information about both the conditional independencies entailed by G, and partial information about the ancestor relations in G. We will briefly describe how PAGs can be used to search for latent variable DAG models, to perform efficient classifications, and to predict the effects of interventions in causal systems. The advantages of the PAGs representation include:

- The space of PAGs is smaller than the space of DAGs they represent, making search over PAGs more feasible than a search over DAGs. For a given set of measured variables, the set of PAGs is finite, whereas the set of DAGs that contain the measured variables but may also contain latent variables is infinite.

- In some cases where large sample (or even population) data would not discriminate between different DAGs, the same data would select a single PAG, which could be used to answer some (but not all) questions about the effects of intervening upon an existing causal structure.
- Quite apart from any causal interpretation of PAGs, in some cases there is a PAG that is a more parsimonious representation of a distribution than any DAG containing the same variables. Hence a PAG may be used to obtain unbiased estimates of population parameters that have lower variance than estimates obtained from any DAG without latent variables.

Using PAGs, we characterize the causal information that can (and in some cases cannot) be obtained from independence and conditional independence relations in a population subject to both sample selection bias and latent variables. Given an oracle (such as a family of statistical tests) for judging population conditional independence relations among a set of recorded variables, we provide an asymptotically reliable algorithm for constructing PAGs, under the set of assumptions described above (and described again more precisely in sections I and II). The algorithm is exponential in the worst case, but feasible for sparse graphs with up to 100 variables. We will also describe the results of a simulation study on the reliability of the algorithm.

II. Representation of Selection Bias

We distinguish two different reasons why a sample distribution may differ from the population distribution from which it is drawn. The first is simply the familiar phenomenon of sample variation, or as we shall say, **sample bias**: the frequency distribution of a finite random sample of variables in a set **V** does not in general perfectly represent the probability distribution over **V** from which the sample is drawn. The second reason is that causal relationships between variables in **V,** on the one hand**,** and the mechanism by which individuals in the sample are selected from a population, on the other hand, may lead to differences between the expected parameter values in a sample and the population parameter values. In this case we will say that the differences are due to **selection bias**. Sampling bias tends to be remedied by drawing larger samples; selection bias does not.

We will not consider the problems of sample bias in this paper (except in the simulation studies); we will always assume that we are dealing with an idealized selected subpopulation of infinite size, but one which may be selection biased.

For the purposes of representing selection bias, following Cooper (1995) we assume that for each **measured** random variable A, there is a binary random variable S_A that is equal to one if the value of A has been recorded, and is equal to zero otherwise. (We will say that a variable is measured if its value is recorded for any member of the sample.) If $\mathbf{V}$ is a set of variables, we will always suppose that $\mathbf{V}$ can be partitioned into three sets: the set $\mathbf{O}$ (standing for observed) of measured variables, the set $\mathbf{S}$ (standing for selection) of selection variables for $\mathbf{O}$, and the remaining variables $\mathbf{L}$ (standing for latent). Although this representation allows for the possibility that some units have missing values for some variables and not others, the algorithms for causal inference that we will describe assume that we are using only the data for the subset of the sample in which all of the units have no missing data for any of the measured variables (i.e. $\mathbf{S} = \mathbf{1}$). Since in some circumstances this reduces the usable sample dramatically (or even to zero) it would obviously be desirable to make use of the full sample; how to do this is an open research problem.

In the marginal distribution over a subset $\mathbf{X}$ of $\mathbf{O}$ in a selected subpopulation, the set of selection variables $\mathbf{S}$ has been conditioned on, since its value is always equal to $\mathbf{1}$ in the selected subpopulation. Hence for disjoint subsets $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ of $\mathbf{O}$, we will assume that we cannot determine whether $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{Y}$, but that we can determine whether $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$. ($\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{Y}$ means $\mathbf{X}$ is independent of $\mathbf{Z}$ given all values of $\mathbf{Y}$. If $\mathbf{Y}$ is empty, we simply write $\mathbf{X} \perp\!\!\!\perp \mathbf{Z}$. If the only member of $\mathbf{X}$ is X, then we write $X \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{Y}$ instead of $\{X\} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{Y}$. $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$ means $\mathbf{X}$ is independent of $\mathbf{Z}$ given all values of $\mathbf{Y}$, and the value $\mathbf{S} = \mathbf{1}$.) There may be cases in which all of the variables in $\mathbf{S}$ always take on the same value; this corresponds to the case where there are no missing values in the sample. In such cases we will represent the selection with a single variable S.

The three causal DAGs for a given population shown in Figure 1 illustrate a number of different ways in which selection variables can be related to non-selection variables. The causal DAG in (i) would occur, for example if the members of the population whose X values were recorded and the members of the population whose Y values were recorded were randomly selected by flips of a pair of independent coins. The DAG in (ii) would occur if the flip of a single coin was used to choose units which would have both their X and Y values recorded (i.e. $S_X = S_Y$ and there are no missing values in the sample). The

DAG in (iii) would occur if, for example, X is years of education, and people with higher X values respond to a questionnaire about their education--and thus appear in the sample--more often than people with lower X values. We do not preclude the possibility that a variable Y can be cause of S_X for some variable $X \neq Y$, nor do we preclude the possibility that S_X can be a cause as well as an effect, respectively, of one or more different variables.

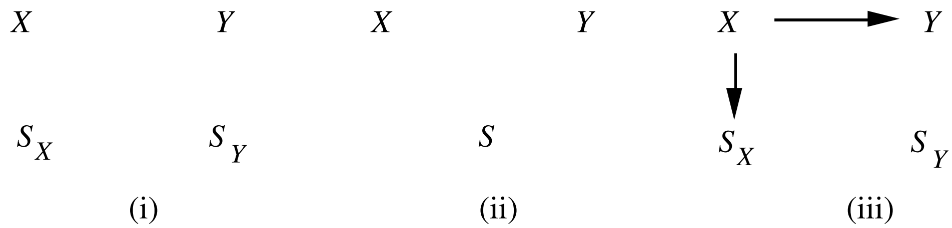


Figure 1

The causal DAG that represents the *intelligence* and *sex drive* example described in section I is shown in Figure 2.

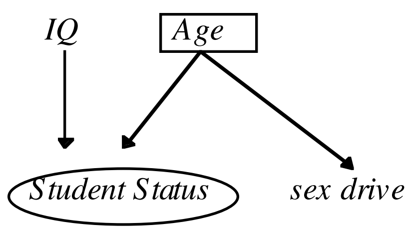


Figure 2

The causal inferences that we make rest on three different assumptions. The Causal Markov and Causal Faithfulness Assumptions are described in the Introduction. The Population Inference Assumption described below is the third assumption we make that guarantees the asymptotic correctness of the causal inference procedures described in the following sections.

Consider the case where one is interested in causal inferences about the whole population from the selected subpopulation. The notion of a causal graph, as we have defined it is relative to a set of variables and a population. Hence the causal graph of the whole population and the causal graph of the selected subpopulation can conceivably be different. For example, if a drug has an effect on people with black hair, but no effect on people with brown hair, then there is an edge from drug to outcome in the first subpopulation, but not in the second. Because of this, in order to draw causal conclusions about either the whole population or the unselected subpopulation (e.g. the black haired subpopulation) from the causal graph of the selected subpopulation (e.g. the brown haired subpopulation), we will make the following assumption:

Population Inference Assumption: If **V** is a causally sufficient set of variables, then the causal DAG relative to **V** in the population is identical with the causal DAGs relative to **V** in the selected subpopulation and the unselected subpopulation.

This is the sort of assumption that is routinely made when, for example, the results of drug trials conducted in Cleveland are generalized to the rest of the country. Of course, there may be examples where the assumption is less plausible. For example, a drug may have no effect on outcome in men, but have an effect on women.

There are some subtleties about the application of these assumptions to different sets of variables and different populations which are explained in more detail in the Appendix, but are not needed in order to understand the rest of the paper.

III. Using Partial Ancestral Graphs

Let us consider several different sets of conditional independence and dependence relations, and what they can tell us about the causal DAGs that generated them, under a variety of different assumptions.

Given a causal graph G over a set of variables **V**, we will say there is no selection bias if and only if for any three disjoint sets of variables **X**, **Y**, and **Z** included in **V\S**, G entails $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$ if and only if G entails $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{Y}$. This happens, for example, when the variables in **S** are causally unconnected to any other variables in **V**. (Note that this does not in general entail that the *distributions* in the selected subpopulation and the population are the same; it just entails that the same conditional independence relations holds in both.) In that case, when we depict a DAG in a figure we will omit the variables in **S**, and edges that have an endpoint in **S**.

For a given DAG G, and a partition of the variable set **V** of G into observed (**O**), selection (**S**), and latent (**L**) variables, we will write G(**O**,**S**,**L**). We assume that the only conditional independence relations that can be tested are those among variables in **O** conditional on any subset of **O** when $\mathbf{S} = \mathbf{1}$; we will call this the set of **observable** conditional independence relations. If **X**, **Y**, and **Z** are included in **O**, and $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$, then we say it is an **observed** conditional independence relation. Let **Cond** be a set of conditional independence relations among the variables in **O**. A DAG G(**O**,**S**,**L**) is in **O-Equiv**(**Cond**) just when G(**O**,**S**,**L**) entails that $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$ if and only if

$\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid \mathbf{Y}$ is in **Cond**. If $G'(\mathbf{O},\mathbf{S'},\mathbf{L'})$ entails that $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid (\mathbf{Y} \cup (\mathbf{S'} = \mathbf{1}))$ if and only if $G(\mathbf{O},\mathbf{S},\mathbf{L})$ entails that $\mathbf{X} \perp\!\!\!\perp \mathbf{Z} \mid (\mathbf{Y} \cup (\mathbf{S} = \mathbf{1}))$, then $G'(\mathbf{O},\mathbf{S'},\mathbf{L'})$ is in **O-Equiv**(G).

Imagine now that a researcher does not know the correct causal DAG, but can determine whether an observed conditional independence relation is in **Cond**, perhaps by performing statistical tests of conditional independence on the selected subpopulation. (As we will see later, because many of the conditional independencies in **Cond** entail other members of **Cond**, only a small fraction of the membership of **Cond** actually need be tested.) From this information alone, and the Causal Markov Assumption, the Causal Faithfulness Assumption, and the Population Inference Assumption, the most he or she could conclude is that the true causal DAG is some member of **O-Equiv**(**Cond**). This information by itself is not very interesting, unless the members of **O-Equiv**(**Cond**) all share some important features. The examples below show that sometimes the members of **O-Equiv**(**Cond**) do share important features.

Our strategy for finding PAGs even when there may be latent variables or selection bias, is a generalization of the strategy without selection bias described in Spirtes, et al., 1993. *A* is an **ancestor** of *B* in DAG *G* when there is a directed path from *A* to *B*, or *A* = *B*. We will construct from **Cond** a graphical object called a **partial ancestral graph** (PAG)[2], using the Causal Markov, Causal Faithfulness, and Population Inference Assumptions. The PAG represents information about which variables are or not ancestors of other variables in all of the DAGs in **O-Equiv**(**Cond**). If *A* is an ancestor of *B* in all DAGs in **O-Equiv**(**Cond**), then although from **Cond** we cannot tell exactly which DAG in **O-Equiv**(**Cond**) is the true causal DAG, because we know that all of the DAGs in **O-Equiv**(**Cond**) contain a directed path from *A* to *B* we can reliably conclude that in the true causal DAG *A* is a (possibly indirect) cause of *B*. This strategy is represented schematically in Figure 3. In the following examples we will apply this strategy to particular sets of observed conditional independence relations. We will also show what features of DAGs can be reliably inferred, and what features cannot.

[2] A similar object was called a partially oriented inducing path graph (POIPG) in Spirtes et al. 1993. PAGs in effect represent a kind of equivalence class of what Wermuth et al.(1994) call summary graphs.

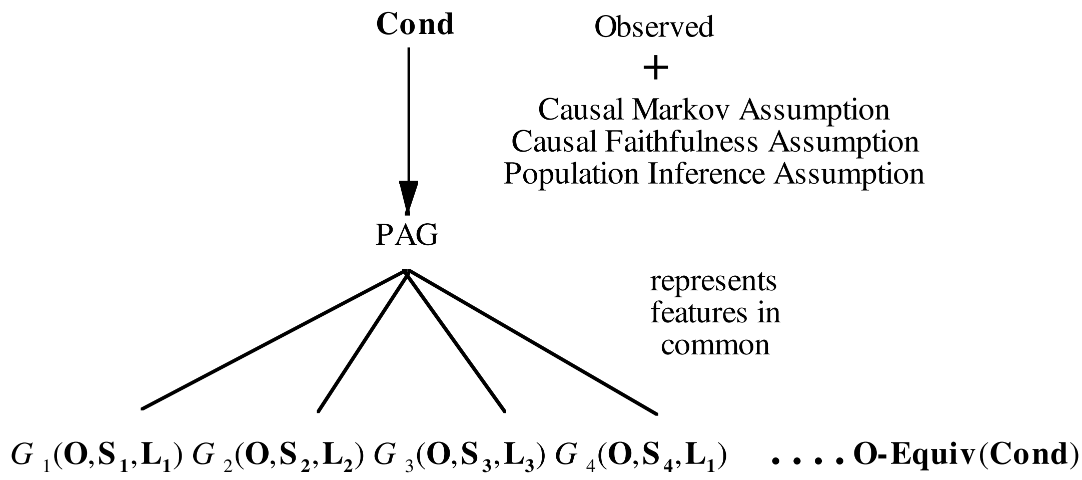


Figure 3

The formal definition of a PAG is given below. There are three kinds of endpoints an edge in a PAG can have: "-", "o", or ">". These can be combined to form the following four kinds of edges: $A \rightarrow B$, $A \leftrightarrow B$, A o→ B, or A o—o B. Let "*" be a meta-symbol that stands for any of the three kinds of endpoints. More formally:

A PAG π **represents** a DAG G(**O**,**L**,**S**) if and only:

1. The set of variables in π is **O**.
2. If there is any edge between A and B in π, it is one of the following kinds: $A \rightarrow B$, A o→ B, A o—o B, or$A \leftrightarrow B$.
3. There is at most one edge between any pair of vertices in π.
4. A and B are adjacent in π if and only if for every subset **Z** of **O**\{A,B} G does not entail that A and B are independent conditional on $\mathbf{Z} \cup \mathbf{S}$.
5. An edge between A and B in π is oriented as $A \rightarrow B$ only if A is an ancestor of B but not **S** in every DAG in **O-Equiv**(G).
6. An edge between A and B in π is oriented as A *→ B only if B is not an ancestor of A or **S** in every DAG in **O-Equiv**(G).
7. A *—$\underline{*\ B *}$—* C in π only if in every DAG in **O-Equiv**(G) either B is an ancestor of C, or A, or **S**. (Suppose that A and B are adjacent, and B and C are adjacent, and A and C are not adjacent, and the edges in the PAG are not both into B, i.e. the PAG does not contain A *→ B ←* C. Then the underlining of B should be assumed to be present, although we do not explicitly put the underlining in π.)

A "o" on the end of an edge places no restriction on the ancestor relationships. Note that more than one PAG can represent a DAG G(**O**,**L**,**S**). Two such PAGs π_1 and π_2 that represent $G(\mathbf{O},\mathbf{L},\mathbf{S})$ have the same adjacencies, but are differently oriented in the sense that π_1 may contain a "o" where π_2 contains a ">" or a "–", or vice-versa. Examples of PAGs are shown in the following subsections.

A. Example 1

We will start out with a very simple example, in which the set of observed conditional independence relations is not very informative. (For simplicity, in all of the following examples we assume that all of the variables in **S** take on the same value, and hence can be represented by a single variables *S*.) For example, suppose first that the set $\mathbf{Cond_1}$ of observed conditional independence relations is empty, i.e. $\mathbf{Cond_1} = \varnothing$. We now want to find out what DAGs are in **O-Equiv**($\mathbf{Cond_1}$). Let **V** be a set of causally sufficient variables. Suppose that we assume or know from background knowledge that $\mathbf{O} = \{A,B\}$ is causally sufficient and there is no selection bias. (In general it is not possible to test these assumption from observational data alone.) Under these assumptions there are exactly two DAGs that entail $\mathbf{Cond_1}$, labeled (i) and (ii) in Figure 4. In general, when there are no latent variables and no selection bias, there is an edge between *A* and *B* if and only if for any subset **X** of $\mathbf{O}\backslash\{A,B\}$, $G(\mathbf{O},\mathbf{S},\mathbf{L})$ entails that *A* and *B* are dependent given **X**.

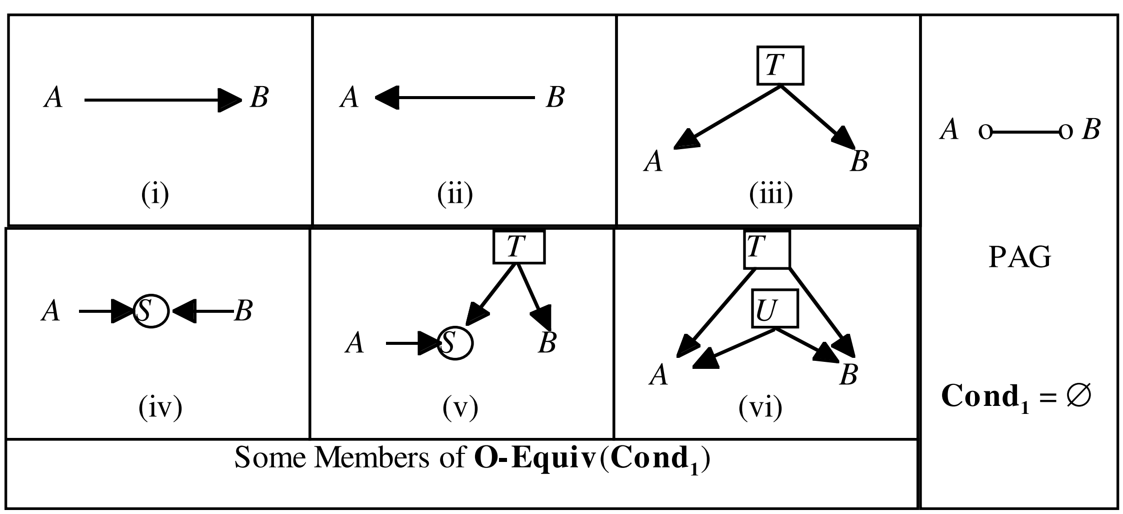


Figure 4

Now suppose that there are latent variables but no selection bias, and that the set of measured variables $\mathbf{O} = \{A,B\}$. Then, if we do not limit the number of latent variables in a DAG, there are an infinite number of DAGs that entail $\mathbf{Cond_1}$ many of which do not contain an edge between *A* and *B*. Two such DAGs are shown in (iii) and (vi) of Figure 4.

(Latent variables in **L** are represented by variables in boxes.) The examples in (iii) and (vi) of Figure 4 show that when there are latent variables it is not the case that there is an edge between *A* and *B* if and only if for all subsets **X** of **O**\{*A*,*B*}, *G*(**O**,**S**,**L**) entails that *A* and *B* are dependent given **X**. (Recall that if there is no selection bias then *A* and *B* are dependent given **X** ∪ (**S = 1**) if and only if *A* and *B* are dependent given **X.**)

Finally, let us consider the case where there are latent variables and selection bias. Examples of DAGs in **O-Equiv**(**Cond**$_1$) with selection bias are shown in (iv) and (v) of Figure 4.

The DAGs in **O-Equiv**(**Cond**$_1$) seem to have little in common, particularly when there is the possibility of both latent variables and selection bias. While there are a great variety of DAGs in **O-Equiv**(**Cond**$_1$), it is not the case that every DAG is in **O-Equiv**(**Cond**$_1$). For example, a DAG *G*(**O**,**S**,**L**) with no edges at all is not in **O-Equiv**(**Cond**$_1$).

If **Cond**$_1$ is observed, it is not possible to determine anything about the ancestor relationships between *A* and *B* in the causal DAG describing the population. We represent this information in a partial ancestral graph with the edge *A* o—o *B*. The "o" on each end of the edge means that the PAG does not specify whether or not *A* is an ancestor of *B*, or *B* is an ancestor of *A*. (Since there are DAGs in **O-Equiv**(**Cond**$_1$) in which *A* is an ancestor of *B*, and others in which *B* is an ancestor of *A*, every PAG which represents **O-Equiv**(**Cond**$_1$) has and "o" on each end of the edge between *A* and *B*.)

B. Example 2

Let **O** = {*A*,*B*,*C*,*D*} and **Cond**$_2$ = $\{D \perp\!\!\!\perp \{A,B\} \mid C, A \perp\!\!\!\perp B\}$ and all of the other conditional independence relations entailed by these. Once again the simplest case is when it is assumed that there are no latent variables and no selection bias. In that case the only DAG that entails **Cond**$_2$ is (i) in Figure 5.

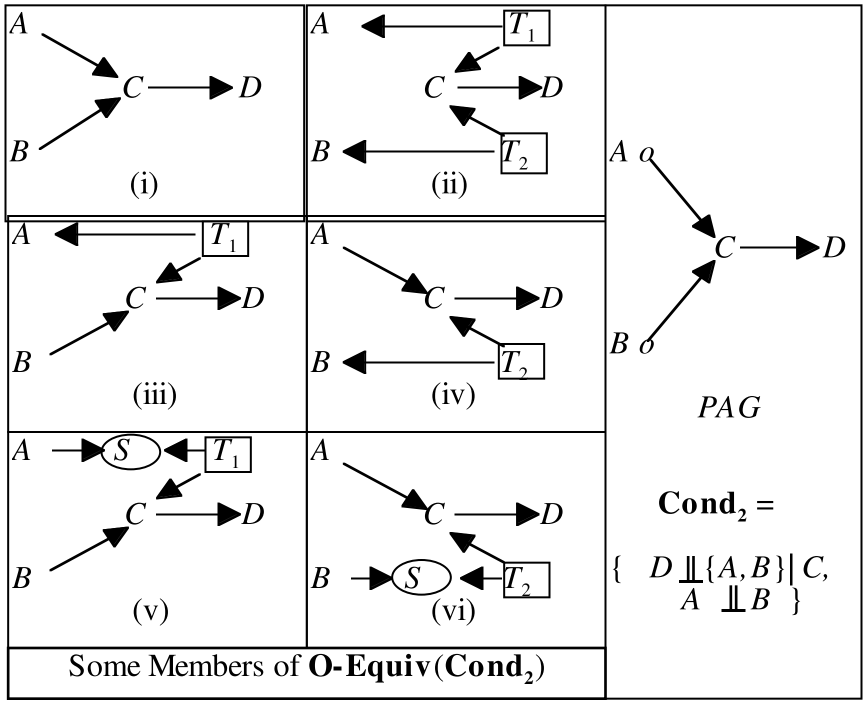


Figure 5

Now suppose that we consider DAGs with latent variables so **V** ≠ **O**, but no selection bias. In that case if there is no upper limit to the number of latent variables allowed, then there are an infinite number of DAGs in **O-Equiv**(**Cond$_2$**), several of which are shown in (ii), (iii) and (iv) of Figure 5.

Suppose that we now consider DAGs with selection bias. (v) and (vi) of Figure 5 show some examples of DAGs that are in **O-Equiv**(**Cond$_2$**) that have selection bias and latent variables.

Is there anything that all of the DAGs in Figure 5 have in common? In any of the DAGs in **O-Equiv**(**Cond$_2$**), for each of the pairs <*A*,*D*>, <*B*,*D*>, and <*A*,*B*>, there is a subset **Z** of **O** such that the pair is independent conditional on $\mathbf{Z} \cup \{S\}$. This is represented in the PAG by the lack of edges between *A* and *D*, between *B* and *D*, and between *A* and *B*. Moreover, in some of the DAGs in Figure 5, *A* is an ancestor of *C*, while in others it is not. Note that in none of them is *C* an ancestor of *A* or any member of **S**. It can be shown that in none of the DAGs in **O-Equiv**(**Cond$_2$**) is *C* an ancestor of *A* or of any member of **S**. In the PAG representing **O-Equiv**(**Cond$_2$**) we represent this by *A* o→ *C*.

The "o" on the *A* end of the *A* o→ *C* edge means the PAG does not say whether or not *A* is an ancestor of *C*; and the ">" on the *C* end of the edge means that *C* is not an ancestor of A or any member of **S** in *all* of the DAGs in **O-Equiv**(**Cond$_2$**). It is also the case that in all of the DAGs in Figure 5, C is an ancestor of D but not of any member of **S**, and D is not an ancestor of C or any member of **S**. It can be shown that in all of the DAGs in **O-Equiv**(**Cond$_2$**) C is an ancestor of D but not of any member of **S** and D is not an ancestor of C or any member of **S**. These facts are represented in the PAG by the edge between C and D having a ">" at the D end and a "–" at the C end.

There is an important distinction between the conditional distribution of D on $C = c$, and the distribution that results when C is forced (by intervening on the structure) to have the value c. The latter quantity depends upon the causal relations between C and D. If C is a cause of D, then forcing the value c on C will in general have an effect on the value of D, while if C is an effect of D, then forcing a value of c on C will not have an effect on the value of D. See Spirtes et al. (1993) and Pearl (1995) for details. In this particular case, it is possible to make both qualitative and quantitative predictions about the effects on the value of D of interventions that set the value of C from the PAG and the measured conditional distribution of D on $C = c$. This is because every DAG in **O-Equiv**(**Cond$_2$**) makes the same quantitative prediction about the effects of intervening to set the value of C to c. The details of the algorithm for making this prediction are given in Spirtes, Glymour and Scheines (1993). In this case we can determine that the only source of dependency between C and D are directed paths from C to D, so if $P(D \mid C = c)$ is the conditional distribution of D on C in the population, and C is forced to have the value c, then the new distribution of D will be $P(D \mid C = c)$.

C. Example 3

Finally, consider an example in which **Cond$_3$** = $\{D \perp\!\!\!\perp \{A,B\} \mid C, A \perp\!\!\!\perp \{C,D\}\}$. There is no DAG in **O-Equiv**(**Cond$_3$**) in which both **V** = **O** and there is no selection bias. Hence we can conclude that the DAG that entails **Cond$_3$** either contains a latent variable or there is selection bias or both. (i) and (ii) of Figure 6 are examples of DAGs with latent variables that entail **Cond$_3$**. Note that in each of them, there is a latent cause of B and C, B is not an ancestor of C or any member of **S**, and C is not an ancestor of B or any member of **S**. As long as there is no selection bias, these properties can be shown to hold of any DAG in **O-Equiv**(**Cond$_3$**).

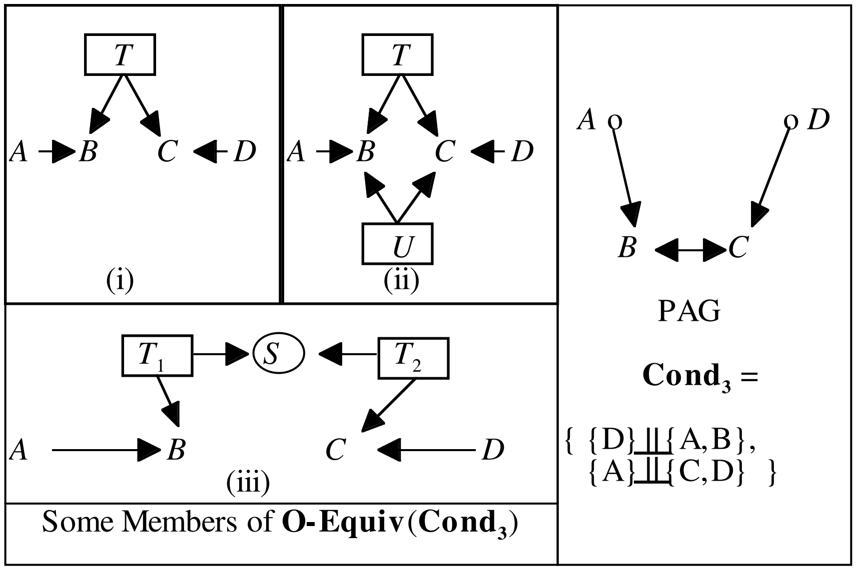


Figure 6

Suppose now that we also consider DAGs with selection bias. (iii) of Figure 6 is an example of a DAG with selection bias that entails **Cond$_3$**. Note that (iii) in Figure 6 does not contain a latent common cause of *C* and *B*. However, in each of the DAGs in Figure 5 *B* is not an ancestor of *C* or any member of **S**, and *C* is not an ancestor of *B* or any member of **S**; these properties can be shown to hold of any DAG in **O-Equiv**(**Cond$_3$**), even when there are latent variables and selection bias. Hence in the PAG we have an edge $B \leftrightarrow C$. Thus, if the conditional independence relations in **Cond$_3$** are ever observed, it can be reliably concluded that even though there may be latent variables and selection bias, regardless of the causal connections of the latent variables and selection variables to other variables, in the causal DAG that generated **Cond$_3$**, *B* is not a direct or indirect cause of *C* and *C* is not a direct or indirect cause of *B*.

Suppose P(**O**) is a distribution that has just the conditional independence relations in **Cond$_3$**. The PAG in Figure 6 can be parameterized in such a way that it represents P(**O**), and is more parsimonious (its parameterization is lower dimensional) than any DAG that contains just the variables in **O** and represents P(**O**). (For example in the linear case, the PAG can be given a complete orientation with all "o" ends removed, and interpreted as a linear structural equation models with correlated errors. See Spirtes, et al. 1996 for details.) Hence the PAG can be used to find an unbiased estimator of the population parameters that has lower variance than any unbiased estimator based on a DAG with the same set of variables. Even if one is not interested in predicting the effects of interventions, but merely seeks to find a parsimonious representation of a distribution in order to classify or diagnose members of a population, the PAG in Figure 6 has advantages over any DAG with the same variables.

IV. Summary of PAG Theorems

Note that it follows from the definition of a PAG and the assumed acyclicity of the directed graphs, that there are no edges A — B in a PAG, and no directed cycles in a PAG. (PAGs can also be used to represent directed cyclic graphs. See Richardson 1996).

Informally, a directed path in a PAG is a path that contains only "→" edges pointing in the same direction.

Theorem 1: If π is a partial ancestral graph, and there is a directed path U from A to B in π, then in every DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ with PAG π, there is a directed path from A to B, and A is not an ancestor of $\mathbf{S}$.

(This follows directly from the definition of an "→" edge in a PAG.

A **semi-directed path from** A to B in a partial ancestral graph π is an undirected path U from A to B in which no edge contains an arrowhead pointing towards A, (i.e. there is no arrowhead at A on U, and 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). Theorems 4, 5, and 6 give information about what variables appear on causal paths between a pair of variables A and B, i.e. information about how those paths could be blocked.

Theorem 2: If π is a partial ancestral graph, and there is no semi-directed path from A to B in π that contains a member of $\mathbf{C}$, then every directed path from A to B in every DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ with PAG π that contains a member of $\mathbf{C}$ also contains a member of $\mathbf{S}$.

Theorem 3: If π is a partial ancestral graph of DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$, and there is no semi-directed path from A to B in π, then every directed path from A to B in every DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ with PAG π contains a member of $\mathbf{S}$.

Theorem 4: If π is a partial ancestral graph, and every semi-directed path from A to B contains some member of $\mathbf{C}$ in π, then every directed path from A to B in every DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ with PAG π contains a member of $\mathbf{S} \cup \mathbf{C}$.

V. An Algorithm for Constructing PAGs

We have seen that a PAG contains valuable information about the causal relationships between variables; it also represents conditional independence relations in the margin, and can be used for classification. However, the number of observable conditional independence relations grows exponentially with the number of members of **O**. In addition, some of the independence relations are conditional on large sets of variables, and often these cannot be reliably tested on reasonable sample sizes. Is it feasible to construct a PAG from data?

The Fast Causal Inference (FCI) algorithm constructs PAGs that are correct even when selection bias may be present (under the Causal Markov Assumption, the Causal Faithfulness Assumption, the Population Inference Assumption, and the assumption that conditional independence relations can be reliably tested). The description in Spirtes et al. 1993 did not allow the possibility of selection bias. If the possibility of selection bias is allowed, the algorithm described there gives the correct output, (called a POIPG in Spirtes et al. 1993) but the conclusions that one can draw from the PAG are the slightly different ones described in section IV, rather than those described in Spirtes et al. 1993.) Since the algorithm decides which conditional independence tests to perform, we will assume that for each **X**, **Y**, and **Z** included in **O**, the algorithm has some method for reliably determining if **X** is independent of **Y** given $\mathbf{Z} \cup (\mathbf{S} = \mathbf{1})$ in the distribution $P(\mathbf{V})$; we will call this method an **oracle** for $P(\mathbf{V})$ over **O** given $\mathbf{S} = \mathbf{1}$. In practice, the oracle can be a statistical test of conditional independence (which is of course not completely reliable on finite sample sizes.)

Theorem 5: If $P(\mathbf{V})$ is faithful to $G_1(\mathbf{O},\mathbf{S_1},\mathbf{L_1})$, and the input to the FCI algorithm is an oracle for $P(\mathrm{V})$ over **O** given $\mathbf{S} = \mathbf{1}$, the output is a PAG of $G_1(\mathbf{O},\mathbf{S_1},\mathbf{L_1})$.

Even if one drops the assumptions relating causal DAGs to probability distributions, then the output PAG is still a parsimonious representation of the marginal of the distribution.

In the worst case the number of times the FCI algorithm consults the oracle is exponential in the number of variables (as is any correct algorithm whose output is a function of the answers of a conditional independence oracle). Even when the maximum number of vertices any given vertex is adjacent to is held fixed, in the worse case the algorithm is exponential in the number of variables. In light of this the title "Fast Causal Inference Algorithm" is perhaps somewhat over-optimistic; however, on simulated data it can often be run on up to 100 variables provided the true graph is sparse. This is because it is (usually) not necessary to examine the entire set of observable conditional independence relations; many conditional independence relations are entailed by other conditional independence relations. The FCI algorithm relies on this fact to test a relatively small set of conditional independence relations, and test independence relations conditional on as few variables as possible.

The FCI algorithm can be divided into two parts. First the adjacencies in the PAG are found, and then the edges are oriented. First we will describe how the adjacencies are found.

A. Fast Causal Inference Algorithm - Adjacencies

The details of the adjacency phase of the FCI algorithm are stated at the end of this section. Here we will give an informal description and motivation for the steps of the algorithm.

There is a very simple, but slow and unreliable way of determining when two variables in a PAG are adjacent. Start off with a complete undirected graph. By definition, two variables A and B in a PAG for $G(\mathbf{O},\mathbf{L},\mathbf{S})$ are adjacent if and only if there is no subset $\mathbf{Z}$ of $\mathbf{O}\backslash\{A,B\}$ such that A and B are entailed to be independent given $\mathbf{Z} \cup \mathbf{S}$. So, for each pair of variables A and B, and each subset $\mathbf{Z}$ of $\mathbf{O}\backslash\{A,B\}$ one could simply ask the oracle if A and B are entailed to be independent given $\mathbf{Z} \cup \mathbf{S}$. The edge between A and B is removed if and only if the oracle answers yes to any of these questions. The problems with this algorithm are that the number of questions asked of the oracle grows exponentially with the number of variables in $\mathbf{O}$, and in practice, the oracle is unreliable if the number of variables in $\mathbf{Z}$ is large. Clearly it is desirable to ask as few questions of the oracle as possible, and to ask questions in which $\mathbf{Z}$ is as small as possible.

Since when the algorithm asks an oracle a question, it is trying to limit the size of the conditioning sets, it makes sense to first ask the oracle about independencies conditional on the empty set, then independencies conditional on sets with one variable, then independencies conditional on sets with two variables, etc. However, this still leads to asking unnecessary questions of the oracle.

The strategy that the FCI algorithm adopts for avoiding asking such unnecessary questions of the oracle is based on the following idea. Suppose the original unknown graph is $G(\mathbf{O},\mathbf{L},\mathbf{S})$ in Figure 7 (i), where for purposes of illustration there is no selection bias, so we do not need to condition on $\mathbf{S}$. If we applied the strategy described above, we would first create the complete undirected graph shown in Figure 7(ii). After we asked the oracle if W is independent of X, Y, and Z (in each case receiving the answer "yes"), we would obtain the result shown in Figure 7 (iii). At this point, although the graph created is not a PAG for $G(\mathbf{O},\mathbf{L},\mathbf{S})$ (because it contains the wrong adjacencies), and we have very incomplete information about $G(\mathbf{O},\mathbf{L},\mathbf{S})$, it is easy to show that W does not lie on any path between X and Z in $G(\mathbf{O},\mathbf{L},\mathbf{S})$. Hence if X and Z are independent conditional on any subset of $\mathbf{O}\backslash\{A,B\}$ that contains W, they are also independent given some other subset of $\mathbf{O}\backslash\{A,B\}$ that does not contain W. So there is never any need to ever ask the oracle if X and Z are independent given any subset containing W. For example, the algorithm never asks whether $\mathbf{X}$ and $\mathbf{Z}$ are independent given W. This reduces the number of questions asked of the oracle, and limits the size of the conditioning sets.

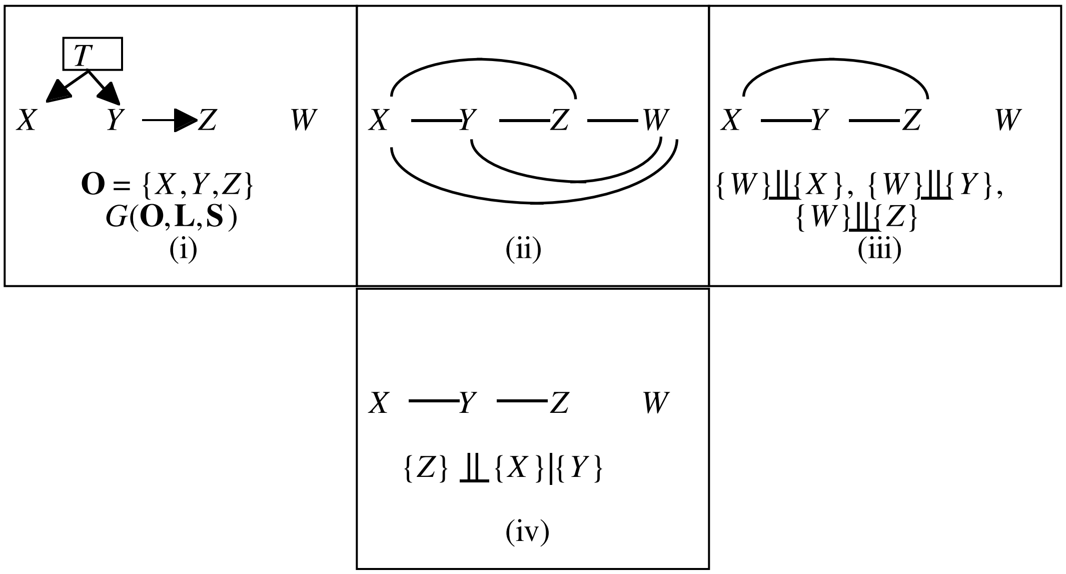


Figure 7

The adjacency phase of the FCI algorithm, which is stated at the end of this section, contains four steps, A), B), C) and D).

Step A) of the algorithm simply creates a complete undirected graph.

In step B), in searching for a subset $\mathbf{Z}$ of $\mathbf{O}\backslash\{A,B\}$ such that A and B are independent conditional on $\mathbf{Z} \cup \mathbf{S}$, the algorithm restricts the search to subsets of variables that are adjacent to A in the undirected graph it has constructed thus far, or subsets of variables adjacent to B. If there were no latent variables or selection bias in G, no more questions would need to be asked of the oracle in order to determine the correct set of adjacencies in the PAG. Unfortunately, if there are latent variables or selection bias, some further questions are needed. This is done in step D) of the FCI algorithm.

Consider the DAG $G(\mathbf{O},\mathbf{L},\mathbf{S})$ shown in Figure 8 (i). The PAG for $G(\mathbf{O},\mathbf{L},\mathbf{S})$ is shown in Figure 10 (ii). In the PAG, X_3 is not adjacent to either X_1 or X_5. However, in $G(\mathbf{O},\mathbf{L},\mathbf{S})$, the only subset $\mathbf{Z}$ of $\mathbf{O}$ such that X_1 and X_5 are independent conditional on $\mathbf{Z} \cup \mathbf{S}$, contains X_3. Hence we need to consider asking independence questions conditional on sets of variables that contain variables not adjacent to either X_1 or X_5. The algorithm constructs a set of variables called **Possible-D-Sep**(A,B,π), which is a function of A, B, and the graphical object π constructed by the algorithm thus far which has the following property: if A and B are independent conditional on any subset of $\mathbf{O}\backslash\{A,B\} \cup \mathbf{S}$, then they are independent given some subset of **Possible-D-Sep**(A,B,π) or **Possible-D-Sep**(B,A,π).

A, B, and C form a **triangle** in a graph or a PAG if and only if A and B are adjacent, B and C are adjacent, and A and C are adjacent.V is in **Possible-D-Sep**(A,B,π) in π if and only if there is an undirected path U between A and B in π such that for every subpath X *—* Y *—* Z of U either Y is a collider on the subpath, or X, Y, and Z form a triangle in π.. In Figure 8 (vi), **Possible-D-Sep**(X_1,$X_5$$\pi$) = **Possible-D-Sep**($X_5$,$X_1$, π) = $\{X_2,X_3,X_4\}$. Thus in step D) of the algorithm, the only independence questions that are asked of the oracle for a given pair of variables A and B are conditional on subsets of **Possible-D-Sep**(A,B,π) or **Possible-D-Sep**(B,A,π).

The construction of **Possible-D-Sep**(A,B,π) requires some limited orientation information about the edges in the PAG. Step C) performs some orientation of the PAG, so that the membership of **Possible-D-Sep**(A,B,π) can be calculated in step D). The orientation principles used in step C) are essentially the same as those used in step F) of the orientation phase of the algorithm. Step F) will be discussed in the next section, so we will not discuss step C) here.

When the algorithm removes an edge between A and B, it does so because it has found some subset $\mathbf{Z}$ of $\mathbf{O}\backslash\{A,B\}$ such that A and B are independent conditional on $\mathbf{Z} \cup \mathbf{S}$. The subset $\mathbf{Z}$ is recorded in **Sepset**(A,B) and **Sepset**(B,A). This information is used later in the orientation phase of the algorithm. Because each edge is removed at most once, **Sepset**(A,B) contains at most one subset of $\mathbf{O}\backslash\{A,B\}$. In the algorithm, **Adjacencies**(Q,X) is the set of vertices that are adjacent to X in Q. **Adjacencies**(Q,X) changes as the algorithm progresses, because the algorithm removes edges from Q. (However, **Possible-D-Sep** is calculated only once, and remains fixed, even as the graph changes.

Fast Causal Inference Algorithm - Adjacencies

A). Form the complete undirected graph Q on the vertex set $\mathbf{V}$.
B). $n = 0$.
repeat
  repeat
    select an ordered pair of variables X and Y that are adjacent in Q such that **Adjacencies**(Q,X)\{Y} that contains at least n members,

repeat

select a subset **T** of **Adjacencies**(Q,X)\{Y} with n members, and if X and Y are independent given $\mathbf{T} \cup \mathbf{S}$ delete the edge between X and Y from Q, and record **T** in **Sepset**(X,Y) and **Sepset**(Y,X)

until all subsets of **Adjacencies**(Q,X)\{Y} of size n have been checked for independence given $\mathbf{T} \cup \mathbf{S}$ or there is no edge between X and Y;

until all ordered pairs of adjacent variables X and Y such that **Adjacencies**(Q,X)\{Y} has at least n members have been selected;

$n = n + 1$;

until for each ordered pair of adjacent vertices X, Y, **Adjacencies**(Q,X)\{Y} has fewer than n members.

C). Let π_0 be the undirected graph resulting from step B). Orient each edge as "–". For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in π_0 but the pair A, C are not adjacent in π_0, orient A — B — C as $A \rightarrow B \leftarrow C$ if and only if B is not in **Sepset**(A,C).

D). Let π_1 be the undirected graph resulting from step C.) For each pair of variables A and B adjacent in π_1, if there is a subset **T** of **Possible-D-SEP**(A,B, π_1)\{A,B} or of **Possible-D-SEP**(B,A, π_1)\{A,B} such that A and B are independent conditional on $\mathbf{T} \cup \mathbf{S}$, remove the edge between A and B from π_1, and record **T** in **Sepset**(A,B) and **Sepset**(B,A).

E.) Orient each edge as "o—o". Call this graph π_2.

Figure 8 illustrates the application of the adjacency phase of the FCI algorithm to DAG G(**O**,**L**,**S**). We show only those steps which make changes to the PAG being created.

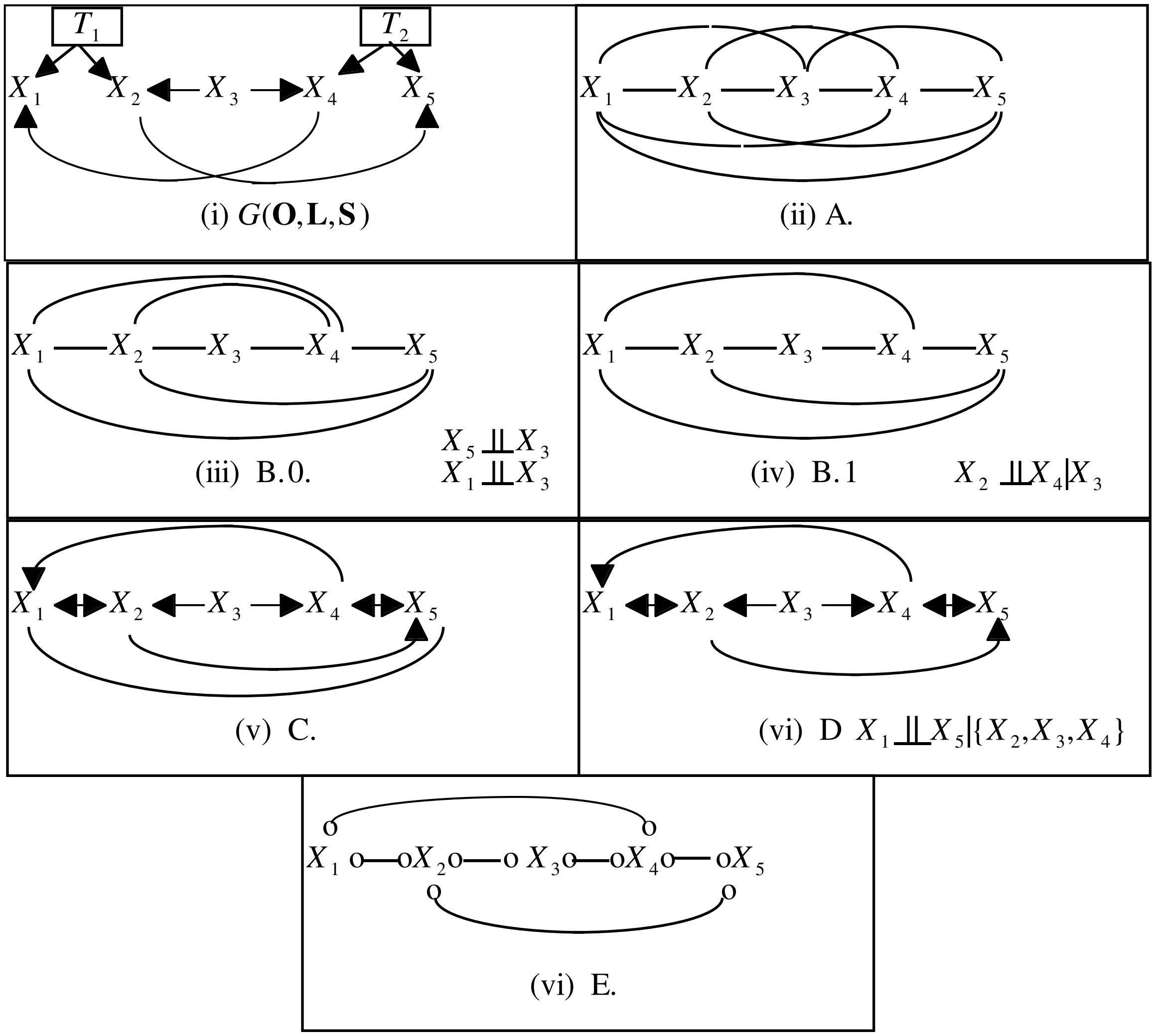


Figure 8

B. Fast Causal Inference Algorithm - Orientations

Step F) states that for each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in π_2 but the pair A, C are not adjacent in π_2, orient A *—* B *—* C as A *$\rightarrow$ B $\leftarrow$* C if and only if B is not in **Sepset**(A,C). The intuition behind rule F) is the following. It is easy to see if a DAG G with a set of variables $\mathbf{V}$ contains $A \rightarrow B \leftarrow C$, and A and C are not adjacent, then the path between A and C entails that A and C are dependent given every subset of $\mathbf{V}\backslash\{A,C\}$ that contains B. Alternatively, if G contains $A \rightarrow B \rightarrow C$, $A \leftarrow B \leftarrow C$, or $A \leftarrow B \rightarrow C$, and A and C are not adjacent, then A and C are entailed to be dependent given any subset of $\mathbf{V}\backslash\{A,C\}$ that does not contain B. This property generalizes to PAGs as well. Hence, if a PAG contains A *—* B *—* C and A and C are not adjacent in the PAG, a PAG can be oriented as A *$\rightarrow$ B $\leftarrow$* C if and only if **Sepset**(A,C) does not contain B. (In the version of the algorithm that we implemented for the simulation studies in this paper, we have actually replaced step F) by a more complicated step which is more reliable on small samples.)

The proofs of correctness of the orientation rules in step G) of the algorithm are all inductive arguments that show that the n+1st application of an orientation rule is correct if the first n application of the orientation rules are correct.

G(i) states that if there is a directed path from A to B, and an edge A *—* B, orient A *—* B as A *→ B. This is correct because if there is a directed path from A to B in π_2, and π_2 has been oriented correctly thus far, then A is an ancestor of B in every member of the **O**-equivalence class of G(**O**,**S**,**L**). Because each member of the **O**-equivalence class of G(**O**,**S**,**L**) is acyclic, it follows that B is not an ancestor of A in any member of the **O**-equivalence class of G(**O**,**S**,**L**). Hence the edge can be oriented as A *→ B. (In the version of the algorithm that we implemented for the simulation studies in this paper, we have actually deleted step G (i) because, although theoretically correct it is expensive to calculate, and leads to errors on small samples.)

G(ii) states that if P *→ $\underline{M}$ *—* R then orient as P *→ $M \to R$. The underlining means that M is an ancestor of either P or R in every member of the **O**-equivalence class of G(**O**,**S**,**L**). P *→ M means that M is not an ancestor of P in any member of the **O**-equivalence class of G(**O**,**S**,**L**). It follows that M is an ancestor of R in every member of the **O**-equivalence class of G(**O**,**S**,**L**). Hence the edge can be oriented as $M \to R$.

G(iii) states that if B is a collider along <A,B,C> (i.e. A *→ B ←* C) in π_2, B is adjacent to D, and D is in **Sepset**(A,C), then orient B *—* D as B ←* D. Suppose that D is in **Sepset**(A,C). Because the PAG contains orientation information about every member of the **O**-equivalence class of G(**O**,**S**,**L**), in some cases it is possible to show from even a partially oriented PAG that X and Y are entailed to be dependent conditional on **Z** in every member of the **O**-equivalence class of G(**O**,**S**,**L**). Suppose that there were some member G'(**O**,**S'**,**L'**) of the **O**-equivalence class of G(**O**,**S**,**L**) in which B was an ancestor of D. We have shown that in that DAG, A and C are entailed to be dependent conditional on any set containing D. But this is a contradiction, because D is in **Sepset**(A,C), A and C are entailed to be independent given **Sepset**(A,C) in every member of the **O**-equivalence class of G(**O**,**S**,**L**), and **Sepset**(A,C) does not contain D. Hence B is not an ancestor of D in any member of the **O**-equivalence class of G(**O**,**S**,**L**).

Rule G(iv) states that if B ←*C, $B \to D$, and D o—* C, then orient as D ←* C. By hypothesis, B is not an ancestor of C, but is an ancestor of D. Hence D is not an ancestor of C.

Rule G(v) is somewhat complicated, and not often applied, so the reader may wish to skip the following discussion of it. G(v) uses definite discriminating paths to orient edges in the PAG. The concept of a definite discriminating path is illustrated in Figure 9, and defined more formally below.

Consider the graph shown in Figure 9. All of the orientations shown can be derived without using step G(v) of the FCI algorithm. We have proved that in every member $G'(\mathbf{O},\mathbf{S'},\mathbf{L'})$ of the **O**-equivalence class of $G(\mathbf{O},\mathbf{S},\mathbf{L})$, if $G'(\mathbf{O},\mathbf{S'},\mathbf{L'})$ entails that A and C are entailed to be independent conditional on $\mathbf{Z} \cup \mathbf{S}$, and $G'(\mathbf{O},\mathbf{S'},\mathbf{L'})$ does not entail that A and C are independent conditional on any proper subset of $\mathbf{Z} \cup \mathbf{S'}$, then **Sepset**($A$,$C$) contains B if and only if the edges between B and C, and B and D do not collide at B. This orientation rule, when applied repeatedly, can lead to “long distance” orientations, i.e. a conditional independence relation between A and C can orient edges into B, even though the shortest path from A to B is arbitrarily long. The sorts of paths that have to exist in order to perform this “long distance” orientation are called definite discriminating paths. (In a partial ancestral graph π, U is a **definite discriminating path for** B if and only if U is an undirected path between X and Y, B is the predecessor of Y on U, $B \neq X$, every vertex on U except for the endpoints and possibly B is a collider on U, for every vertex V on U except for the endpoints there is an edge $V \rightarrow Y$, and X and Y are not adjacent.). An example of a definite discriminating path is given in Figure 9.

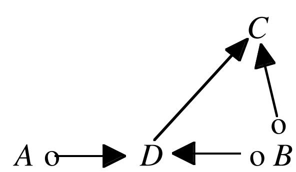


Figure 9: <A, D,B,C> is a definite discriminating path for B

Fast Causal Inference Algorithm - Orientations

F. For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in π_2 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 **Sepset**(A,C).

G. repeat

(i) If there is a directed path from A to B, and an edge A *—* B, orient A *—* B as A *→ B,

(ii) else if P *→ $\underline{M}$ *—* R then orient as P *→ $M \rightarrow R$.,

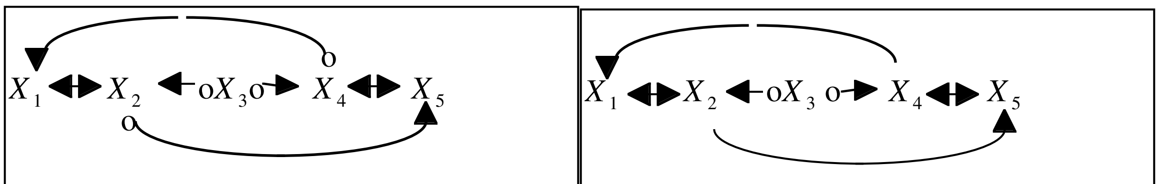
(iii) else if B is a collider along <A,B,C> in π_2, A is not adjacent to C, B is adjacent to D, and D is a non-collider along <A,D,C>, then orient B *—* D as $B \leftarrow * D$,

(iv) else if $B \leftarrow *C$, $B \rightarrow D$, and D o—* C, orient as $D \leftarrow * C$;

(v) If U is a definite discriminating path between A and C for B in π_2, D is adjacent to C on U, and D, B, and C for a triangle, then if B is in **Sepset**(A,C) then mark B as a non-collider on subpath D *—<u>* B *</u>—* C else orient D *—* B *—* C as D *→ $B \leftarrow * C$.

until no more edges can be oriented.

Figure 10 shows the application of the orientation part of the FCI algorithm to the example begun in Figure 8.



(i) F. (ii) G(ii).

Figure 10

We do not know whether the orientation rules of the FCI algorithm are complete in the sense that if any edge A o—* B occurs in the output, then in some member of the set of DAGs represented by the PAG, A is an ancestor of B, and in some other member of the set of DAGs represented by the PAG A is not an ancestor of B. However, we have proved that the FCI algorithm provides enough orientations so that any two DAGs represented by the output are in the same **O-equivalence** class. See Spirtes and Verma (1992) for details.

VI. Simulation Study

We ran some preliminary simulation studies of the FCI algorithm which were intended to show how making various variables latent or selection variables would change what information could theoretically be inferred from population information, and how much sample bias would affect the actual performance of the algorithm.

We used 10,000 cases generated pseudo-randomly by Cooper(1992) from the Alarm network, shown in Figure 11. The ALARM network was developed to model an emergency medical system (Beinlich, et al. 1989). The 37 variables are all discrete, taking 2, 3 or 4 distinct values. There are 46 edges in the DAG. In most instances a directed arrow indicates that one variable is regarded as a cause of another. The physicians who built the network also assigned it a probability distribution: each variable *V* is given a probability distribution conditional on each vector of values of the variables having edges directed into *V*. This data has been used to test several different discovery algorithms. (Spirtes et al. 1993, Cooper and Herskovits 1992, Chickering 1994) The interpretation of the variables is not relevant to the study described here.

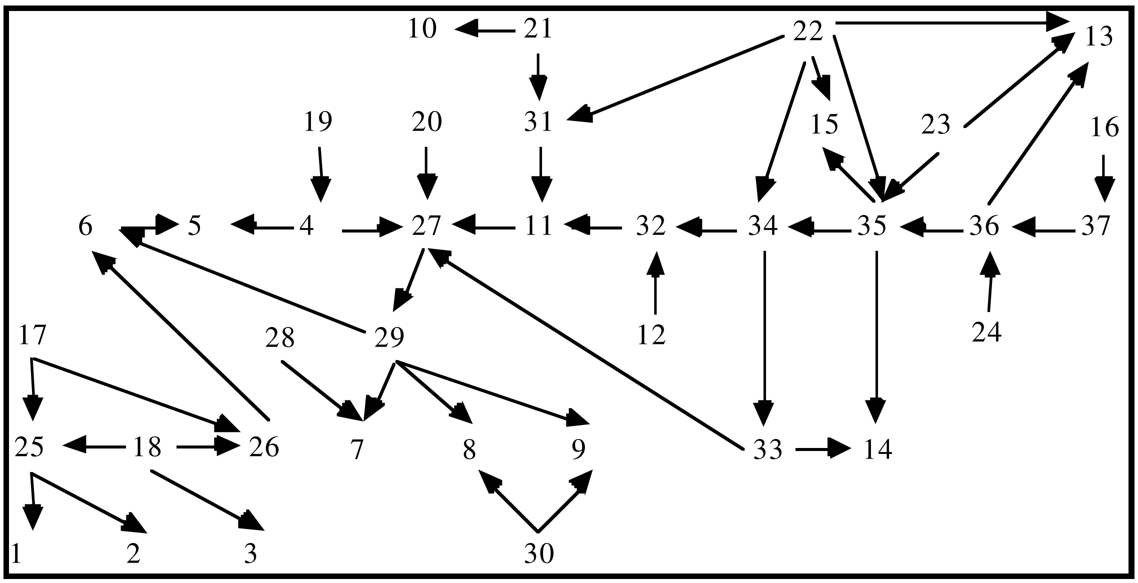


Figure 11

We scored the PAGs output by the algorithm in the following way. For each ordered pair of variables *A* and *B* the PAG either entails nothing about whether *A* is an ancestor of *B*, or it entails that *A* is an ancestor of *B*, or it entails that *A* is not an ancestor of *B*. We count the number of ordered pairs for which the output PAG entails that *A* is an ancestor of *B*, the percentage of times the entailment is correct, the number of ordered pairs for which the output PAG entails that *A* not is an ancestor of *B*, and the percentage of times the entailment is correct. For purposes of comparison, we note that in the Alarm DAG that there are 122 ordered pairs of distinct variables <*A*,*B*> such that *A* is an ancestor of *B* (18.32% of the ordered pairs), and 544 ordered pairs of distinct variables <*A*,*B*> such that *A* is not an ancestor of *B* (81.68% of the ordered pairs.) Of course, making a variable or variables latent and conditioning on a value of a selection variable will reduce the number of ancestor pairs.

We ran the algorithm on 6 different versions of the Alarm network, variously obtained by treating some of the Alarm variables as latent, and by selecting on values of some of the Alarm variables. The results are shown in Table 1. A variable A is made latent by simply removing all of its value from the original data set. A variable B is made a selection variably by choosing a subpopulation which all share the same B value Since conditioning on a B value reduces the sample size, we chose by hand selection variable values that did not reduce the sample size too much. In no case was the sample size reduced below 6000. We ran the algorithm with no latent variables or selection variables, with variable 29 made latent (abbreviated as 29L in Table 1), with variables 29 and 22 made latent, with variable 29 latent and 8 a selection variable (abbreviated by 8S in Table 1), with 29 latent and 1 a selection variable, and with 29, 22, and 4 latent and 8 and 1 as selection variables.

Results of Simulation Studies				
Model	Number of Ancestor Relations Predicted	% Ancestor Relations Predicted Correct	Number of Non-Ancestor Relations Predicted	% Non-Ancestor Relations Predicted Correct
Alarm	62	100.00	1088	96.97
29L	21	100.00	1119	90.80
29L, 22L	25	100.00	1059	91.60
29L, 8S	43	97.67	1082	92.05
29L, 1S	31	83.87	1117	86.75
29L, 22L, 4L, 8S, 1S	23	86.96	861	91.52

Table 1

Often, when the output PAG incorrectly states that A is *not* an ancestor of B the mistake would produce only small errors in predicting the effects on B of intervening on A, because the influence of A on B is very weak. For example, in the last simulation test we did, the output PAG incorrectly stated that 32 is not an ancestor of 6. However, 6 and 32 are almost independent; they pass a test of independence at the .01 significance level. So for the purposes of predicting the effects of intervention, this particular error is not important. However, we have not yet systematically calculated how important the errors that algorithm makes are for prediction.

VII. Future Work

The FCI algorithm could be improved in several ways. First, when the results of the statistical tests that it performs are conflict in the sense that they are not compatible with any DAG, it could make more intelligent compromises based on what the preponderance of the evidence is. Second, there has been some progress recently in heuristic greedy DAG searches based upon maximizing some model score such as the posterior probability of the minimum description length. (Cooper and Herskovits 1992, Heckerman et al. 1994, Chickering et al. 1995.) Combining an independence test algorithm for DAG search (essentially a special case of FCI) and greedy DAG searches based upon maximizing a score has proved successful in simulation studies (Spirtes and Meek 1995, Singh and Valorta 1993). An analogous strategy might improve the accuracy of the FCI algorithm, although the task of calculating scores for a PAG faces a number of obstacles.

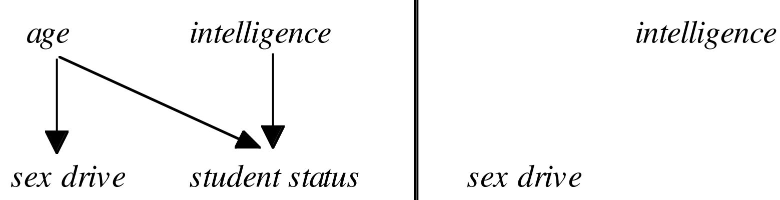
The FCI algorithm will also be tested on a wider variety of DAGs, and empirical examples.

VIII. Appendix

A. *The Assumptions*

The Causal Markov Assumption does not hold for every set of variables, nor for every subpopulation. Consider the following example. Suppose that the DAG in (i) is an accurate description of the causal relations in a population *Pop* for the set of variables **V** = {*age*, *intelligence*, *sex drive*, *student status*}and the Causal Markov Assumption and the Causal Faithfulness Assumption hold in *Pop* for **V**.

Let **V'** = {*intelligence*, *sex drive*}. The causal DAGs for the sets of variables **V** and **V'** are shown in (i) and (ii) respectively of Figure 12.



(i) Causal DAG for **V** in *Pop* and *Pop'*

(ii) Causal DAG for **V'** in *Pop* and *Pop'*

Figure 12

There is no edge between *intelligence* and *sex drive* in (ii) because *intelligence* is not a cause of *sex drive* and *sex drive* is not a cause of *intelligence.* Note that although (i) and (ii) are different DAGs, the causal facts represented in (ii) are a subset of the causal facts represented in (i).

It follows from the Causal Markov Assumption and the Causal Faithfulness Assumption for **V** in *Pop*, that *sex drive* and *intelligence* are dependent in the subpopulation *Pop'* of college students. (Causal Faithfulness entails that *sex drive* and *intelligence* are dependent conditional on some value of *student status*, and because student status is binary, it follows that *sex drive* and *intelligence* are dependent on both values of *student status*).

The Causal Markov Assumption is not true in *Pop'* for **V'**, because *sex drive* and *intelligence* are dependent in the subpopulation *Pop'* of college students, i.e. they are not independent given the parents in (ii) of sex drive or of intelligence (i.e. not independent given the empty set). Indeed because the Causal Markov and Causal Faithfulness Assumptions hold in *Pop* for **V**, the Causal Markov Assumption is *entailed* to fail in *Pop'* for **V'**.

As a consequence, in general we will *not* assume that for the set of measured variables, and the subpopulation from which the sample was drawn, that the Causal Markov and Causal Faithfulness Assumptions hold. Rather we will assume, as in this example, that there is a larger set of variables that includes the measured variables, and a larger population that includes the subpopulation from which the sample was drawn such that:

- the causal DAG in the expanded set of variables for the expanded population satisfies the Causal Markov and Causal Faithfulness Assumptions;
- the causal structure in the subpopulation is the same as the causal structure in the expanded population.

In general, there will be more than one way of expanding the set of variables and the population so that the conditions described above are satisfied; corresponding to these different sets of variables and populations will be different causal DAGs. (For example, instead of adding just *age* and *student status* to **V'**, we could add *age* , *student status*, and some irrelevant variable such as *eye color*, with no edges between *eye color* and any of the other variables.) This does not matter, because the conclusions that we will draw will be true of *all* of the causal DAGs that satisfy the conditions we have laid down.

The assumptions that (i) the **S = 1** subpopulation from which the sample is drawn is part of a population *Pop* in which the Causal Mark and Causal Faithfulness Assumptions hold for some causally sufficient set of variables **V** and (ii) the causal structures relative to **V** in *Pop* and the **S = 1** subpopulation are the same, are sufficient (but not necessary) conditions for the following assumption:

Selection Bias Causal Assumption: For each set of variables **O**, and each population *Pop* such that **S = 1**, there is a causally sufficient set of variables **V** such that $\mathbf{O} \cup \mathbf{S} \subseteq \mathbf{V}$ and for all $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{O}$, $\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid (\mathbf{C} \cup (\mathbf{S} = \mathbf{1}))$ in *Pop* if and only if the causal DAG G relative to **V** in *Pop* entails that $\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid (\mathbf{C} \cup (\mathbf{S} = \mathbf{1}))$ in *Pop*

The Selection Bias Causal Assumption is sufficient for the asymptotic correctness of the methods of inference described in this paper.

B. Proofs

In this section we will prove all of the theorems in the main body of the paper. In order to simplify the proofs, the theorems are not proved in the order they were stated.

In the usual graph theoretic definition, a graph is an ordered pair <**V**,**E**> where **V** is a set of vertices, and **E** is a set of edges. The members of **E** are pairs of vertices (an ordered pair in a directed graph and an unordered pair in an undirected graph). For example, the edge $A \rightarrow B$ is represented by the ordered pair <*A*,*B*>. In directed graphs the ordering of the pair of vertices representing an edge in effect marks an arrowhead at one end of the edge. For our purposes we need to represent a larger variety of marks attached to the ends of undirected edges. In general, we allow that the end of an edge can be marked “out of” by “–”, or can be marked with “>”, or can be marked with an “o”.

In order to specify completely the type of an edge, therefore, we need to specify the variables and **marks** at each end. For example, the left end of "A o$\rightarrow$ B" can be represented as the ordered pair [A, o][3], and the right end can be represented as the ordered pair [B, >]. We will also call [A, o] the A end of the edge between A and B. The first member of the ordered pair is called an endpoint of an edge, e.g. in [A, o] the endpoint is A. The entire edge consists of a set of ordered pairs that represent both of the endpoints,

[3]It is customary to represent the ordered pair *A*, *B* with angle brackets as <*A*, *B*>, but for endpoints of an edge we use square brackets so that the angle brackets will not be misread as arrowheads.

e.g. {[*A*, o], [*B*, >]}. The edge {[*B*, >],[*A*, o]} is the same as {[*A*, o],[*B*, >]} since it doesn't matter which end of the edge is listed first.

Note that a directed edge such as $A \rightarrow B$ has a mark "–"at the *A* end.

We say a **graph** is an ordered triple $<\mathbf{V},\mathbf{M},\mathbf{E}>$ where **V** is a non-empty set of vertices, **M** is a non-empty set of marks, and **E** is a set of sets of ordered pairs of the form $\{[V_1,M_1],[V_2,M_2]\}$, where V_1 and V_2 are in **V**, $V_1 \neq V_2$, and M_1 and M_2 are in **M**. If $G = <\mathbf{V},\mathbf{M},\mathbf{E}>$ we say that *G* is **over V**.

We distinguish the following kinds of edges. An edge $\{[A, -],[B,>]\}$ is a **directed edge** from *A* to *B*, and is written $A \rightarrow B$ or $A \leftarrow B$. An edge $\{[A, -],[B, -]\}$ is an **undirected edge** between *A* and *B*, and is written $A — B$. An edge $\{[A, >],[B, >]\}$ is a **bidirected edge** between *A* and *B*, and is written $A \leftrightarrow B$. An edge {[*A*, o],[*B*, >]} is a **partially directed edge** between *A* and *B*, and is written *A* o→ *B* or *A* ←o *B*. An edge {[*A*, o],[*B*, o]} is a **nondirected edge** between *A* and *B*, and is written *A* o—o *B*. Vertices X, Y, and Z are in a **triangle** in graph *G* if and only if *A* and *B* are adjacent, *B* and *C* are adjacent, and *A* and *C* are adjacent.

For a directed edge $A \rightarrow B$, *A* is the **tail** of the edge and *B* is the **head**; the edge is **out of** *A* and **into** *B*, and *A* is **parent** of *B* and *B* is a **child** of *A*. A sequence of edges $<E_1,...,E_n>$ in *G* is an **undirected path** if and only if there exists a sequence of vertices $<V_1,...,V_{n+1}>$ such that for $1 \leq i \leq n$ E_i has endpoints V_i and V_{i+1}, and $E_i \neq E_{i+1}$. An empty sequence of edges with an associated sequence of vertices $<V_1>$ is an **empty path** between V_1 and V_1. A path *U* is **acyclic** if no vertex appears more than once in the corresponding sequence of vertices. We will assume that an undirected path is acyclic unless specifically mentioned otherwise. A sequence of edges $<E_1,...,E_n>$ in *G* is a **directed path** *D* **from** V_1 to V_n if and only if there exists a sequence of vertices $<V_1,..., V_{n+1}>$ such that for $1 \leq i \leq n$, there is a directed edge $V_i \rightarrow V_{i+1}$ on *D*. If there is an acyclic directed path from *A* to *B* or $B = A$ then *A* is an **ancestor** of *B*, and *B* is a **descendant** of *A*. If **Z** is a set of variables, *A* is an **ancestor** of **Z** if and only if it is an ancestor of a member of **Z**, and similarly for **descendant**. If **X** is a set of vertices in directed acyclic graph *G*(**O**,**S**,**L**), let **Ancestors**(*G*,**X**) be the set of all ancestors of members of **X** in *G*(**O**,**S**,**L**). (If the context makes clear what graph is being referred to, we will simply write **Ancestors**(**X**).)

In a **directed graph**, all of the edges are directed edges. A directed graph is **acyclic** if and only if it contains no directed cyclic paths. A vertex *V* is a **collider** on an undirected path *U* if and only if *U* contains a pair of distinct edges adjacent on the path and into *V*.

The **orientation** of an acyclic undirected path between A and B is the set consisting of the A end of the edge on U that contains A, and the B end of the edge on U that contains B.

If U is an undirected path that is a sequence of edges $<E_1,\ldots,E_n>$, and U' is a subsequence of the edges in U that is also an undirected path, then U' is a **subpath** of U. Note that if U is a cyclic undirected path, and U contains edges $E_i = X \rightarrow Y$ and $E_j = Y \rightarrow Z$ that are not adjacent on U, then a subpath U' may leave out all of the edges between E_i and E_j. If U is acyclic, then there for any two vertices on U there is a unique subpath of U between the two vertices. If U is an acyclic undirected path between X and Y, and U contains distinct vertices A and B, then $U(A,B)$ is the unique subpath of U between A and B.

For three disjoint sets of variables **A**, **B**, and **C**, **A** is **d-separated** from **B** given **C** in DAG G, if and only if there is an undirected path from some member of **A** to a member of **B** such that every collider on that path is either in **C** or has a descendant in **C**, and every non-collider on the path is not in **C**. For three disjoint sets of variables **A**, **B**, and **C**, **A** is **d-connected** to **B** given **C** in DAG G if and only if **A** is not d-separated from **B** given **C**. Geiger, Pearl, and Verma have shown that G entails **A** is independent of **B** given **C** if and only if **A** is d-separated from **B** given **C** in G. See Pearl(1988).

Lemma 1: In a directed acyclic graph G over **V**, if X and Y are not in **Z**, and there is a sequence H of distinct vertices in **V** from X to Y, and there is a set T of undirected paths such that

(i) for each pair of adjacent vertices V and W in H there is a unique undirected path in T that d-connects V and W given $\mathbf{Z}\backslash\{V,W\}$, and

(ii) if a vertex Q in H is in **Z**, then the paths in T that contain Q as an endpoint collide at Q, and

(iii) if for three vertices V, W, Q occurring in that order in H the d-connecting paths in T 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 paths in T that contain X are into (out of) X then U is into (out of) X, and similarly for Y.

Proof. Let U' be the concatenation of all of the paths in T in the order of the sequence H. U' may not be an acyclic undirected path, because it may contain some vertices more than once. Let U be the result of removing all of the cycles from U'. If each edge in U' that contains X is into (out of) X, then U is into (out of) X, because each edge in U is an edge in U'. Similarly, if each edge in U' that contains Y is into (out of) Y, then U is into (out of) Y, because each edge in U is an edge in U'. We will prove that U d-connects X and Y given $\mathbf{Z}$.

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

First we prove that every member R of $\mathbf{Z}$ that is on U is a collider on U. If there is an endpoint edge containing R on U then it is into R because by assumption the paths in T containing R collide at R. If an edge on U is an internal edge with endpoint R 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 $\mathbf{Z}\backslash\{A,B\}$, and R is in $\mathbf{Z}$. All of the edges on paths in T are into R, and hence the subset of those edges that occur on U are into R.

Next we show that every collider R on U has a descendant in $\mathbf{Z}$. R is not equal to either of the endpoints X or Y, because the endpoints of a path are not colliders along the path. If R is a collider on any of the paths in T then R has a descendant in $\mathbf{Z}$ because it is an edge on a path that d-connects two variables A and B not equal to R given $\mathbf{Z}\backslash\{A,B\}$. If R is a collider on two endpoint edges then it has a descendant in $\mathbf{Z}$ by hypothesis. Suppose then that R is not a collider on the path in T between A and B, and not a collider on the path in T between C and D, but after cycles have been removed from U', 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 $\mathbf{Z}$. Hence R has a descendant in $\mathbf{Z}$. ∴

Lemma 2: If G is a directed acyclic graph, R is d-connected to Y given $\mathbf{Z}$ by undirected path U, and W and X are distinct vertices on U not in $\mathbf{Z}$, then $U(W,X)$ d-connects W and X given $\mathbf{Z}$.

Proof. Suppose G is a directed acyclic graph, R is d-connected to Y given $\mathbf{Z}$ by undirected path U, and W and X are distinct vertices on U not in $\mathbf{Z}$. Each non-collider on $U(W,X)$ except for the endpoints is a non-collider on U, and hence not in $\mathbf{Z}$. Every collider on $U(W,X)$ has a descendant in $\mathbf{Z}$ because each collider on $U(W,X)$ is a collider on U, which d-connects R and Y given $\mathbf{Z}$. It follows that $U(W,X)$ d-connects W and X given $\mathbf{Z} = \mathbf{Z}\backslash\{W,X\}$. ∴

Lemma 3: If G is a directed acyclic graph, R is d-connected to Y given $\mathbf{Z}$ by undirected path U, there is a directed path D from R to X that does not contain any member of $\mathbf{Z}$, and X is not on U, then X is d-connected to Y given $\mathbf{Z}$ by a path U' that is into X. If D does not contain Y, then U' is into Y if and only if U is.

Proof. Let D be a directed path from R to X that does not contain any member of $\mathbf{Z}$, and U an undirected path that d-connects R and Y given $\mathbf{Z}$ and does not contain X. Let Q be the point of intersection of D and U that is closest to Y on U. Q is not in $\mathbf{Z}$ because it is on D.

If D does contain Y, then $Y = Q$, and $D(Y,X)$ is a path into X that d-connects X and Y given $\mathbf{Z}$ because it contains no colliders and no members of $\mathbf{Z}$.

If D does not contain Y then $Q \neq Y$. $X \neq Q$ because X is not on U and Q is. By Lemma 2$U(Q,Y)$ d-connects Q and Y given $\mathbf{Z}\backslash\{Q,Y\} = \mathbf{Z}$. Also, $D(Q,X)$ d-connects Q and X given $\mathbf{Z}\backslash\{Q,X\} = \mathbf{Z}$. $D(Q,X)$ is out of Q, and Q is not in $\mathbf{Z}$. By Lemma 1 there is a path U' that d-connects X and Y given $\mathbf{Z}$ that is into X. If Y is not on D, then all of the edges containing Y in U' are on $U(Q,Y)$, and hence by Lemma 1 U' is into Y if and only if U is.
$\therefore$

U is an **inducing path** between A and B in DAG $G(\mathbf{O}, \mathbf{S}, \mathbf{L})$ relative to $\mathbf{O}$ given $\mathbf{S}$ if and only if there is an undirected path from A to B such that every collider on U has a descendant in $\{A,B\} \cup \mathbf{S}$, and no non-collider on U is in $\mathbf{O} \cup \mathbf{S}$. For example, all of the paths between A and B in Figure 4 are inducing paths relative to $\mathbf{O}$ given $\mathbf{S}$. The following theorems generalize Verma and Pearl (1990).

Lemma 4: In directed graph $G(\mathbf{O},\mathbf{S},\mathbf{L})$, if there is an inducing path between A and B that is out of A and into B, then for any subset $\mathbf{Z}$ of $\mathbf{O}\backslash\{A,B\}$ there is an undirected path C that d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ that is out of A and into B.

Proof**.** Let U be an inducing path between A and B that is out of A and into B. Every member of $\mathbf{O} \cup \mathbf{S}$ on U except for the endpoints is a collider, and every collider is an ancestor of either A or B or a member of $\mathbf{S}$.

If every collider on U has a descendant in $\mathbf{Z} \cup \mathbf{S}$, then let $C = U$. C d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ because every collider has a descendant in $\mathbf{Z} \cup \mathbf{S}$, and no non-collider is in $\mathbf{Z} \cup \mathbf{S}$. C is out of A and into B.

Suppose that not every collider on U has a descendant in $\mathbf{Z} \cup \mathbf{S}$. Let R be the collider on U closest to A that does not have a descendant in $\mathbf{Z} \cup \mathbf{S}$, and W be the collider on U closest to A. $R \neq A$ and $R \neq B$ because A and B are not colliders on U.

Suppose first that $R = W$. R is not in $\mathbf{Z} \cup \mathbf{S}$ because R has no descendant in $\mathbf{Z} \cup \mathbf{S}$. There is a directed path from R to B that does not contain A, because otherwise there is a cycle in $G(\mathbf{O},\mathbf{S},\mathbf{L})$. B is not on $U(A,R)$. $U(A,R)$ d-connects A and R given $\mathbf{Z} \cup \mathbf{S}$, and is out of A. By Lemma 3 there is a d-connecting path C between A and B given $\mathbf{Z} \cup \mathbf{S}$ that is out of A and into B.

Suppose then that $R \neq W$. Because U is out of A, W is a descendant of A. W has a descendant in $\mathbf{Z} \cup \mathbf{S}$ by definition of R. It follows that every collider on U that is an ancestor of A has a descendant in $\mathbf{Z} \cup \mathbf{S}$. Hence R is an ancestor of B, and not of A. B is not on $U(A,R)$. $U(A,R)$ d-connects A and R given $\mathbf{Z} \cup \mathbf{S}$ and is out of A. By hypothesis, there is a directed path D from R to B that does not contain A or any member of $\mathbf{Z} \cup \mathbf{S}$. By Lemma 3, there is a path that d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ that is out of A and into B. $\therefore$

Lemma 5: If $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is a directed acyclic graph, and there is an inducing path U between A and B that is into A and into B then for every subset $\mathbf{Z}$ of $\mathbf{O}\backslash\{A,B\}$ there is an undirected path C that d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ that is into A and into B.

Proof**.** If every collider on U has a descendant in $\mathbf{Z} \cup \mathbf{S}$, then U is a d-connecting path between A and B given $\mathbf{Z} \cup \mathbf{S}$ that is into A and into B. Suppose then that there is a collider that does not have a descendant in $\mathbf{Z} \cup \mathbf{S}$. Let W be the collider on U closest to A that does not have a descendant in $\mathbf{Z} \cup \mathbf{S}$. Suppose that W is the source of a directed path D to B that does not contain A. B is not on $U(A,W)$. $U(A,W)$ is a path that d-connects A and W given $\mathbf{Z} \cup \mathbf{S}$, and is into A. By Lemma 3, there is an undirected path C that d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ and is into A and into B. Similarly, if R is the closest collider to B on U that does not have a descendant in $\mathbf{Z} \cup \mathbf{S}$, and R is the source of a directed path D to A that does not contain B, then by Lemma 3, A and B are d-connected given $\mathbf{Z} \cup \mathbf{S}$ by an undirected path into A and into B.

Suppose then that the collider W on U closest to A that does not have a descendant in $\mathbf{Z} \cup \mathbf{S}$ is not the source of a directed path to B that does not contain A, and that the collider R on U closest to B that does not have a descendant in $\mathbf{Z} \cup \mathbf{S}$ is not the source of a directed path to A that does not contain B. The subpath of U from W to A does not contain B or a member of $\mathbf{Z} \cup \mathbf{S}$, and the subpath of U from R to B does not contain A or a member of $\mathbf{Z} \cup \mathbf{S}$. It follows that there exist two colliders E and F on U such that there is a directed path from E to A that does not contain B, there is a directed path from F to B that does not contain A, F is between E and B, and every collider between E and F is an ancestor of a member of $\mathbf{Z} \cup \mathbf{S}$. $U(E,F)$ d-connects E and F given $(\mathbf{Z} \cup \mathbf{S})\backslash\{E,F\}$ because no member of $\mathbf{O} \cup \mathbf{S}$ is a non-collider on $U(E,F)$ except for the endpoints, and every collider on $U(E,F)$ has a descendant in $\mathbf{Z} \cup \mathbf{S}$. The directed path from E to A d-connects E

and A given $(\mathbf{Z} \cup \mathbf{S})\backslash\{E,A\}$ and the directed path from F to B d-connects F and B given $(\mathbf{Z} \cup \mathbf{S})\backslash\{F,B\}$. By Lemma 1 there is an undirected path that d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ that is into A and into B. $\therefore$

Lemma 6: If $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is a directed acyclic graph and U is an inducing path out of both A and B then every collider on U is an ancestor of a member of $\mathbf{S}$.

Proof**.** Let W be the collider on U closest to A and let R be the collider on U closest to B. Since U is out of A and $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is acyclic, W is not an ancestor of A. Similarly, R is not an ancestor of B. Let Q be the collider on U closest to A that is not an ancestor of a member of $\mathbf{S}$.

Suppose that $W = Q$. Hence Q is an ancestor of B. $Q \neq R$ since R is not an ancestor of B. R is not an ancestor of A, since otherwise there would be a cycle in $G(\mathbf{O},\mathbf{S},\mathbf{L})$. Thus R is an ancestor of a member of $\mathbf{S}$ and Q is an ancestor of a member of $\mathbf{S}$. This is a contradiction. The argument for $R = Q$ is similar.

Suppose that $W \neq Q$ and $R \neq Q$. Because $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is acyclic, either W or R is an ancestor of $\mathbf{S}$. Without loss of generality, suppose that W is an ancestor of $\mathbf{S}$. If R is an ancestor of A then R is an ancestor of $\mathbf{S}$ since A is an ancestor of W. If R is not an ancestor of A, then R is an ancestor of a member of $\mathbf{S}$ from the definition of inducing path. In either case R is an ancestor of $\mathbf{S}$. Since W and R are ancestors of $\mathbf{S}$, A and B are ancestors of $\mathbf{S}$ and therefore Q is an ancestor of $\mathbf{S}$ $\therefore$

Lemma 7: If $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is a directed acyclic graph over $\mathbf{V}$, and there is an inducing path U between A and B that is out of A and out of B then for every subset $\mathbf{Z}$ of $\mathbf{O}\backslash\{A,B\}$ there is an undirected path C that d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ that is out of A and out of B.

Proof. Let U be an inducing path out of both A and B. By Lemma 6 every collider on U has a descendant in $\mathbf{Z} \cup \mathbf{S}$. Thus U d-connects A and B given $\mathbf{Z} \cup \mathbf{S}$ and is out of both endpoints. $\therefore$

Lemma 8: If $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is a directed acyclic graph over $\mathbf{V}$ and an undirected path U in G d-connects A and B given $((\mathbf{Ancestors}(G,\{A,B\} \cup \mathbf{S}) \cap \mathbf{O}) \cup \mathbf{S})\backslash\{A,B\}$ then U is an inducing path between A and B.

Proof. If there is a path U that d-connects A and B given $((\mathbf{Ancestors}(G,\{A,B\} \cup \mathbf{S}) \cap \mathbf{O}) \cup \mathbf{S})\backslash\{A,B\}$ then every collider on U is an ancestor of a member of $((\mathbf{Ancestors}(G,\{A,B\} \cup \mathbf{S}) \cap \mathbf{O}) \cup \mathbf{S})\backslash\{A,B\}$, and hence has a descendant in $\{A,B\} \cup \mathbf{S}$. 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 or $\mathbf{S}$. If U d-connects A and B given $((\mathbf{Ancestors}(G,\{A,B\} \cup \mathbf{S}) \cap \mathbf{O}) \cup \mathbf{S})\backslash\{A,B\}$, then every member of $((\mathbf{Ancestors}(G,\{A,B\} \cup \mathbf{S}) \cap \mathbf{O}) \cup \mathbf{S})\backslash\{A,B\}$ that is on U, except for the endpoints, is a collider. Since every vertex on U is in $\mathbf{Ancestors}(G,\{A,B\} \cup \mathbf{S})$, every member of $\mathbf{O}$ that is on U, except for the endpoints, is a collider. Every member of $\mathbf{S}$ on U is a collider. Hence U is an inducing path between A and B. $\therefore$

Lemma 9: $G(\mathbf{O},\mathbf{S},\mathbf{L}))$ entails that for all subsets $\mathbf{X}$ of $\mathbf{O}$, A is dependent on B given $(\mathbf{X} \cup \mathbf{S})\backslash\{A,B\}$ if and only if there is an inducing path between A and B.

Proof. This follows from Lemma 4, Lemma 5, Lemma 7, and Lemma 8 $\therefore$

If $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is a directed acyclic graph, and in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ there is a sequence of vertices M (each of which is in $\mathbf{O}$) starting with A and ending with C, and a set of paths F such that for every pair of vertices I and J adjacent in M in that order there is exactly one inducing path W between I and J in F, and if $J \neq C$ then W is into J, and if $I \neq A$ then W is into I, and I and J are ancestors of $\{A,C\} \cup \mathbf{S}$, then F is an **inducing sequence** between A and C.

Lemma 10: If $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is a directed acyclic graph and there is an inducing sequence F between A and B in $G(\mathbf{O},\mathbf{S},\mathbf{L})$, then in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ a subpath of the concatenation of the paths in F is an inducing path T between A and C such that if the path in F between A and its successor in M is into A then U is into A, and if the path in F between C and its predecessor in M is into C then U is into C.

Proof**.** Suppose that in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ there is a sequence M of vertices in $\mathbf{O}$ starting with A and ending with C, and a set of paths F such that for every pair of vertices I and J adjacent in M there is exactly one inducing path W between I and J, and if $J \neq C$ then W is into J, and if $I \neq A$ then W is into A, and I and J are ancestors of either A or C or $\mathbf{S}$. Let T' be the concatenation of the paths in F. T' may not be an acyclic undirected path because it might contain undirected cycles. Let T be an acyclic undirected subpath of T' between A and C. We will now show that except for the endpoints, every vertex in $\mathbf{O} \cup \mathbf{S}$ on T is a collider, and every collider on T is an ancestor of $\{A,C\} \cup \mathbf{S}$.

If V is a vertex in $\mathbf{O} \cup \mathbf{S}$ that is on T but that is not equal to A or C, every edge on every path in F is into V. Hence, every edge on T that contains V is into V because the edges on T are a subset of the edges on inducing paths in F.

Let R and H be the endpoints of a path W in F. We will now show that every vertex on W is an ancestor of $\{A,C\} \cup \mathbf{S}$. By hypothesis, R is an ancestor of $\{A,C\} \cup \mathbf{S}$, and H is an ancestor of $\{A,C\} \cup \mathbf{S}$. Because W is an inducing path, every collider on W is an ancestor of $\{R,H\} \cup \mathbf{S}$, and hence an ancestor of $\{A,C\} \cup \mathbf{S}$. Every non-collider on W is either an ancestor of R or H, or an ancestor of a collider on W. Hence every vertex on W is an ancestor of either $\{A,C\} \cup \mathbf{S}$. It follows that every collider on T is an ancestor of $\{A,C\} \cup \mathbf{S}$, because the vertices on T are a subset of the vertices on paths in T'.

By definition, T is an inducing path between A and C. Suppose the path in F between A and its successor is into A. If the edge on T with endpoint A is on the path in F 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 is on an inducing path in which A is not an endpoint of the path, then T is into A because A is in $\mathbf{O}$, and hence a collider on every inducing path for which it is not an endpoint. Similarly, T is into C if in F the path between C and its predecessor is into A. $\therefore$

Lemma 11: If $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is a directed acyclic graph, A and B are in $\mathbf{O}$, and $G(\mathbf{O},\mathbf{S},\mathbf{L})$ contains an inducing path U between A and B that is out of A and into B, and A is not an ancestor of $\mathbf{S}$, then there is a directed path from A to B in $G(\mathbf{O},\mathbf{S},\mathbf{L})$.

Proof. Suppose that A is not an ancestor of $\mathbf{S}$, and U is an inducing path between A and B that is out of A and into B. If U does not contain a collider, then U is a directed path from A to B. If U does contain a collider, let D be the first collider after A. By definition of inducing path, there is either a directed path from D to B, D to a member of $\mathbf{S}$, or D to A. There is no path from D to A because there is no cycle in $G(\mathbf{O},\mathbf{S},\mathbf{L})$. There is no directed path from D to a member of $\mathbf{S}$, because A is an ancestor of D, but not an ancestor of a member of $\mathbf{S}$. Hence there is a directed path from D to B. Because U is out of A, and D is the first collider after A, there is a directed path from A to D. Hence there is a directed path from A to B. $\therefore$

$V \in$ **D-SEP**(A,B) in DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ if and only if there is a sequence of vertices $U \equiv \langle A\ldots V\rangle$ in $\mathbf{O} \cap$ **Ancestors**$(\{A,B\} \cup \mathbf{S})$ such that

1. there is an inducing path between every consecutive pair of vertices on U
2. with the exception of the endpoints every vertex on U is not an ancestor of the vertices preceding or succeeding it in the sequence U nor an ancestor of $\mathbf{S}$.

Lemma 12: If there is some subset $\mathbf{W} \subseteq \mathbf{O}\backslash\{A,B\}$ such that A and B are d-separated by $\mathbf{W} \cup \mathbf{S}$ then A and B are d-separated given **D-SEP**$(A,B) \cup \mathbf{S}$.

Proof. Suppose there is some subset $\mathbf{W} \subseteq \mathbf{O}\backslash\{A,B\}$ such that A and B are d-separated by $\mathbf{W} \cup \mathbf{S}$, but A and B are d-connected given **D-SEP**$(A,B) \cup \mathbf{S}$. Let $P \equiv \langle A,\ldots B\rangle$ be a path d-connecting A and B given **D-SEP**$(A,B) \cup \mathbf{S}$.

If every vertex in **O** on P occurs as a collider then every observed vertex on P is an ancestor of A or B or **S** (since **D-SEP**(A,B) $\cup$ **S**)$\subseteq$ **Ancestors**($\{A,B\} \cup$ **S**).) Hence in this case P constitutes an inducing path between A and B, and so there is no subset **W** $\subseteq$ **O**$\backslash\{A,B\}$ such that A and B are d-separated by some **W** $\cup$ **S**.

Hence there is some vertex $O \in$ **O**, such that O is a non-collider on P. Suppose without loss that O is the first such vertex on P. We will now show that $O \in$ **D-SEP**(A,B) $\cup$ **S**.

Consider the subpath $P(A,O)$. Let $<C_1,\ldots C_m>$ denote the (possibly empty) sequence of colliders on $P(A,O)$, which are ancestors of **D-SEP**(A,B), but not ancestors of **S**. Hence there is a directed path (possibly of length 0) from $C_i \rightarrow \ldots \rightarrow D_i$, where $D_i \in$ **D-SEP**(A,B). Let F_i be the first measured vertex on the path $C_i \rightarrow \ldots \rightarrow D_i$; such an F_i is guaranteed to exist since $D_i \in$ **O**.

We will now show that there is an inducing path between F_i and F_{i+1}. It follows that no F_i is an ancestor of **S**, because no C_i is an ancestor of **S**. Consider the path Q_i formed by concatenating the directed path $F_i \leftarrow \ldots \leftarrow C_i$, the subpath $P(C_i,C_{i+1})$, and the directed path $C_{i+1} \rightarrow \ldots F_{i+1}$. It follows from the construction, that with the exception of F_i and F_{i+1} the only measured vertices on Q_i are on $P(C_i,C_{i+1})$. Moreover, since O is the first non-collider on P that is in **O**, every measured vertex on $P(C_i,C_{i+1})$ is a collider. Hence by construction of the sequence $<C_1,\ldots C_m>$, every measured vertex on $P(C_i,C_{i+1})$ is an ancestor of **S**. Hence Q_i is an inducing path.

Similarly, by concatenating the path $P(A,C_1)$ and the path $C_1 \rightarrow \ldots \rightarrow F_1$, and by concatenating the path $F_m \leftarrow \ldots \leftarrow C_m$ and $P(C_m,O)$, we may form inducing paths Q_0 and Q_m between A and F_1, and between F_m and O respectively.

Note that all of the inducing path Q_k that we have constructed are into the F_i vertices. The sequence $R \equiv <A \equiv F_0,F_1,\ldots F_m,F_{m+1} \equiv O>$ thus consists of a sequence each of which is an ancestor of A or B or **S**, and such that each consecutive pair in the sequence is connected by an inducing path. Hence this sequence satisfies the first condition necessary to show that $O \in$ **D-SEP**(A,B).

We now construct a subsequence of $T \equiv <\mathrm{A} \equiv F_{\alpha(0)},F_{\alpha(1)}\ldots F_{\alpha t)} \equiv O>$ satisfying property (ii) as follows:

1. Let $\alpha(0) = 0$, so $F_{\alpha(0)} = F_0 \equiv \mathrm{A}$.
2. Let $\alpha(1)$ be the largest η such that there is an inducing path between $F_{\alpha(0)}$ and F_η which is into F_η if there is such an inducing path, else let $F_{\alpha(1)} \equiv O$.

3. Let $\alpha(k+1)$ be the largest $\eta > k$ such that there is an inducing path between $F_{\alpha(k)}$ and F_η which is into $F_{\alpha(k)}$ and, if $\eta < m+1$, into F_η.
4. If $\alpha(k) = m+1$ then stop.

(Note that at each stage in the construction, so long as $\alpha(k) < m+1$, there is guaranteed to be some η such that there is an inducing path between $F_{\alpha(k)}$ and F_η which is into $F_{\alpha(k)}$ and, if $\eta < m+1$, into F_η, since, for $i > 0$ there is an inducing path between F_i and F_{i+1}, which is into F_i, and for $i+1 < m+1$, into F_{i+1}.)

We will now show that for $i \neq 0$, and $i \neq m+1$, $F_{\alpha(i)}$ is not an ancestor of $F_{\alpha(i-1)}$ or $F_{\alpha(i+1)}$. Suppose that $F_{\alpha(i)}$ is an ancestor of $F_{\alpha(i-1)}$ or $F_{\alpha(i+1)}$. By construction there is an inducing path between $F_{\alpha(i)}$ and $F_{\alpha(i-1))}$ which is into $F_{\alpha(i)}$, likewise there is an inducing path between $F_{\alpha(i)}$ and $F_{\alpha(i+1)}$ which is into $F_{\alpha(i)}$. Hence if $F_{\alpha(i)}$ is an ancestor of $F_{\alpha(i-1)}$ or $F_{\alpha(i+1)}$ then there is an inducing path between $F_{\alpha(i-1)}$ and $F_{\alpha(i+1)}$ which is into both $F_{\alpha(i-1)}$ and $F_{\alpha(i+1)}$ (unless $F_{\alpha(i-1)}$ or $F_{\alpha(i+1)}$ is an endpoint). But in that case $\alpha(i)$ is not the largest η such that there is an inducing path between $F_{\alpha(i-1)}$ and F_η which is into $F_{\alpha(i-1)}$ and, if $\eta < m+1$, into F_η. This is a contradiction. $\therefore$

Lemma 13: If π_0 is the partially oriented graph constructed in step C) of the Fast Causal Inference Algorithm for $G(\mathbf{O},\mathbf{S},\mathbf{L})$, A and B are in $\mathbf{O}$, and A is not an ancestor of B in G', then every vertex in **D-SEP**(A,B) in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ is in **Possible-D-SEP**(A,B, π_0).

Proof**.** Suppose that A is not an ancestor of B. If V is in **D-SEP**(A,B), in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ there is a sequence of vertices $U \equiv \langle A \ldots V \rangle$ in $\mathbf{O} \cap$ **Ancestors**$(\{A,B\} \cup \mathbf{S})$, an inducing path between every consecutive pair of vertices on U with the exception of the endpoints, and every vertex on U is not an ancestor of the vertices preceding or succeeding it in the sequence U. The proof is by induction on the length of U. If the length of U is 1, **Possible-D-SEP**(A,B, π_0) includes **D-SEP**(A,B) because it contains all edges adjacent to A in π_0. Suppose that each vertex V that is in **D-SEP**(A,B)because of a sequence U of no more than length n, V is in **Possible-D-SEP**(A,B,π_0). Let W be a vertex in **D-SEP**(A,B) because of a sequence U of length $n + 1$, and X is the predecessor of W in U, and Y is the predecessor of X in U. If there is an inducing path between W and Y, then W is in **Possible-D-SEP**(A,B,π_0). Suppose that there is no inducing path between W and Y. Because X is not an ancestor of Y or W, the inducing path between W and X is into X. Similarly, the inducing path between Y and X is into X. Hence X is not in any set that d-separates Y and W. Y and W are d-separated given a subset of $\mathbf{O} \backslash \{W,Y\}$ because there is no inducing path between them. Hence, step C) of the FCI algorithm orients the edge between W and X into X. It follows that W is in **Possible-D-SEP**(A,B,π_0). $\therefore$

Because π_0 contains more edges than the output PAG, the orientations in π may not be correct for the two following reasons. First, an edge that is in π_0, but not in the output PAG, may hide some collider along a path. Second, a vertex may be oriented as a collider in π_0, but not in the output PAG, because of a "collision" involving an edge in π that does not occur in the output PAG. However, either of these mistakes in orientation in π simply makes **Possible-D-SEP**(A,B,π) larger, and so it still includes **D-SEP**(A,B,G').

Lemma 14: Suppose that in a graph G(**O**,**S**,**L**), X, Y, Z $\in$ **O**, and Y is not an ancestor of X or Z or **S**. If there is a set **Q** $\subseteq$ **O**, containing Y, such that for every subset **T** $\subseteq$ **Q**\{Y}, X and Z are d-connected given **T** $\cup$ **S**, then X and Z are d-connected given **Q** $\cup$ **S.**

Proof: Let **T***= (**Ancenstors**({X,Z} $\cup$ **S**)) $\cap$ **Q**. Now **T*** $\subseteq$ **Q**, but since, by hypothesis, Y$\notin$ **Ancenstors**({X,Z} $\cup$ **S**), it follows that **T*** $\subseteq$ **Q**\{Y}. Hence, again, by hypothesis, there is a path P d-connecting X and Z given **T*** $\cup$ **S**. By the definition of a d-connecting path, every vertex on P is either an ancestor of X or Z, or **T*** $\cup$ **S**. Since **T*** $\subseteq$ (**Ancenstors**({X,Z} $\cup$ **S**)) $\cap$ **Q**, it follows that every vertex on P is in **Ancenstors**({X,Z} $\cup$ **S**). Since no vertex in **Q\T*** is in **Ancenstors**({X,Z} $\cup$ **S**), it follows that no vertex in **Q\T*** lies on P. But since **T*** $\cup$ **S** $\subseteq$ **Q** $\cup$ **S**, the only way in which P could fail to d-connect X and Z given **Q** $\cup$ **S** would be if some vertex in (**Q** $\cup$ **S**)\(**T** *$\cup$ **S**) = **Q\T*** lay on the path. Hence P still d-connects X and Z given **Q** $\cup$ **S**.$\therefore$

In a graph G(**O**,**S**,**L**) if X and Z are d-separated given **Q**$\cup$**S**, ({X,Z}$\cup$**Q**$\subseteq$**O**), and for any proper subset **T**$\subset$**Q**, X and Z are d-connected given **T**$\cup$**S**, then **Q** is a **minimal d-separating set** for X and Z in G(**O**,**S**,**L**).

Corollary: If Y is in a minimal d-separating set for X and Z in G(**O**,**S**,**L**), then Y is an ancestor of X or Z or **S** in G(**O**,**S**,**L**).

Proof: Suppose for a contradiction that there were a minimal d-separating set **Q** for X and Z, which contained some vertex Y $\notin$ **Ancenstors**({X,Z} $\cup$ **S**). It would then follow from Lemma 14, and the definition of a minimal d-separating set, that X and Z were d-connected given **Q** $\cup$ **S**, which is a contradiction.$\therefore$

Theorem 5: If P(V) is faithful to G_1(**O**,**S**$_1$,**L**$_1$), and the input to the FCI algorithm is an oracle for P(**V**) over **O** given **S** = **1**, the output is a PAG of G_1(**O**,**S**$_1$,**L**$_1$).

Proof. The adjacencies are correct by Lemma 12 and Lemma 13. We will now show that the orientations are correct. The proof is by induction on the number of applications of orientation rules in the repeat loop of the Fast Causal Inference Algorithm.. Let the object constructed by the algorithm after the n^{th} iteration of the repeat loop be π'_n (Note that according to the notation of the algorithm, $\pi'_0 = \pi_2$.) Note that each set **Sepset**(A,C) is a minimal d-connecting set, because if there were any subset **X** of **Sepset**(A,C) such that **X** $\cup$ **S** d-separated A and C, then the algorithm would have found **X** at an earlier stage and made **Sepset**(A,C) equal to **X**.

Base Case: Suppose that the only orientation rule that has been applied is that if A *—* B *—* C in π'_0, but A and C are not adjacent in π'_0, A *—* B *—* C is oriented as A *$\rightarrow$ B $\leftarrow$* C if B is not a member of **Sepset**(A,C) and as A *—$\underline{\text{* } B \text{ *}}$—* C if B is a member of **Sepset**(A,C). Suppose first that B is not in **Sepset**(A,C), and that contrary to the hypothesis that B is an ancestor of A or C or **S**. If G(**O,S,L**) is an arbitrary member of **O**-Equiv($\boldsymbol{G}_1$($\mathbf{O}$,$\mathbf{S}_1$,$\mathbf{L}_1$)), there is an inducing path between A and B and an inducing path between B and C. Hence there are is a path U_1 that d-connects A and B given **Sepset**(A,C)\\{A} $\cup$ **S** and a path U_2 that d-connects B and C given **Sepset**(A,C)\\{C} $\cup$ **S**. If U_1 and U_2 do not collide at B then, by Lemma 1, A and C are d-connected given **Sepset**(A,C) $\cup$ **S**, which is a contradiction. If U_1 and U_2 do collide at B, and B is an ancestor of **S**, then, by Lemma 1, A and C are d-connected given **Sepset**(A,C) $\cup$ **S**, which is a contradiction. If B is an ancestor of A or C but not of **S**, then either there is a directed path D from B to C that does not contain A or **S**, or there is a directed path D from B to C that does not contain A or **S**. Suppose without loss of generality that the latter is the case. It follows that D d-connects B and C given **Sepset**(A,C)\\{C} $\cup$ **S** and is out of B. It follows by Lemma 1 that A and C are d-connected given **Sepset**(A,C) $\cup$ **S**, which is a contradiction.

Suppose next that B is in **Sepset**(A,C), but that B is not an ancestor of A or C or **S**. It follows from Lemma 11 that the inducing paths in G(**O,S,L**) are both into B. It follows from Lemma 1 that A and C are d-connected given **Sepset**(A,C) $\cup$ **S**, which is a contradiction. Hence, B is an ancestor of A or C or of **S**.

Induction Case: Suppose π'_n is a partial ancestral graph of G(**O**,**S**,**L**). We will now show that π'_{n+1} is a partial ancestral graph of G(**O**,**S**,**L**).

Case 1: There is a directed path from D from A to B and an edge A *—* B in π'_n, so A *—* B is oriented as A *$\rightarrow$ B. By the induction hypothesis, there is a directed path from A to B in G(**O**,**S**,**L**). Hence B is not an ancestor of A in G(**O**,**S**,**L**).

Case 2: If B is a collider along <A,B,C> in π_2, A is not adjacent to C, B is adjacent to D, and D is a non-collider along <A,D,C>, then orient B *—* D as B $\leftarrow$* D. Because D is a non-collider along <A,D,C>, D is an ancestor of A or C or **S.** If B is an ancestor of D, then B is an ancestor of A or C or **S**. But because B is a collider along $\langle A,B,C\rangle$, it is not an ancestor of A or C or **S**. Hence B is not an ancestor of D.

Case 3: If $P \ast\!\!\rightarrow \underline{M \ast}\!\!-\!\!\ast R$ then the orientation is changed to $P \ast\!\!\rightarrow M \rightarrow R$. By the induction hypothesis, if $P \ast\!\!\rightarrow \underline{M \ast}\!\!-\!\!\ast R$ in π'_n, then in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ M is an ancestor of either P or R or **S**. Because $P \ast\!\!\rightarrow M$ in π'_n, M is not an ancestor of P or S. Hence M is an ancestor of R..

Case 4: If $B \leftarrow\!\!\ast\, C$, $B \rightarrow D$, and D o—$\ast$ C, then orient as $D \leftarrow\!\!\ast\, C$. By the induction hypothesis, B is not an ancestor of C, but is an ancestor of D. Hence D is not an ancestor of C.

Case 5: If U is a definite discriminating path between A and C for B in π'_n, and D is adjacent to C on U, and D, B, and C form a triangle, then if B is in **Sepset**(A,C) then mark B as a non-collider on subpath $D \ast\!\!-\!\!\underline{\ast\ B\ \ast}\!\!-\!\!\ast C$ else orient $D \ast\!\!-\!\!\ast B \ast\!\!-\!\!\ast C$ as $D \ast\!\!\rightarrow B \leftarrow\!\!\ast\, C$.

There are two cases. First suppose that B is in **Sepset**(A,C). Suppose, contrary to the hypothesis that B is not an ancestor of C or D or **S** in $G(\mathbf{O},\mathbf{S},\mathbf{L})$. Because **Sepset**($A$,$C$) is a minimal d-connecting set, and B is in **Sepset**(A,C) then B is an ancestor of A or C or **S**. Because it is not an ancestor of C or **S** it is an ancestor of A. Because there are inducing paths between B and C, and B and D in $G(\mathbf{O},\mathbf{S},\mathbf{L})$, but B is not an ancestor of C or D or **S**, it follows from Lemma 11 that the inducing paths between D and B, and between B and C, are both into B. The directed edge from each vertex X_i on U (except for A) to C in π', entails that X_i is an ancestor of C but not of **S** in $G(\mathbf{O},\mathbf{S},\mathbf{L})$. So, the inducing path in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ corresponding to a bi-directed edge between X_i and X_{i+1} on U is into X_i and X_{i+1}. Hence in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ there is in inducing sequence between A and C. Hence, by Lemma 10 in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ there is an inducing path between A and C, which is a contradiction.

Suppose next that B is not in **Sepset**(A,C). First we will show that every vertex along U except for the endpoints is an ancestor of **Sepset**(A,C) in $G(\mathbf{O},\mathbf{S},\mathbf{L})$. Suppose, contrary to the hypothesis that some vertex on U is not an ancestor of **Sepset**(A,C), and let W be the closest such vertex on U to B. It follows that in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ there is a sequence of vertices $\langle A,X_1,\ldots,X_n,W\rangle$ such that each pair of vertices X_i and X_{i+1} that are adjacent in the sequence are d-connected given **Sepset**(A,C)$\backslash\{X_i,X_{i+1}\}$ (because of the existence of the inducing path into X_i and X_{i+1}), and if a pair of paths d-connects X_{i-1} and X_i, and X_i and X_{i+1} respectively, they collide at X_i. By hypothesis, in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ X_i is an ancestor of **Sepset**(A,C), and W is an ancestor of C but not of **Sepset**(A,C). It follows there is a path D from W to C that d-connects W and C given **Sepset**(A,C)$\backslash\{W\}$ in $G(\mathbf{O},\mathbf{S},\mathbf{L})$. By Lemma

1 it follows that A and C are d-connected given **Sepset**(A,C) in G(**O**,**S**,**L**), which is a contradiction.

Since every vertex along U except for the endpoints is an ancestor of **Sepset**(A,C) in G(**O**,**S**,**L**), it follows that there is a sequence of vertices $\langle A, X_1, \ldots, X_n, B \rangle$ such that each pair of vertices X_i and X_{i+1} that are adjacent in the sequence are d-connected given **Sepset**(A,C)\\{X_i,X_{i+1}}, and if a pair of paths d-connects X_{i-1} and X_i, and X_i and X_{i+1} respectively, they collide at X_i. Since each of the X_i has a descendant in **Sepset**(A,C) in G(**O**,**S**,**L**), A and B are d-connected given **Sepset**(A,C) in G(**O**,**S**,**L**)

Suppose, contrary to the hypothesis, that B is an ancestor of C. There are two cases. If B is an ancestor of **Sepset**(A,C), then in G(**O**,**S**,**L**) A and B are d-connected given **Sepset**(A,C), and B and C are d-connected given **Sepset**(A,C), and B is an ancestor of **Sepset**(A,C) but not in **Sepset**(A,C). It follows from Lemma 1 that A and C are d-connected given **Sepset**(A,C), which is a contradiction. If in G(**O**,**S**,**L**) B is not an ancestor of **Sepset**(A,C) but is an ancestor of C, then in G(**O**,**S**,**L**) there is a directed path D from B to C that contains no member of **Sepset**(A,C). By Lemma 1 it follows that B and C are d-connected given **Sepset**(A,C) in G(**O**,**S**,**L**) which is a contradiction. It follows that B is not an ancestor of C. Because in G(**O**,**S**,**L**) B is not an ancestor of C, but D is an ancestor of C by hypothesis, B is not an ancestor of D. $\therefore$

Theorem 1: If π is a partial ancestral graph, and there is a directed path U from A to B in π, then in every DAG G(**O**,**S**,**L**) with PAG π, there is a directed path from A to B and A is not an ancestor of **S**.

Proof**.** By Theorem 5, for each directed edge between M and N in U there is a directed path from M to N in G(**O**,**S**,**L**) and M is not an ancestor of **S**. The concatenation of the directed paths in G(**O**,**S**,**L**) contains a subpath that is a directed path from A to B in G(**O**,**S**,**L**). Because there is a directed edge between A and its successor on U, A is not an ancestor of **S**. $\therefore$

A **semi-directed path from** A to B in partial ancestral graph π is an undirected path U from A to B in which no edge contains an arrowhead pointing towards A, that is, there is no arrowhead at A on U, and 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.

Lemma 15: If π is a partial ancestral graph of directed acyclic graph G(**O**,**S**,**L**), there is a directed path D from A to B in G(**O**,**S**,**L**) that does not contain any member of **S**, and **C** is the set of vertices in **O** on D, then there is a semi-directed path from A to B in π that contains just the members of **C**.

Proof. Suppose there is a directed path D from A to B in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ that does not contain any member of **S**. Let **C** be the set of vertices in **O** on D, and D' in π be the sequence of edges between vertices in **O** along D in the order in which they occur on D. Let X and Y be any pair of vertices adjacent on D' for which X is between A and Y or $X = A$. Because X is an ancestor of Y in $G(\mathbf{O},\mathbf{S},\mathbf{L})$, the edge in π between X and Y has either a "o" or "–" at the X end of the edge. Hence D' is a semi-directed path from A to B in π that contains the members of **C**.

Theorem 2: If π is a partial ancestral graph, and there is no semi-directed path from A to B in π that contains a member of **C**, then every directed path from A to B in every DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ with PAG π that contains a member of **C** also contains a member of **S**.

Proof. This follow from Lemma 15. $\therefore$

Theorem 3: If π is a partial ancestral graph of DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$, and there is no semi-directed path from A to B in π, then every directed path from A to B in every DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ with PAG π contains a member of **S**.

Proof. By Lemma 15, if there is a directed path from A to B in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ that contains no member of **S**, there is a semi-directed path from A to B in π. $\therefore$

Theorem 4: If π is a partial ancestral graph, and every semi-directed path from A to B contains some member of **C** in π, then every directed path from A to B in every DAG $G(\mathbf{O},\mathbf{S},\mathbf{L})$ with PAG π contains a member of $\mathbf{S} \cup \mathbf{C}$.

Proof. Suppose that U is a directed path in $G(\mathbf{O},\mathbf{S},\mathbf{L})$ from A to B that does not contain a member of **C** or **S**. Then by Lemma 15 there is a semi-directed path from A to B in π that does not contain any member of **C**. $\therefore$

IX. Bibliography

Beinlich, I., Suermondt, H., Chavez, R., and Cooper, G. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks, in Proc. Second European Conference on Artificial Intelligence in Medicine, London, England. 247-256.

Bollen, K. (1989). Structural Equations with Latent Variables. Wiley, NY.

Chickering, D., Geiger, D., and Heckerman, D. (1994) "Learning Bayesian networks: Search methods and experimental results". Preliminary papers of the fifth international workshop on Artificial Intelligence and Statistics.

Cooper, G., and Herskovits, E. (1992). "A Bayesian Method for the Induction of Probabilistic Networks from Data." Machine Learning, 9, 309-347.

Cooper, G. (1995). “Causal Discovery from Data in the Presence of Selection Bias”. Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statistics, Fort Lauderdale, FL, January 5-7, 1995, pp. 140-150.

Epstein, R. (1987) A History of Econometrics, North-Holland, New York.

Heckerman, D., Geiger, D., and Chickering, D. (1994). “Learning Bayesian Networks: The Combination of Knowledge and Statistical Data.” Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp. 293-302, Morgan Kaufmann.

Lauritzen, S., Dawid, A., Larsen, B., Leimer, H. (1990). "Independence Properties of Directed Markov Fields." Networks, 20, 491-505.

Meek, C. (1995) “Strong Completeness and Faithfulness in Bayesian Networks”. Uncertainty in Artificial Intelligence: Proceedings of the XI Conference. Morgan Kaufmann.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. Morgan and Kaufman, San Mateo.

Pearl, J. (1995). “Causal Diagrams for empirical research.” Biometrika, 82, 669-710.

Richardson, T. (1996) “A Discovery Algorithm for Directed Cyclic Graphs”, *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, Eric Horvits and Finn Jensen (Eds.), Morgan Kaufmann Publishers, Inc., San Mateo, CA.

Singh, M. and Valtorta, M. (1993), An algorithm for the construction of Bayesian network structures from data, Proceedings of the Ninth conference on Uncertainty in Artificial Intelligence 259-265, Morgan Kaufmann.

Spirtes P., Glymour C., Scheines, R. (1990). Causality from probability. Conference Proceedings: Advanced Computing for the Social Sciences, Williamsburgh, VA.

Spirtes, P., Glymour, C., and Scheines, R. (1991). An algorithm for fast recovery of sparse causal graphs. Social Science Computer Review, vol. 9, 1991, pp. 62-72.

Spirtes, P., and Verma, T. (1992) Equivalence of Causal Models with Latent Variables, Carnegie Mellon University Department of Philosophy Technical Report Phil-33.

Spirtes, P., Glymour, C., and Scheines, R. (1993).Causation, Prediction, and Search. Springer-Verlag Lecture Notes in Statistics 81, NY.

Spirtes, P. (1992). Building causal graphs from statistical data in the presence of latent variables", forthcoming in Proceedings of the IX International Congress on Logic, Methodology, and the Philosophy of Science, B. Skyrms, ed., Uppsala, Sweden, 1991.

Spirtes, P. and Meek, C. (1995) Learning Bayesian Networks with Discrete Variables from Data", in Proceedings of The First International Conference on Knowledge Discovery and Data Mining, ed. by Usama M. Fayyad and Ramasamy Uthurusamy, AAI Press, pp. 294-299.

Spirtes, P., Richardson, T., Meek, C., Scheines, R., and Glymour, C. (1996) Using D-separation to Calculate Zero Partial Correlations in Linear Models with Correlated Errors, Carnegie Mellon University Department of Philosophy Technical Report Phil-72.

Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. Proc. Sixth Conference on Uncertainty in AI. Association for Uncertainty in AI, Inc., Mountain View, CA, 220-227.

Verma, T. and Pearl, J. (1991). Equivalence and synthesis of causal models. Technical Report R-150, Cognitive Systems Laboratory, University of California, Los Angeles.

Wermuth, N., Cox, D., and Pearl, J. (1994). "Explanations for multivariate structures derived from univariate recursive regressions", Report 94-1, University of Mainz, 1994.

Wright, S. (1934). The method of path coefficients. Ann. Math. Stat. 5, 161-215.