Discovering Cyclic Causal Structure

by

Thomas Richardson

February 1996

Report CMU-PHIL-68

Carnegie Mellon

Philosophy
Methodology
Logic

Pittsburgh, Pennsylvania 15213-3890

Discovering Cyclic Causal Structure

Thomas Richardson[1]
Tetrad Group
Philosophy Department
CMU

Abstract

This paper is concerned with the problem of making causal inferences from observational data, when the underlying causal structure may involve feedback loops. In particular, making causal inferences under the assumption that the causal system which generated the data is linear and that there are no unmeasured common causes (latent variables). Linear causal structures of this type can be represented by non-recursive linear structural equation models.

I 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. A simple modification of the algorithm can be used)as a decision procedure (whose runtime is polynomial in the number of vertices) for determining when two directed graphs (cyclic or acyclic) entail the same set of conditional independence relations.

After proving that the algorithm is correct I then show that it is also complete in the sense that if two linear structural equation models are used as conditional independence *oracles' for the discovery algorithm, then the algorithm will give the same output only if every conditional independence entailed by one model is entailed by the other and vice versa. Another way of saying this is that the algorithm can be used as a decision procedure for determining Markov equivalence of directed cyclic graphs; if the conditional independencies associated with two cyclic graphs result in the same output from the algorithm, when used as input, then the two graphs are equivalent.

[1] thank P. Spirtes, C. Glymour, R. Scheines & C. Meek for helpful conversations. Research supported by NSF grant 9102169.

§1 Linear Feedback and Non-Recursive Structural Equation Models

§1.1 Linear Structural Equation Models

In a Structural Equation Model (SEM) the variables are divided into two disjoint sets: the error variables, and the substantive variables. Associated to each substantive variable V there is a unique error term ε_V. In a *linear* SEM each substantive variable V is written as a linear function of other substantive variables and ε_V. A linear SEM also specifies a joint distribution over the error terms.

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. I 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 (e.g. not infinite). In addition I will consider only linear SEMs with error terms that are jointly independent. This corresponds to the assumption that there are no unmeasured common causes in the structure that generated the data.

Since, in this discussion, I am not concerned with first moments, each variable can be expressed as a deviation from its mean without loss of generality.

The following is an example of a non-recursive linear SEM:

$$X=\varepsilon_X \qquad Y=\varepsilon_Y$$
$$A=\alpha_1 \cdot X+\alpha_2 \cdot B+\varepsilon_A$$
$$B=\beta_1 \cdot Y+\beta_2 \cdot A+\varepsilon_B$$

The ε_V's are jointly independent standard normal error terms.

A structural equation model is said to be *recursive.* if for some ordering of the variables the matrix of coefficients is in lower triangular form.

1.2 Graphs

There is a directed graph, naturally associated with a given linear SEM, by the rule that there is an edge from X to Y (X→Y) if and only if the coefficient of X, in the equation for Y, is non-zero. By convention, error terms are not included in the graph. Hence the graph relating to the model above is (here the error terms are omitted, being assumed jointly independent):

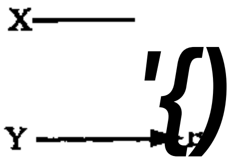


Figure 1

A linear SEM with a jointly independent distribution over the error terms constitutes a parameterization of its associated graph. It is easy to see that the linear SEM associated with an **acyclic** graph will be a recursive structural equation model.

1.3 Linear Entailment

A directed graph containing disjoint sets of variables X, Y, and Z[2] *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 the non-zero linear coefficients and all distributions of the exogenous variables in which they have positive variances and are jointly independent It is important to note that in any particular SEM with directed graph £ there may be conditional independencies which hold even though they are not linearly entailed by *Q*. However, if a zero-correlation holds for some, but not all, parameterizations of *Q,* then the set of parameterizations in which this 'extra' conditional independence holds, is of zero Lebesgue measure over the set of all parameter value assignments to the non-zero linear coefficents.

In the example of the graph given in Figure 1 there are two conditional independence facts that are linearly entailed by the model: X JL Y and X JL Y I {A,B}.[3]

1.4 Conditional Independencies and d-separation.

Verma and Pearl (see Pearl 1988) provided a rule for calculating the conditional independence relations linearly entailed by an acyclic graph *Q*. They showed that a certain 'path' condition, called 'd-connection[1]', held between disjoint sets of vertices in an acyclic graph if and only if that graph linearly entailed a conditional dependence relation between those sets of vertices. The notion of 'd-connection' requires a few preliminary graphical definitions:

Definition: Edge, Parent, Child

An arrow from A to B (A-»B), in a directed graph is called an *edge from A to* B. An arrow from A to B (A->B) or from B to A (B->A) are both called an *edge between A* and B. If there is an edge <A,B> in *Q* then A is a *parent* of B, and B is a *child* of B.

---

[2]**We use bold face letters (X) to denote sets of variables.**

[3]**'X JL Y I Z' means that 'X is independent of Y given Z'.**

Definition: Directed Path

A sequence of **distinct** edges $<E_u...E_n>$ **in** Q **is a** ***directed*** *path* **P** from Vi to V_n+i if and **only** if there exists a sequences of vertices <Vi,... V_n+i> s.t. for $1 \leq i \leq n$, <Vi,Vi_{+1}> = *El* In this case P is said to be a directed path *from* V_h*to* V_{n+1}.

Definition: Ancestor and Descendant

If there is a directed path from A to B, or A = B, **then** A is said to be **an** ancestor of B, and B is said to be a descendant of B. (Thus 'ancestor' is the transitive, reflexive closure of the 'parent' relation, and likewise with 'descendant' and 'child')

Definition: Undirected Path

A sequence of distinct edges <*E*\,...,E_n> in *Q* is an *undirected path* if and only if there exists a sequences of vertices <V_1,...V_n.|-i> s.t. for $1 \leq i \leq n$ either <Vi+!,Vj> = *E*\ or **<V_i,V_{i+1}> = E_i.**

Definition: Collider (Non-collider) Relative to Edges or a Path.

Given three vertices A, B and C such that there is an edge between A and B, and between B and C, then if the edges 'collide' at B, B is said to be a *collider* between A and C, *relative to these edges* i.e. A-*B<- C.

Definition: d-connection

In a graph *Q*, an undirected path U between distinct vertices X and Y not in some set Z, *d-connects* X and Y, given Z, if and only if (i) every collider on U has a descendant in Z, and (ii) any vertex U on U in Z is a collider on U.

For disjoint sets of vertices X, Y and Z, if there is a path which d-connects some vertex X e X and *Ye* Y given Z, then X and Y are said to be d-connected given Z. If there is no such path which d-connects a vertex Xe X with some vertex *Yety* given Z, then X and Y are said to be d-separated by Z.

Verma and Pearl showed that d-separation characterized precisely the independence and conditional independence relations linearly entailed by an acyclic graph:

Theorem (Verma and Pearl, 1990): In an **acyclic** graph *Q*, for disjoint vertex sets X, Y and Z in the graph, X is d-separated from Y by Z if and only if *Q* linearly entails that X JL Y I Z.

Subsequently Spirtes (1995), building upon an idea in Haavelmo (1943) showed that this relation, d-separation, also characterized the independencies that are linearly entailed by a cyclic graph (a similar result was proved independently by Koster (1994)):

Theorem (Spirtes): In a (cyclic or acyclic) graph Q_9 for disjoint sets of vertices, X, Y, Z, X and Y are d-separated given Z, if and only if *Q* linearly entails X JL **Y** I Z.

Spirtes has further shown that d-separation corresponds to vanishing zero partial correlations:

Theorem (Spirtes): In a linear SEM $\mathcal{L}$ with jointly independent error variables and (cyclic or acyclic) directed graph $\mathcal{G}$ 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 $\mathcal{G}$ if and only if $\mathcal{L}$ linearly entails that $\rho_{XY.\mathbf{Z}} = 0$.

§2 Discovery

§2.1 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 variables (i.e. there may be unmeasured variables that are the direct cause of at least two measured variables.) Here I consider the problem of discovering features of the causal graph under the assumption that it may be cyclic, but there are no latent variables. Future research is needed on the problem of discovering the causal graph when it may be cyclic *and* there may be latent variables.

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

§2.2 The Causal Independence Assumption

The most fundamental assumption relating causality and probability that I 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 uncorrelated.

This assumption makes it possible 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 it does not. If A and B are correlated in the experimental population, the experimenter concludes that A does cause B, which is an application of the Causal Independence Assumption.

Since in this paper I consider only linear SEMs without correlated errors or latent variables, it follows from the Causal Independence Assumption that if A and B are correlated then either A causes B, or B causes A (or both). It then follows from the Theorem of Spirtes cited earlier that whether a partial correlation is linearly entailed to be zero by a given SEM can be determined by applying d-separation to the associated graph.

§2.3 The Faithfulness Assumption

In addition to the zero partial correlations that are entailed for *all* linear parameterizations of a graph, there may be zero partial correlations that hold only for some *particular* parameterizations of a graph. For example, suppose Figure 2 is the directed graph of a SEM that describes the relations among Tax Rate, the Economy, and Tax Revenues.

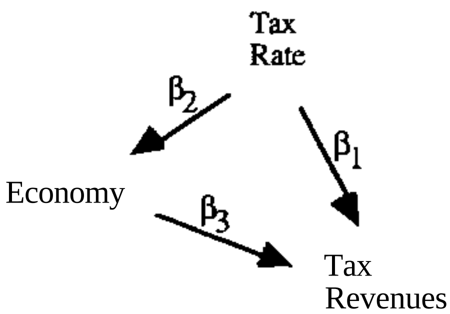


Figure 2. Economic Model

In this case there are no vanishing partial correlation constraints entailed for all values of the free parameters. But if $Pi = -(p_2 \text{ x } p_3)$, then Tax Rate and Tax Revenues are uncorrelated. The SEM postulates a direct effect of Tax Rate on Revenue (pi), and an indirect effect through the Economy (P_2 x p_3). The parameter constraint indicates that these effects *exactly* offset each other, leaving no total effect whatsoever. In such a case the population is said to be **unfaithful** to the graph of the causal structure that generated it. A distribution is **faithful** to a directed graph *Q,* if each partial correlation that is zero in the distribution is entailed to be zero by *(j.*

Causal Faithfulness Assumption: If the directed graph *Q* associated with a SEM correctly describes the causal structure in the population, then each partial correlation that is zero in the population distribution is entailed to be zero by *Q.*

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 $\rho_{AB \cdot C} \neq 0$, (because **it is not linearly entailed to equal** zero for **all** values of the free parameters.) 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 graph, the set of linear parameterizations of the graph that lead to violations of linear faithfulness are Lebesgue measure zero in the space of all parameterizations. 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 3. 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 *Q*. So, if it is possible to perform statistical tests of zero partial correlations, then this information can be exploited to draw conclusions about the d-separation relations in g_y and then to reconstruct as much information about *Q* as possible. Henceforth I will speak of reconstructing features of *Q* from d-separation relations, and from zero partial correlation interchangeably, since given my assumptions, these are equivalent.

Of course the number of distinct d-separation relations grows exponentially with the number of variables in the 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 described in the next section chooses the subset of d-separation relations that it needs to reconstruct features of *g* as it goes along. Therefore I assume that it 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 Causal Independence and Causal Faithfulness assumptions with respect to causal graph *Q*. (The algorithm is correct for any distribution for which a d-separation oracle is available; however, the only case for which such an oracle is known, when the graph is cyclic, is the linear case.)

§2.4 Output Representation - Partial Ancestral Graphs (PAGs)

In general, it is not possible to reconstruct a unique graph *Q* given information only about its d-separation relations, because there may be more than one graph that contains exactly the same set of d-separation relations. Thus directed (cyclic or acyclic) graphs can be partitioned into d-separation equivalence classes:

Definition: Equiv(^)
Two directed graphs *Q*, *Q* * are said to be *equivalent* if they both linearly entail the same set of independencies and conditional independencies. The set of directed graphs equivalent to a given graph £7 is denoted by *Equiv(Q)*.

Richardson(1994b, 1995) presents a polynomial-time algorithm for determining when two graphs are d-separation equivalent to each other; a simpler algorithm is presented in Section 4. (Note that there is a stronger sense of equivalence, linear statistical equivalence between two graphs, which holds when every distribution described by a linear parameterization of one graph can also be described by a linear parameterization of the other graph, and vice-versa. In the acyclic case it is known that^d-separation equivalence is equivalent to linear statistical equivalence, but it is not known if this is so for cyclic graphs.)

The members of Equiv(^) always have certain features in common. I now introduce the formalism with which the features common to all graphs in **Equiv(^),** for some fixed *Q*^ will be represented. A PAG is an extended graph, consisting of a set of vertices V, a set of edges between vertices, and a set of edge-endpoints, two for each edge, drawn from the set {o, -, >}. In addition, pairs of edge endpoints may be connected by underlining, or dotted underlining. In the following definition, which provides a semantics for PAGs, '*' is used as a meta-symbol indicating the presence of any one of {o, -, >}.

Definition: **Partial Ancestral Graph (PAG)**

Ψ is a PAG for Directed Cyclic Graph $\mathcal{G}$ 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 $\mathcal{G}$ given any subset $\mathbf{W} \subseteq \mathbf{V}\backslash\{A,B\}$.

(ii) If there is an edge in Ψ, A—* B, out of A (not necessarily into B), then in every graph in **Equiv**($\mathcal{G}$), A is an ancestor of B .

(iii) If there is an edge in Ψ, A*–>B, into B, then in every graph in **Equiv**($\mathcal{G}$), B is **not** an ancestor of A.

(iv) If there is an underlining A*—<u>*B*</u>—*C in Ψ then B is an ancestor of (at least one of) A or C in every graph in **Equiv**($\mathcal{G}$).

(v) If there is an edge from A to B, and from C to B, (A→B←C), then the arrow heads at B are joined by dotted underlining, thus A—>B<—C, only if in every graph in **Equiv**($\mathcal{G}$) 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—*.

Observe that condition (i) differs from the other five conditions in providing necessary *and* sufficient condtions on **Equiv**($\mathcal{G}$) 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 graph $\mathcal{G}$. Although they all have the same edges, they do not necessarily have the same endpoints. Some of the PAGs provide more information than others about causal structure, e.g. they have fewer 'o's at the end of edges.[4] Some PAGs (providing less information) represent graphs from different Markov equivalence classes. However, the PAGs output by the discovery algorithm I present, provide sufficient information so as to ensure that graphs with the features described by a particular PAG all lie in one d-separation equivalence class.

Since every clause in the definition refers only to **Equiv**($\mathcal{G}$), it follows that if Ψ is a PAG for Directed Cyclic Graph $\mathcal{G}$, and $\mathcal{G}^* \in$ **Equiv**($\mathcal{G}$), then Ψ is also a PAG for $\mathcal{G}^*$. This is not surprising since, as the output of the discovery algorithm, the PAG is designed to represent features common to all graphs in the equivalence class. Hence a PAG Ψ produced by the algorithm represents a unique d-separation equivalence class.

As a consequence, this shows that the set of features described by a PAG is rich enough to enable us to distinguish between any two equivalence classes i.e. there is some feature common to all graphs in one equivalence class that is not true of all graphs in the other

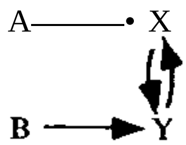
[4]If one PAG has a '>' at the end of an edge, then every other PAG for the same graph either has a '>' or a 'o' in that location. Similarly if one PAG has a '–' at the end of an edge then every other PAG either has a '–' or an 'o' in that location.

equivalence class, **and this difference can be expressed by a difference in the PAGs** representing those equivalence classes.

Example:

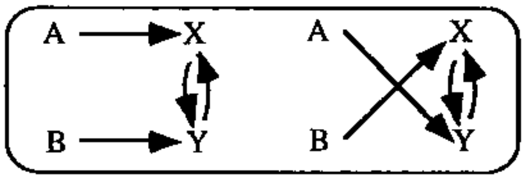
Suppose *Q* is as follows:

G



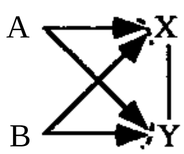
In this case it can be shown that **Equiv(^)** includes (only) two graphs:

Equiv(0



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

PAG for *Q* given by algorithm



Observe that the PAG tells us the following facts about **Equiv(^):**[5]

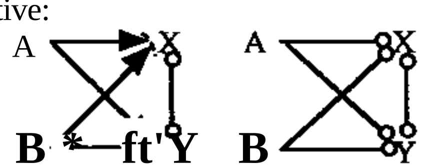
(a) X is an ancestor of Y, and Y is an ancestor of X in every graph in **Equi**v(£).

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

(c) In every graph in **Equiv(^)** both A and B are ancestors of^C and Y.

Note that not every edge in the PAG appears in every graph in **Equiv(^).** 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 graph represented by the PAG in which that edge is present.[6] This example is atypical in that the PAG given by the algorithm contains no V endpoints; however, it shows how much information a PAG may provide. Notice that the following are also PAGs for *Q* though they are less informative:

Other PAGs for *Q*



[5]This is not an exhaustive list. e.g. 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(^)** are both of these edges present. Likewise with the dotted line connecting the arrowheads of the B-»Y, and A—>Y edges.

[6]See previous footnote.

The algorithm I present does not always give the most informative PAG for a given graph *Q* in that there may be features common to all graphs in the 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[1] in the sense that if the conditional independence oracles for two different graphs cause the algorithm to produce the same PAG as output, then the two graphs are d-separation equivalent, i.e. the oracles always give the same answer. This has the consequence that any extensions of the CCD algorithm, say to make it produce a more informative PAG, will not need to make any further consultations of the d-separation oracle; the answer to any potential query can already be inferred (in polynomial time) by lookng at the PAG that the algorithm outputs.

§3 Discovery Algorithm for Cyclic Graphs (Without Latents)

Two minor definitions are required to state the algorithm:

Definiton: p-adjacent in a PAG

Two vertices, X and Y in a PAG are *p-adjacent* if there is an edge between them, X — Yin the PAG.[7]

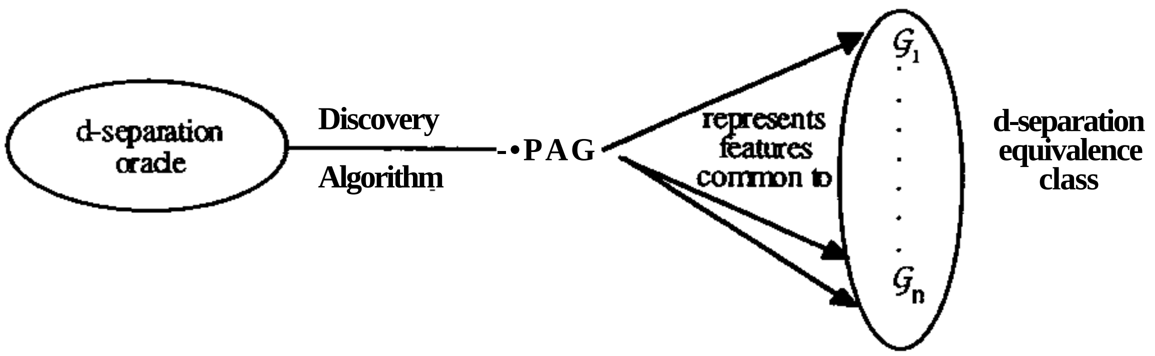
Definition: Adjacencies0F,X)

For PAG *F, AdjacenciesCF,X) is a function giving the set of variables Y s.t. there is an edge X—Y in *F.

*F is a dynamic object in the algorithm that changes as the algorithm progresses, and hence AdjacenciesCF,X) also changes as the algorithm progresses.

The Cyclic Causal Discovery (CCD) Algorithm

The overall strategy for discovery is shown below :



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

CCD Algorithm

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

Output: A PAG for *Q*.

¶A a) Form the complete undirected PAG VF, i.e. for each pair of variables A and B, *P contains the edge A o—o B.

b) n = 0.

repeat

repeat

select an ordered pair of variables X and Y that are p-adjacent in *F such that the number of vertices in AdjacenciesOF,X)\{ Y} is greater than or equal to n;

repeat

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

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

until every subset S of AdjacenciesOF,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 AdjacenciesCF,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, AdjacenciesOF,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 *¥ but the pair A, C are not p-adjacent in XP, orient A*—*B*—*C as A—>B<—C if and only if B is not in Sepset<A,C>; orient A*—*B*—*C as A*—*B*—*C if and only if B is 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 ∉ **Sepset**<A,Y>

(i) If **Sepset** < A,Y > $\subsetneq$ **Sepset** < A,X > then orient X o—*Y as X<—Y

(ii) Else if **Sepset**<A,X> is not a subset of **Sepset**<A,Y>, then orient X o—*Y as X<—Y if A and X are d-connected given **Sepset**<A,Y>

¶**D**. For each vertex V in Ψ form the following set: X∈ **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 an ordered triple <A,B,C> such that A—>B<—C but A and C are not p-adjacent, and **Local**(Ψ,A)\(**Sepset**<A,C> ∪ {B,C}) has greater than or equal to m vertices.

repeat

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

until every subset **T** ⊆ **Local**(Ψ,A)\(**Sepset**<A,C> ∪ {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>.

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> ∪ {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> ∪ {B}) has fewer than m vertices.

1fE. If there is a quadruple <A,B,C,D> of distinct vertices such that

(i) A—>B<—C in ¥

(ii) A—>D<—C or A—>D<—C in *¥

(iii) B and D are p-adjacent in *F

then orient B*-oD as B—>D in 4* if D is not in SupSepset<A,B,O

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

1fF. For each quadruple <A,B,C,D> of distinct vertices such that

(i) A—>B<—C in *V*

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

if A and D are d-connected given SupSepset<A,B,C> u {D}, then orient B*—°D as B—>D in ¥

§3.2 Soundness and Completeness

Theorem 1 (Soundness)
Given as input an oracle for d-separation relations in the (cyclic or acyclic) graph *Q,* then the output is a PAG *¥ for *Q.*

Theorem 1 is proved by showing that each section of the algorithm makes correct inferences from the answers of the d-separation oracle applied to *Q,* The proof is given in Section 5.

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, I will test the sensitivity of the algorithm to these factors on simulated data.

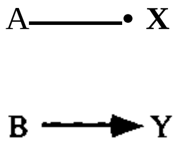
Theorem 2 (d-separation Completeness)

If the CCD algorithm, when given as input d-separation oracles for the graphs £1, £2 produces as output PAGs *Fi, ¥2 respectively, then 4^ is identical to ¥2 if and only if *Qi* and *Q2* are d-separation equivalent, i.e. £2 G **Equiv(^)** and vice versa.

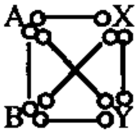
The proof is based on the characterization of equivalence, mentioned earlier, in Richardson (1994b). (It follows directly from Theorem 1 that if *Q* and *Q2* are equivalent then *Fi is identical to Ψ'2.)

§3.3 Trace of CCD **Algorithm**

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

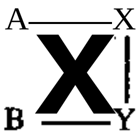


Initial Complete Undirected PAG *F:



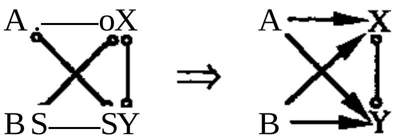
Section 1JA:

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> = 0. This is the only pair of vertices that are not p-adjacent in this graph hence after flA *F is as follows:



Section 1JB

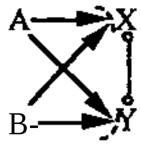
Since Xg Sepset<A,B> and Ye Sepset<A,B>, Ao-oXo-oB and Ao-oYo-oB are oriented respectively as A->X<-B and A->Y<-B. Thus flB performs the following orientation:



Section flC - Performs no orientations in this case.

Section 1JD

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. Thus after flD, the PAG Y is as foUows:

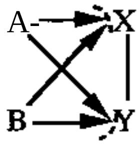


SectionfflE

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 flE. Since Ye SupSepset<A,X,B>, the edge Xo-oY is oriented as Y—oX. Since Xe SupSepset<A,Y,B>, this edge is further oriented as Y—X.

Section 5fF - Performs no orientations in this case, hence the PAG that is output is:

PAGY



§3.4 Complexity of CCD Algorithm

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

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

The number of d-separation tests performed by Step flA of the CCD algorithm will, in a worst case, be bounded by

$$\text{Total no. of tests of oracle consultations in } \wedge A \leq 2\cdot\binom{n}{2}\sum_{i=0}^{k}\binom{n-2}{i} \leq \frac{(k+1)n^2(n-2)^{k+1}}{k!}.$$

where n = number of vertices in $\mathcal{G}$, k = MaxAdj(^).

Since MaxAdj(^) ≤ (MaxDegree(^))^2, hence with MaxDegree(£)=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 graph in which every pair of vertices in the graph, which are not p-adjacent, are only d-separated given all vertices p-adjacent to one of them.

Step flB performs no additional tests of d-separation

Step flC 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 flD performs is bounded by

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

where m = $\max_{X \in V}$ llLocalCP^QI in flD. Since m ≤ (MaxDegree(^))^2, this step is O(n^{r2+4}). Again this is a loose bound.
Step flE performs no tests of d-separation, while step flF performs at most one test for each quadruple satisfying the conditions. Hence this step is O(n^4), (though in many graphs there may be very few quadruples satisfying all four conditions).

§3.5 **Partial Ancestral Graphs (PAGs) and Parially Oriented Inducing Path Graphs (POIPGs)**

The extended graphs which I introduce here - Partial Ancestral Graphs - use a superset of the set of symbols used by Spirtes[1] Partially Oriented Inducing Path Graphs (POIPGs), (See Spirtes *et aL>* 1993) but the *graphical* interpretation of the orientations given to edges is different.[8] After formulating the PAG construct I conjectured that the output of Spirtes' FCI algorithm, for making causal inference in the presence of latent variables and selection bias (See Spirtes, Meek and Richardson 1995), which is a POIPG, could be interpreted directly as a PAG. Shortly thereafter Spirtes proved this conjecture;[9] he has now adopted the PAG interpretation of the FCI algorithm output.
A direct Corollary of Spirtes[1] result is that PAGs can be used to represent the d-separation equivalence class for directed *acyclic* graphs with *latent* variables.[10] It is an open question whether or not the set of symbols is sufficiently rich to allow us to represent the class of cyclic graphs with latent variables.

§4 d-separation Equivalence

Since the CCD algorithm is d-separation complete, the orientation rules in the algorithm may be used to construct a d-separation equivalence algorithm. Below I present an algorithm that, given as input a directed cyclic or acyclic graph *Q* will produce as output the same PAG that the CCD algorithm outputs given only a d-separation oracle for *Q*. However, this algorithm, unlike the CCD algorithm, runs in time polynomial in the number of vertices, even if MaxDegree(^) is not kept fixed. Thus this algorithm can be used to test for d-separation equivalence of two graphs in polynomial time.

Specifically, whereas in a PAG an arrow head at the A end of an edge A<-*B signifies that A is not an ancestor of B in any graph in the equivalence class, in a POIPG it indicates information about the orientation of what Spirtes calls inducing paths between A and B. Similar differences apply to the significance of the tail end of an edge.

[9]Many of the steps of this proof were proved earlier, see §6.8 of *Causation, Prediction and Search,* Spirtes *etaL,* 1994.

[10]In fact, a subset of the symbols present in PAGs will suffice: the conditional independencies entailed by the A—>B<—C are not entailed by any directed acyclic graph, with or without latent variables.

Definition: Parents(X), Children(X), An(X) and Descendants(X)

Parents(X) = {V I 3X e X, V is a parent of X in £}, Children(X) = {V I 3X e X, V is a child of X in £}, An(X) = {V I 3X *e* X, V is an ancestor of X in *Q}y* and Descendants(X) = {V I 3X G X, and V is a descendant of X in *Q)*

Cyclic PAG-Frotn-Graph Algorithm

Input: Directed Cyclic or Acyclic graph *Q*

Output: The CCD PAG NP for *Q*.

fla Form the complete undirected PAG *F, whch has an edge *o-o* between every pair of vertices in the vertex set V.

For each ordered pair of vertices <A,B> form the following sets:

S_A,B = Children(A) n An({A,B})
T_A,B = (Parents($S_{A,B}$ u {A}) u S_{A3})XDescendants(Children(A)nChildren(B)) u {A,B})
For each ordered pair <A,B>:
If A and B are d-separated given T_{AI}B in *Q* then record $T_{A,B}$ in Sepset<A,B> and Sepset <B,A> and remove the edge Ao-oB from *F.
else if A and B are d-separated given $T_{B,A}$ in *Q* then record TB,$_A$ in Sepset<A,B> and Sepset <B,A> and remove the edge Ao-oB from ^{x}¥.

1fb For each triple of vertices A,B,C such that the pair A, B and the pair B, C are each p-adjacent in *F but the pair A, C are not p-adjacent in ^{V}F, orient A*—*B*—*C as A—>B<—C if and only if B is not in Sepset<A,C>; orient A*-$_r$*B*—*C as A*—*B*—*C if and only if B is in Sepset<A,C>.

^Jc For each triple of vertices <A,X,Y> in *¥ such that

(a) A is not p-adjacent to X or Y in *F

(b) X and Y are p-adjacent in *¥

(c) X « Sepset<A,Y>

Orient X a-*Y as X<—Y if A and X are d-connected given Sepset<A,Y>

^fd For each triple <A,B,C> or <C,B,A> such that A—>B<—C, A and C are not p-adjacent, form the following set:

Q_A,B,C = Children(A) n An({ **A,B,C})**
R_A,B,C = (Parents($Q_{A,B}$,c u {A}) u $Q_{A>B}$,c)XDescendants(Children(A)nChildren(C)) u **{A,C})**
If A and C are d-separated given $R_{A>}$B,C U {B} then orient A—>B<—C as A—>B<—C, and record R_{A},B,C u {B} in SupSepset<A,B,C>.

¶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 ¥,

(iii) B and D are p-adjacent in ^{x}¥,

then orient B*-oD as B—>D in NK if D is not in SupSepset<A,B,C>
else orient B*-oD as B*—D in *F 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 *P

(ii) D is not p-adjacent to both A and C in *F, and

if A and D are d-connected given SupSepset<A,B,C> u {D}, then orient B*—oD as B —>D in *F.

I do not include the proof that this algorithm is correct, but it is very similar to the proof that the CCD algorithm itself is correct. The main difference between the two algorithms lies in the fact the CCD algorithm must search for the Sepset and SupSepset sets, testing many different candidates, whereas the PAG-from-graph algorithm is faced with the much simpler task of constructing these sets, given the graph itself.

Since, by Theorem 2, given two graphs $Q\backslash_y$ *Qi* as input, the CCD algorithm will produce the same PAG as output if and only if Q_x and Q_2 are d-separation equivalent, the algorithm given above provides a procedure for deciding the d-separation equivalence of two directed graphs. Moreover the algorithm is of complexity O(n^7) where n is the number of vertices in the graph. This algorithm is significantly faster than the procedure presented in Richardson (1994b) which was O(n^9).

In addition, if a directed cyclic graph *Q* is provided as input to the PAG-from-graph algorithm, then it is also possible to tell from the execution of the algorithm, whether or not there is a directed acyclic graph that is d-separation equivalent to *Q:* There is a DAG *(f* d-separation equivalent to *Q* if and only if steps flc-flf perform no orientations. This follows from the fact that the combination of d-separation relations that the rules in flc-flf require are not entailed by any DAG (See Richardson 1994, 1994b). _

§5 Proofs

§5.1 Proof of Theorem 1

Theorem 1: (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 *F for *Q*.

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

<u>Sections 1TA-1TB</u>

Lemma 1

Given a PAG *F for graph *Q*, if at least one of the following holds:

(i) X is a parent of Y in *g*, or

(ii) Y is a parent of X in *g*, or

(iii) There is some vertex Z which is a child of both X and Y, such that Z is an ancestor of either X or Y (or both)

then X and Y are p-adjacent in *F, i.e. X and Y are d-connected given any subset SeV\{X,Y} of the other vertices in *g*.

Proof: If (i) holds then the path X->Y d-connects X and Y given any subset SeV\{X,Y}, hence X and Y are p-adjacent in any PAG *F for graph *g*. The case in which (ii) holds is equally trivial: X<—Y is a d-connecting path given any set ScVy{X,Y}.
If (iii) holds there is a common child (Z) of X and Y which is an ancestor of X or Y; therefore either there is a directed path X—>Z—>Ai->...A_n—>Y ($n \geq 0$), or there is a directed path Y—>Z->Ai—»...A_n—>X. Suppose without much loss of generality that it is the former. Let S be an arbitrary subset of the other variables (SeV\{X,Y}). There are two cases to consider:

Case 1: S n {Z, Ai...A_n}^ 0; in this case X-»Z <—Y is a d-connecting path.

Case 2: S n {Z, Ai...A_n}= 0; then X-»Z—>Ai-^...A_n—>Y is a d-connecting path. .\

Lemma 2

In a graph *g*, with vertices V, if the following hold[11]:

(i) X is not a parent of Y in *Q*

(ii) Y is not a parent of X in Q_y and

(iii) there is no vertex Z s.t. Z is a common child of X and Y, and an ancestor of X or Y

---

[11] **i.e. None of the conditions in the antecedent of Lemma 1 hold.**

then for any set **Q**, X and Y are d-separated given **T**, defined as follows:

$S = \text{Children}(X) \cap \text{Ancestors}(\{X,Y\} \cup Q)$

$T = (\text{Parents}(S \cup \{X\}) \cup S) \setminus (\text{Descendants}(\text{Children}(X) \cap \text{Children}(Y)) \cup \{X,Y\})$.

Proof: Every vertex in **S** is an ancestor of X or Y or **Q**. Every vertex in **T** is either a parent of X, a vertex in **S**, or a parent of a vertex in **S**, hence every vertex in **T** is an ancestor of X or Y or **Q**. It is sufficient to prove that if (i), (ii) and (iii) hold then X and Y are d-separated given **T**.

Suppose, on the contrary that there is a path d-connecting X and Y given **T**.
Let W be the first vertex on the path from X to Y. (It follows from (i) and (ii) that W≠Y.)
There are two cases to consider:

Case 1: The path contains X←W...Y.

Subcase A: W is not a descendant of a common child of X and Y.
If W is not a descendant of a common child, then W∈**T** (Since W is a parent of X). Thus since W is a non-collider on the path, the path is not d-connecting given **T**

Subcase B: W is a descendant of a common child of X and Y.
In this case since X is a child of W, it follows that X is a descendant of some common child Z of X and Y. But this is contrary to the assumption that (iii) holds.

Case 2: The path contains X→W... Y.

Subcase A: W is not a descendant of a common child of X and Y.
Let V be the next vertex on the path.

Sub-subcase a: The path contains X→W←V...Y.
If this path is d-connecting then some descendant of W is in **T**, but then some descendant of W is an ancestor of X or Y or **Q**. Hence W is an ancestor of X, Y or **Q**. So if some descendant of W is in **T**, then W is in S. Moreover, since W is (by hypothesis) not a descendant of a common child, V ≠ Y.
Now V is a parent of W, and W∈S. Moreover V is not a descendant of a common child since in that instance W would also be a descendant of a common child, contrary to hypothesis. X ≠ V ≠ Y, so V∈**T**. Thus V occurs as a non-collider, but V ∈ **T**, hence the path fails to d-connect given **T**.

Sub-subcase b: The path contains X→W→V...Y.
If some path X→W→V...Y d-connects given **T** then W is either an ancestor of Y or some vertex in **T**. However if W is an ancestor of some vertex in **T**, then W is an ancestor of X, Y or **Q**, since every vertex in **T** is an ancestor of X, Y or **Q**. Hence W∈S, and thus since W is (by hypothesis) not a descendant of a common child of X and Y, and X ≠ W ≠ Y, W ∈**T**. Since W occurs as a non-collider on this path, it

follows that any path X-*W-»V…Y fails to d-connect given T. (This allows for the possibility that V=Y).

Subcase B: W is a descendant of a common child.

Thus Descendants (W) n T = 0, since descendants of W are also descendants of common children of X and Y and so cannot occur in T.

Since no descendant of W has been conditioned on, if W occurs on a d-connecting path then W is a non-collider. We can show that any other vertex on such a d-connecting path must be a non-collider: Suppose that there is a collider on the path, then take the first collider on the path after W, let us say <A, B, C>, so that the path now takes the form: X—>W-^V-»...-^...->A-->B^-C...Y. Since <A, B, C> is the first collider after V, it follows that B is a descendant of W. But if the path is d-connecting then some descendant of B, say D, has been conditioned on, i.e. De T. But then since D is a descendant of B, and B is a descendant of W, De Descendants(W) which is a contradiction since Descendants (W) n T = 0.

As there are no colliders on the path it follows that W is an ancestor of Y. But then W is a descendant of a common child of X and Y, *and* an ancestor of Y. But this contradicts (iii).

This completes the proof of Lemma 2. .*.

Corollary A

Given a graph *Q,* and PAG *F for *Q,* X and Y are p-adjacent in *¥ if and only if one of the following holds in *Q:*

(i) X is a parent of Y, or

(ii) Y is a parent of X, or

(iii) there is some vertex Z which is a child of both X and Y, such that Z is an ancestor of either X or Y (or both)

Proof: 'If is proved by Lemma 1. 'Only if follows from Lemma 2 with Q=0 by contraposition.

Definition: p-adjacent in a graph *Q*

Corollary A gives necessary and sufficient conditions on a graph *Q* for a pair of vertices to be p-adjacent in any PAG for *Q.* Thus it makes sense to speak of a pair of vertices X, Y being *p-adjacent in graph Q,* where this means that at least one of (i), (ii) and (iii) holds.

It follows from Corollary A that a pair of vertices are p-adjacent in *Q* if and only if they are p-adjacent in every PAG for *Q.* For this reason I will often refer to a pair of variables as p-adjacent without specifying whether I am referring to the graph or the PAG.

Corollary B

In a graph $\mathcal{G}$, if X and Y are d-separated by some set **R**, then X and Y are d-separated by a set **T** in which every vertex is an ancestor of X or Y. Furthermore, either **T** is a subset of the vertices p-adjacent to X or X is an ancestor of Y.

Proof: Since X and Y are d-separated by some set **R**, X and Y are not p-adjacent in $\mathcal{G}$. Apply Lemma 2, with $\mathbf{Q}=\emptyset$. In that case

$S = \text{Children}(X) \cap \text{Ancestors}(\{X,Y\})$

$T = (\text{Parents}(S \cup \{X\}) \cup S) \setminus (\text{Descendants}(\text{Children}(X) \cap \text{Children}(Y)) \cup \{X,Y\})$

It follows from Lemma 2 that X and Y are d-separated given **T**. Every vertex in **S** is an ancestor of X or Y. Every vertex in **T** is either a parent of X, a vertex in S, or a parent of a vertex in S, hence every vertex in **T** is an ancestor of X or Y.

It only remains to show that either **T** is a subset of the vertices p-adjacent to X or X is an ancestor of Y in $\mathcal{G}$. Every vertex in **T** is either a parent of X, a child of X, or a parent V of some vertex C in S, where C is also a child of X. Any vertex in the first two categories is clearly p-adjacent to X. Since C is in S, C is an ancestor of X or Y. 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 3

In a graph $\mathcal{G}$, if A and B are not p-adjacent, then either A and B are d-separated given a set $\mathbf{T_A}$ of vertices p-adjacent to A or by a set $\mathbf{T_B}$ of vertices p-adjacent to B.

Proof: By Corollary B to Lemma 2, if A and B are not p-adjacent then A and B are d-separated given $\mathbf{T_A}$ where:

$S_A = \text{Children}(A) \cap \text{Ancestors}(\{A,B\})$

$T_A = (\text{Parents}(S \cup \{A\}) \cup S) \setminus (\text{Descendants}(\text{Children}(A) \cap \text{Children}(B)) \cup \{A,B\})$,

Case 1: A is not an ancestor of B

From Corollary B to Lemma 2, since A is not an ancestor of B, $\mathbf{T_A} \subseteq \{X \mid X \text{ p-adjacent to } A\}$.

Case 2: B is not an ancestor of A.

It follows again by symmetry that A and B are d-separated given $\mathbf{T_B}$, where $\mathbf{T_B}$ is defined symmetrically to $\mathbf{T_A}$ in Case 1.

Case 3: B is an ancestor of A and A is an ancestor of B.

Now any vertex V in $\mathbf{T_A}$ is either a child of A, a parent of A or a parent of some vertex C in $\mathbf{S_A}$, which is itself a child of A. 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. Any vertex in $\mathbf{S_A}$ is an ancestor of A or B. Since A is an ancestor of B, and B is an ancestor of A, it follows that every vertex in $\mathbf{S_A}$ is an ancestor of A, hence every vertex in T_A is p-adjacent to A. [Note that it is also the case that every vertex in T_B is p-adjacent to B.] /.

Suppose that the input to the algorithm is a d-separation oracle for a directed graph *Q*. To find a set which d-separates some pair of variables A and B in *Q* the algorithm tests subsets of the vertices which are p-adjacent to A in *F, and subsets of vertices which are p-adjacent to B in ¥ to see if they d-separate A and B. Since the vertices which are p-adjacent to A and B in *Q* are at all times a subset of the vertices p-adjacent to A and B in *x*¥[12] it follows from Lemma 3 that step flA is guaranteed to find a set which d-separates A and B, if any set d-separates A and B in *Q*.

Section IIB

The next lemma gives 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 the CCD algorithm).

Lemma 4 Suppose that in a graph *Q*, Y is not an ancestor of X or Z or R. If there is a set S, RczS, such that YeS and every proper subset T s.t. RcTczS, not containing Y, d-connects X and Z then S d-connects X and Z in *Q*.
Proof Let T*= Ancestors({X,Z}uR)nS. Now, R^T*, and T* is a proper subset of S, so by hypothesis there is a d-connecting path, P, conditional on T*. By the definition of a d-connecting path, every element on P is either an ancestor *oij* one of the endpoints, or T*. Moreover, by definition, every element in T* is an ancestor of X or Z or R. Thus every element on the path P is an ancestor of X or Z or R. Since neither Y nor any element in S\T* is an ancestor of X or Z or R, it follows that no vertex in S\T* lies on P. Since T* dS the only way in which P could fail to d-connect given S would be if some element of S\T* lay on the path (every collider active given T* will remain active given S). Hence P still d-connects X and Z given S. /.

Definition: Minimal d-separating Set
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.

---

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

The following corollary is **useful here:**

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

Proof: The corollary follows immediately from Lemma 4, with R = 0 via contraposition. /.

This shows that the unshielded non-collider orientation rule in flB is correct If A and B, and B and C are p-adjacent, but **Sepset(A,C)** contains B, then it follows from the nature of 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 proof which follows makes frequent use of the following Lemma which I state here without proof. (It is an extension to the cyclic case of Lemma 3.3.1 in *Causation, Prediction and Search,* Spirtes *et aL,* 1993. See Richardson(1994) for a proof.) The Lemma gives conditions under which a set of 'short[1] d-connecting paths may be put together to form a single path.

CPS **Lemma 3.3.1+** (Richardson 1994b)

In a directed (cyclic or acyclic) graph *Q* 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 = < A=Xo,...X_{n+}i=B>, such that Vi, 0 ≤ i ≤ n, Xi * Xi+i (the Xi are only *pairwise distinct* i.e. not necessarily distinct).

(b) S ⊆ V\{A,B)

(c) £ is a set of undirected paths such that

(i) for each pair of consecutive vertices in R, Xi and X_{i+1}, there is a unique undirected path in S that d-connects Xi and X*+i given S\{Xi, X*+i}.

(ii) if some vertex Xk in R, is in S, then the paths in % , that contain Xk as an endpoint collide at Xk, (i.e. both paths are directed into Xk).

(iii) if for three vertices Xk-i, Xk, Xk+i occurring in R, the d-connecting paths in S between Xk_i and Xk, and Xk and Xk+i, collide at Xk then Xk has a descendant in S.

THEN there is a path U in *g* that d-connects AsXo and B=X_n+i given S.

The following Lemma shows the correctness of the orientation rule in flB:

Lemma 5: 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 £ S.

Proof: Without loss of generality, let us suppose that B is an ancestor of C. It is sufficient to prove that A and C are d-connected conditional on S. There are two cases **to consider,** depending **upon** whether or not some (proper) descendant of B is in S.

Case 1: Some (proper) descendant of B is in S.

It follows from Lemma 1 and the p-adjacency of A and B, that given any set S, A,B,C£ S, there is a d-connecting path from A to B, and likewise a d-connecting path from B to C, conditional on S. Since some descendant of B is in S, **but** B itself is not in S, it follows by a simple application of Lemma 3.3.1+ that A and C are d-connected, since it does not matter whether or not the path from A to B and the path from B to C collide at B.

Case 2: No descendant of B is in S.

It follows from Lemma 1 that there is a path d-connecting A and B. Since no descendant of B has been conditioned on the directed path B—>...-»C is d-connecting. Since Be S, it follows that by Lemma 3.3.1+ that 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 Bg Sepset<A,C>, then B is not an ancestor of A or C, hence A*—*B*—*C should be oriented as A—>B<—C.

Section ¶C

Lemma 6: In a graph *Q*, 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, add Xg S, then A and X are d-separated given S.

Proof: Let X be an ancestor of Y. Let S be any set such that X and Y are d-connected given S, Xg S, and A and Y are d-separated by S. Suppose, for a contradiction, that A and X are d-connected given S, it then follows that there is a d-connecting path P from A to X. There are now two cases:

Case 1: Some descendant of X is in S.

Since XgS, and some descendant of X is in S, it follows from Lemma 3.3.1+ that the d-connecting path from A to X given S and the d-connecting path fronrX to Y given S can be put together to form a d-connecting path from A to Y given S. This is a contradiction since it was assumed that A and Y were d-separated given S.

Case 2: No descendant of X is in S.

In this case since X is an ancestor of Y, there is a d-connecting directed **path** X-^.. .—»Y. Again, by Lemma 3.3.1+ the d-connecting path from A to X and the d-connecting directed path from X to Y can be put together to form a d-connecting path from A to Y

given S. This is again a contradiction since it was assumed that A and Y were d-separated **given S.**

We **have now** shown that under the conditions in the antecedent, S is a d-separating set for A and X. .\

Lemma 7: Let A, X and Y be three vertices in a graph *Q*, such that X and Y are p-adjacent. If there is a set S such that:

(i) X∉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: 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 6 that X is not an ancestor of Y. /.
.[13].\

Step flC simply applies Lemma 7. 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 *F, and

(iii)XeSepset<A,Y>

flC(i) is justified in the following way. Suppose that Sepset<A,Y> **c** Sepset<A,X>. Recall that the search procedure used in *%A* to find Sepset<A,X> tests every subset of Sepset<A,X> to see if it d-separates A and X, before testing Seij[)set<A,X>. In particular, if Sepset<A,Y> e Sepset<A,X>, then A and X are d-connected given Sepset<A,Y>, so taking S = Sepset<A,Y>, we can apply Lemma 7 to orient X *-*Y as X<—Y.

flC(ii) is justified in the following way. Suppose that A and X are d-connected given **Sepset<A,Y>.** Since X £ **Sepset<A,Y>,** setting S = Sepset<A,Y>, we can again apply Lemma 7 to orient X *-*Y as X<—Y.

The condition in UC(ii) that Sepset<A,X> c Sepset<A,Y> is hot needed to make flC(ii) correct (as evidenced by the fact that it plays no role in the justification of the rule); it is included in order to avoid carrying out a redundant test of d-separation. If **Sepset<A,X>** $\underset{\neq}{\subset}$ **Sepset** < A,Y >, then A and X are not d-connected given Sepset<A,Y>. (This is because Yg **Sepset<A,X>.** Hence X*-*Y will eventually by another application of flC(i) be oriented as X—>Y in the PAG. It follows that X is an

[13] We do not need this last fact to prove the correctness of the algorithm, but we include it since it shows the circumstances under which this inference is possible.

ancestor of Y in *Q*. By Lemma 6, since X is an ancestor of Y in *Q*, A and X are not d-connected given **Sepset<A,Y>.) If Sepset<A,Y> = Sepset<A>X>** then there is no need to test whether A and X are d-connected given **Sepset<A,**Y>, because it is already known that they are not d-connected (by definition of Sepset<A,X>).

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 graphs may agree 'locally' on d-separation relations, but disagree on some d-separation relation between distant variables. (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.)

Section HP

This section searches to find 'extra' d-separating sets for unshielded colliders. In the acyclic case a triple of 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.[14] 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 5 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 be inferred 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 8: If in a graph *Q*, 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: If in a graph *Q*, X and Y are p-adjacent, Y and Z are p-adjacent, but X and Z are not p-adjacent, Y is not an ancestor of X or Z, and there is some set S such that Ye S, and X and Z are d-separated given S, then Y is not a descendant of a common child of X andZ.

[14]This is also true even in the acyclic case with latent variables.

Lemma 9: If in graph $\mathcal{G}$, 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:

$\mathbf{S}$ = Children(X) $\cap$ Ancestors($\{X,Y,Z\}$)

$\mathbf{T}$ = (Parents($\mathbf{S} \cup \{X\}$) $\cup$ $\mathbf{S}$)\(Descendants (Children(X)$\cap$Children(Z)) $\cup$ $\{X,Z\}$)

Further, if X and Y, and Y and Z are p-adjacent then $Y \in \mathbf{T}$.

Proof: It follows from Lemma 2, with $\mathbf{Q}=\{Y\}$ that X and Z are d-separated given $\mathbf{T}$. All that remains is to show that $Y \in \mathbf{T}$. There are three cases to consider here:

Case 1: Y is a child of X.

If Y is a child of X, then since Y is an ancestor of Y, $Y \in \mathbf{S}$. In both cases since Y is not a descendant of a common child of X and Z, $Y \in \mathbf{T}$.

Case 2: Y is a parent of X

Since Y is a parent of X and Y is not a descendant of a common child of X and Z, $Y \in \mathbf{T}$.

Case 3: X and Y have a common child C that is an ancestor of X or Y

Since C is a child of X and an ancestor of X or Y, $C \in \mathbf{S}$. Since Y is a parent of C, and Y is not a descendant of a common child of X and Z then $Y \in \mathbf{T}$. $\therefore$

Lemma 10: If X and Z are d-separated by some set $\mathbf{R}$, then for all sets $\mathbf{Q} \subseteq$ Ancestors($\mathbf{R} \cup \{X,Z\}$)\$\{X,Z\}$, X and Z are d-separated by $\mathbf{R} \cup \mathbf{Q}$.

Proof: Suppose for a contradiction that there is a path $\mathbf{P}$ d-connecting X and Z given $\mathbf{R} \cup \mathbf{Q}$. It follows that every vertex on $\mathbf{P}$ is an ancestor of either X, Z, or $\mathbf{R} \cup \mathbf{Q}$. Since $\mathbf{Q} \subseteq$ Ancestors($\mathbf{R} \cup \{X,Z\}$) it follows that every vertex on $\mathbf{P}$ is an ancestor of X, Z or $\mathbf{R}$. Let A be the collider furthest from X on $\mathbf{P}$ which is an ancestor of X and not $\mathbf{R}$ (or X if no such collider exists), let B be the first collider after A on $\mathbf{P}$ which is an ancestor of Z and not $\mathbf{R}$ (or Z if no such collider exists). Clearly the paths $X \leftarrow \ldots \leftarrow A$, and $B \rightarrow \ldots \rightarrow Z$ are d-connecting given $\mathbf{R}$, since by the definition of A and B, no vertex on these paths is in $\mathbf{R}$. In addition the subpath of $\mathbf{P}$ between A and B is also d-connecting given $\mathbf{R}$ since every collider is an ancestor of $\mathbf{R}$, and no non-collider lies in $\mathbf{R}$, since, by hypothesis $\mathbf{P}$ d-connects given $\mathbf{R} \cup \mathbf{Q}$. It follows, by Lemma 3.3.1+, that there is a path d-connecting X and Z given $\mathbf{R}$. This is a contradiction. $\therefore$

The search in section ¶D considers each triple $A \rightarrow B \leftarrow C$ in Ψ, A and C are not p-adjacent, in turn, and attempts to find a set $\mathbf{R}$ which is a subset of **Local**(Ψ,A)\{C} such that A and C are d-separated given $\mathbf{R} \cup \{B\} \cup$ **Sepset**<A,C>. It follows from Lemma 8 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 9 that in this case there is some subset, the set $\mathbf{T}$ 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 $\mathbf{T} \subseteq$ **Local**(Ψ,X). Since **Sepset**<A,C> is a minimal d-separating set for A and C,

it follows that Sepset<A,C> e Ancestors ({A,C})\{A,C} (c Ancestors (T u{A,C}). Hence by Lemma 10, T u Sepset<A,C> also d-separates A and C.

The reader may wonder why flD tests sets of the form T u Sepset<A,C>, (where **T c LocalCF,A)),** instead of just testing sets of the form **T c LocalCF,A));** Lemma 9 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 10 it follows that any set **T e LocalCF,A)** which d-separates A and C, is such that T u Sepset<A,C> also d-separates A and C, but the reverse is not true. In particular the smallest set T such that T u 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.[15]

One more lemma is required to explain why the algorithm begins the search in flD with m=l, and does not test T=0:

Lemma 11: 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 u(Y).

Proof: Corollary A to Lemma 2 implies that 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, 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 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 3.3.1+ that these paths can be joined to form a single d-connecting path, hence X and Z are d-connected given S u{Y}. /.

This completes the proof that step flD of the algorithm will succeed in findings set which d-separates A and C, and contains B, for each triple A—>B<—C if any such set exists.

[15] In some cases the cardinality of the smallest set (T u Sepset<A,Q>) 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.)

Section HE

The following Lemma provides the justification of flE where A—>B<—C, A—>**D**<—C, and **D** is not in **SupSepset<A,B,C>,** in which case B*-oD is oriented as B—>D.

Lemma 12: If in a **PAG** *W* for *Q*, X—>V<—Z, X—>W<—Z, X and Z are not p-adjacent, and W is an ancestor of V in *Q*, then any set S such that VeS, 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 Ve S, but WgS, it follows by Lemma 3.3.1+ that d-connecting paths from X to W given S and from W to Z given S can be put together 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 Corollary A to Lemma 1) since X is p-adjacent to W and W is p-adjacent to Z. This is a contradiction. /.

Note: In fact the converse to this lemma is also true: If every d-separating set containing V also contains W, then W is an ancestor of V.[16]

Proof of Converse to Lemma 12: It is sufficient to prove that if W is not an ancestor of V then there is some set which d-separates X and Z, but does not contain W.

It follows from Lemma 9 and the assumption that X—>V<—Z in the PAG ^{V}F, that there is some set R containing V which d-separates X and Z. Let S be any subset of R such that Ve S, and S d-separates X and Z, but there is no subset TcS, such that VeT and X and Z are d-separated by T. (Such a set S, is guaranteed to exist.) It follows from Lemma 4, that every vertex in S is an ancestor of X, Z or V. Since by hypothesis X—>W<—Z in the PAG, W is not an ancestor of X or Z. If in addition W is not'4n ancestor of V, then it follows that W £ S. (Since W is not an ancestor of X,Vor Z.)

Thus if W is not an ancestor of V, then there is some set, namely S, which d-separates X and Z, and contains V, but does not contain W. .\

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, this inference can be justified as follows:

If D is in **SupSepset<A,B,C>** then it follows from Lemma 4, and the nature of the search for **SupSepset<A,B,O**[17] that D is an ancestor of {B} u **Sepset<A,C>.** Since Sepset<A,C> is a minimal d-separating set for A and C, moreover, every vertex in **Sepset<A,O** is an ancestor of A or C, thus if D is in **SupSepset<A,B,C>,** then D is an ancestor of **A,C** or B. However, since there are arrowheads at D on the edges from A to

---

[16]The converse is staled separately since it is not required in any of the proof that follows.

[17]Namely the fact that section flD looks for the smallest superset of {B} u Sepset<A,C>, which d-separates A and C.

D, and C to D, it follows that D is not an ancestor of A or C, hence D is an ancestor of B. Thus it is correct to orient B*–*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 follows from the discussion and corollaries following Lemma 8, that since A and C are d-connected by any set **S** that contains D, (and does not contain A or C), 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, 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*–*D should be oriented as B<—D.

Note Since neither **Sepset** nor **SupSepset** are consulted in making this last inference, this case might better be termed a 'propagation rule', rather than an 'orientation rule'.

<u>Section ¶F</u>

A and C are d-separated by **SupSepset**<A,B,C>, and B∈**SupSepset**<A,B,C>. Hence by Lemma 10, if D is an ancestor of B, then A and C are d-separated by **SupSepset**<A,B,C> ∪ {D}. Hence by contraposition, if A and C are d-connected given **SupSepset**<A,B,C> ∪ {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 that B is an ancestor of D. Thus B*—*D should be oriented as B—>D in Ψ.

This completes the proof of the correctness of the CCD algorithm. ∴

§5.2 Proof of Theorem 2: d-separation Completeness

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 $\mathcal{G}_1$ that results in output Ψ_1, and the second input to the CCD algorithm is a d-separation oracle for $\mathcal{G}_2$ that results in output Ψ_2, and Ψ_1 and Ψ_2 are identical, then $\mathcal{G}_1$ and $\mathcal{G}_2$ are d-separation equivalent. I shall do this by proving that when d-separation oracles for $\mathcal{G}_1$ and $\mathcal{G}_2$ are used as input to the CCD algorithm and produce the same PAG as output, then $\mathcal{G}_1$, and $\mathcal{G}_2$ 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 graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are d-separation equivalent to one another if and only if they satisfy these 5 conditions.

A number of extra definitions are required in order to state the Cyclic Equivalence Theorem :

Definition: Unshielded Conductor and Unshielded Non-Conductor

In a (cyclic or acyclic) graph £, a triple of vertices <A,B,C> is said to form an *unshielded conductor* if:

(i) A and B are p-adjacent, B and C are p-adjacent, A and C are not p-adjacent

(ii) B is an ancestor of A or C

If <A,B,C> satisfies (i), but B is not an ancestor of A or C, then <A,B,O is said to be an *unshielded non-conductor.*

Definition: Unshielded Perfect and Imperfect Non-Conductors

In a (cyclic or acylcic) graph *Q,* a triple of vertices <A,B,C> is said to be an *unshielded perfect non-conductor* if:

(i) A and B are p-adjacent, B and C are p-adjacent, but A and C are not p-adjacent.

(ii) B is not an ancestor of A or C.

(iii) B is a descendant of a common child of A and C.

If <A,B>C> satisfies (i) and (ii) but B is not a descendant of a common child of A and C, then <A,B,C> is said to be an *unshielded imperfect non-conductor.*

Definition: Itinerary

If <Xo,Xi,...,X_n+i> is a sequence of distinct vertices s.t. Vi 0 ≤ i ≤ n, *X* and Xj+i are p-adjacent then <Xo,Xi,...X_n+i> is said to be an *itinerary..*[1]*

Definition: Mutually Exclusive Unshielded Conductors with respect to an Itinerary

If <XQ,.. .X_n+i> is an itinerary such that: y;

(i) Vt 1≤ t ≤ n, <X_t-i> X_t, X_t+i> is an unshielded conductor,

(ii) Vk 1≤ k ≤ n, X^-i is an ancestor of X^, and X^+i is an ancestor of X^, and

(iii) Xo is *not* a descendant of Xi, and X_n is *not* an ancestor of X_n+j, then <Xo,Xi,X2> and <X_n_i,X_n,X_n+i> are *mutually exclusive (m.e.) unshielded conductors on the itinerary* <XQ,...X_n+i>.

Definition: Uncovered Itinerary

If <X_0,...,X_n+i> is an itinerary such that Vij 0 ≤i < j-l < j ≤ n+1 Xi and Xj are not p-adjacent in the graph then *<Xo,...X_n+i>* 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.

---

[18]Thus itinerary is to PAG, as undirected path is to directed graph.

Cyclic Equivalence Theorem (Richardson 1994b)

Graphs *Q* and *(ji* are d-separation **equivalent if and only** if **the** following five **condtions** hold:

CET(I) *(j* and *Qi* have the same p-adjacencies.

CET(II) *Qi* and *Qi* have the same unshielded elements i.e.

(Ha) the same unshielded conductors, and

(lib) the same unshielded 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=<A,B,C,...X,Y,Z> in *Qi* if and only if <A,B,C> and <X,Y,Z> are m.e. conductors on some uncovered itinerary Q=<A,B,C,...X,Y,Z>in ^.

CET(IV) If <A,X,B> and <A,Y,B> are unshielded imperfect non-conductors (in *Q* and £2), then X is an ancestor of Y in *Q* if and only if X is an ancestor of Y in £72.

CET(V) If <A,B,C> and <X,Y,Z> are mutually exclusive conductors on some uncovered itinerary P E=<A,B,C,...X,Y,Z> $_{anc}$j <A,M,Z> is an unshielded imperfect non-conductor (in *Q* and *Qi),* then M is a descendant of B in *Q* iff M is a descendant of B in *Qi.*

Lemma 13: Given a sequence of vertices <Xo>—X_{n+}i> in a directed graph £ having the property that Vk, $0 \le k \le n$, Xk is an ancestor of Xk+i, and Xk is p-adjacent to Xk+i there is a subsequence of the Xi's, which we label the Yj's having the following properties:

(a) XQ^YO

(b) Vj, Yj is an ancestor of Yj_{+}i

(c) Vj,k If j < k, Yj and Yk are p-adjacent in the graph if and only if k=j+l. i.e. the only Yk's which are p-adjacent are those that occur consecutively.

Proof: The Yk's can be constructed as follows:

LetYo = X_o.

Let Yfc+i = X-q where T| is the greatest h > j such that Xh is p-adjacent to Xj where Xj=Yk-

Property (a) is immediate from the construction. Property (b) follows from the transitivity of the ancestor relation, and the fact that the Y^'s are a subsequence of the Xi's. It is also clear, from the construction that if k=j+l then Yj and Yk are p-adjacent. Moreover, if Yj=Xa[19] and Yfc=Xp are p-adjacent, and j < k, then it follows again from the construction that if Yj+i=Xy> then p ≤ y, so k ≤ j+1. (Since the Yk's are a subsequence of the Xi's.) Hence Yj+i=Yk..'.

[19] That is, the *f** vertex in the sequence of Y vertices is the *(Xth* vertex in the sequence of X vertices.

Lemma 14: Let ***Q*** and ***Q%*** **be** two **graphs satisfying CET(I)-(III). Suppose there** is a directed path Di—»...D_n, in *Q*. Let Do be a vertex distinct from Di,...,D_n, s.t. Do is p-adjacent to Di in *Qi* and *Qi,* Do is not p-adjacent to D2,...D_n in *Qi* or £2 and Do is not a descendant of Di in *Q* or *Qi.* It then follows that Di is an ancestor of D_n in *Qi.*

Proof: By induction on n.
Base Case: n=2
Since, by hypothesis, Do is not p-adjacent to D2, it follows that <Do, Di, D2> forms an unshielded conductor in *Q* (since Di is an ancestor of D2). Hence this triple of vertices also forms an unshielded conductor in £72, by CET(IIa). Hence Di is an ancestor of Do or D2 in *Cft.* Since, by hypothesis Di is not an ancestor of Do in £2, it follows that Di is an ancestor of D2 in *Cfa.*

Inductive Case: Suppose that the hypothesis is true for paths of length n.
It follows from Lemma 13 that there is a subsequence <D_a(opDo,D_a(i),D_a(2)...D_a($_r$)=D_n> such that the only p-adjacent vertices are those that occur consecutively, and in *Q* each vertex is an ancestor of the next vertex in the sequence. Moreover, since, by hypothesis, Do is not p-adjacent to D2,...D_n, it follows that D_a(i)^Di. Since *Q* and *Q2* satisfy CET(I), they have the same p-adjacencies, hence in *Qi* the only vertices that are p-adjacent are those that occur consecutively in the sequence. Suppose, for a contradiction that D_a($_r$_i) is not an ancestor of D^) in *Cfi.* Let s be the smallest j such that D_a(j) is not an ancestor of D_a(j_!) in *Cft,.* (Such a j exists since D_a(i)=D! and D_a(O)=Do is not a descendant of Di.) It then follows that <D_a($_s$_i),D_a($_s$),D_a($_s$+i)> and <Do(r-2)>Da(r-i)>D_a($_r$)> are mutually exclusive conductors on the unshielded itinerary <D_a($_s$_i),...D_a($_r$)>. But these two triples are not mutually exclusive in *Q* since D_a($_r$_j) is an ancestor of D_a($_r$) in *Q* hence *Q* and *Qi* fail to satisfy CET(III), which is a contradiction.
Hence D_a($_r$_i) is an ancestor of $D_{a(r)}$ in *(fr-* It then follows from the induction hypothesis that Di is an ancestor of D_a($_r$)=D_n. .\

Theorem 2: (d-separation Completeness)
If the CCD algorithm, when given as input d-separation oracles for the graphs *Q*, *Qi* .produces as output PAGs *Fi, $¥_2$ respectively, then ¥1 is identical to ¥2 if and only if *Qi* and *Qi* are d-separation equivalent, i.e. *Qi* ^{e} **Equiv(^)** and vice versa.

Proof: We will show that if two graphs, *Q* and £2 are *not* d-separation equivalent, then the PAG discovered by the CCD algorithm, given *Q* and *Cp,* as input would differ in some respect.

It follows from the cyclic equivalence theorem that if *Q* and *§2* are not d-separation equivalent, then they fail to satisfy one or more of the five conditions CET(I)-(V). Let *Fi

and ¥2 **denote, respectively, the PAGs output by the CCD algorithm when given** ***Qx*** **and** ***Qi*** **as input**

Case 1: *Q* and *Qi* fail to satisfy CET(I)
In this case the two graphs have different p-adjacencies. Let us suppose without loss of generality that there is some pair of variables, X and Y which are p-adjacent in *Qi* and not p-adjacent in *Qi.* Since X and Y are p-adjacent in *Q*, X and Y are d-connected conditional upon any subset of the other vertices. Hence there is an edge between X and **Y in** Ψ_1.
Since X and Y are not p-adjacent in £72, there is some subset S, (X,Y£ S) such that X and Y are d-separated in *Qi* given S. It follows from Lemma 3, that X and Y are d-separated by a set of variables T, such that either T is a subset of the vertices p-adjacent to X, or T is a subset of the vertices p-adjacent to Y. It follows that in step HA of the CCD algorithm the edge between X and Y in *F would be removed. Since edges are not added back in at any later stage of the algorithm, there is no edge in HKQ between X and Y. Hence *Fi and 4*2 are different.

Case 2: *Qi* and *fr* fail to satisfy CET(IIa)
We assume that *Q* and £2 satisfy CET(I). In this case the two graphs have different unshielded non-conductors. Without much loss of generality, suppose that there is some triple of vertices <X,Y,Z> such that in *Q*, Y is an ancestor of X or Z, while in (72 Y is not an ancestor of either X or Z.

If Y is an ancestor of X or Z then it follows from the corollary to Lemma 4 that every set which d-separates X and Z includes Y. Hence Ye Sepset(X,ZA in *Q*. Thus <X,Y,Z> is marked as X*-*Y*-*Z in step flB. It then follows from the correctness of the algorithm that in W_u either X*->Y—*Z, X*—Y<-*Z, or X*-*Y*-*Z.
If Y is not an ancestor of X or Z in *Qi,* then Y is not in any minimal d-separating set for X and Z. In particular Yg **Sepset(X,Z)** for *Qi.* Again it follows from the correctness of the algorithm that <X,Y,Z> is oriented as X*->Y<-*Z, or X*->Y<-*Z in $*F_2$- Thus *¥\ and Ψ_2 are different.

Case 3: *Qi* and *Qi* fail to satisfy CET(IIb)
We assume that *Q* and *Q2* satisfy CET(I), CET(IIa). In this case the two graphs have different unshielded imperfect non-conductors, i.e. there is some triple <X,Y,Z> such that it forms an unshielded non-conductor in both *Q* and *Q2-,* but in one graph Y is a descendant of a common child of X and Z, while in the other graph it is not. Let us assume that Y is a descendant of a common child of X and Z in *Q*, while in (72 it is not

It follows from Lemma 8 that in $Q\backslash_9$ X and Z are d-connected given any subset containing Y. In this case the search in CCD section flD will fail to find any set SupsepseKX,Y,Z>. Hence in *Fi **<X,Y,Z> will** be **oriented** as **X*->Y<-*Z** (i.e. **without dotted underlining).** If Y is not a descendant of a common child of X and Z in £72, then it follows from Lemma 9 and Lemma 10, that there is some subset **T** of **LocalCP2,X),** such that X and Z are d-separated given T u (Y) u **Sepset<X,Z>.** Section flD will find such a set T, and hence <X,Y,Z> will be oriented as X*-≥Y≤-*Z in ¥2. Since no subsequent orientation rule removes or adds dotted underlining, it follows immediately that *Fi and ¥2 are different.

Case 4: *Qi* and *(Ji* fail to satisfy CET(III)
We assume that Q_x and Q_2 satisfy CET(I), CET(IIa), CET(IIb). In this case the two graphs have the same p-adjacencies, and unshielded conductors, perfect non-conductors, and imperfect non-conductors. However, the two graphs have different mutually exclusive conductors. Hence in both *Q* and *Qi* there is an uncovered itinerary, <Xo,...X_n+i> such that every triple <Xk-i,Xk,Xk+i> (l≤k≤n) on this itinerary is a conductor, but in one graph <XQ,XI,X2> and <X_{n}_i,X_n,X_{n+}i> are mutually exclusive, i.e. Xi is not an ancestor of Xo, and X_n is not an ancestor of X_n+i, while in the other they are not mutually exclusive. Let us suppose without loss of generality that <Xo,Xi,X2> and <X_{n}_i,X_n,X_{n+}i> are mutually exclusive in $Q\backslash_9$ while in £72 they are not.
It follows from the definition of m.e. conductors that in *Q*, the vertices Xi,...X_n, inclusive are *not* ancestors of Xo or X_n+i. Hence {Xi,...X_n} n Sepset(Xo,X_n+i) = 0, since Sepset(Xo,X_n+i) is minimal, and so is a subset of Ancestors(Xo,X_{n+}i). (Here we refer to Sepset(Xo,X_n+i) calculated for *Q\.)* For the same reason Descendants({Xi,..., X_n}) ^ Sepset(Xo,X_n+i) = 0. It follows from the definition of m.e. conductors on an itinerary that X^ is an ancestor of X^+i (1 ≤ k < n), thus there is a directed path *P^=* Xfc—»...—>Xj_{c+}i. Since no descendant of Xi,...X_n is in Sepset(Xo,X_n+i), each of the directed paths Pk d-connects each vertex Xk to its successor Xk+i (1 ≤ k < n), conditional on Sepset(Xo,X_n+i). In addition, since *XQ* and Xi are p-adjacent there is some path Q d-connecting Xo and Xi given Sepset(Xo,X_n+i). Since each Pi is out of Xi (i.e. the path goes Xi->...-»X0, by applying Lemma 3.3.1+, with 3:={Q,Pi,...P_n}, R = <X_0>.-.X_n>, S = Sepset(Xo>Xn+i) a path d-connecting Xo and X_n given Sepset(Xo,X_n+i) can be constructed. A symmetric argument shows that Xi and X_n+i are also d-connected given Sepset(Xo,X_n+i). It then follows that the edges Xo*—*Xi and X_n*—*X_n+i are oriented as X_o*—>Xi and X_n<—*X_{n+1} in ^ by stage flC 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 *Fi. (Subsequent stages of the algorithm only add '-' and V endpoints, not V endpoints. If either of the arrowhead at Xi or X_n were replaced with a '-' the algorithm would be incorrect.)
Since by hypothesis, <Xo>Xi,X_2> and <X_n-i,X_n,X_{n+}i> are not mutually exclusive in £2, either Xi is an ancestor of Xo, or X_n is an ancestor of X_{n+i}, it follows from the

correctness of the orientation rules in the CCD algorithm that the edges Xo*—*Xi and X_n*—*X_n+i will not both be oriented as Xo*—>Xi **and** X_n<—*X_n+i **in *F_2.** Thus Yi and ^2 will once again be different.

Case 5: *Qi* and *Qi* fail to satisfy either CET(TV) or CET(V)

We assume that Q_x and g_2 satisfy CET(I)-(HI).[20] If *Qx* and Q_2 fail to satisfy either CET(IV) or CET(V), then in either case we have the following situtation: There is some sequence of vertices in *Q* and *'(ji*<Xo,Xi,...X_n, X_n+i>, [21] satisfying the following:

(a) if i > j then X* and Xj are p-adjacent if and only if i=j+1,

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

(c)Vk, 1 ≤ k ≤ n, Xk-u and X^+i are ancestors of X^.

In addition there is some vertex V, p-adjacent to Xo and X_n+i in *Q* and Q_2 , not an ancestor of XQ or X_n+i in *Q* or *Qi* and not a descendant of a common child of XQ and X_n+i in *Q* or *Cp,*. As explained in Case 3, this implies in both of the PAGs *Fi and 4'2> Xo—>y<—x_{n+}i.

Since *Qi* and *Qi* fail to satisfy CET(IV) or CET(V), in one graph V is a descendant of $X_{1?}$ while in the other graph V is not a descendant of Xi. Let us suppose without loss of generality that V is a descendant of Xi in *Q* , and V is not a descendant of Xi in *Qi.* As in previous cases it is sufficient to show that if *Fi and 4*2 are the CCD PAGs corresponding to *Q* and *Q2* respectively, then *Fi and ^2 are different. We may suppose, again without loss of generality that V is the closest such vertex to any X^ (1≤ k ≤ n) in *Qi,* in the sense that the shortest directed path PsX^—>...-»V in *Q* contains at most the same number of vertices as the shortest directed path in *Q* from any X^ (l≤ k ≤ n) to some other vertex V satisfying the conditions on V.

V

Claim: Let W be the first vertex on P which is p-adjacent to V, (both in *Q* and *Qi* since by CET(I) *(J* and £72 have the same p-adjacencies). We will show that the assumption that V is the closest such vertex to any X^ (in *Qi)* together with the assumption that *Q* and *Q2* satisfy CET(I)-(III) imply that W is a descendant of Xi in *Qi.* We prove this by showing that every vertex in the directed subpath P(Xk, W)=Xk—>...W in *Q* is also a descendant of Xi in <#.

Proof of Claim: By induction on the vertices occuring on the path **P(Xk, W).**

Base: X^. By hypothesis X^ is a descendant of Xi in both *Q* and £72.

---

[20] 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.

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

Inductive Case: Y_r; where $P(X_k, W) \equiv X_k \to Y_1 \to \ldots \to Y_r \to \ldots Y_t \equiv W$. **By the induction hypothesis, for** $s < r$, Y_s **is a descendant of** X_1 **in** $\mathcal{G}_2$. **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 $\mathcal{G}_1$ there is a directed path $X_0 \to \ldots X_k \to Y_1 \to \ldots Y_r$, by Lemma 13 it then follows that there is some subsequence of this sequence of vertices, $Q \equiv \langle X_0, \ldots Y_r \rangle$ such that consecutive vertices in Q are p-adjacent, but only these vertices 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. $Q \equiv \langle X_0, D \ldots Y_r \rangle$ where D is the first vertex in the subsequence after X_0, hence either $D \equiv X_\kappa$ $(1 \le \kappa \le k)$ or $D \equiv Y_\mu$, $(1 \le \mu < r)$. Since in either case D is a descendant of X_1 in both $\mathcal{G}_1$ and $\mathcal{G}_2$, (either by the inductive hypothesis or by the hypothesis of case 5), but X_0 is not a descendant of X_1 in $\mathcal{G}_1$ or $\mathcal{G}_2$ it follows that D is not an ancestor of X_0 in $\mathcal{G}_1$ or $\mathcal{G}_2$. Hence we may apply Lemma 14, to deduce that Y_r is a descendant of D. Hence Y_r is a descendant of X_1, 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 $\mathcal{G}_1$ 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 $\mathcal{G}_1$. Moreover, since X_0 and X_{n+1} are not p-adjacent, $\langle X_0, Y_r, X_{n+1} \rangle$ forms an unshielded non-conductor in $\mathcal{G}_1$. Hence $\langle X_0, Y_r, X_{n+1} \rangle$ forms an unshielded non-conductor in $\mathcal{G}_2$, since by hypothesis $\mathcal{G}_1$ and $\mathcal{G}_2$ satisfy CET(IIa). So Y_r is not an ancestor of X_0 or X_{n+1} in $\mathcal{G}_1$ or $\mathcal{G}_2$. Further, since Y_r is an ancestor of V in $\mathcal{G}_1$ and V is not a descendant of a common child of X_0 and X_{n+1} in $\mathcal{G}_1$, it follows that Y_r is not a descendant of a common child of X_0 and X_{n+1} in $\mathcal{G}_1$. Thus $\langle X_0, Y_r, X_{n+1} \rangle$ forms an unshielded imperfect non-conductor in $\mathcal{G}_1$, hence also in $\mathcal{G}_2$, since $\mathcal{G}_1$ and $\mathcal{G}_2$ satisfy CET(I),(IIa) and (IIb).[22] Now, if Yr were not a descendant of X_1 in $\mathcal{G}_2$, then Y_r would satisfy the conditions on V, yet be closer to X_k than V (Y_r occurs before V on the shortest directed path from X_k to V in $\mathcal{G}_1$). This is a contradiction, hence Y_r is a descendant of X_k in $\mathcal{G}_2$.

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 $\mathcal{G}_1$, W is an ancestor of V, hence it follows from the correctness of the algorithm in Ψ_1 this edge is oriented as Wo–*V or W—*V.

---

[22]If $\mathcal{G}_1$ and $\mathcal{G}_2$ satisfy CET(I), CET(IIa), CET(IIb) then they have the same p-adjacencies, the same unshielded conductors and the same unshielded perfect non-conductors. Thus all other unshielded triples which are not conductors or perfect non-conductors are imperfect non-conductors. Hence $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same imperfect non-conductors.

In ⪡j2, however, since Xi is not an ancestor of V, but, as we have just shown Xi is an ancestor of W, it follows that W is not an ancestor of V. There are now two cases to consider:

Subcase 1: n=l, and W=Xi

In this case Xo—>Xi<—X2, in *F_2 (and ¥1). Supsepset(Xo,V,X2) is the smallest set containing Sepset(Xo,X2)u{ V} which d-separates Xo and X2, in the sense that no subset of Supsepset (Xo,V,X2) which contains Sepset(Xo,X_2) u{V} d-separates Xo and X_2. It follows from Lemma 4 (with R =Sepset(Xo,X_2) u {V}) that every vertex in Supsepset(X_0,V,X_2) is an ancestor of X_o, X_2 or Sepset(X_0,X_2) u{V}. Since every vertex in Sepset(Xo,X2) is an ancestor of Xo or X2, it follows that every vertex in Supsepset(Xo,V,X2) is an ancestor of Xo, X2 or V. Xi is not an ancestor of Xo or X2, or V in *§2*- Hence in step flD of the algorithm given a d-separation oracle for £72 as input Xi£ Supsepset(X_0,V,X_2). Thus step flE of the CCD algorithm will orient Wo-*V in ¥2 as W<-*V (unless they have already been oriented this way in a previous stage of the algorithm). Thus *Fi and ¥2 are not equivalent

Case 2: n>l,orW^Xi.

Claim: X_o and X_{n+}i are d-connected given Supsepset(X_0,V,X_{n+1})u{ W} in *fy.*

Proof: We have already shown that W is a descendant of Xi, and so also of X_n in *Q* and *CJ2.* Since in both *Q* and *Q2* Xo is p-adjacent to Xi, but Xi is not an ancestor of Xo, it follows that Xo is an ancestor of Xi. Hence in both *Q* and *Qz* there is a directed path PQ from Xo to Xi on which every vertex except for Xo is a descendant of Xi. (In the case Xo—>Xi, the last assertion is trivial. In the case where Xo and Xi have a common child that is an ancestor of Xo or Xi, and Xi is not an ancestor of XQ, it merely states a property of the path Xo~^C-^...Xi, where C is a common child of Xo and Xi.) Since W is a descendant of Xi, it follows that there is a directed path Pi from Xi to W. Concatenating Po and Pi we construct a directed path P* from Xo to W on which every vertex except Xo is a descendant of Xi. Since Xi is not an ancestor of Xo, X_n+i or V, it follows that no vertex on P*, except Xo, is an ancestor of Xo, X_{n+}i or V. Similarly we can construct a path from Q* from X_n+i to W on which no vertex, except X_n+i, is an ancestor of Xo,X_n+i orV.

Since every vertex in Supsepset(Xo,V,X_n+i) is an ancestor of Xo, X_{n+}i or Sepset(Xo,X_{n+}i)u{ V}, it follows as before that every vertex in Supsepset(Xo,V^X_n+i) is an ancestor of Xo,X_n+i or V. Thus no vertex in Supsepset(Xo,V,X_n+i) lies on P* or Q* (Xo, X_{n+}i £ Supsepset(Xo,V,X_n+i) by definition). It now follows by Lemma 3.3.1+that we can concatenate P* and Q* to form a path which d-connects Xo and X_{n+}i given Supsepset(X_0,V,X_{n+}i)u{W}.

It follows directly from this claim that step flF of the CCD algorithm will orient V*-oW as V—>W in ¥2 (unless they have already been oriented this way in a previous stage of the algorithm). Hence *Fi and ¥2 are different.

Since Cases 1-5 exhaust the possible ways in which *Q* and *Qi* may fail to satisfy CET(I)-(V), this completes the proof that the CCD algorithm locates the d-separation equivalence class. .*.

References

AHO, A.V., HOPCROFT, J.E., & ULLMAN, J.D., (1974) The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading Mass.

BOLLEN, K. (1989) Structural Equations with Latent Variables, Wiley, New York.

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. *Scandinavian Journal of Statistics* 17, 333-353.

GEIGER, D. (1990). Graphoids: a qualitative framework for probabilistic inference. PhD dissertation, Univ. 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.

KIIVERI, H. and SPEED, T.P. (1982). Structural Analysis of multivariate data: A review. In *Sociological Methodology, 1982* (S. Leinhardt, ed.) 209-289. Jossey Bass, San Francisco.

KOSTER, J.T.A.(1994) Markov Properties of Non-Recursive Causal Models. MS.

LAURTTZEN, S., DAWID, A., LARSEN, B., LEIMER, H. (1990). "Independence Properties of Directed Markov Fields." *Networks* , 20, 491-505.

LEE, S. (1987) Model Equivalence in Covariance Structure Modelling. Ph.D. Thesis, Department of Psychology, Ohio State University.

MASON, S. (1956). Feedback Theory: Further properties of signal flow graphs. *Proceedings Institute ofRadio Eng.* 44 920-926.

MUTHÉN, B. (1984) A general structural equation model, with dichotomous, ordered categorical and continuous latent variable indicators. *Psychometrika* 49: 115-132

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

RICHARDSON, T. (1994a). Equivalence in Non-Recursive Structural Equation Models. ProceedingsrCo/npsfatf *94* , Physica Verlag

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

SOBEL, M. E. (1993) Comment in *Statistical Science* Vol.8 3 p.271

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., (1994) Conditional Independence in Directed Cyclic Grapical Models Representing Feedback or Mixtures. Philosophy, Methodology and Logic Technical Report 59, Carnegie Mellon University

SPIRTES, P., GLYMOUR C. and SCHEINES R. (1993), Causation, Prediction and Search. Lecture Notes in Statistics, Springer-Verlag.

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

WERMUTH, N. (1992) On Block-Recursive Linear Regression Equations, *Revista Brasileira de Probabilidade e Estatistica* 6, pp. 1-56 (with discussion)

WHITTAKER, J. (1990). Graphical Models in Applied Multivariatq/Statistics, Wiley, NY.