Automated Discovery of Linear Feedback Models

by

Thomas Richardson and Peter Spirtes

October 1996

Report CMU-PHIL-75

Philosophy
Methodology
Logic

Pittsburgh, Pennsylvania 15213-3890

Automated Discovery of Linear Feedback Models[1]

by

Thomas Richardson and Peter Spirtes

1. Introduction

The introduction of statistical models represented by directed acyclic graphs (DAGs) has proved fruitful in the construction of expert systems, in allowing efficient updating algorithms that take advantage of conditional independence relations (Pearl, 1988, Lauritzen *et al.* 1988), and in inferring causal structure from conditional independence relations (Spirtes and Glymour, 1991, Spirtes, Glymour and Scheines, 1993, Pearl and Verma, 1991, Cooper, 1992). As a framework for representing the combination of causal and statistical hypotheses, DAG models have shed light on a number of issues in statistics ranging from Simpson's Paradox to experimental design (Spirtes, Glymour and Scheines, 1993). The relations of DAGs with statistical constraints, and the equivalence and distinguishability properties of DAG models, are now well understood, and their characterization and computation involves three properties connecting graphical structure and probability distributions: (i) a local directed Markov property, (ii) a global directed Markov property, and (iii) factorizations of joint densities according to the structure of a graph (Lauritizen, *et al.*, 1990).

Recursive structural equation models are one kind of DAG model. However, non-recursive structural equation models are not DAG models, and are instead naturally represented by directed *cyclic* graphs in which a finite series of edges representing influence leads from a vertex representing a variable back to that same vertex. Such graphs have been used to model feedback systems in electrical engineering (Mason, 1953, 1956), and to represent economic processes (Haavelmo, 1943, Goldberger, 1973). In contrast to the acyclic case, almost nothing general is known about how directed cyclic graphs (DCGs) represent conditional independence constraints, or about their equivalence

[1] Research for this paper was supported by the National Science Foundation through grant 9102169 and the Navy Personnel Research and Development Center and the Office of Naval Research through contract number N00014-93-1-0568. We are indebted to Clark Glymour, Richard Scheines, Christopher Meek, and Marek Druzdel for helpful conversations. We also wish to thank anonymous referees for helpful comments, corrections, simplifications, and clarifications.

or identifiability properties, or about characterizing classes of DCGs from conditional independence relations or other statistical constraints. This paper addresses all of these issues. The issues turn on how the relations among properties (i), (ii) and (iii) essential to the acyclic case generalize—or fail to generalize—to directed cyclic graphs and associated families of distributions. It will be shown that when DCGs are interpreted by analogy with DAGs as representing functional dependencies with independently distributed noises or "error variables," the equivalence of the fundamental global and local Markov conditions characteristic of DAGs no longer holds, even in linear systems. For linear systems associated with DCGs with independent errors or noises, a characterisation of conditional independence constraints is obtained, and it is shown that the result generalizes in a natural way to systems in which the error variables or noises are statistically dependent.

We also present a correct polynomial time (on sparse graphs) discovery algorithm for linear cyclic models that contain no latent variables. This algorithm outputs a representation of a class of non-recursive linear structural equation models given observational data as input. Under the assumption that all conditional independencies found in the observational data are true for structural reasons rather than because of particular parameter values, the algorithm discovers causal features of the structure which generated the data. (Discovery algorithms for directed acyclic graphs based upon similar assumptions are described in Spirtes et al. 1993, and Pearl and Verma 1991.)The remainder of this paper is organized as follows: Section 2 defines relevant mathematical ideas and gives some necessary technical results on DAGs and DCGs. Section 3 obtains results for non-recursive linear structural equations models. Section 4 describes a discovery algorithm. Section 5 describes some open research problems. All proofs are in Section 6.

2. Directed Graphs and Probability Distributions

A **directed graph** (DG) is an ordered pair of a finite set of vertices **V**, and a set of directed edges **E**. (We place sets of variables and defined terms in boldface.) A directed edge from A to B is an ordered pair of distinct vertices <A,B> in **V** (depicted as A $\rightarrow$ B) in which A is the **tail** of the edge and B is the **head**; the edge is **out of** A and **into** B, and A is a **parent** of B and B is a **child** of A; also A and B are **adjacent**. A sequence of edges $<E_1,...,E_n>$ in a directed graph G is an **undirected path** if and only if there exists a sequence of vertices $<V_1,...,V_{n+1}>$ such that for $1 \leq i \leq n$ either $<V_i,V_{i+1}> = E_i$ or $<V_{i+1},V_i> = E_i$ and $E_i \neq E_{i+1}$. A sequence of edges $<E_1,...,E_n>$ in a directed graph G is a

directed path if and only if there exists a sequence of vertices $<V_1,...,V_{n+1}>$ such that for $1 \leq i \leq n$ $<V_i,V_{i+1}> = E_i$. A (directed or undirected) path U is **acyclic** if no vertex occurring on an edge in the path occurs more than once. 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. A directed graph is **acyclic** if and only if it contains no directed cyclic paths.[2]

A directed acyclic graph (DAG) G with a set of vertices **V** can be given two distinct interpretations. On the one hand, such graphs can be used to represent causal relations between variables, where an edge from A to B in G means that A is a direct cause of B relative to **V**. A **causal graph** is a DAG given such an interpretation. Here we take the concept of "direct cause relative to a set of variables" to be primitive. There is a large philosophical literature that attempts to define various causal relations (see e.g. Sosa 1975). However, for the theorems in this paper, such definitions are not needed. The key assumptions we make are the ones *relating* causal relations to probability distributions, and these are stated and justified in section 4.

On the other hand, a DAG with a set of vertices **V** can also represent a set of probability measures over **V** (where the members of **V** are both the vertices of the graph and random variables). Following the terminology of Lauritzen *et al.* (1990) say that a probability measure over a set of variables **V** satisfies the **local directed Markov property** for a directed acyclic graph (or DAG) G with vertices **V** if and only if for every W in **V**, W is independent of **V**\(**Descendants**(W) $\cup$ **Parents**(W)) given **Parents**(W), where **Parents**(W) is the set of parents of W in G, and **Descendants**(W) is the set of descendants of W in G. (Note that the vertices do not merely *index* the random variables; rather the random variables are the vertices of the graph. A vertex is its own ancestor and descendant, although not its own parent or child.) A DAG G **represents** the set of probability measures which satisfy the local directed Markov property for G.

The use of DAGs to simultaneously represent a set of causal hypotheses and a family of probability distributions extends back to the path diagrams introduced by Sewell Wright (1934). Variants of probabilistic DAG models were introduced in the 1980's in Wermuth (1980), Wermuth and Lauritzen (1983), Kiiveri, Speed, and Carlin (1984), Kiiveri and Speed (1982), and Pearl (1988). In Section 4 we will present assumptions which link the two interpretations of directed graphs.

---

[2]An undirected path is often defined as a sequence of vertices rather than a sequence of edges. The two definitions are essentially equivalent for acyclic directed graphs, because a pair of vertices can be identified with a unique edge in the graph. However, a cyclic graph may contain more than one edge between a pair of vertices. In that case it is no longer possible to identify a pair of vertices with a unique edge.

Pearl(1988) defines a global directed Markov property that has been shown to be equivalent to the local directed Markov property for DAGs, and can be used to calculate the consequences of the local directed Markov property. (See e.g. Lauritzen *et al.* 1990.[3]) Several preliminary notions are required. Vertex X is a **collider** on an acyclic undirected path U in directed graph G if and only if there are two adjacent edges on U directed into X (e.g. A $\rightarrow$ X $\leftarrow$ B). Every other vertex on U is a **non-collider** on U. In a directed graph G, if X and Y are not in **Z**, then an acyclic undirected path U **d-connects** X and Y given **Z** if and only if every collider on U has a descendant in **Z**, and no non-collider on U is in **Z**. For three disjoint sets **X**, **Y**, and **Z**, **X** and **Y** are **d-connected** given **Z** in G if and only if there is a path U that d-connects some X in **X** to some Y in **Y** given **Z**. For three disjoint sets **X**, **Y**, and **Z**, **X** and **Y** are **d-separated** given **Z** in G if and only if **X** is not d-connected to **Y** given **Z**. A probability distribution P satisfies the global directed Markov property for directed graph G if and only if for any three disjoint sets of variables **X**, **Y**, and **Z**, if **X** is d-separated from **Y** given **Z** in G, then **X** is independent of **Y** given **Z** in P.

The following theorems relate the global directed Markov property to factorizations of a density function. Denote a density function over **V** by $f(\mathbf{V})$, where for any subset **X** of **V**, $f(\mathbf{X})$ denotes the marginal density of $f(\mathbf{V})$. If $f(\mathbf{V})$ is the density function for a probability measure over a set of variables **V** and **An**(**X**) is the set of ancestors of members of **X** in directed graph G, say that $f(\mathbf{V})$ **factors according to directed graph** G with vertices **V** if and only if for every subset **X** of **V**,

$$f(\mathbf{An}(\mathbf{X})) = \prod_{\mathrm{V} \in \mathbf{An}(\mathbf{X})} g_{\mathrm{V}}(\mathrm{V}, \mathbf{Parents}(\mathrm{V}))$$

where each g_{V} is a non-negative function.

The following result was proved in Lauritzen *et al.* (1990). (A more precise description of the weak assumptions that need to be made about the underlying probability spaces and densitites is given in Lauritzen et al. 1990.)

Theorem 1: If **V** is a set of random variables with a probability measure P that has a density function $f(\mathbf{V})$, then $f(\mathbf{V})$ factors according to DAG G if and only if P satisfies the global directed Markov property for G.

---

[3] However, in Section 3 we show that the local and global directed Markov properties are not equivalent for cyclic directed graphs.

As in the case of acyclic graphs, the existence of a factorization according to a cyclic directed graph G does entail that a measure satisfies the global directed Markov property for G. The proof given in Lauritzen *et al.* (1990) for the acyclic case carries over essentially unchanged to the cyclic case. (Lauritzen *et al.* use a different definition of d-separation that is equivalent to Pearl's in both the cyclic and the acyclic case.)

Theorem 2: If **V** is a set of random variables with a probability measure P that has a density function $f(\mathbf{V})$ and $f(\mathbf{V})$ factors according to directed (cyclic or acyclic) graph G, then P satisfies the global directed Markov property for G.

However, unlike the case of acyclic graphs, if a probability measure over a set of variables **V** satisfies the global directed Markov property for cyclic graph G and has a density function $f(\mathbf{V})$, it does not follow that $f(\mathbf{V})$ factors according to G, even if $f(\mathbf{V})$ is positive. (We thank an anonymous referee for pointing this fact out.)

3. Non-recursive Linear Structural Equation Models

The problem considered in this section is to investigate the generalization of the Markov properties to linear, non-recursive structural equation models. First we must relate the social scientific terminology to graphical representations, and clarify the questions.

The variables in a structual equation model (SEM) can be divided into two sets, the "error" variables and the "substantive" variables. Corresponding to each substantive variable X_i is an equation expressing X_i as a *linear* function of the direct causes of X_i plus a unique error variable ε_i where the linear coefficient of each variable that is not an error variable is a *free* parameter. (We will not consider non-linear models in this paper. For a discussion of non-linear cyclic models see Spirtes 1995.) Since we have no interest in first moments, without loss of generality each variable can be expressed as a deviation from its mean.

Consider, for example, two SEMs S_1 and S_2 over $\mathbf{X} = \{X_1, X_2, X_3\}$, where in both SEMs X_1 is a direct cause of X_2 and X_2 is a direct cause of X_3. The structural equations[4] in Figure 1 are common to both S_1 and S_2.

[4] We realize that it is slightly unconventional to write the trivial equation for the exogenous variable X_1 in terms of its error, but this serves to give the error variables a unified and special status as providing all the exogenous sources of variation for the system.

$$X_1 = \varepsilon_1$$

$$X_2 = \beta_{21} X_1 + \varepsilon_2$$

$$X_3 = \beta_{32} X_2 + \varepsilon_3$$

Figure 1: Structural Equations for SEMs S_1 and S_2

where β_{21} and β_{32} are free parameters ranging over real values, and ε_1, ε_2 and ε_3 are error variables. In addition suppose that ε_1, ε_2 and ε_3 are distributed as multivariate normal. In S_1 we will assume that the correlation between each pair of distinct error variables is fixed at zero. The free parameters of S_1 are $\theta_1 = <\beta, \mathbf{P}>$, where β is the set of linear coefficients $\{\beta_{21}, \beta_{32}\}$ and **P** is the set of variances of the error variables. We will use $\Sigma_{S1}(\theta_1)$ to denote the covariance matrix parameterized by the vector θ_1 for model S_1, and occasionally leave out the model subscript if the context makes it clear which model is being referred to. If all the pairs of error variables in a SEM S are uncorrelated, we say S is a SEM with **uncorrelated errors**.

S_2 contains the same structural equations as S_1, but in S_2 we will allow the errors between X_2 and X_3 to be correlated, i.e., we make the correlation between the errors of X_2 and X_3 a free parameter, instead of fixing it at zero, as in S_1. In S_2 the free parameters are $\theta_2 = <\beta, \mathbf{P'}>$, where β is the set of linear coefficients $\{\beta_{21},\beta_{32}\}$ and **P'** is the set of variances of the error variables and the correlation between ε_2 and ε_3. If the correlations between any of the error variables in a SEM are not fixed at zero, we will call it a SEM with **correlated errors**.[5]

If the coefficients in the linear equations are such that the substantive variables are a unique linear function of the error variables alone, the set of equations is said to have a **reduced form**. A linear SEM with a reduced form also determines a joint distribution over the substantive variables. We will consider only linear SEMs which have coefficients for which there is a reduced form, all variances and partial variances among the substantive variables are finite and positive, and all partial correlations among the substantive variables are well defined.

It is possible to associate with each SEM with uncorrelated errors a directed graph that represents the causal structure of the model and the form of the linear equations. For example, the directed graph associated with the substantive variables in S_1 is $X_1 \rightarrow X_2 \rightarrow$

[5]We do not consider SEMs with other sorts of constraints on the parameters, e.g., equality constraints.

X_3, because X_1 is the only substantive variable that occurs on the right hand side of the equation for X_2, and X_2 is the only substantive variable that appears on the right hand side of the equation for X_3. We generally do not include error variables in the causal graph associated with a SEM unless the errors are correlated. When the distinction is relevant to the discussion, we enclose measured variables in boxes, latent variables in circles, and leave error variables unenclosed.

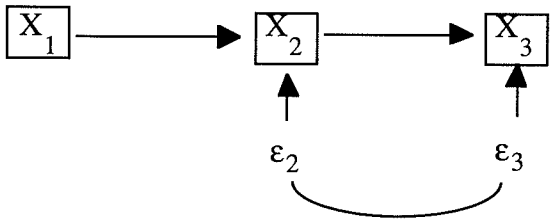


Figure 2. SEM S_2 with correlated errors

The typical path diagram that would be given for S_2 is shown in Figure 2. This is not strictly a directed graph because of the curved line between error variables ε_2 and ε_3, which indicates that ε_2 and ε_3 are correlated. It is generally accepted that correlation is to be explained by some form of causal connection. Accordingly if ε_2 and ε_3 are correlated we will assume that either ε_2 causes ε_3, ε_3 causes ε_2, some latent variable causes both ε_2 and ε_3, or some combination of these. In other words, curved lines are an ambiguous representation of a causal connection.

A SEM is said to be **recursive** (an RSEM) if its directed graph is acyclic; otherwise it is **non-recursive**.[6]

A SEM containing disjoint sets of variables **X**, **Y**, and **Z** **linearly entails** that **X** is independent of **Y** given **Z** if and only if **X** is independent of **Y** given **Z** for all values of free parameters in the SEM. A DG *G* containing disjoint sets of variables **X**, **Y**, and **Z** **linearly entails** that **X** is independent of **Y** given **Z** if and only if the SEM with DG *G* and no correlated errors linearly entails that **X** is independent of **Y** given **Z**. Similarly we may say that a SEM containing X, Y, and **Z**, where $X \neq Y$ and X and Yare not in **Z**, **linearly entails** that $\rho_{XY.\mathbf{Z}} = 0$, if and only if $\rho_{XY.\mathbf{Z}} = 0$ for all values of free parameters in the SEM (where $\rho_{XY.\mathbf{Z}}$ is the partial correlation of X and Y given **Z**.) DG *G* **linearly entails** that $\rho_{XY.\mathbf{Z}} = 0$ if and only if the SEM with DG *G* and no correlated errors linearly

[6] Note that this use of cyclic directed graphs to represent feedback processes represents an extension of the causal interpretation of directed graphs.

entails $\rho_{XY.Z} = 0$. It follows from Kiiveri and Speed (1982) that if the error variables are jointly independent, then any distribution that forms a linear, recursive SEM with a directed acyclic graph G satisfies the local directed Markov property for G. One can therefore apply d-separation to the DAG in a linear, recursive SEM to compute the conditional independencies and zero partial correlations it linearly entails. The d-separation relation provides a polynomial (in the number of vertices) time algorithm for deciding whether a given conditional independence relation or vanishing partial correlation is linearly entailed by a SEM with a given DAG.

Linear non-recursive structural equation models (linear SEMs) are commonly used in the econometrics literature to represent feedback processes that have reached equilibrium.[7] Corresponding to a set of non-recursive linear equations is a cyclic graph, as the following example from Whittaker (1990) illustrates.

$$X_1 = \varepsilon_{X1}$$

$$X_2 = \varepsilon_{X2}$$

$$X_3 = \beta_{31}X_1 + \beta_{34}X_4 + \varepsilon_{X3}$$

$$X_4 = \beta_{42}X_2 + \beta_{43}X_3 + \varepsilon_{X4}$$

$\varepsilon_{X1}, \varepsilon_{X2}, \varepsilon_{X3}, \varepsilon_{X4}$ are jointly independent and normally distributed

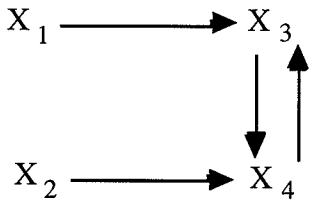


Figure 3: Example of Non-recursive SEM

Theorem 3 and Theorem 4 state that the set of conditional independence relations (and hence, zero partial correlations) linearly entailed by a SEM correspond to the d-separation relations in the associated directed graph, even in the case of cyclic graphs. (Theorem 3 was independently proved by J. Koster in Koster 1995.)

Theorem 3: The probability measure P over the substantive variables of a linear SEM L (recursive or non-recursive) with jointly independent error variables satisfies the global

[7]Cox and Wermuth (1993), Wermuth and Lauritzen(1990) and (indirectly) Frydenberg(1990) consider a class of linear models they call *block recursive*. The block recursive models overlap the class of SEMs, but they are neither properly included in that class, nor properly include it. Frydenberg (1990) presents necessary and sufficient conditions for the equivalence of two block recursive models. The graphs of SEMs without correlated errors are a subclass of the reciprocal graphs introduced in Koster (1995).

directed Markov property for the directed (cyclic or acyclic) graph G of L, i.e. if **X**, **Y**, and **Z** are disjoint sets of variables in G and **X** is d-separated from **Y** given **Z** in G, then **X** and **Y** are independent given **Z** in P.

Theorem 4: In a linear SEM L with jointly independent error variables and directed (cyclic or acyclic) graph G containing disjoint sets of variables **X**, **Y** and **Z**, if **X** is not d-separated from **Y** given **Z** in G then L does not linearly entail that **X** is independent of **Y** given **Z**.

Applying Theorem 3 and Theorem 4 to a linear SEM with the directed graph in Figure 3, the conditional independence relations linearly entailed are: X_1 is independent of X_2; X_1 is independent of X_2 given X_3 and X_4. It is easy to see from Theorem 3 and Theorem 4 that in a linear SEM L with jointly independent error variables and (cyclic or acyclic) directed graph G containing substantive variables X, Y and **Z**, where $X \neq Y$ and **Z** does not contain X or Y, X is d-separated from Y given **Z** in G if and only if L linearly entails that $\rho_{XY.Z} = 0$ (even if the error terms are not normally distributed).

As in the acyclic case, d-separation provides a polynomial time procedure for deciding whether a linear SEM with a cyclic graph linearly entails a conditional independence or vanishing partial correlation.

In DAGs the global directed Markov property entails the local directed Markov property, because a variable V is d-separated from its non-parental non-descendants given its parents. However, this is not always the case in cyclic graphs. For example, in Figure 3, X_4 is not d-separated from its non-parental non-descendant X_1 given its parents X_2 and X_3, so the local directed Markov property does not hold.[8]

There is also a way to decide which partial correlations are entailed to be zero by a SEM with correlated errors, such as S_2 (Figure 2). This is done by first creating a directed graph G with latent variables but no correlated errors, and then applying d-separation to G to determine if a zero partial correlation is entailed. The latent variable directed graph G (without correlated errors) that we will assocate with a SEM S with correlated errors is created in the following way. Start with the usual graphical representation of S, that contains undirected lines connecting correlated errors (e.g. SEM S_2 in Figure 2). For each

[8] We are indebted to C. Glymour for pointing out that the local Markov condition fails in Whittaker's model. Indeed, there is *no* acyclic graph (even with additional variables) that linearly entails all and only conditional independence relations linearly entailed by Figure 3, although the directed cyclic graph of Figure 3 is equivalent to one in which the edges from X_1 to X_3 and X_2 to X_4 are replaced, respectively, by edges from X_1 to X_4 and from X_2 to X_3.

pair of error variables ε_i and ε_j connected by an undirected edge, introduce a new latent variable T_{ij}, and edges from T_{ij} to X_i and X_j. Finally remove all of the error variables from the graph. When this process is applied to SEM S_2, the result is shown in Figure 4.

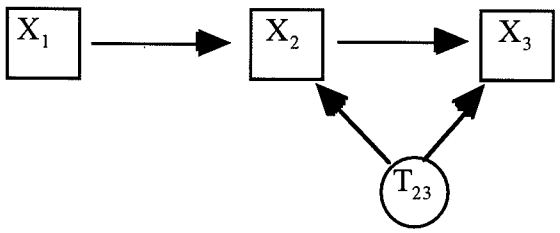


Figure 4. SEM S_2': Correlated Errors in S_2 Replaced by Latent Common Cause

In a SEM like S_2, with correlated errors, one can decide whether $\rho_{X1,X3.X2}$ is entailed to be zero by determining whether $\{X_1\}$ and $\{X_3\}$ are d-separated given $\{X_2\}$ in the directed graph in Figure 4. In this way the problem of determining whether a SEM with correlated errors entails a zero partial correlation is reduced to the already solved problem of determining whether a SEM without correlated errors entails a zero partial correlation. (In general if S is a SEM with correlated errors, and S' is the SEM with uncorrelated errors and the latent variable directed graph associated with S, it is *not* the case that for every instantiation θ_1 of the free parameters of S there is an instantiation θ_2 of the free parameters of S' such that $\Sigma_S(\theta_1) = \Sigma_{S'}(\theta_2)$. We are making the weaker claim that d-separation applied to *G* correctly describes which zero partial correlations are linearly entailed by S. See Spirtes et al. 1996a.)

4. The Discovery Problem

Suppose that we are given data sampled from a population whose causal structure is correctly described by some non-recursive structural equation model **M**. Is it possible to discover the causal graph of **M** from the data, or at least recover some features of the causal graph from the data? In Spirtes *et al.* (1995) the problem of discovering features of the causal graph is considered under the assumption that it is acyclic, but that there may be latent common causes (i.e. there may be unmeasured variables that are the direct cause of at least two measured variables.) Here we will consider the problem of discovering features of the causal graph under the assumption that it may be cyclic, but there are no

latent common causes. Future research is needed on the problem of discovering the causal graph when it may be cyclic *and* there may be latent common causes.

In order to make inferences about causal relations from a sample distribution it is necessary to introduce some axioms that link probability distributions to causal relations. The two assumptions that we will make are the Causal Independence and Causal Faithfulness Assumptions, described in the next two subsections.

4.1. The Causal Independence Assumption

The most fundamental assumption relating causality and probability that we will make is the following:

Causal Independence Assumption: If A does not cause B, and B does not cause A, and there is no third variable that causes both A and B, then A and B are independent.

This assumption allows us to draw a *causal* conclusion from *statistical* data and lies at the foundation of the theory of randomized experiments. If the value of A is randomized, the experimenter knows that the randomizing device is the sole cause of A. Hence the experimenter knows B did not cause A, and that there is no third variable which causes both A and B. This leaves only two alternatives: either A causes B or A and B are independent. If A and B are dependent in the experimental population, the experimenter concludes that A does cause B, which is an application of the Causal Independence Assumption.

The Causal Independence Assumption entails that if two error variables, such as ε_2 and ε_3 in Figure 2 are correlated there is a latent common cause of X_2 and X_3 responsible for the correlation. In other words, when X_2 and X_3 have correlated errors, we assume that the distribution over X_2 and X_3 is the marginal of some other distribution including a finite number of latent causes of X_2 and X_3 in which the error variables are uncorrelated. Since we are making the assumption that there are no latent common causes, it follows that the error variables of the causal graph are uncorrelated. The correctness of the d-separation criterion for deciding which partial correlations are linearly entailed to be zero by a SEM with an associated directed graph *G* then follows from Theorem 3 and Theorem 4.

4.2. The Faithfulness Assumption

In addition to the zero partial correlations that are entailed for *all* values of the free parameters of a SEM with a given directed graph, there may be zero partial correlations that hold only for some *particular* assignments of values to the parameters. For example,

suppose Figure 5 is the directed graph of a SEM that describes the relations among Tax Rate, the Economy, and Tax Revenues, where β_1, β_2 and β_3 are free parameters.

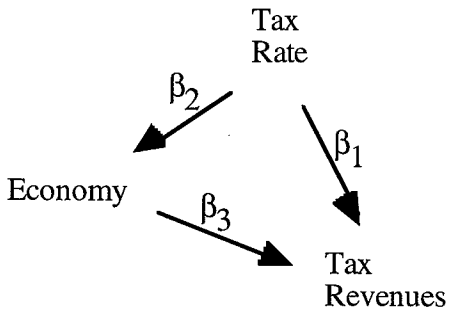


Figure 5. Economic Model

In this case there are no vanishing partial correlation constraints entailed for all values of the free parameters. But if in the population $\beta_1 = -(\beta_2 \times \beta_3)$, then Tax Rate and Tax Revenues are uncorrelated. The SEM postulates a direct effect of Tax Rate on Revenue (β_1), and an indirect effect through the Economy ($\beta_2 \times \beta_3$). The parameter constraint indicates that these effects *exactly* offset each other in the population, leaving no total effect whatsoever. In such a case we say that the distribution is **unfaithful** to the directed graph of the causal structure that generated it. A distribution is **faithful** to a directed graph *G* if each vanishing partial correlation in the distribution is linearly entailed by *G* (i.e. entailed for all values of the free parameters of the SEM with directed graph *G* and no correlated errors).

Causal Faithfulness Assumption: If the directed graph associated with a SEM M correctly describes the causal structure in the population, and θ_{pop} are the population parameter values, then if $\rho_{XZ.Y} = 0$ in $\Sigma_M(\theta_{pop})$, M linearly entails that $\rho_{XZ.Y} = 0$.

The faithfulness assumption limits the SEMs considered to those in which population constraints are entailed by structure, not by particular values of the parameters. If one assumes faithfulness, then if A and B are *not* d-separated given **C**, then $\rho_{A,B.C} \neq 0$, (because it is not linearly entailed to equal zero.) Faithfulness should not be assumed when there are deterministic relationships among variables, or equality constraints upon free parameters, since either of these can lead to violations of the assumption. Some form of the assumption of faithfulness is used in every science, and amounts to no more than the belief that an improbable and unstable cancellation of parameters does not hide real causal influences. When a theory cannot explain an empirical regularity save by invoking

a special parameterization, most scientists are uneasy with the theory and look for an alternative.

It is also possible to give a personalist Bayesian argument for assuming faithfulness. For any directed graph, the set of linear parameterizations of the directed graph that lead to violations of linear faithfulness are Lebesgue measure zero. Hence any Bayesian whose prior over the parameters is absolutely continuous with Lebesgue measure, assigns a zero prior probability to violations of faithfulness. Of course, this argument is not relevant to those Bayesians who place a prior over the parameters that is not absolutely continuous with Lebesgue measure and assigns a non-zero probability to violations of faithfulness.

The assumption of faithfulness guarantees the asymptotic correctness of the Cyclic Causal Discovery (CCD) algorithm described in Section 4.4. It does *not* guarantee that on samples of finite size this algorithm is reliable.

Given the Causal Independence Assumption, an assumption of no latent variables, a linearity assumption, and the Causal Faithfulness assumption, it follows that in a distribution P generated by a causal structure represented by a directed graph G, $\rho_{XY.Z} = 0$ if and only if X is d-separated from Y given Z in G. So if we can perform statistical tests of zero partial correlations then we can use that information to draw conclusions about the d-separation relations in G, and then to reconstruct as much information about G as possible. Henceforth we will speak of reconstructing features of G from d-separation relations, and from zero partial correlation interchangeably, since given our assumptions, these are equivalent. We assume that the discovery algorithm has access to a **d-separation oracle** that correctly answers questions about d-separation relations in G. In practice, of course, the oracle is some kind of statistical test of the hypothesis that a particular partial correlation is zero in a population that satisfies the global Markov and faithfulness properties for G. (The algorithm is correct for any distribution for which a d-separation oracle is available, but because in the case where the functional relations between variables are non-linear, non-recursive d-separation is not a sufficient condition for conditional independence, the only case we know of in which such an oracle is available is the linear case.)

Of course the number of distinct d-separation relations grows exponentially with the number of variables in the directed graph. Therefore it is important to discover the features of G from a subset of the set of all d-separation relations. The CCD algorithm that we describe below chooses the subset of d-separation relations that it needs to reconstruct features of G as it goes along.

4.3. Output Representation – Partial Ancestral Graphs (PAGs)

In general, it is not possible to reconstruct a unique directed graph G given information only about its d-separation relations, because there may be more than one directed graph in which exactly the same set of d-separation relations hold. Two directed graphs G, G^* are said to be **d-separation equivalent** if the same set of d-separation relations holds in both directed graphs. The set of directed graphs d-separation equivalent to a given directed graph G is denoted by **Equiv**(G). (Note that there is a stronger sense of equivalence, which we will call linear statistical equivalence between two directed graphs G and G' which holds when for every instantiation θ_1 of the free parameters of SEM S with directed graph G and no correlated errors, there is an instantiation θ_2 of the free parameters of SEM S' with directed graph G' and no correlated errors, such that $\Sigma_S(\theta_1) = \Sigma_{S'}(\theta_2)$, and vice versa. In the acyclic case it is known that d-separation equivalence implies linear statistical equivalence, but it is known that this is not so for cyclic graphs.)

The members of **Equiv**(G) always have certain features in common. We now introduce the formalism with which we will represent features common to all directed graphs in **Equiv**(G) for some fixed G. A partial ancestral graph (PAG) is an extended graph consisting of a set of vertices **V**, and a set of edges between vertices, where there may be the following kinds of edges: A $\leftrightarrow$ B, A o—o B, A — B, A o$\rightarrow$ B, A $\leftarrow$o B, A $\rightarrow$ B, A $\leftarrow$ B, A o— B, and A —o B (The A $\leftrightarrow$ B, A $\leftarrow$o B, and A o$\rightarrow$ B edges appear only in PAGs for directed graphs with latent variables. Because in this paper we are considering only directed graphs without latent variables, none of these types of edges occur in the PAGs we consider here.) We say that the A endpoint of an A $\rightarrow$ B, A — B, or A —o B edge is "–"; the A endpoint of an A $\leftrightarrow$ B, A $\leftarrow$o B, or. A $\leftarrow$ B edge is "<"; and we say the A endpoint of an A o—o B, A o$\rightarrow$ B, or A o— B edge is "o". The conventions for the B endpoints are analogous. In addition pairs of edge endpoints may be connected by underlining, or dotted underlining (illusttated below). A partial ancestral graph for G contains partial information about the ancestor relations in G, namely only those ancestor relations common to all members of **Equiv**(G). In the following definition, which provides a semantics for PAGs we use "*" as a meta-symbol indicating the presence of any one of {o, –, >}, e.g. A—* B represents any of the following edges: A — B, A $\rightarrow$ B, or A —o B.

Partial Ancestral Graphs (PAGs)[9]

Ψ is a PAG for directed graph G with vertex set **V**, if and only if

(i) There is an edge between A and B in Ψ if and only if A and B are d-connected in G given any subset $\mathbf{W} \subseteq \mathbf{V}\backslash\{A,B\}$.

(ii) If there is an edge in Ψ out of A (not necessarily into B), i.e. A—* B, then A is an ancestor of B in every directed graph in **Equiv**(G).

(iii) If there is an edge in Ψ into B, i.e. A*–>B, then in every directed graph in **Equiv**(G), B is **not** an ancestor of A.

(iv) If there is an underlining A*—*B*—*C in Ψ then B is an ancestor of (at least one of) A or C in every directed graph in **Equiv**(G).

(v) If there is an edge from A into B, and from C into B, (A—> B<— C), then the arrow heads at B are joined by dotted underlining (A—>B<—C) only if in every directed graph in **Equiv**(G) B is not a descendant of a common child of A and C.

(vi) Any edge endpoint not marked in one of the above ways is left with a small circle thus: o—*.

Two vertices, X and Y, in a directed cyclic graph G are **p-adjacent** if there is an edge between them, X*—*Y, in any (hence every) PAG for G. It follows directly from the definitions that a pair of vertices X, Y are p-adjacent in G if and only if X and Y are d-connected given every subset of the other vertices in$G$.

Observe that condition (i) in the definition of the PAG differs from the other five conditions in providing necessary *and* sufficient conditions on **Equiv**(G) for a given symbol, in this case an edge, to appear in a PAG. The other five conditions merely state necessary conditions. For this reason there are in fact many different PAGs for a directed graph G. Although they all have the same p-adjacencies, the edges may be of different types. Some of the PAGs provide more information than others about causal structure, e.g. they have fewer 'o's at the end of edges.[10]

---

[9] The extended graphs which we introduce here - Partial Ancestral Graphs - use a superset of the set of symbols used by Partially Oriented Inducing Path Graphs (POIPGs) descaribed in Spirtes *et al.* (1993) but the *graphical* interpretation of the orientations given to edges is different. However, it has been shown in Sprites *et al.* (1996) that a POIPG can be interpreted directly as a PAG. A direct corollary is that PAGs can be used to represent the d-separation equivalence class for directed *acyclic* graphs with *latent* variables. It is an open question whether or not the set of symbols is sufficiently rich to allow us to represent d-separation classes of cyclic graphs with latent variables.

[10] If one PAG for a graph G has a '>' at the end of an edge, then every other PAG for the same graph either has a '>' or a 'o' in that location. Similarly if one PAG for a graph G has a '–' at the end of an edge then every other PAG for the same graph either has a '–' or an 'o' in that location.

If Ψ is a PAG for directed graph G, we also say that Ψ **represents** G. Since every clause in the definition refers only to **Equiv**(G), it follows that if Ψ is a PAG for directed graph G, and $G^* \in$ **Equiv**(G), then Ψ is also a PAG for G^*. This is not surprising since, as the output of the discovery algorithm we present below, the PAG is designed to represent features common to all directed graphs in the d-separation equivalence class. However, some PAGs may represent directed graphs from different d-separation equivalence classes. This leaves open the possibility that an algorithm might output the same PAG given directed graphs from different d-separation classes as input. However, any PAG output by the discovery algorithm we present provides sufficient information to ensure that the algorithm never outputs the same PAG given oracles for two directed graphs unless those directed graphs are d-separation equivalent. Hence the algorithm provides a 1-1 mapping from d-separation equivalence classes into PAGs.

The set of features described by a PAG is rich enough to enable us to distinguish between any two d-separation equivalence classes, i.e. there is some set of features common to all directed graphs in one d-separation equivalence class that is not true of all directed graphs in another d-separation equivalence class, and this difference can be expressed by a difference in the PAGs representing those d-separation equivalence classes.

Example:
Suppose G is as follows:

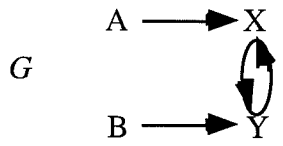


Figure 6

In this case it can be shown that **Equiv**(G) contains (only) two directed graphs:

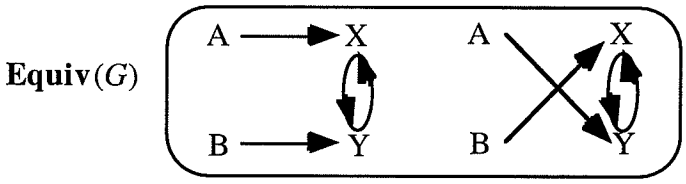


Figure 7

The PAG which the CCD algorithm outputs given as input an oracle for deciding conditional independence facts in G, is:

PAG for **Equiv**(G)

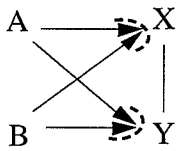


Figure 8

Observe that the PAG tells us the following facts about **Equiv**(G):[11]

(a) X is an ancestor of Y, and Y is an ancestor of X in every directed graph in **Equiv**(G).

(b) In no directed graph in **Equiv**(G) is X or Y an ancestor of A or B.

(c) In every directed graph in **Equiv**(G) both A and B are ancestors of X and Y.

Note that not every edge in the PAG appears in every directed graph in **Equiv**(G). This is because an edge in the PAG indicates only that the two variables connected by the edge are d-connected given any subset of the other variables. In fact it is possible to show something stronger, namely that if there is an edge between two vertices in a PAG, then there is some directed graph represented by the PAG in which that edge is present.[12]

This example is atypical in that the PAG given by the algorithm contains no 'o' endpoints; however it shows how much information a PAG may provide. Notice that the following are also PAGs for G though they are less informative.

Other PAGs for **Equiv**(G)

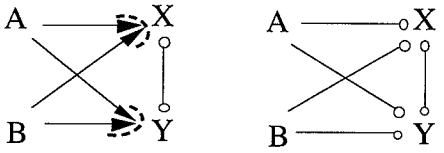


Figure 9

The CCD algorithm we describe does not always give the most informative PAG for a given directed graph G in that there may be features common to all directed graphs in the d-separation equivalence class which are not captured by the PAG that the algorithm outputs. In this sense the algorithm is not complete. However, the algorithm is **d-separation complete** in the sense that if the d-separation oracles for two different directed graphs cause the algorithm to produce the same PAG as output then the two directed graphs are d-separation equivalent.

---

[11]This is not an exhaustive list. For example, the presence of the dotted line connecting the arrowheads on the A —>X, and B—>X edges, tells us that in no graph in Equiv(G) are both of these edges present. Likewise with the dotted line connecting the arrowheads of the B—>Y, and A—>Y edges.

[12]See footnote 10.

The following definition is required to state the algorithm. For graph Ψ, **Adjacencies**(Ψ,X) is a function giving the set of variables Y s.t. there is an edge X*—*Y in Ψ. [13] Ψ is a dynamic object in the algorithm that changes as the algorithm progresses, and hence **Adjacencies**(Ψ,X) also changes as the algorithm progresses. A trace of the algorithm on a simple example is given in section 4.8.

4.4. The Cyclic Causal Discovery (CCD) Algorithm

The overall strategy for discovery is shown in Figure 10.

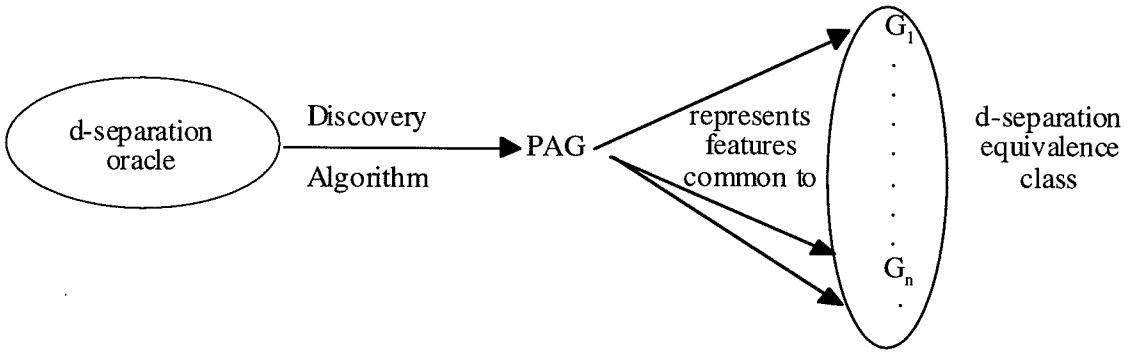


Figure 10

Note that once the following algorithm adds a "—", ">", or "<" endpoint to an edge it never removes or changes it; similarly once it adds underlining (dotted or not) it never removes it or changes it. For each pair of variables X and Y the set **Sepset**(X,Y) is assigned a values at most once by the algorithm. For some pairs of variables the algorithm does not assign a value to **Sepset**(X,Y), but in those cases, the values are not needed by the algorithm. Similar remarks hold for **Supsetset**(X,Y,Z). The algorithm correctly creates PAGs for acyclic as well as cyclic graphs.

CCD Algorithm

Input: An oracle for answering questions of the form: "Is X d-separated from Y given set **Z**, (X,Y∉ **Z**) in directed graph *G*?"

Output: A PAG for *G*.

[13]Here as elsewhere '*' as a meta-symbol indicating any of the three ends -, o, >.

¶**A** a) Form the complete graph Ψ, such that between every pair of variables A and B there is an edge Ao—oB in Ψ.

b) n = 0.

repeat

repeat

select an ordered pair of variables X and Y such that there is an edge Xo—oY in Ψ and the number of vertices in **Adjacencies**(Ψ,X)\{Y} is greater than or equal to n;

repeat

select a subset **S** of **Adjacencies**(Ψ,X)\{Y} with n vertices;

if X and Y are d-separated given **S** delete edge Xo—oY from Ψ and set **Sepset**(X,Y) = **S** and **Sepset**(Y,X) = **S**;

until every subset **S** of **Adjacencies**(Ψ,X)\{Y} with n vertices has been selected or some subset **S** has been found for which X and Y are d-separated given **S**;

until all ordered pairs of p-adjacent vertices X and Y such that **Adjacencies**(Ψ,X)\{Y} has greater than or equal to n vertices have been selected;

n = n + 1;

until for each ordered pair of p-adjacent vertices X, Y, **Adjacencies**(Ψ,X)\{Y} has less than n vertices.

¶**B.** For each triple of vertices A,B,C such that the pair A,B and the pair B,C are each p-adjacent in Ψ but the pair A, C are not p-adjacent in Ψ, then:

(i) orient A*—*B*—*C as A—>B<—C if and only if B $\notin$ **Sepset**<A,C>;

(ii) orient A*—*B*—*C as A*—*B*—*C if and only if B $\in$ **Sepset**<A,C>.

¶**C.** For each triple of vertices <A,X,Y> in Ψ such that

(a) A is not p-adjacent to X or Y in Ψ,

(b) X and Y are p-adjacent in Ψ,

(c) X $\notin$ **Sepset**<A,Y>

if A and X are d-connected given **Sepset**<A,Y> then orient X o—o Y or X o— Yas X<—Y

¶**D.** For each vertex V in Ψ form the following set: X$\in$ **Local**(Ψ,V) if and only if X is p-adjacent to V in Ψ, or there is some vertex Y such that X—>Y<—V in Ψ. (Local(Ψ,V) is calculated once for each vertex V and does not change as the algorithm progresses.)

m = 1.

repeat

repeat

select a pair of variables {A,C} and a third variable B such that A and C are not p-adjacent, A—> B<—C, and **Local**(Ψ,A)\(**Sepset**<A,C> $\cup$ {B,C}) has greater than or equal to m vertices.

repeat

select a set **T** $\subseteq$ **Local**(Ψ,A)\(**Sepset**<A,C> $\cup$ {B,C}) with m vertices; if A and C are d-separated given **T** $\cup$ **Sepset**<A,C> $\cup$ {B} then orient the triple A—>B<—C as A—>B<—C, and record **T** $\cup$ **Sepset**<A,C> $\cup$ {B} in **SupSepset**<A,B,C> and **SupSepset**<C,B,A>.

until every subset **T** $\subseteq$ **Local**(Ψ,A)\(**Sepset**<A,C> $\cup$ {B,C}) with m vertices has been selected or a d-separating set for A and C has been recorded in **SupSepset**<A,B,C> and **SupSepset**<C,B,A>.

until all triples such that A—>B<—C, (i.e. not A—>B<—C), A and C are not p-adjacent, and **Local**(Ψ,A)\(**Sepset**<A,C> $\cup$ {B,C}) have greater than or equal to m vertices have been selected.

m = m +1.

until each ordered triple <A,B,C> such that A—>B<—C but A and C are not p-adjacent, is such that **Local**(Ψ,A)\(**Sepset**<A,C> $\cup$ {B,C}) has fewer than m vertices.

¶**E.** If there is a quadruple <A,B,C,D> of distinct vertices such that

(i) A—>B<—C in Ψ

(ii) A—>D<—C or A—>D<—C in Ψ

(iii) B and D are p-adjacent in Ψ

then orient B o—o D or B —o D as B—>D in Ψ if D is not in **SupSepset**<A,B,C>

else orient B*—oD as B*—D in Ψ if D is in **SupSepset**<A,B,C>.

¶**F.** For each quadruple <A,B,C,D> of distinct vertices such that

(i) A—>B<—C in Ψ

(ii) D is not p-adjacent to both A and C in Ψ

(iii) B and D are p-adjacent in Ψ

if A and C are a pair of vertices d-connected given **SupSepset**<A,B,C> $\cup$ {D}, then orient the edge B o—o D or B —o D as B—>D in Ψ.

Notes concerning the operation of the CCD Algorithm:

(¶**A**) The search in ¶A looks for d-separating sets for pairs of vertices X, Y in *G*. If such a set is found then it is recorded in **Sepset**(X,Y), and the edge between X and Y in Ψ is deleted. It can be shown (see proof of Theorem 5) that if X and Y are not p-adjacent in *G*, then ¶A is guaranteed to find a set which d-separates X and Y. Consequently at the end of ¶A there is an edge between a pair of vertices V and W in Ψ if and only if V and W are p-adjacent in *G*. Since, further, all edges in Ψ at this point take the form o–o, at this point Ψ is a PAG for *G*, though not a very informative one.

¶A always tests every subset of a given set before testing that set itself. It can be shown (see Lemma 6, Corollary 2) that as a consequence every vertex in **Sepset**(X,Y) is an ancestor of either X or Y in every directed graph in **Equiv**(*G*). Note that **Sepset**(X,Y) is set at most once: the algorithm removes the edge between X and Y in Ψ, as soon as a d-separating set for X and Y is found, and only attempts to find such a d-separating set if there is still an edge between X and Y in Ψ.

(¶**B**) In section ¶B each triple of vertices <A,B,C> in Ψ, such that there is an edge between A and B, and B and C, but there is no edge between A and C is either oriented as A—>B<—C or as A*—*B*—*C. The orientation rule makes use of the property (mentioned above) that every vertex in **Sepset**(A,C) is an ancestor of A or C. The rule also uses the fact that if A and B, and B and C are p-adjacent, but A and C are not p-adjacent, and B is an ancestor of A or C then B occurs in every set which d-separates A and C (See Lemma 7). Note that the premise in ¶B that there is no edge between A and C in Ψ ensures that **Sepset**(A,C) exists and has been set in ¶A. The proof of correctness for the algorithm implicitly shows that this rule can never lead to contradictory conclusions (e.g. a graph containing A—>B<—C) as long as the d-separation oracle gives correct answers about d-separation in directed graph *G*.)

(¶**C**) Section ¶C performs additional orientations in Ψ. The rule applies to certain triples of vertices <A,X,Y>, where X and Y are p-adjacent, but A is not p-adjacent to X or Y. The rule infers from the existence of a d-connecting path from A to X given **Sepset**(A,Y), (X∉ **Sepset**(A,Y)) and the absence of a d-connecting path from A to Y given **Sepset**(A,Y), that X is not an ancestor of Y. The inference is based on the idea that if X were an ancestor of Y then the d-connecting path from A to X could be 'extended' to a d-connecting path between A and Y, given **Sepset**(A,Y). Note again that the condition that there is no edge between A and Y ensures that **Sepset**(A,Y) has been set in ¶A.

(¶D) In section ¶D, the algorithm considers each triple <A,B,C> which is then oriented as A—>B<—C in Ψ, and attempts to find a set **Z** which d-separates A and C, but contains {B}$\cup$ **Sepset**(A,C). If such a set is found then it is recorded in **Supsepset**<A,B,C>, and dotted underlining is added, linking the arrowheads at B thus: A—>B<—C.

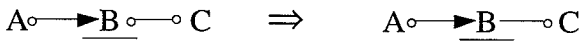
Since ¶D looks for the smallest superset of {B}$\cup$ **Sepset**(A,C), it can be proved (see Lemma 6) that every vertex in **Supsepset**<A,B,C> is an ancestor of A, B or C in every directed graph in **Equiv**(*G*). (This makes use of the analogous property, mentioned above, that **Sepset**(A,C) $\subset$ **An**({A,C}) in every directed graph in **Equiv**(*G*).)

Note that ¶D looks for **Supsepset**<A,B,C> only if A—>B<—C in Ψ, A and C are not p-adjacent, and there is no underlining at B. Since underlining is added at B if a set which satisfies the conditions on **Supsepset**<A,B,C> is found, it follows that **Supsepset**<A,B,C> is set at most once by the algorithm.

(¶E & ¶F) These last two sections make additional inferences concerning ancestor relations by examining **Supsepset**<A,B,C>. Both rules make use of the fact that **Supsepset**<A,B,C> $\subset$ **An**({A,B,C}) as mentioned above. Note that antecedent (i) in ¶E and ¶F ensures that **Supsepset**<A,B,C> exists and has been set by ¶D of the algorithm.

4.5. Propagation Rules

There are many inferences that are validated by the semantics of a PAG, without referring to the d-separation oracle. For example the following inference rule:



The underlining at B asserts that B is an ancestor of A or C, while the arrowhead at B on the A—>B edge asserts that B is not an ancestor of A, hence B is an ancestor of C. We shall call such inferences *propagation* rules, since they 'propagate' information that is already present in the PAG. The CCD algorithm as it stands includes almost no such propagation rules.[14] The development of a complete set of such propagation rules is an area for future research.

It will follow from the completeness theorem (Theorem 7) that all of the structural information about the directed graph that can ever be obtained from the oracle can be

[14] In certain special instances rules ¶C, ¶E and ¶F may redundantly consult the d-separation oracle, in the sense that the answer to the query could be inferred from orientations that are already present in the PAG. In such cases these rules behave as propagation rules. (We have not removed these redundant tests because, so far as we can see, this would involve a substantial increase in computational complexity.)

obtained by applying propagation rules (which do not require further oracle consultation) to the output of the CCD algorithm. If any of the steps of the algorithm were omitted, this would no longer be the case, i.e. in certain cases further consultation of the oracle would be needed in order to find the most informative PAG.

4.6. Soundness

Theorem 5: (Soundness) Given as input an oracle for d-separation relations in the (cyclic or acyclic) directed graph G, the output of the CCD algorithm is a PAG Ψ for G.

Theorem 5 is proved by showing that each section of the algorithm makes correct inferences about the structure of G from the answers of the d-separation oracle for G.

In practice, an approximation to a d-separation oracle can be implemented as a statistical test that the corresponding partial correlation vanishes. As the sample size increases without limit, if the significance level of the statistical test is systematically lowered, then the probabilities of both Type I and Type II error for the test approach zero, so that the statistical test is correct with probability one. Of course, this does not guarantee that the CCD algorithm as implemented is reliable on realistic sample sizes. The reliability of the algorithm depends upon the following factors:

1. Whether the Causal Independence Assumption holds (i.e. there are no latent variables).
2. Whether the Causal Faithfulness Assumption holds.
3. Whether the distributional assumptions made by the statistical tests hold.
4. The power of the statistical tests against alternatives.
5. The significance level used in the statistical tests.

In the future, we will test the sensitivity of the algorithm to these factors on simulated data.

4.7. Completeness

The statement of the algorithm in §4.4 does not specify completely an order in which sets are to be tested in ¶A and ¶D: it is only required that no set may be tested until all of the sets of smaller cardinality have been tried, but the order in which sets of the same cardinality are to be tested is unspecified.

If such an order is specified, and the process of selecting sets is deterministic then it follows that given oracles for two d-separation equivalent directed graphs G_1 and G_2, the algorithm will generate the same PAG. This is because, relative to a fixed order of

selecting subsets, the output is determined entirely by the responses of the oracle, and for d-separation equivalent directed graphs the oracle, by definition, will give the same responses.

However, this leaves open the possibility that different orderings of the oracle consultations might generate different PAGs, given the same directed graph (or d-separation equivalent directed graphs) as input. In fact this may occur in certain circumstances: selecting sets in a different order may result in a different PAG as output. It can in fact be shown that the operation of sections ¶A, ¶B, ¶D and ¶E will be unaffected by the order in which subsets of the same cardinality are selected. However, ¶C and ¶F may orient more edges under some orderings than others.

In spite of this it is still the case that if, given oracles for two directed graphs the CCD algorithm produces the same PAG as output then the directed graphs are d-separation equivalent. This remains true even if the PAGs were generated by different implementations of the algorithm, which selected subsets differently:[15]

Theorem 7 (d-separation Completeness) If the CCD algorithm, when given as input d-separation oracles for the directed graphs G_1, G_2 produces as output PAGs Ψ_1, Ψ_2 respectively, then Ψ_1 is identical to Ψ_2 only if G_1 and G_2 are d-separation equivalent, i.e. $G_2 \in \mathbf{Equiv}(G_1)$ and vice versa.

The proof is based on the characterization of d-separation equivalence in Richardson (1994b).

As argued above, relative to a fixed, deterministic method for selecting subsets, the converse to Theorem 7 also holds: oracles for d-separation equivalent directed graphs will produce the same PAG as output from the algorithm. Hence the CCD algorithm, together with a fixed method of selecting sets, will produce the same PAG as output if and only if given oracles for d-separation equivalent directed graphs as input.

4.8. Trace of CCD Algorithm

The following illustrates the operation of the algorithm given as input a d-separation oracle for the following directed graph:

---

[15] This is not in conflict with the statement that different implementations may produce different PAGs. If Ψ_1 and Ψ_2 are different PAGs for the same graph resulting from different implementations, then any edge endpoint oriented with a '–' or a '>' in Ψ_1 but with a 'o' in Ψ_2 could also be oriented in Ψ_2 by applying a propagation rule (see §4.5) to Ψ_2 (and vice versa).

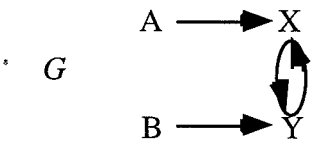


Figure 11

Initial Graph Ψ:

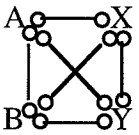


Figure 12

Section ¶A:

Since A and B are d-separated given the empty set, the algorithm removes the edge between A and B and records **Sepset**<A,B> = **Sepset**<B,A> = ∅. This is the only pair of vertices that are d-separated given a subset of the other variables. Hence after ¶A Ψ, which is now a PAG for *G*, is as follows:

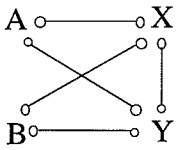


Figure 13

Section ¶B

Since X∉ **Sepset**<A,B> and Y∉ **Sepset**<A,B>, Ao–oXo–oB and Ao–oYo–oB are oriented respectively as A–>X<–B and A–>Y<–B. The state of Ψ at the end of ¶B is shown in Figure 14.

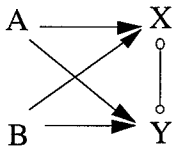


Figure 14

Section ¶C No orientations are performed in this case.

Section ¶D

Since A and B are d-separated given {X,Y}, the algorithm records **SupSepset**<A,X,B> = **SupSepset**<A,Y,B> = {X,Y}, and it orients A–>X<–B as A–>X<–B, and A–>Y<–B as A–>Y<–B. The state of PAG Ψ after ¶D is shown in Figure 15.

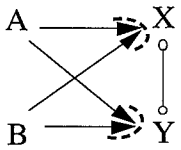


Figure 15

Section ¶E

The quadruple <A,B,X,Y> is such that (i) A–>X<–B, (ii) A–>Y<–B, (iii) X and Y are p-adjacent, thus it satisfies the conditions in section ¶E. Since Y∈ **SupSepset**<A,X,B>, the edge Yo–o X is oriented as Y—oX. Since X∈ **SupSepset**<A,Y,B>, this edge is further oriented as Y—X.

Section ¶F No orientations are performed in this case, hence the PAG that is output is:

PAG Ψ

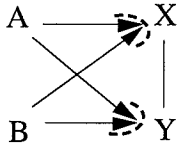


Figure 16

4.9. Complexity of CCD Algorithm

Let $\text{MaxDegree}(G) = \max_{Y \in \mathbf{V}} \left|\{X \mid Y \leftarrow X, \text{ or } X \leftarrow Y \text{ in } G\}\right|$,

and $\text{MaxAdj}(G) = \max_{Y \in \mathbf{V}} \left|\{X \mid X \text{ is p-adjacent to } Y \text{ in any PAG for } G\}\right|$

The number of d-separation tests performed by Step ¶A of the CCD algorithm will, in a worst case, be bounded as follows:

$$\begin{matrix}\text{Total number of} \\ \text{oracle consultations in ¶A}\end{matrix} \leq 2 \cdot \binom{n}{2} \sum_{i=0}^{k} \binom{n-2}{i} \leq \frac{(k+1)n^2(n-2)^{k+1}}{k!}.$$

where n = number of vertices in G, and k = MaxAdj(G). Since MaxAdj(G) ≤ (MaxDegree(G))^2, with MaxDegree(G) = r this step is $O(n^{r^2+3})$. It should be stressed that even as a worst case complexity bound this is a very loose one; the bound presumes that there is a directed graph in which for every pair of vertices (X,Y), not p-adjacent in the directed graph, X and Y are only d-separated given all vertices adjacent to X or all vertices adjacent to Y.

Step ¶B performs no additional tests of d-separation.

Step ¶C performs at most one d-separation test for each triple satisfying the conditions given. Thus this step is $O(n^3)$.

In a worst case the number of tests of d-separation that Step ¶D performs is bounded by

$$\text{Total number of oracle consultations in } \P D \leq \binom{n}{3}\sum_{i=0}^{m}\binom{n-3}{i} \leq \frac{(m+1)n^3(n-3)^{m+1}}{m!}$$

where $m = \underset{Y \in V}{Max}|\{X \mid X \in \text{Local}(\Psi,Y)\}$ in ¶D. Since $m \leq (\text{MaxDegree}(G))^2$, this step is $O(n^{r^2+4})$. Again this is a loose bound.

Step ¶E performs no tests of d-separation, while step ¶F performs at most one test for each quadruple satisfying the conditions. Hence this step is $O(n^4)$, (though in many directed graphs there may be very few quadruples satisfying all four conditions).

Thus overall the algorithm is of complexity $O(n^{r^2+4})$.

5. Conclusion

These results raise a number of interesting questions whose answers may be of practical importance. Are there other parameterizations of directed cyclic graphs which entail the global Markov condition? Richardson (1995) gives a polynomial time algorithm for deciding whether two directed cyclic graphs are d-separation equivalent, based on the characterization of d-separation equivalence given in Theorem 6. Spirtes and Verma (1992) gives a polynomial time algorithm for deciding whether two directed acyclic graphs with latent variables are d-separation equivalent over the subset of measured variables. Is there a polynomial algorithm for determining when two arbitrary directed graphs (cyclic or acyclic) have the same set of d-separation relations over a common subset of variables **O**? As we have seen there are correct, polynomial time algorithms for inferring features of sparse directed graphs (cyclic or acyclic) from a probability distribution when there are no latent common causes. There are similarly correct, but not polynomial time, algorithms for inferring features of directed acyclic graphs from a probability distribution even when there may be latent common causes (see Spirtes, 1992, Spirtes, Glymour and Scheines, 1993, and Spirtes, Meek, and Richardson 1995, Pearl and Verma 1991). Are there comparable algorithms for inferring features of directed graphs (cyclic or acyclic) from a probability distribution even when there may be latent common causes?

6. Proofs

6.1. Proof of Theorem 3

Some of the proofs are simplified by using a graphical relation (which we will call "Lauritzen d-separation") shown in Lauritzen *et al.* (1990) to be equivalent to Pearl's d-separation relation defined in Section 2. Several preliminary definitions are needed to define Lauritzen d-separation. An **undirected graph** is an ordered pair of a finite set of vertices **V**, and a set of undirected edges **E**. An undirected edge between A and B is an unordered pair of distinct vertices {A,B} in **V**. A sequence of edges $<E_1,...,E_n>$ in an undirected graph H is an **undirected path** if and only if there exists a sequence of vertices $<V_1,...,V_{n+1}>$ such that for $1 \leq i \leq n$ $\{V_i,V_{i+1}\} = E_i$ and $E_i \neq E_{i+1}$. Let $G(\mathbf{X})$ be the 'induced' directed subgraph of directed graph G that contains only vertices in **X**, with an edge from A to B in $G(\mathbf{X})$ if and only if there is an edge from A to B in G. Moral(G) **moralizes** a directed graph G if and only if Moral(G) is an undirected graph with the same vertices as G, and a pair of vertices X and Y are adjacent in Moral(G) if and only if either X and Y are adjacent in G, or they have a common child in G. In an undirected graph H, if **X**, **Y**, and **Z** are disjoint sets of vertices, then **X** is **separated** from **Y** given **Z** if and only if every undirected path between a member of **X** and a member of **Y** contains a member of **Z**. If **X**, **Y** and **Z** are disjoint sets of variables, **X** and **Y** are **Lauritzen d-separated** given **Z** in a directed graph G just when **X** and **Y** are separated given **Z** in Moral(G(**An**(**X** $\cup$ **Y** $\cup$ **Z**))).

Since some of the vertices in the proofs are defined as satisfying certain properties in the graph, if A and B are vertices, we write A≡B when A and B are different names for the same vertex. If there is an undirected path U containing vertices A and B in directed graph G, and there is only one subpath of U between A and B, then U(A,B) is the subpath of U between A and B.

Lemma 1: In a directed graph G with vertices **V**, if **X**, **Y**, and **Z** are disjoint subsets of **V**, and **X** is d-connected to **Y** given **Z** in G, then **X** is d-connected to **Y** given **Z** in an acyclic directed subgraph of G.

Proof. Suppose that U is an undirected path that d-connects X and Y given **Z**, and C is a collider on U. Let *length*(C,**Z**) be 0 if C is a member of **Z**; otherwise it is the length of a shortest directed path from C to a member of **Z**. Let *size*(U) equal the number of colliders on U plus the sum over all colliders C on U of *length*(C,**Z**). U is a **minimal d-connecting path** between X and Y given **Z**, if U d-connects X and Y given **Z** and there is no other path U' that d-connects X and Y given **Z** such that *size*(U') < *size*(U). If there is a path

that d-connects X and Y given **Z** there is at least one minimal d-connecting path between X and Y given **Z.**

Suppose **X** is d-connected to **Y** given **Z.** Then for some X in **X** and Y in **Y**, there is a minimal d-connecting path U between X and Y given **Z**. It follows immediately from the definition of a d-connecting path that U is acyclic. First we will show that no shortest acyclic directed path D_i from a collider C_i on U to a member of **Z** intersects U except at C_i by showing that if such a point of intersection exists then U is not minimal, contrary to our assumption. See Figure 17.

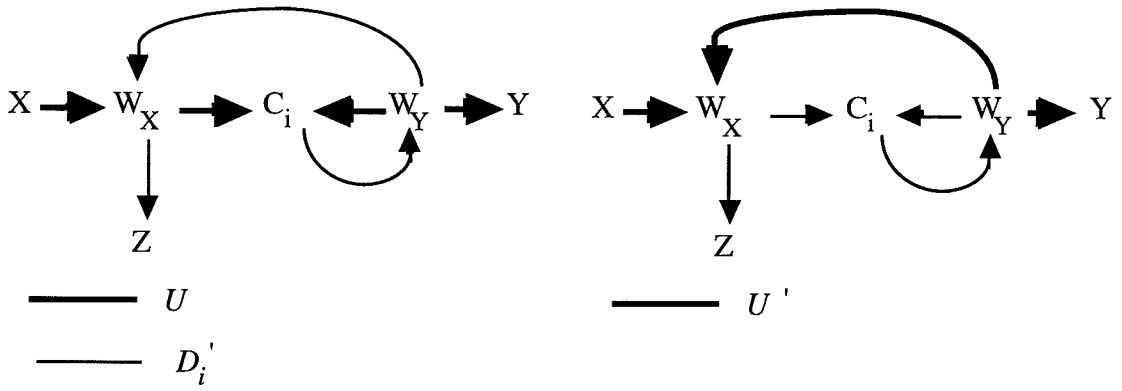


Figure 17

Form the path U' in the following way. If D_i intersects U at a vertex other than C_i then let W_X be the vertex closest to X on U that is on both D_i and U, and W_Y be the vertex closest to Y on U that is on both D_i and U. Suppose without loss of generality that W_X is after W_Y on D_i. Let U' be the concatenation of $U(X,W_X)$, $D_i(W_Y,W_X)$, and $U(W_Y,Y)$. It is now easy to show that U' d-connects X and Y given **Z**, and $size(U') < size(U)$ because U' contains no more colliders than U and a shortest directed path from W_X to a member of **Z** is shorter than D_i. Hence U is not minimal, contrary to the assumption.

Next, we will show that if U is minimal, then it does not contain a pair of colliders C and D such that a shortest directed path from C to a member of **Z** intersects a shortest path from D to a member of **Z**. Suppose this is false. See Figure 18.

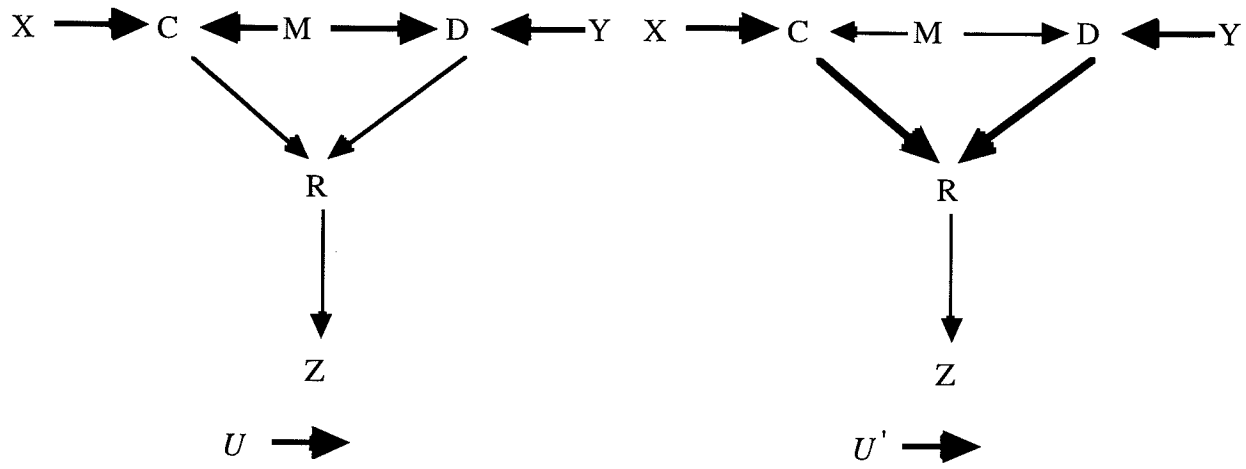


Figure 18

Let D_1 be a shortest directed acyclic path from C to a member of **Z** that intersects D_2, a shortest directed acyclic path from D to a member of **Z**. Let the vertex on D_1 closest to C that is also on D_2 be R. Let U' be the concatenation of U(X,C), D_1(C,R), D_2(D,R), and U(Y,D). It is now easy to show that U' d-connects X and Y given **Z** and $size(U') < size(U)$ because C and D are not colliders on U', the only collider on U' that may not be on U is R, and the length of a shortest path from R to a member of **Z** is less than the length of a shortest path from D to a member of **Z**. Hence U is not minimal, contrary to the assumption.

For each collider C on a minimal path U that d-connects X and Y given **Z**, a shortest directed path from C to a member of **Z** does not intersect U except at C, and does not intersect a shortest directed path from any other collider D to a member of **Z**. It follows that the directed subgraph consisting of U and a shortest directed acyclic path from each collider on U to a member of **Z** is acyclic. ∴

Lemma 2 (Lauritzen *et al.*, 1990): In a directed (cyclic or acyclic) graph G, disjoint sets of variables **X**, **Y** and **Z**, **X** and **Y** are Pearl d-connected given **Z** if and only if **X** and **Y** are Lauritzen d-connected given **Z**.

Lauritzen *et al.*, originally proved this for the acyclic case, but the proof goes over essentially unchanged to the cyclic case. Since Lauritzen d-separation and Pearl d-separation are equivalent, henceforth we will simply refer to "d-separation" when the context makes clear which definition is being used.

Theorem 3: The probability measure P over the substantive variables of a linear SEM L (recursive or non-recursive) with jointly independent error variables satisfies the global directed Markov property for the directed (cyclic or acyclic) graph G of L, i.e. if **X**, **Y**, and **Z** are disjoint sets of variables in G and **X** is d-separated from **Y** given **Z** in G, then **X** and **Y** are independent given **Z** in P.

Proof. Let **Err(X)** be the set of error variables corresponding to a set of substantive variables **X**. In order to distinguish the density function for **V** from the density function for the error variables we will use $f_{\mathbf{V}}$ to represent the density function (including marginal densities) for the former and $f_{\mathbf{Err}}$ to represent the density function of the latter. If **V** is the set of variables in G, then by hypothesis,

$$f_{\mathbf{Err}}(\mathbf{Err}(\mathbf{V})) = \prod_{\varepsilon \in \mathbf{Err}(\mathbf{V})} f_{\mathbf{Err}}(\varepsilon)$$

It is possible to integrate out the error variables not in **Err(An(X))** and obtain

$$f_{\mathbf{Err}}(\mathbf{Err}(\mathbf{An}(\mathbf{X}))) = \prod_{\varepsilon \in \mathbf{Err}(\mathbf{An}(\mathbf{X}))} f_{\mathbf{Err}}(\varepsilon)$$

Because for each variable X in **V**, X is a linear function of its parents in G plus a unique error variable ε_X, it follows that ε_X is a linear function g_X of X and the parents of X in G. Hence **Err(An(X))** is a function of **An(X)**. Following Haavelmo (1943) it is possible to derive the density function for the set of variables **An(X)** by replacing each ε_X in $f_{\mathbf{Err}}(\varepsilon_X)$ by $g_X(X,\mathbf{Parents}(X))$ and multiplying by the absolute value of the Jacobian:

$$f_{\mathbf{V}}(\mathbf{An}(\mathbf{X})) = \prod_{X \in \mathbf{An}(\mathbf{X})} f_{\mathbf{Err}}(g_X(X,\mathbf{Parents}(X))) \times |J|$$

where J is the Jacobian of the transformation. Because the transformation is linear, the Jacobian is a constant. All of the terms in the multiplication are non-negative because they are either a density function or a positive constant. It follows from Theorem 2 that if **X** and **Y** are d-separated given **Z** then **X** and **Y** are independent given **Z**. ∴

6.2. Proof of Theorem 4

Theorem 4: In a linear SEM L with jointly independent error variables and directed (cyclic or acyclic) graph G containing disjoint sets of variables **X**, **Y** and **Z**, if **X** is not d-separated from **Y** given **Z** in G then L does not linearly entail that **X** is independent of **Y** given **Z**.

Proof. Suppose that **X** is not d-separated from **Y** given **Z**. By Lemma 1, if **X** is not d-separated from **Y** given **Z** in a cyclic directed graph *G*, then there is some acyclic directed subgraph *G'* of *G* in which **X** is not d-separated from **Y** given **Z**. Geiger and Pearl (1988) have shown that if **X** is not d-separated from **Y** given **Z** in a directed acyclic graph, then there is some distribution represented by the directed acyclic graph in which **X** is not independent of **Y** given **Z**, and it has been shown (Spirtes, Glymour and Scheines, 1993) that there is in particular an instantiation of a linear parameterization of a SEM with directed graph *G* and no correlated errors in which **X** is not independent of **Y** given **Z**. If P satisfies the global directed Markov property for *G'* it also satisfies it for *G* because every d-connecting path in *G'* is a d-connecting path in *G*. Hence there is a distribution represented by *G* in which **X** is not independent of **Y** given **Z**. ∴

6.3. Proof of Theorem 5

Theorem 5: (Soundness) Given as input an oracle for testing d-separation relations in the directed (cyclic or acyclic) graph *G*, then the output is a PAG Ψ for *G*.

Proof. The proof proceeds by showing that each section of the CCD algorithm makes correct inferences from the answers given by the d-separation oracle for *G*, to the structure of *G* (and hence any directed graph in **Equiv**(*G*)).

Section ¶A

Lemma 3: Let *G* be a directed graph with vertex set **V**, and X, Y∈ **V**. The following are equivalent:

(a) $\exists \mathbf{Z} \subseteq \mathbf{V} \backslash \{X,Y\}$ such that X and Y are d-separated given **Z**, i.e. X and Y are not p-adjacent.

(b) {X,Y} is not an edge in Moral(*G*(**An**({X,Y}))).

(c) None of the following conditions hold in *G*:

(i) X is a parent of Y

(ii) Y is a parent of X

(iii) X and Y have a common child C that is an ancestor of X or Y.

Proof:

(a)⇒(b) Observe that Moral(G(**An**({X,Y}))) is a subgraph of Moral(G(**An**({X,Y}∪**Z**))). The hypothesis implies that {X,Y} is not an edge in Moral(G(**An**({X,Y∪**Z**}))). Hence it is also not an edge in Moral(G(**An**({X,Y}))).

(b)⇔(c) By definition of the operation of graph moralization on G(**An**({X,Y})): there is an edge between X and Y in Moral(G(**An**({X,Y}))) if and only if either there is an edge between X and Y in G(**An**({X,Y})) and thus in G, i.e. (i) or (ii) holds, or X and Y have a common child C in G(**An**({X,Y})), in which case (iii) holds.

(c)⇒(a) Take **Z**=**An**({X,Y})\{X,Y}. By definition, every vertex in Moral(G(**An**({X,Y}))) is an ancestor of X or Y. Since (c)⇒(b) there is no edge between X and Y in Moral(G(**An**({X,Y}))). Thus there is a vertex in **Z** lying on every path from X to Y in Moral(G(**An**({X,Y}))) ≡ Moral(G(**An**(**Z**∪{X,Y}))). Hence X and Y are d-separated given **Z**. ∴

Corollary 1: In directed graph G, if X and Y are p-adjacent then either X is an ancestor of Y, or Y is an ancestor of X (or both).

Proof: This follows immediately from the previous Lemma: if X and Y are p-adjacent then either (i) X is a parent of Y, (ii) Y is a parent of X, or (iii) X and Y have a common child C that is an ancestor of X or Y (or some combination). ∴

Lemma 4: In directed graph G, if X and Y are not p-adjacent then X and Y are d-separated given $\mathbf{T}_{X;Y}$={V| V is adjacent to X in Moral(G(**An**({X,Y}))) }.
Further, either $\mathbf{T}_{X;Y} \subseteq$ {V | V is p-adjacent to X in G} or X is an ancestor of Y in G.

Proof: Since X and Y are not p-adjacent it follows from Lemma 3 that there is no edge between them in Moral(G(**An**({X,Y}))). Hence every path from X to Y in Moral(G(**An**({X,Y}))) contains at least two edges. Hence the vertex closest to X on any path is in $\mathbf{T}_{X;Y}$. So X and Y are d-separated given $\mathbf{T}_{X;Y}$.

We now show that either $\mathbf{T}_{X;Y} \subseteq$ {V | V is p-adjacent to X in G} or X is an ancestor of Y in G. By the definition of graph moralization, in G every vertex in $\mathbf{T}_{X;Y}$ is either (a) a parent of X, (b) a child of X, or (c) a parent V of some vertex C, where C is also a child of X and an ancestor of X or Y. Any vertex in the first two categories is clearly p-adjacent to X. If C is an ancestor of X, then V is p-adjacent to X. If C is an ancestor of Y, then X is an ancestor of Y.∴

Lemma 5: In a directed graph G, if A and B are not p-adjacent then either A and B are d-separated by a set of vertices all of which are p-adjacent to A, or by a set of vertices all of which are p-adjacent to B.

Proof. Let $\mathbf{T}_{A;B}$, and $\mathbf{T}_{B;A}$ be defined as in Lemma 4. It follows from this Lemma that A and B are d-separated given $\mathbf{T}_{A;B}$ and A and B are d-separated given $\mathbf{T}_{B;A}$. There are three cases to consider:

Case 1: A is not an ancestor of B.
From Lemma 4, since A is not an ancestor of B, $\mathbf{T}_{A;B} \subseteq \{V \mid V \text{ p-adjacent to } A\}$.

Case 2: B is not an ancestor of A. Symmetrical to Case 1.

Case 3: B is an ancestor of A, and A is an ancestor of B. Any vertex V in $\mathbf{T}_{A;B}$ is either a child of A, a parent of A, or a parent of some vertex C, which is also a child of A and an ancestor of A or B. Clearly vertices in the first two categories are p-adjacent to A; as before, vertices in the last category are p-adjacent to A if C is an ancestor of A. Since C is an ancestor of A or B, and B is an ancestor of A, consequently C is an ancestor of A. (Note that in this case every vertex in $\mathbf{T}_{B;A}$ is also p-adjacent to B.) ∴

Suppose that the input to the algorithm is a d-separation oracle for a directed graph *G*. To find a set which d-separates some pair of variables A and B in *G* the algorithm tests subsets of **Adjacencies** (Ψ,A) and subsets of **Adjacencies** (Ψ,B) to see if they d-separate A and B. Since the vertices which are p-adjacent to A in *G* are at all times a subset of **Adjacencies** (Ψ,A),[16] and likewise the vertices p-adjacent to B are always a subset of **Adjacencies** (Ψ,B), it follows from Lemma 5 that step ¶A is guaranteed to find a set which d-separates A and B, if any set d-separates A and B in *G*. Clearly the order in which subsets of **Adjacencies** (Ψ,A) and **Adjacencies** (Ψ,B) of a fixed cardinality are tested in ¶A will not affect whether or not a d-separating set for a given pair of variables is found: the above argument shows that the search in ¶A is guaranteed to find some d-separating set for A and B if such exists (i.e. A and B are not p-adjacent). However, *which* d-separating set the search finds first may be influenced by the ordering of the tests in ¶A.[17]

Section ¶B

The next lemma and corollary give an important property of d-separating sets that are found through a search which never tests a set unless it has already tested every proper subset of that set (as in ¶A of the CCD algorithm).

---

[16]This is because if a pair of vertices X,Y are p-adjacent in *G* then no set is found which d-separates them, and hence the edge between X and Y in Ψ is never deleted.

[17] In this regard note that there may be vertices in **Sepset**(A,B) that are not p-adjacent to A or B. This is because although, in searching for **Sepset**(A,B) only subsets of **Adjacencies** (Ψ,A) and **Adjacencies** (Ψ,B) are tested, there may be vertices which are in these sets on account of edges in Ψ that have yet to be deleted at that point in the search, i.e. vertices which are not p-adjacent to A or B.

Lemma 6: Suppose that in a directed graph G, Y is not an ancestor of X or Z or **R**. If there is a set **S**, such that $\mathbf{R} \subset \mathbf{S}$, $Y \in \mathbf{S}$, and for every set **T** s.t. $\mathbf{R} \subseteq \mathbf{T} \subseteq \mathbf{S}\backslash\{Y\}$ X and Z are d-connected given T in G , then **S** d-connects X and Z in G.

Proof. Let $\mathbf{T}^* = \mathbf{An}(\{X,Z\} \cup \mathbf{R}) \cap \mathbf{S}$. Since by assumption $Y \notin \mathbf{An}(\{X,Z\} \cup \mathbf{R})$, $Y \notin \mathbf{T}^*$. Now, $\mathbf{R} \subseteq \mathbf{T}^*$, and $\mathbf{T}^* \subseteq \mathbf{S}\backslash\{Y\}$, so by hypothesis there is a d-connecting path, P, between X and Z, conditional on $\mathbf{T}^*$. By the definition of a d-connecting path every vertex on P is either an ancestor of one of the endpoints, or $\mathbf{T}^*$. Moreover, by definition, every vertex in $\mathbf{T}^*$ is an ancestor of X or Z or **R**. Thus every vertex on the path P is an ancestor of X or Z or **R**. Since neither Y nor any vertex in $\mathbf{S}\backslash\mathbf{T}^*$ is an ancestor of X or Z or **R**, it follows that no vertex in $\mathbf{S}\backslash\mathbf{T}^*$ lies on P. Since $\mathbf{T}^* \subset \mathbf{S}$ the only way in which P could fail to d-connect given **S** would be if some vertex in $\mathbf{S}\backslash\mathbf{T}^*$ lay on the path. Hence P still d-connects X and Z given **S**. ∴

In a directed graph G , if X and Y are d-separated given **S**, and are d-connected given any proper subset of **S**, then **S** is a **minimal d-separating** set for X and Y in G.

The following corollary is useful here:

Corollary 2: In a directed graph G, if **S** is a minimal d-separating set for X and Y, then any vertex in **S** is an ancestor of X or Y in G.

Proof. The corollary follows immediately from Lemma 6, with $\mathbf{R} = \varnothing$ via contraposition. ∴

This shows that orientation rule ¶B(ii) is correct. If A and B, and B and C are p-adjacent, but **Sepset**(A,C) contains B, then we know from the search procedure that A and C are not d-separated given any subset of **Sepset**(A,C). It follows that B is an ancestor of A or C. Hence A*—*B*—*C should be oriented as A*—*B*—*C in the PAG.

The following Lemma shows the correctness of the orientation rule ¶B(i):

Lemma 7: In a directed graph G, if A and B are p-adjacent, B and C are p-adjacent, and B is an ancestor of A or C then A and C are d-connected given any set **S**, s.t. $A,B,C \notin \mathbf{S}$.

Proof. Since A and B, and B and C are p-adjacent in G it follows from Lemma 3 that {A,B} and {B,C} are edges in Moral(G(**An**({A,B}))) and Moral(G(**An**({B,C}))) respectively, hence also in Moral(G(**An**({A,B,C}∪**S**))). If $B \in \mathbf{An}(\{A,C\})$, then **An**({A,B,C}∪**S**) = **An**({A,C}∪**S**), hence {A,B} and {B,C} are edges in Moral(G(**An**({A,C}∪**S**))). If $B \notin \mathbf{S}$ then A—B—C is a path circumventing **S** in Moral(G(**An**({A,C}∪**S**))) hence A and C are d-connected given **S**. ∴

It follows by contraposition that if A and B are p-adjacent, B and C are p-adjacent, A and C are d-separated given **Sepset**<A,C>, and B$\notin$ **Sepset**<A,C>, then B is not an ancestor of A or C, hence A*—*B*—*C should be oriented as A*—>B<—*C in the PAG. It then follows from Corollary 1 that A is an ancestor of B, and C is an ancestor of B, hence these edges are oriented as A—>B<—C.

Section ¶C

Lemma 8: In a directed graph G, suppose X is an ancestor of Y. If there is a set **S** such that A and Y are d-separated given **S**, X and Y are d-connected given **S**, and X$\notin$ **S**, then A and X are d-separated given **S**.

Proof. Suppose for a contradiction that A and X are d-connected given **S**. In that case there is a path P between A and X in Moral(G(**An**({A,X}$\cup$**S**))) on which there is no vertex in **S**. Since, by hypothesis X and Y are d-connected given **S**, there is a path Q between A and X in Moral(G(**An**({X,Y}$\cup$**S**))) on which there is no vertex in **S**. Since {X,Y}$\cup$**S** and {A,X}$\cup$**S** are subsets of {A,X,Y}$\cup$**S** path P and path Q exist in Moral(G(**An**({A,X,Y}$\cup$**S**))). Since X is an ancestor of Y, **An**({A,X,Y}$\cup$**S**) = **An**({A,Y}$\cup$**S**). Thus P and Q exist in Moral(G(**An**({A,Y}$\cup$**S**))). Since P and Q intersect at least once (at X), and do not contain any vertices in **S**, it follows that there is a path R from A to Y in Moral(G(**An**({A,Y}$\cup$**S**))), which also does not contain any vertices in **S**. But this is a contradiction. ∴

Lemma 9: Let A, X and Y be three vertices in a directed graph G, such that X and Y are p-adjacent. If there is a set **S** such that:

(i) X$\notin$ **S**,

(ii) A and Y are d-separated given **S**, and

(iii) A and X are d-connected given **S**,

then X is not an ancestor of Y.

Proof. Suppose that there is such a set **S**. If X and Y are p-adjacent then X and Y are d-connected by every subset of the other variables. In particular X and Y are d-connected given **S**. Since **S** d-separates A and Y but d-connects A and X, it follows from Lemma 8 by contraposition that X is not an ancestor of Y.∴

Step ¶C simply applies Lemma 9. Suppose that <A,X,Y> is a triple such that:

(i) A is not p-adjacent to X or Y,
(ii) X and Y are p-adjacent in Ψ, and
(iii) $X \notin$ **Sepset**<A,Y>.

¶C is justified in the following way. Suppose that A and X are d-connected given **Sepset**<A,Y>. Since $X \notin$ **Sepset**<A,Y>, setting **S** = **Sepset**<A,Y>, we can apply Lemma 9 to orient X o—o Y or X o— Y as X<–*Y. It then follows by Corollary 1 that Y is an ancestor of X, hence the edge is oriented as X<—Y.

It is a feature of this orientation rule that X and Y may be arbitrarily far from A. Rules of this type are needed by a cyclic discovery algorithm, because, as was shown in Richardson (1994b), two cyclic directed graphs may agree 'locally' on d-separation relations, but disagree on some d-separation relation between distant variables.[18]

We state without proof the following Lemma, used subsequently in the proof, which is an easy generalization of Lemma 3.3.1 in Spirtes *et al.* (1993). The Lemma states conditions under which a set of 'short' d-connecting paths may be put together to form a single d-connecting path.

Lemma 10: (Richardson 1994b)
In a directed (cyclic or acyclic) graph *G* over a set of vertices **V**, if the following conditions hold:
(a) *R* is a sequence of vertices in **V** from A to B, $R \equiv < A \equiv X_0, \ldots X_{n+1} \equiv B>$, such that $\forall i, 0 \leq i \leq n, X_i \neq X_{i+1}$ (the X_i are only *pairwise distinct* , i.e. not necessarily distinct),
(b) $\mathbf{Z} \subseteq \mathbf{V} \backslash \{A,B\}$,
(c) **T** is a set of undirected paths such that
(i) for each pair of consecutive vertices in *R*, X_i and X_{i+1}, there is a unique undirected path in **T** that d-connects X_i and X_{i+1} given $\mathbf{Z} \backslash \{X_i , X_{i+1}\}$,
(ii) if some vertex X_k in *R*, is in **Z**, then the paths in **T** that contain X_k as an endpoint collide at X_k, (i.e. all such paths are directed into X_k)
(iii) if for three vertices X_{k-1}, X_k, X_{k+1} occurring in *R*, the d-connecting paths in **T** between X_{k-1} and X_k, and X_k and X_{k+1}, collide at X_k then X_k has a descendant in **Z**,
then there is a path *U* in *G* that d-connects $A \equiv X_0$ and $B \equiv X_{n+1}$ given **Z** that contains only edges occurring in **T**.

Section ¶D This section searches to find 'extra' d-separating sets for triples oriented as X—>Y<—Z by ¶B (where X and Z are not p-adjacent). In the acyclic case, a triple of

[18] Whether or not such rules will ever be used on real data, in which 'distant' variables are generally found to be independent by statistical tests is another question.

vertices X*—*Y*—*Z, where X and Y are p-adjacent, Y and Z are p-adjacent, but X and Z are not p-adjacent, either has the property that every d-separating set for X and Z contains Y, or that every d-separating set for X and Z does not contain Y. However, in the cyclic case it is possible for X and Z to be d-separated by one set containing Y, and one set not containing Y. We already know from Lemma 7 that if X and Z are d-separated by some set which does not contain Y, then Y is not an ancestor of X or Z. What can we infer if in addition X and Z are also d-separated by a set which contains Y? This is answered by the next Lemma and Corollary.

Lemma 11: In a directed graph *G*, Y is a descendant of a common child of X and Z then X and Z are d-connected by any set containing Y.

Proof. Suppose that Y is a descendant of a common child C of X and Z. Then the path X→C←Z d-connects X and Z given any set containing Y. ∴

Corollary 3: If in a directed graph *G*, with vertices X, Y and Z, if there is some set **S** such that Y∈**S**, and X and Z are d-separated given **S**, then Y is not a descendant of a common child of X and Z.

It follows from Lemma 12 that if <X,Y,Z> is a triple such that X and Z are d-connected given any set containing Y, and d-separated by some set not containing Y, then Y is a descendant of a common child of X and Z.

Lemma 12: In directed graph *G*, if X and Z are not p-adjacent, and Y is not a descendant of a common child of X and Z, then X and Z are d-separated by the set **T**, defined as follows:

T = {V | V is adjacent to X in Moral(*G*(**An**({X,Y,Z})))}.

Further, if X and Y are p-adjacent then Y∈**T**.

Proof: Since X and Z are not p-adjacent it follows by Lemma 3 that X and Z are not adjacent in Moral(*G*(**An**({X,Z}))). As Y is not a descendant of a common child of X and Z, it then follows that X and Z are not adjacent in Moral(*G*(**An**({X,Y,Z}))). Hence Z∉**T** and every path from X to Z in Moral(*G*(**An**({X,Y,Z}))) contains some vertex in **T**. Thus X and Z are d-separated given **T**.

If X and Y are p-adjacent in *G* then Y is adjacent to X in Moral(*G*(**An**(X,Y))), and therefore in Moral(*G*(**An**({X,Y,Z}))). Thus Y∈**T**.∴

Lemma 13: In directed graph *G*, if X and Z are d-separated by some set **R**, then for all sets **Q** ⊆ **An**(**R** ∪ {X,Z})\{X,Z}, X and Z are d-separated by **R** ∪ **Q**.

Proof. If $\mathbf{Q} \subseteq \mathbf{An}(\mathbf{R} \cup \{X,Z\})\backslash\{X,Z\}$ then $\mathbf{An}(\mathbf{R} \cup \{X,Z\}) = \mathbf{An}(\mathbf{R} \cup \mathbf{Q} \cup \{X,Z\})$. It follows that Moral($G$(**An**(**R**$\cup$ {X,Z}))) = Moral(G(**An**(**R**$\cup$ **Q**$\cup$ {X,Z})))). The result then follows via the (Lauritzen) definition of d-connection..∴

The search in section ¶D considers in turn each triple A—>B<—C in Ψ, A and C not p-adjacent, and attempts to find a set **R** which is a subset of **Local**(Ψ,A)\{B,C} such that A and C are d-separated given **R** $\cup$ {B} $\cup$ **Sepset**<A,C>. It follows from Lemma 11, that if there is some set which d-separates A and C, and contains B, then B is not a descendant of a common child of A and C. It then follows from Lemma 12 that in this case there is some subset, the set **T** given in the Lemma, which contains B, d-separates A and C and in which every vertex is either a parent of A, a child of A, or a parent of a child of A and so **T** $\subseteq$ **Local**(Ψ,X). Since **Sepset**<A,C> is a minimal d-separating set for A and C, it follows that **Sepset**<A,C> $\subseteq$ **An**({A,C})\{A,C} ($\subseteq$ **An**(**T** $\cup${A,C}). Hence by Lemma 13, **T** $\cup$ **Sepset**<A,C> also d-separates A and C.

The reader may wonder why ¶D tests sets of the form **T** $\cup$ **Sepset**<A,C>, (where **T** $\subseteq$ **Local**(Ψ,A)), instead of just testing sets of the form **T** $\subseteq$ **Local**(Ψ,A)); Lemma 12 shows that a search of the latter kind would succeed in finding a d-separating set for A and C which contained B. The answer is that from Lemma 13 we know that any set **T** $\subseteq$ **Local**(Ψ,A) which d-separates A and C is such that **T** $\cup$ **Sepset**<A,C> also d-separates A and C, but the reverse is not true. In particular the smallest set **T** such that **T** $\cup$ **Sepset**<A,C> d-separates A and C may be considerably smaller than the smallest set **T** which d-separates A and C alone, hence the search is significantly faster.[19]

We require one more lemma to explain why we initialize m = 1, and do not test **T** = ∅.

Lemma 14: In directed graph G, if X and Y are p-adjacent, Y and Z are p-adjacent, X and Z are not p-adjacent, Y is not an ancestor of X or Z, and **S** is a minimal d-separating set for X and Z then X and Z are d-connected given **S** $\cup$ {Y}.

Proof. According to Lemma 3, if X and Y are p-adjacent then either X→Y, Y→X or X→C←Y, where C is an ancestor of X or Y. Thus under the hypothesis that Y is not an ancestor of X it follows that X is an ancestor of Y. Moreover, it follows that there is a directed path P from X to Y, on which every vertex except X is a descendant of Y, and hence on which every vertex except X is not an ancestor of X or Z. (In the case X→Y,

[19]In some cases the cardinality of the smallest set (**T** $\cup$ **Sepset**<A,C>) may be greater than the cardinality of the smallest **T**; but this is not true in general, and since we only intend to discover linear models this is insignificant. (With discrete models conditioning on a large set of variables in a conditional independence test may reduce dramatically the power of the test.)

the last assertion is trivial. In the other case it merely states a property of the path X→C→...Y, where C is a common child of X and Y.) Likewise there is a path Q from Z to Y on which every vertex except Z is not an ancestor of X or Z.

If **S** is a minimal d-separating set for X and Z every vertex in **S** is an ancestor of X or Z, (and X,Z ∉ **S**). Hence no vertex on P or Q is in **S**. It follows that P d-connects X and Y given **S**, and Q d-connects Y and Z given **S**. It then follows from Lemma 10 that these paths can be joined to form a single d-connecting path, hence X and Z are d-connected given **S** ∪ {Y}..∴

This completes the proof that step ¶D of the algorithm will succeed in finding a set which d-separates A and C, and contains B, for each triple A—>B<—C in the PAG, if any such set exists. A number of the subsequent proofs make use of the following consequence: For *every* triple A, B, C such that Ψ contains A—>B<—C, A and C are not p-adjacent in Ψ, and B is not a descendant of a common child of A and C, ¶D orients A—>B<—C as A—>B<—C.

Section ¶E

The following Lemma provides the justification of ¶E where A—>B<—C, A—>D<—C, and D is not in **SupSepset**<A,B,C>, in which case B o—o D or B —o D is oriented as B—>D.

Lemma 15: If in a PAG Ψ for G, X—>V<—Z, X—>W<—Z, X and Z are not p-adjacent, and W is an ancestor of V in G, then any set **S** such that V∈ **S**, and X and Z are d-separated by **S**, also contains W.

Proof. Suppose there were some d-separating set **S** for X and Z which contained V and did not contain W. Then, since W is an ancestor of V and V∈ **S**, but W∉ **S**, it follows by Lemma 10 that we could put together a d-connecting path from X to W given **S** and from W to Z given **S** to form a new d-connecting path from X to Z given **S** (irrespective of whether or not these paths collide at W). Such d-connecting paths between X and W, and between W and Z exist (by Lemma 3) since X is p-adjacent to W and W is p-adjacent to Z. This is a contradiction..∴

In the case in which A—>B<—C, A—>D<—C, and D is in **SupSepset**<A,B,C> the algorithm orients B*—oD as B*—D, the inference can be justified as follows. If D is in **SupSepset**<A,B,C> then it follows from Lemma 6 and the fact that section **¶D** looks for the smallest superset of {B} ∪ **Sepset**<A,C> which d-separates A and C that D is an ancestor of {B} ∪ **Sepset**<A,C>. Since **Sepset**<A,C> is a minimal d-separating set for A

and C, every vertex in **Sepset**<A,C> is an ancestor of A or C. Thus if D is in **SupSepset**<A,B,C>, D is an ancestor of A, C or B. However, since there are arrowheads at D on the edges from A to D, and C to D in Ψ, it follows that D is not an ancestor of A or C, and hence D is an ancestor of B. Thus it is correct to orient B *—o D as B*—D.

In the case in which A—>D<—C in Ψ, (A and C are not p-adjacent and there is no dotted line A—>D<—C), it does not matter whether D is in **SupSepset**<A,B,C> or not. A and C are d-connected by any set **S** that contains D but does not contain A or C (because of thelack of underlining in the edge pair A—> D <—C). It follows from Lemma 12 by contraposition that D is a descendant of a common child of A and C. Moreover since A and C are d-separated by some set containing B (because of the underlining in the edge pair A—>B<—C), B is not a descendant of a common child of A and C. Hence B is not a descendant of D. Thus in the case where in Ψ, A—>B<—C, A—>D<—C, B and D are p-adjacent, B o—o D or B —o D should be oriented as B—>D.

Section ¶F

A and C are d-separated by **SupSepset**<A,B,C>, and B$\in$ **SupSepset**<A,B,C>. Hence by Lemma 13, if D is an ancestor of B, then A and C are d-separated by **SupSepset**<A,B,C> $\cup$ {D}. Hence by contraposition, if A and C are d-connected given **SupSepset**<A,B,C> $\cup$ {D} then D is not an ancestor of B. (In fact, it follows that D is not an ancestor of A,B or C.) Since D is not an ancestor of B, but B and D are p-adjacent it follows by Corollary 1 that B is an ancestor of D. Thus B o—o D or B —o D should be oriented as B—>D in Ψ.

This completes the proof of the correctness of the CCD algorithm. $\therefore$

6.4. Proof of Theorem 7

In order to prove the d-separation completeness of the CCD algorithm, all that is required is to show that whenever the first input to the CCD algorithm is a d-separation oracle for G_1 that results in output Ψ_1, and the second input to the CCD algorithm is a d-separation oracle for G_2 that results in output Ψ_2, and Ψ_1 and Ψ_2 are identical, then G_1 and G_2 are d-separation equivalent. We shall do this by proving that when d-separation oracles for G_1 and G_2 are used as input to the CCD algorithm and produce the same PAG as output, then G_1, and G_2 satisfy the five conditions of the Cyclic Equivalence Theorem CET(I)-(V) (given below) with respect to one another. It has already been shown in Richardson(1994b) that two directed graphs G_1 and G_2 are d-separation equivalent to one another if and only if they satisfy these 5 conditions. These conditions lead directly to a

polynomial-time ($O(n^9)$)$\equiv O(n^3e^4)$) algorithm, for determining whether or not two directed cyclic graphs are d-separation equivalent, see Richardson (1994b, 1995).

Before stating the Cyclic Equivalence Theorem we require a number of extra definitions. In a directed graph G, call a triple of vertices <A,B,C> an **unshielded triple** if A and B are p-adjacent, B and C are p-adjacent, but A and C are not p-adjacent.

Call an unshielded triple a **conductor** if B is an ancestor of A or C, otherwise, if B is not an ancestor of A or C, call it a **non-conductor**. (Note that it follows from Corollary 1 that if <A,B,C> is a non-conductor then A and C are ancestors of B.) Call a non-conductor **perfect** if B is a descendant of a common child of A and C, otherwise call it **imperfect**.

If $<X_0,X_1,\ldots X_{n+1}>$ is a sequence of distinct vertices s.t. $\forall i\ 0 \leq i \leq n$, X_i and X_{i+1} are p-adjacent then we will refer to $<X_0,X_1,\ldots X_{n+1}>$ as an **itinerary**.
If $<X_0,\ldots X_{n+1}>$ $(n \geq 2)$ is an itinerary such that:
(i) $\forall t\ 1 \leq t \leq n$, $<X_{t-1}, X_t, X_{t+1}>$ is a conductor,
(ii) $\forall k\ 1 \leq k \leq n$, X_{k-1} is an ancestor of X_k, and X_{k+1} is an ancestor of X_k, and
(iii) X_0 is *not* a descendant of X_1, and X_n is *not* an ancestor of X_{n+1},
then $<X_0,X_1,X_2>$ and $<X_{n-1},X_n,X_{n+1}>$ are **mutually exclusive (m.e.) conductors on the itinerary** $<X_0,\ldots X_{n+1}>$.[20]

If $<X_0,\ldots X_{n+1}>$ is an itinerary such that $\forall i,j\ 0 \leq i < j-1 < j \leq n+1$ X_i and X_j are not p-adjacent in the directed graph then we say that $<X_0,\ldots X_{n+1}>$ is an **uncovered itinerary**, i.e. an itinerary is uncovered if the only vertices on the itinerary which are p-adjacent to other vertices on the itinerary, are those that occur consecutively on the itinerary.

Theorem 6: (**Cyclic Equivalence Theorem,** Richardson 1994b) Directed graphs G_1 and G_2 are d-separation equivalent if and only if the following five conditions hold:

CET(I) G_1 and G_2 have the same p-adjacencies,

CET(II) G_1 and G_2 have (a) the same conductors, and (b) the same perfect non-conductors,

CET(III) For all triples <A,B,C> and <X,Y,Z>, <A,B,C> and <X,Y,Z> are m.e. conductors on some uncovered itinerary $P \equiv <A,B,C,\ldots X,Y,Z>$ in G_1 if and only if <A,B,C> and <X,Y,Z> are m.e. conductors on some uncovered itinerary $Q \equiv <A,B,C,\ldots X,Y,Z>$ in G_2,

CET(IV) If <A,X,B> and <A,Y,B> are imperfect non-conductors (in G_1 and G_2), then X is an ancestor of Y in G_1 if and only if X is an ancestor of Y in G_2,

[20] Note that a pair of m.e. conductors on an uncovered itinerary are a generalization of a non-conductor. In both cases there is a set of vertices "in the middle" that are not ancestors of the vertices at the "ends".

CET(V) If <A,B,C> and <X,Y,Z> are mutually exclusive conductors on some uncovered itinerary **P** ≡<A,B,C,...X,Y,Z> and <A,M,Z> is an imperfect non-conductor (in G_1 and G_2), then M is a descendant of B in G_1 iff M is a descendant of B in G_2.

Lemma 16: Given a sequence of vertices $<X_0,...X_{n+1}>$ in a directed graph G having the property that $\forall k$, $0 \leq k \leq n$, X_k is an ancestor of X_{k+1}, and X_k is p-adjacent to X_{k+1} there is a subsequence of the X_i's, which we label the Y_j's having the following properties:

(a) $X_0 \equiv Y_0$

(b) $\forall j$, Y_j is an ancestor of Y_{j+1}

(c) $\forall j,k$ If $j < k$, Y_j and Y_k are p-adjacent in the directed graph if and only if $k = j+1$. i.e. the only Y_k's which are p-adjacent are those that occur consecutively.

Proof. The Y_k's can be constructed as follows:

Let $Y_0 \equiv X_0$.

Let $Y_{k+1} \equiv X_\eta$ where η is the greatest $h > j$ such that X_h is p-adjacent to X_j where $X_j \equiv Y_k$.

Property (a) is immediate from the construction. Property (b) follows from the transitivity of the ancestor relation, and the fact that the Y_k's are a subsequence of the X_i's. It is also clear, from the construction that if $k = j+1$ then Y_j and Y_k are p-adjacent. Moreover, if $Y_j \equiv X_\alpha$[21] and $Y_k \equiv X_\beta$ are p-adjacent, and $j < k$, then it follows again from the construction that if $Y_{j+1} \equiv X_\gamma$, then $\beta \leq \gamma$, so $k \leq j+1$. (This is because the Y_k's are a subsequence of the X_i's.) Hence $Y_{j+1} \equiv Y_k$. $\therefore$

Lemma 17: Let G_1 and G_2 be two directed graphs satisfying CET(I)–(III). Suppose there is a directed path $D_1 \rightarrow ... D_n$, in G_1. Let D_0 be a vertex distinct from $D_1,...,D_n$, s.t. D_0 is p-adjacent to D_1 in G_1 and G_2, D_0 is not p-adjacent to $D_2,...D_n$ in G_1 or G_2 and D_0 is not a descendant of D_1 in G_1 or G_2. It then follows that D_1 is an ancestor of D_n in G_2.

Proof. It follows from Lemma 16 that in G_1 there is a subsequence $<D_{\alpha(0)} \equiv D_0, D_{\alpha(1)}, D_{\alpha(2)}...D_{\alpha(m)} \equiv D_n>$ such that the only p-adjacent vertices are those that occur consecutively, and each vertex is an ancestor of the next vertex in the sequence. Since G_1 and G_2 satisfy CET(I), they have the same p-adjacencies, hence also in G_2 the only vertices in the subsequence that are p-adjacent are those that occur consecutively. Moreover, since, by hypothesis, D_0 is not p-adjacent to $D_2,...D_n$ in G_1 or G_2 it follows that $D_{\alpha(1)} \equiv D_1$ in G_1 and G_2.

---

[21] That is, the j^{th} vertex in the sequence of Y vertices is the α^{th} vertex in the sequence of X vertices.

Suppose, for a contradiction that some vertex $D_{\alpha(k-1)}$ is not an ancestor of its successor $D_{\alpha(k)}$ in the sequence in G_2. Let r be the smallest $k \leq m$ such that $D_{\alpha(k-1)}$ is not an ancestor of $D_{\alpha(k)}$ in G_2. Let s be the greatest $j \leq r-1$ such that $D_{\alpha(j)}$ is not an ancestor of $D_{\alpha(j-1)}$ in G_2. (Such a j exists since $D_{\alpha(1)} \equiv D_1$ and $D_{\alpha(0)} \equiv D_0$ is not a descendant of D_1.)

There are now two cases: $s = r-1$ or $s < r-1$.

If $s = r-1$ then the unshielded triple $<D_{\alpha(s-1)}, D_{\alpha(s)} \equiv D_{\alpha(r-1)}, D_{\alpha(r)}>$ is a non-conductor in G_2, since $D_{\alpha(s)} \equiv D_{\alpha(r-1)}$ is not an ancestor of $D_{\alpha(s-1)}$ or $D_{\alpha(r)}$. But in G_1, by hypothesis, $D_{\alpha(r-1)}$ is an ancestor of $D_{\alpha(r)}$ hence $<D_{\alpha(s-1)}, D_{\alpha(s)} \equiv D_{\alpha(r-1)}, D_{\alpha(r)}>$ is a conductor in G_2. But this is a condtradiction since G_1 and G_2 have the same conductors by CET(IIa).

If $s < r-1$ then it follows that $<D_{\alpha(s--1)}, D_{\alpha(s)}, D_{\alpha(s+1)}>$ and $<D_{\alpha(r-2)}, D_{\alpha(r-1)}, D_{\alpha(r)}>$ are mutually exclusive conductors on the uncovered itinerary $<D_{\alpha(s-1)}, \ldots D_{\alpha(r)}>$ in G_2. But these two triples are not mutually exclusive in G_1 since $D_{\alpha(r-1)}$ is an ancestor of $D_{\alpha(r)}$ in G_1; hence G_1 and G_2 fail to satisfy CET(III), which is a contradiction.

It follows that $D_{\alpha(r-1)}$ is an ancestor of $D_{\alpha(r)}$ in G_2. $\therefore$

Theorem 7: (d-separation Completeness) If the CCD algorithm, when given as input d-separation oracles for the directed graphs G_1, G_2 produces as output PAGs Ψ_1, Ψ_2 respectively, then Ψ_1 is identical to Ψ_2 only if G_1 and G_2 are d-separation equivalent, i.e. $G_2 \in \mathbf{Equiv}(G_1)$ and vice versa.

Proof. We will show that if two directed graphs, G_1 and G_2 are *not* d-separation equivalent, then the PAGs output by the CCD algorithm, given d-separation oracles for G_1 and G_2 as input, would differ in some respect.

It follows from the Cyclic Equivalence Theorem that if G_1 and G_2 are not d-separation equivalent, then they fail to satisfy one or more of the five conditions CET(I)-(V).

Case 1: G_1 and G_2 fail to satisfy CET(I).

In this case the two directed graphs have different p-adjacencies. It has already been established (Theorem 5) that the CCD algorithm outputs a PAG. It follows from clause (i) of the definition that G_1 and G_2 have different p-adjacencies if and only if the corresponding PAGs, Ψ_1 and Ψ_2 possess different adjacencies.

Case 2: G_1 and G_2 fail to satisfy CET(IIa). We assume that G_1 and G_2 satisfy CET(I). In this case the two directed graphs have different conductors and hence different non-conductors. Thus we may assume, without loss of generality, that there is some

unshielded triple of vertices <X,Y,Z> such that in G_1, Y is an ancestor of X or Z, while Y is not an ancestor of either X or Z in G_2.

If Y is an ancestor of X or Z then it follows from Lemma 7 that every set which d-separates X and Z contains Y. Hence Y∈**Sepset**(X,Z) in G_1. It then follows from ¶B(ii) that in Ψ_1, X*–*Y*–*Z.

If Y is not an ancestor of X or Z in G_2, then Y is not in any minimal d-separating set for X and Z. In particular Y∉ **Sepset**(X,Z) for G_2. Again it follows from the correctness of the algorithm that <X,Y,Z> is oriented as X–>Y<–Z or X–>Y<–Z in Ψ_2. Thus Ψ_1 and Ψ_2 are different.

Case 3: G_1 and G_2 fail to satisfy CET(IIb). We assume that G_1 and G_2 satisfy CET(I), CET(IIa). In this case the two directed graphs have different imperfect non-conductors, i.e. there is some triple <X,Y,Z> such that it forms a non-conductor in both G_1 and G_2, but in one directed graph Y is a descendant of a common child of X and Z, while in the other directed graph it is not. Let us assume that Y is a descendant of a common child of X and Z in G_1, while in G_2 it is not.

It follows from Lemma 11 that in G_1, X and Z are d-connected given any subset containing Y. In this case the search in CCD section ¶D will fail to find any set **Supsepset**<X,Y,Z>. Hence <X,Y,Z> will be oriented as X –>Y<– Z (i.e. without dotted underlining) in Ψ_1.

If Y is not a descendant of a common child of X and Z in G_2, then it follows from Lemma 12 and Lemma 13 that there is some subset **T** of **Local**(Ψ_2,X), such that X and Z are d-separated given **T** ∪ {Y} ∪ **Sepset**<X,Z>. Section ¶D will find such a set **T**, and hence <X,Y,Z> will be oriented as X–>Y<–Z in Ψ_2. Since no subsequent orientation rule removes or adds dotted underlining, it follows that Ψ_1 and Ψ_2 are different.

Case 4: G_1 and G_2 fail to satisfy CET(III). We assume that G_1 and G_2 satisfy CET(I), CET(IIa), CET(IIb). In this case the two directed graphs have the same p-adjacencies, and the same conductors, and perfect non-conductors. However, the two directed graphs have different mutually exclusive conductors. Hence in both G_1 and G_2 there is an uncovered itinerary, $<X_0,\ldots X_{n+1}>$ such that every triple $<X_{k-1},X_k,X_{k+1}>$ $(1 \leq k \leq n)$ on this itinerary is a conductor, but in one directed graph $<X_0,X_1,X_2>$ and $<X_{n-1},X_n,X_{n+1}>$ are mutually exclusive, i.e. X_1 is not an ancestor of X_0, and X_n is not an ancestor of X_{n+1}, while in the other they are not mutually exclusive. Let us suppose without loss of generality that $<X_0,X_1,X_2>$ and $<X_{n-1},X_n,X_{n+1}>$ are mutually exclusive in G_1, while in

G_2 they are not, and that no pair of mutually exclusive conductors on a shorter uncovered itinerary have this property.

From the definition of a pair of m.e. conductors it follows that in G_1 the vertices $X_1,...X_n$, inclusive are *not* ancestors of X_0 or X_{n+1}. Hence $\{X_1,...X_n\} \cap$ **Sepset**$(X_0,X_{n+1}) = \varnothing$, since **Sepset**$(X_0,X_{n+1})$ is minimal, and so is a subset of **An**$(\{X_0,X_{n+1}\})$. (Here, **Sepset**(X_0,X_{n+1}) is calculated for G_1.) For the same reason Descendants$(\{X_1,...,X_n\}) \cap$ **Sepset**$(X_0,X_{n+1}) = \varnothing$. It follows from the definition of a pair of m.e. conductors on an itinerary that X_k is an ancestor of X_{k+1} $(1 \leq k < n)$, thus there is a directed path $P_k \equiv X_k \rightarrow ... \rightarrow X_{k+1}$ in G_1. Since no descendant of $X_1,...,X_n$ is in **Sepset**(X_0,X_{n+1}), each of the directed paths P_k d-connects each vertex X_k to its successor X_{k+1} $(1 \leq k < n)$, conditional on **Sepset**(X_0,X_{n+1}). In addition, since X_0 and X_1 are p-adjacent there is some path Q d-connecting X_0 and X_1 given **Sepset**(X_0,X_{n+1}). Since each P_i is out of X_i (i.e. the path goes $X_i \rightarrow ... \rightarrow X_{i+i}$), by applying Lemma 10, with **T** = $\{Q,P_1,...P_n\}$, $R = \langle X_0,...X_n \rangle$, and **S** = **Sepset**$(X_0,X_{n+1})$ it follows that we can form a path d-connecting X_0 and X_n given **Sepset**(X_0,X_{n+1}). A symmetric argument shows that X_1 and X_{n+1} are also d-connected given **Sepset**(X_0,X_{n+1}). It then follows that the edges X_0*—*X_1 and X_n*—*X_{n+1} are oriented as X_0—>X_1 and X_n<—X_{n+1} in Ψ_1 by stage ¶C of the CCD algorithm (unless they have already been oriented this way in a previous stage of the algorithm). Thus again, by the correctness of the algorithm these arrowheads will be present in Ψ_1. (Subsequent stages of the algorithm only add '–' and '>' endpoints, not 'o' endpoints. If either of the arrowheads at X_1 or X_n were replaced with a '–' the algorithm would be incorrect.)

Since by hypothesis, no pair of conductors on the uncovered itinerary $\langle X_0...X_{n+1} \rangle$ are mutually exclusive in G_2, it follows from Lemma 17 that either X_1 is an ancestor of X_0, or X_n is an ancestor of X_{n+1}. It then follows from the correctness of the orientation rules in the CCD algorithm that the pair of edges X_0*—*X_1 and X_n*—*X_{n+1} will not both be oriented as X_0*—>X_1 and X_n<—*X_{n+1} in Ψ_2. Thus Ψ_1 and Ψ_2 will once again be different.

Case 5: G_1 and G_2 fail to satisfy either CET(IV) or CET(V). We assume that G_1 and G_2 satisfy CET(I)–(III).[22] If G_1 and G_2 fail to satisfy either CET(IV) or CET(V), then in

[22]The conditions under which CET(IV) or CET(V) fail are quite intricate precisely because the assumption that CET(I)-(III) are satisfied implies that the graphs agree in many respects.

either case we have the following situation: There is some sequence of vertices in G_1 and G_2 $<X_0,X_1,\ldots X_n, X_{n+1},V>$, [23] satisfying the following:

(a) if $i > j$ then X_i and X_j are p-adjacent if and only if $i = j+1$,

(b) X_1 is not an ancestor of X_0, and X_n is not an ancestor of X_{n+1},

(c) $\forall k$, $1 \leq k \leq n$, X_{k-1}, and X_{k+1} are ancestors of X_k,

(d) $<X_0,V,X_{n+1}>$ is an imperfect non-conductor, and

(e) in one directed graph V is a descendant of X_1, while in the other directed graph V is not a descendant of X_1.

As explained in Case 3, condition (d) implies that in both Ψ_1 and Ψ_2, $X_0 \text{—>} V \text{<—} X_{n+1}$.

Let us suppose without loss of generality that V is a descendant of X_1 in G_1, and V is not a descendant of X_1 in G_2. As in previous cases it is sufficient to show that if Ψ_1 and Ψ_2 are CCD PAGs corresponding to G_1 and G_2 respectively, then Ψ_1 and Ψ_2 are different. We may suppose, again without loss of generality that V is the closest such vertex to any X_k ($1 \leq k \leq n$) in G_1, in the sense that a shortest directed path $P \equiv X_k \rightarrow \ldots \rightarrow V$ in G_1 contains at most the same number of vertices as a shortest directed path in G_1 from any X_k ($1 \leq k \leq n$) to some other vertex V' satisfying the conditions on V.

Claim: Let W be the first vertex on P which is p-adjacent to V, (both in G_1 and G_2 since by CET(I) G_1 and G_2 have the same p-adjacencies). We will show that the assumption that V is the closest such vertex to any X_k (in G_1) together with the assumption that G_1 and G_2 satisfy CET(I)-(III) imply that W is a descendant of X_1 in G_2. We prove this by showing that every vertex in the directed subpath $P(X_k, W) \equiv X_k \rightarrow \ldots W$ in G_1 is also a descendant of X_1 in G_2.

Proof of Claim: By induction on the vertices occurring on the path $P(X_k, W)$.

Base Case: X_k. By hypothesis X_k is a descendant of X_1 in both G_1 and G_2.

Induction Case: Consider Y_r, where $P(X_k, W) \equiv X_k \rightarrow Y_1 \rightarrow \ldots \rightarrow Y_r \rightarrow \ldots Y_t \equiv W$. By the induction hypothesis, for $s < r$, Y_s is a descendant of X_1 in G_2. Now there are two subcases to consider:

Subcase 1: Not both X_0 and X_{n+1} are p-adjacent to Y_r. Suppose without loss that X_0 is not p-adjacent to Y_r. Since in G_1 there is a directed path $X_0 \rightarrow \ldots X_k \rightarrow Y_1 \rightarrow \ldots Y_r$, by Lemma 16 it then follows that there is some subsequence of this sequence of vertices, $Q \equiv <X_0, \ldots Y_r>$ such that consecutive vertices in Q are p-adjacent, but only these vertices

[23] In the case where CET(IV) fails n=1, while if CET(V) fails, n>1.

are p-adjacent. Moreover, since X_0 is not p-adjacent to Y_r, this sequence of vertices is of length greater than 2, i.e. $Q \equiv <X_0, D \ldots Y_r>$ where D is the first vertex in the subsequence after X_0, hence either $D \equiv X_\kappa$ $(1 \leq \kappa \leq k)$ or $D \equiv Y_\mu$, $(1 \leq \mu < r)$. Since in either case D is a descendant of X_1 in both G_1 and G_2, (either by the induction hypothesis or by the hypothesis of case 5), but X_0 is not a descendant of X_1 in G_1 or G_2 it follows that D is not an ancestor of X_0 in G_1 or G_2. Hence we may apply Lemma 17, to Q to deduce that Y_r is a descendant of D. Hence Y_r is a descendant of X_1 in G_2 since X_1 is an ancestor of D.

Subcase 2: Both X_0 and X_{n+1} are p-adjacent to Y_r. First note that in G_1 the vertex Y_r is a descendant of X_k, and X_k is not an ancestor of X_0 or X_{n+1}. It follows that Y_r is not an ancestor of X_0 or X_{n+1} in G_1 . Moreover, since X_0 and X_{n+1} are not p-adjacent, $<X_0, Y_r, X_{n+1}>$ forms a non-conductor in G_1. Hence $<X_0, Y_r, X_{n+1}>$ forms a non-conductor in G_2, since by hypothesis G_1 and G_2 satisfy CET(IIa). So Y_r is not an ancestor of X_0 or X_{n+1} in G_1 or G_2. Further, since Y_r is an ancestor of V in G_1 and V is not a descendant of a common child of X_0 and X_{n+1} in G_1, it follows that Y_r is not a descendant of a common child of X_0 and X_{n+1} in G_1. Thus $<X_0, Y_r, X_{n+1}>$ forms an imperfect non-conductor in G_1. Since G_1 and G_2 satisfy CET(I), CET(IIa), and CET(IIb), $<X_0, Y_r, X_{n+1}>$ forms an imperfect non-conductor in G_2. Now, if Y_r were not a descendant of X_1 in G_2, then Y_r would satisfy the conditions on V, yet be closer to X_k than V (Y_r occurs before V on a shortest directed path from X_k to V in G_1). This is a contradiction, hence Y_r is a descendant of X_k in G_2.

This completes the proof of the claim. We now show that Ψ_1 and Ψ_2 are different.

Consider the edge W*–*V in Ψ_1. In G_1, W is an ancestor of V, hence it follows from the correctness of the algorithm that in Ψ_1 this edge is oriented as Wo—*V or W—*V. In G_2, however, since X_1 is not an ancestor of V, but, as we have just shown X_1 is an ancestor of W, it follows that W is not an ancestor of V. Further, since W is a descendant of X_1 and so also of X_n, it follows from (b) that W is not an ancestor of X_0 or X_{n+1}. There are now two cases to consider:

Subcase 1: W is p-adjacent to both X_0 and X_{n+1}. Since W is not an ancestor of X_0 or X_{n+1} in G_1 or G_2, $<X_0, W, X_{n+1}>$ is a non-conductor in both G_1 and G_2. Further, since $X_0 \text{—}\underline{\geq V \leq}\text{—} X_{n+1}$ in Ψ_1 (and Ψ_2), and W is an ancestor of V in G_1, it follows that W is not a descendant of a common child of X_0 and X_{n+1} in G_1. Thus $X_0 \text{—}\underline{\geq W \leq}\text{—} X_{n+1}$ in Ψ_1 and hence, by CET(II), also in Ψ_2. **Supsepset**(X_0, V, X_{n+1}) is the smallest set containing **Sepset**$(X_0, X_{n+1}) \cup \{V\}$ which d-separates X_0 and X_{n+1}. It follows from Lemma 6 (with **R** = **Sepset**$(X_0, X_{n+1}) \cup \{V\}$) that **Supsepset**$(X_0, V, X_{n+1}) \subseteq$ **An**(**Sepset**$(X_0, X_{n+1}) \cup \{X_0,$

X_{n+1}, V}). Since **Sepset**(X_0,X_{n+1}) $\subseteq$ **An**({X_0, X_{n+1}}), **Supsepset**(X_0,V,X_{n+1}) $\subseteq$ **An**({X_0, X_{n+1}, V}). We have already shown that W is not an ancestor of X_0, X_{n+1}, or V in G_2. Hence in step ¶D of the algorithm given a d-separation oracle for G_2 as input W$\notin$ **Supsepset**(X_0,V,X_{n+1}). Thus step ¶E of the CCD algorithm will orient W_0 o—o V or an edge W_0 o—V in Ψ_2 as W<—V (unless they have already been oriented this way in a previous stage of the algorithm). Thus Ψ_1 and Ψ_2 are not the same.

Subcase 2:W is not p-adjacent to both X_0 and X_{n+1}.

Claim: X_0 and X_{n+1} are d-connected given **Supsepset**(X_0,V,X_{n+1})$\cup${W} in G_2.

Proof. Since in both G_1 and G_2 X_0 is p-adjacent to X_1, but X_1 is not an ancestor of X_0, it follows from Corollary 1 that X_0 is an ancestor of X_1. Hence in both G_1 and G_2 there is a directed path P_0 from X_0 to X_1 on which every vertex except for X_0 is a descendant of X_1. (In the case $X_0 \rightarrow X_1$, the last assertion is trivial. In the case where X_0 and X_1 have a common child that is an ancestor of X_0 or X_1, and X_1 is not an ancestor of X_0, it merely states a property of the path $X_0 \rightarrow C \rightarrow \ldots X_1$, where C is a common child of X_0 and X_1.) Since W is a descendant of X_1, it follows that there is a directed path P_1 from X_1 to W. Concatenating P_0 and P_1 we construct a directed path P^* from X_0 to W on which every vertex except X_0 is a descendant of X_1. Since X_1 is not an ancestor of X_0, X_{n+1} or V, it follows that no vertex on P^*, except X_0, is an ancestor of X_0, X_{n+1} or V. Similarly we can construct a path from Q^* from X_{n+1} to W on which no vertex, except X_{n+1}, is an ancestor of X_0, X_{n+1} or V.

Since every vertex in **Supsepset**(X_0,V,X_{n+1}) is an ancestor of X_0, X_{n+1} or **Sepset**(X_0,X_{n+1})$\cup${V}, it follows as before that every vertex in **Supsepset**(X_0,V,X_{n+1}) is an ancestor of X_0, X_{n+1} or V. Thus no vertex in **Supsepset**(X_0,V,X_{n+1}) lies on P^* or Q^* (X_0, X_{n+1} $\notin$ **Supsepset**(X_0,V,X_{n+1}) by definition). It now follows by Lemma 10 that we can concatenate P^* and Q^* to form a path R which d-connects X_0 and X_{n+1} given **Supsepset**(X_0,V,X_{n+1})$\cup${W}.

Since W is not p-adjacent to both X_0 and X_{n+1} it follows directly from this claim that step ¶F of the CCD algorithm will orient the edgeV o—o W or V —o W as V—>W in Ψ_2 (unless they have already been oriented this way in a previous stage of the algorithm). Hence Ψ_1 and Ψ_2 are different.

Since Cases 1-5 exhaust the possible ways in which G_1 and G_2 may fail to satisfy CET(I)-(V), this completes the proof. ∴

7. References

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 9, 308-347.

Cox, D.R., and Wermuth, N. (1993) Linear dependencies represented by chain graphs. In Statistical Science, 1993, 8 No.3 , 204-283.

Dawid, A. P. (1979) Conditional Independence in statistical theory (with discussion) Journal Royal Statistical Society Ser. B 41, 1-31. Frydenberg, M., (1990) The chain graph Markov property, Scandinvaian Journal of Statistics, 17, 333-353.

Geiger, D. (1990). Graphoids: a qualitative framework for probabilistic inference. PhD dissertation, Univ. California, Los Angeles.Geiger, D., and Pearl, J., (1988) Logical and Algorithmic properties of Conditional Independence. Technical Report R-97, Cognitive Systems Laboratory, University of California, Los Angeles.

Goldberger, A. S. (1964). Econometric Theory. Wiley, New York.

Haavelmo, T., (1943) The statistical implications of a system of simultaneous equations, Econometrica, 11, 1-12.

Heise D.(1975). Causal Analysis. Wiley, New York.

Kiiveri, H. and Speed, T., (1982) Structural analysis of multivariate data: A review, Sociological Methodology, Leinhardt, S. (ed.). Jossey-Bass, San Francisco.

Kiiveri, H., Speed, T., and Carlin, J., (1984) Recursive causal models, Journal of the Australian Mathematical Society, 36, 30-52.

Koster, J., (1995) Markov Properties of Non-Recursive Causal Models, To appear in the Annals of Statistics, November 1995.

Lauritzen, S., and Spiegelhalter, D., (1988) Local computation with probabilities in graphical structures and their applications to expert systems, Journal of the Royal Statistical Society B, vol. 50, No. 2.

Lauritzen, S., Dawid, A., Larsen, B., Leimer, H., (1990) Independence properties of directed Markov fields, Networks, 20, 491-505.

Mason, S., (1953) Feedback theory-some properties of signal flow graphs, Proceedings of the IRE, 41.

Mason, S., (1956) Feedback theory-further properties of signal flow graphs, Proceedings of the IRE, 44.

Pearl, J., (1986) Fusion, propagation, and structuring in belief networks, Artificial Intelligence 29, 241-88.

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

Pearl, J. and Verma, T. (1991) A theory of inferred causation, in Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference (Morgan Kaufmann, San Mateo, CA).

Richardson, T. (1994a). Equivalence in Non-Recursive Structural Equation Models. Proceedings:Compstat 94 , Physica Verlag.

Richardson, T. (1994b). Properties of Cyclic Graphical Models. MS Thesis, Carnegie Mellon University.

Richardson T.(1995). A polynomial-Time Algorithm for Deciding Markov Equivalence of Directed Cyclic Graphical Models, Technical Report CMU-PHIL-63, Philosophy Department, Carnegie Mellon University.

Sosa, E. (1975). Causation and Conditionals (Oxford University Press, London, England).

Spirtes, P. (1993) Directed Cyclic graphs, Conditional Independence and Non-Recursive Linear Structural Equation Models. Philosophy, Methodology and Logic Technical Report 35, Carnegie Mellon University.

Spirtes, P. and Glymour, C., (1990) Causal Structure Among Measured Variables Preserved with Unmeasured Variables. Technical Report CMU-LCL-90-5, Laboratory for Computational Linguistics, Carnegie Mellon University.

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

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

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

Spirtes, P. Directed Cyclic Graphical Representation of Feedback Models, in Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, ed. by Philippe Besnard and Steve Hanks, Morgan Kaufmann Publishers, Inc., San Mateo, 1995.

Spirtes, P., Meek, C., and Richardson, T.,(1996) Causal Inference in the Presence of Selection Bias, Technical Report CMU-PHIL-64, Philosophy Department, Carnegie Mellon University.

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

Verma, T. & Pearl, J., (1990). On Equivalence of Causal Models. Technical Report R-150, Cognitive Systems Lab., UCLA.

Wermuth, N., (1980) Linear recursive equations, covariance selection and path analysis, Journal of the American Statistical Association, 75, 963-972.

Wermuth, N. and Lauritzen, S., (1983) Graphical and recursive models for contingency tables, Biometrika, 72, 537-552.

Wermuth, N. and Lauritzen, S., (1990) On substantive research hypotheses, conditional independence graphs and graphical chain models, Journal of the Royal Statistical Society, Series B, 52, 21-50.

Whittaker, J., (1990) Graphical Models in Applied Multivariate Statistics (Wiley, New York).

Wright, S. (1934) The method of path coefficients, Annals of Mathematical Statistics 5, 161-215.