An Anytime Algorithm for Causal Inference

Peter Spirtes
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213
ps7z@andrew.cmu.edu

Abstract

The Fast Casual Inference (FCI) algorithm searches for features common to observationally equivalent sets of causal directed acyclic graphs. It is correct in the large sample limit with probability one even if there is a possibility of hidden variables and selection bias. In the worst case, the number of conditional independence tests performed by the algorithm grows exponentially with the number of variables in the data set. This affects both the speed of the algorithm and the accuracy of the algorithm on small samples, because tests of independence conditional on large numbers of variables have very low power. In this paper, I prove that the FCI algorithm can be interrupted at any stage and asked for output. The output from the interrupted algorithm is still correct with probability one in the large sample limit, although possibly less informative (in the sense that it answers “Can’t tell” for a larger number of questions) than if the FCI algorithm had been allowed to continue uninterrupted.

1 INTRODUCTION

One approach to causal inference from observational data represents causal relations by directed acyclic graphs (DAGs) and makes assumptions relating the causal DAG structure to probability distributions. The problem of making causal inferences then becomes the problem of finding the “best” DAG or set of DAGs for a given sample (Spirtes, Glymour, and Scheines 1993, Heckerman, Meek, and Cooper 1999).

There are two distinct types of algorithms that have been proposed for finding the “best” DAG or set of DAGs for a given data set. Both of these types of algorithms share the problem that they are very slow for data sets with large numbers of variables. “Score-based algorithms” assign a score to each causal model (e.g. a Bayesian posterior, an MDL, or BIC score) and perform searches over a large number of different DAGs for the DAG with the highest score (Heckerman, Meek, and Cooper 1999). One problem with this approach is that a conclusion about a causal influence can be reversed if a DAG with a higher score is later found. If the possibility of unmeasured variables is disallowed, the number of possible DAGs grows super-exponentially with the number of variables. Hence heuristic searches are performed, and a very extensive search is needed if a causal conclusion drawn by a score-based algorithm is to be held with any confidence. Moreover, if one allows the possibility of unmeasured variables, scoring the models becomes slow, and the number of potential models is infinite, so it is not clear how the search should be structured.

In contrast, a “constraint-based algorithm” constructs a graphical object that represents features common to all DAGS compatible with the results of a set of statistical tests of conditional independence selected by the algorithm. For example, there are cases in which all causal structures compatible with the results of a set of selected conditional independence tests have the features that “A causes B”, other cases where all agree that “A does not cause B”, and still others in which all agree that “There is an unmeasured common cause of A and B”. If some causal structures in which A causes B are compatible with a given set of conditional independence tests C, and other causal structures in which A does not cause B are compatible with C, then with respect to the question of whether A causes B the algorithm outputs “Can’t tell”. In addition, there are cases in which the size of a causal effect can be estimated in the limit as well. There is a constraint-based algorithm (the Fast Causal Inference, or FCI algorithm) which is correct in the large sample limit with probability one (Spirtes, Glymour, and Scheines 1993, Spirtes, Meek, and Richardson 1999) even if there is a possibility of latent variables and selection bias. Although it does not perform all possible tests of conditional independence among the measured variables, in the worst case, the number of conditional independence tests performed by the algorithm grows exponentially with the number of variables in the data set. Hence, in the worst case the algorithm is not feasible on data sets with large numbers of variables (although in many cases it is feasible with large numbers of variables). In addition, tests of independence conditional on large numbers of variables have very low power and hence the accuracy of the algorithm on small sample sizes decreases if the algorithm requires tests of independence conditional on a large set of variables.

In this paper, I prove that the FCI algorithm can be interrupted at any stage and asked for output, and that the output is still correct in the large sample limit with probability one, although possibly less informative (in the sense that it answers "Can't tell" for a larger number of questions) than if it had been allowed to continue uninterrupted. (The FCI algorithm has an outer loop in which tests of independence conditional on sets of variables of increasing size are considered. "Interrupting at any stage" means stopping the outer loop at any chosen size). Limiting the number of variables in the conditioning set of the independence tests that the FCI algorithm performs not only increases the speed of the algorithm, it also makes the algorithm more reliable on finite samples because the statistical tests with the least power have been eliminated. Call this modified version of the FCI algorithm the anytime FCI algorithm.

2 CAUSAL DAGS

Causal relations among random variables are represented by a DAG. A set of variables **V** is **causally sufficient** if every cause of two members of **V** is also a member of **V**. A DAG G is a **causal DAG** for a causally sufficient set of variables **V** and a population Pop if and only if there is an edge from A to B in G if and only if A is a direct cause of B relative to **V** for some units in Pop.

Causal relationships between a set of variables **V,** on the one hand**,** and the mechanism by which individuals in the sample are selected from a population, on the other hand, may lead to differences between the expected parameter values in a sample and the population parameter values. In this case say that the differences are due to **selection bias**.

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

For a given DAG G, and a partition of the variable set **V** of G into observed (**O**), selection (**S**), and latent (**L**) variables, write *G*(**O**,**S**,**L**). I assume that the only conditional independence relations that can be tested are those among variables in **O** conditional on any subset **X** of **O** when **S**=**1** (which is written as **X**∪(**S**=**1**). Let I(**X**,**Y,Z**) mean **X** is independent of **Z** given **Y**. If **X**, **Y**, and **Z** are included in **O**, and I(**X**,**Y**∪ (**S=1**),**Z**), then say it is an **observed** conditional independence relation.

Say that a distribution P satisfies the **Markov Condition** for a DAG G if in the distribution P each vertex V is independent of the set of vertices which are neither parents nor descendants of V, conditional on the parents of V. A DAG G **entails** a conditional independence relation I(**A**,**B**,**C**) if I(**A**,**B**,**C**) is true in every distribution that satisfies the Markov condition for G. There is a graphical relationship, d-separation, among three disjoint sets of vertices in a DAG, which determines whether or not a DAG G entails I(**A**,**C,B**). A vertex V is a **collider** on an undirected path U if and only if U contains a pair of distinct edges adjacent on the path and into V. For three disjoint sets of variables **A**, **B**, and **C**, **A** is **d-separated** from **B** given **C** in DAG G, if and only if there is an undirected path from some member of **A** to a member of **B** such that every collider on that path is either in **C** or has a descendant in **C**, and every non-collider on the path is not in **C**. For three disjoint sets of variables **A**, **B**, and **C**, **A** is **d-connected** to **B** given **C** in DAG G if and only if **A** is not d-separated from **B** given **C**. Geiger, Pearl, and Verma have shown that G entails I(**A**,**C**,**B**) if and only if **A** is d-separated from **B** given **C** in G. See Pearl(1988). In a DAG G(**O**,**S**,**L**), if **X**, **Y**, and **Z** are included in **O**, and **X** is d-separated from **Z** given **Y**∪(**S=1**) then say it is an **observed** d-separation relation (in the sense that it entails an observed conditional independence relation.) It is possible that the set of observed d-separation relations in two different DAGs are identical, in which case say that they are **O**-equivalent. More formally, if **X**, **Y**, and **Z** are disjoint subsets of **O**, and **X** is d-separated from **Z** given **Y**∪(**S=1**) in G(**O**,**S**,**L**) if and only if **X** is d-separated from **Z** given **Y**∪(**S'=1**) in G'(**O**,**S'**,**L'**) then G'(**O**,**S'**,**L'**) is in **O-Equiv**(G). If two DAGs are **O**-equivalent, then no set of conditional independence tests can distinguish between them. Since the FCI algorithm uses just tests of conditional independence relations to construct its output, the output of the algorithm should be an object that represents an entire **O**-equivalence class of DAGs.

3 PARTIAL ANCESTRAL GRAPHS

Partial ancestral graphs (PAGs) serve a dual purpose. They represent all of the observed d-separation relations in a DAG G(**O**,**S**,**L**) and they represent some of the features common to every member of an **O**-equivalence class of DAGs. There are three kinds of endpoints an edge in a PAG can have: "– ", "o", or ">". These can be combined to form the following four kinds of edges: A → B, A ↔ B, A o→ B, or A o—o B. Let "*" be a meta-symbol that stands for any of the three kinds of endpoints, e.g. "A *→ B" stands for "A o→ B" or "A ↔ B" or "A → B". A PAG π **represents** a DAG G(**O**,**L**,**S**) if and only:

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

Note that an "o" endpoint does not place any restriction on the ancestor relations in G. Hence there can be several different PAGs representing the same DAG G(**O**,**S**,**L**) because where one PAG π may have a ">" or "—" endpoint, or an underlined pair of endpoints, the other PAG π' may have a "o" endpoint or a non-underlined pair of endpoints respectively; if this is the case say that π is **more informative** than π'. An example of a PAG is shown in Figure 1 where **O** = {X,Z,Y,W}, **L** = {L}, and each S ∈ **S** (not shown) has no edges into or out of it.

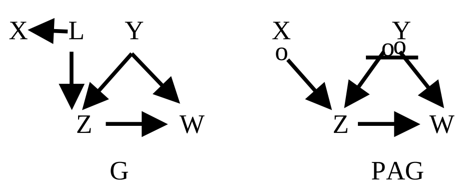


Figure 1

The proofs of the following theorems are in Spirtes, Meek, and Richardson (1999).

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

A **semi-directed path from** A to B in a PAG π is an undirected path U from A to B in which no edge contains an arrowhead pointing towards A, (i.e. there is no arrowhead at A on U, and if X and Y are adjacent on the path, and X is between A and Y on the path, then there is no arrowhead at the X end of the edge between X and Y). Theorem 2, Theorem 3, and Theorem 4 give information about what variables appear on causal paths between a pair of variables A and B, i.e. information about how those paths could be blocked.

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

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

Theorem 4: If π is a partial ancestral graph, and every semi-directed path from A to B contains some member of **C** in π, then every directed path from A to B in every DAG G(**O**,**S**,**L**) represented by PAG π contains a member of **S**∪**C**.

Pearl (1995) showed how in some cases to use the "Causal Calculus" (equivalent to Theorem 7.1 of Spirtes, Glymour, and Scheines 1993) to calculate the effects of interventions from a DAG of completely known structure and a marginal observed distribution, even if the DAG contains latent variables. PAGs represent partial knowledge about DAGs (possibly with latent variables.) There is an algorithm which can take partial knowledge about a DAG which may contain latent variables (in the form of a PAG) and an observed marginal distribution as input, and in some cases calculate the magnitude of an effect of an intervention. See Spirtes, Glymour, and Scheines (1993).

4 FAST CAUSAL INFERENCE ALGORITHM

For details, proofs, and examples for this section see Spirtes, Meek, and Richardson (1999). The fundamental assumption I will make relating causal DAGs to probability distributions is the following:

Selection Bias Causal Assumption: For each set of variables **O**, and each population Pop such that **S**=**1**, there is a causally sufficient set of variables **V** such that **O**∪**S** ⊆ **V** and for all **A**, **B**, **C** ⊆ **O**, I(**A**,(**C**∪(**S**=**1**),**B**) in Pop if and only if the causal DAG G(**O**,**S**,**L**) relative to **V** in Pop entails that I(**A**,(**C**∪(**S**= **1**),**B**) in Pop.

The justification of this assumption, and conditions under which it may fail are discussed in Spirtes, Meek, and Richardson, 1999. One of its consequences is a kind of Markov condition, i.e. the indirect causes of a variable V are screened off from V by its immediate causes.

In practice, for several different families of parameterizations of DAGs there are statistical tests of conditional independence that can be performed. For the purposes of this paper, however, problems of sample size will not be considered. I will assume that there is an oracle that the FCI algorithm can use to determine if I(**A**,(**C**∪(**S**=**1**),**B**) in Pop; in that case say that the oracle is an input for the FCI algorithm.

DAG G(**O**,**S**,**L**) entails that I(**A**,(**C**∪(**S**=**1**),**B**) if and only if A is d-separated from B conditional on **C**∪(**S**=**1**). So given the Selection Bias Causal Assumption, the assumption of an oracle that can determine whether I(**A**,(**C**∪(**S**=**1**),**B**) in Pop. is equivalent to having an oracle that can determine whether **A** is d-separated from **B** conditional on **C**∪**S** in G(**O**,**S**,**L**).

The algorithm starts off with a complete undirected graph between the observed variables. When the algorithm removes an edge between A and B, it does so because it has queried its oracle and received a "yes" answer to the question of whether some subset **Z** of **O**\{A,B} is such that A and B are d-separated conditional on **Z**∪**S**. The subset **Z** is recorded in **Sepset**(A,B) and **Sepset**(B,A). This information is used later in the orientation phase of the algorithm. Because each edge is removed at most once, **Sepset**(A,B) contains at most one subset of **O**\{A,B}. In the algorithm, **Adjacencies**(Q,X) is the set of vertices that are adjacent to X in graph Q. **Adjacencies**(Q,X) changes as the algorithm progresses, because the algorithm removes edges from Q. (However, **Possible-D-Sep** is calculated only once, and remains fixed, even as the graph changes.) When an orientation rule changes the orientation of an edge, e.g. from "A*—* B" to "A*→* B" this means that the orientation of the A endpoint of the edge remains at whatever its current value is, and the the orientation of the B endpoint of the edge is changed to ">".

The following definitions are used in the algorithm. A, B, and C form a **triangle** in a DAG or a PAG if and only if A and B are adjacent, B and C are adjacent, and A and C are adjacent. V is in **Possible-D-Sep**(A,B,π) in a PAG π if and only if there is an undirected path U between A and B in π such that for every subpath X *—* Y *—* Z of U either Y is a collider on the subpath, or X, Y, and Z form a triangle in π. In a partial ancestral graph π, U is a **definite discriminating path for** B if and only if U is an undirected path between X and Y, B is the predecessor of Y on U, B ≠ X, every vertex on U except for the endpoints and possibly B is a collider on U, for every vertex V on U except for the endpoints there is an edge V → Y, and X and Y are not adjacent.

Fast Causal Inference Algorithm
Input: Oracle for G(O,S,L)

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

B). n = 0.

repeat
  repeat
    select an ordered pair of variables X and Y that are adjacent in Q such that **Adjacencies**(Q,X)\{Y} contains at least n members,
      repeat
        select a subset **T** of **Adjacencies**(Q,X)\{Y} with n members, and if X and Y are d-separated given **T** ∪ **S** according to the oracle for G(**O**,**S**,**L**), delete the edge between X and Y from Q, and record **T** in **Sepset**(X,Y) and **Sepset**(Y,X)
      until all subsets of **Adjacencies**(Q,X)\{Y} of size n have been checked for d-separation given **T** ∪ **S** or there is no edge between X and Y;
    until all ordered pairs of adjacent variables X and Y such that **Adjacencies**(Q,X)\{Y} has at least n members have been selected;
  n = n + 1;
until for each ordered pair of adjacent vertices X, Y, **Adjacencies**(Q,X)\{Y} has fewer than n members.

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

D). Let π_1 be the graph resulting from step C.) For each pair of variables A and B adjacent in π_1, if there is a subset **T** of **Possible-D-SEP**(A,B, π_1)\{A,B} or of **Possible-D-SEP**(B,A, π_1)\{A,B} such that A and B are d-separated conditional on **T**∪**S** according to the oracle for G(**O**,**S**,**L**), remove the edge between A and B from π_1, and record **T** in **Sepset**(A,B) and **Sepset**(B,A).

E.) Orient each edge in π_1 as "o—o". Call this graph π_2.

F. For each triple of vertices A, B, C such that the pair A, B and the pair B, C are each adjacent in π_2 but the pair A, C are not adjacent in F', orient A *—* B *—* C as A *→ B ←* C if and only if B is not in **Sepset**(A,C).

G. repeat

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

(ii) else if P *→ M *—* R then orient as P *→ M → R.,

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

(iv) else if X ←* Y, X → Z, and Z o—* Y, orient as Z ←* Y;

(v) If U is a definite discriminating path between F and H for J, K is adjacent to H on U, and K, J, and H forn a triangle, then if J is in **Sepset**(F,H) then mark J as a non-collider on subpath K *—* J *—* H else orient K *—* J *—* H as K *→ J ←* H.

until no more edges can be oriented.

Theorem 5: If the input to the FCI algorithm is an oracle for G(**O**,**S**,**L**), the output is a PAG that represents G(**O**,**S**,**L**).

The output of the FCI algorithm represents ancestor relations common to any DAG that is **O**-equivalent to G. This is because the output of the FCI algorithm is a function of the oracle for G(**O**,**S**,**L**) and hence it produces the same output for any DAG G'(**O**,**S'**,**L'**) that has the same observed d-separation relations as G.

It is not known whether the PAG output by the FCI algorithm captures all of the ancestor relations common to the DAGs in the **O**-equivalence class of G. However, the PAG output by the FCI algorithm does contain enough orientation information to represent the observed d-separation relations in G in the following way. Say B is a **definite non-collider** on undirected path <A,B,C> if and only if either A ← B *—* C, A *—* B → C, or A *—* B *—* C in π. A is a **descendant** of B in a PAG π if and only if there is a directed path (all of the edges on the path are oriented as "→") from A to B in π or A = B. In a PAG π, if X ≠ Y, and X and Y are not in **Z**, then an undirected path U **definitely d-connects** X and Y given **Z** if and only if every collider on U has a descendant in **Z**, every definite non-collider on U is not in **Z**, and every other vertex on U is not in **Z** but has a descendant in **Z**. If **X**, **Y**, and **Z** are disjoint sets of variables, then **X** is definitely d-connected to **Y** given **Z** if and only if some member of **X** is d-connected to some member of **Y** given **Z**.

Theorem 6: If G(**O**,**S**,**L**) is a directed acyclic graph, and π is the output of the FCI algorithm whose input is an oracle for G(**O**,**S**,**L**), and **X**, **Y**, and **Z** are disjoint subsets of **O**, then **X** is d-connected to **Y** given **Z**∪**S** in G(**O**,**S**,**L**) if and only if **X** is definitely d-connected to **Y** given **Z** in π.

5 ANYTIME FAST CAUSAL INFERENCE ALGORITHM

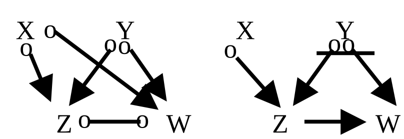
The Anytime Fast Causal Inference Algorithm simply halts the outer repeat loop in step B) of the algorithm at some fixed size n. The result of stopping the algorithm at that point is a PAG which is correct, but possibly less informative than if the algorithm had been allowed to continue running.

Halting the outer repeat loop in step B) of the algorithm at some fixed size n is equivalent to having an oracle that always returns "no" for all conditioning sets of size greater than n. Let an **n-oracle** for G(**O**,**S**,**L**) be an algorithm that on input "**X**, **Y**, **Z**" outputs "no" if |**Z**| > n or **X** is not d-separated from **Y** given **Z**∪**S**, and otherwise outputs "yes". (|**Z**| represents the number of variables in **Z**.)

The sense in which the Anytime Fast Causal Inference Algorithm is correct but possibly less informative than the Fast Causal Inference Algorithm is twofold. First, it correctly represents ancestor relations common to the set of DAGs that that have the same observed d-separation relations as G for all conditioning sets of size less than or equal to n. This condition can be expressed more formally using the following definitions. If **X**, **Y**, and **Z** are disjoint subsets of **O**, and for all |**Y**| ≤ n, **X** is d-separated from **Z** given **Y**∪(**S=1**) in G(**O**,**S**,**L**) if and only if **X** is d-separated from **Z** given **Y**∪(**S'=1**) in G'(**O**,**S'**,**L'**) then G'(**O**,**S'**,**L'**) is in **n-O-Equiv**(G). The definition of a PAG π **n-representing** a DAG G(**O**,**L**,**S**) is the same as the definition of π representing DAG G(**O**,**S**,**L**) except that each occurrence of **O-Equiv**(G) in the definition is replaced with **n-O-Equiv**(G), and clause 4) of the definition is replaced by 4':

4'. A and B are adjacent in π if and only if for every subset **Z** of **O**\{A,B} such that |**Z**| ≤ n, A and B are d-connected conditional on **Z**∪**S** in G(**O**,**S**,**L**)

Suppose that **O** = {X,Z,Y,W}, **L** = {L}, and each S ∈ **S** (not shown) has no edges into or out of it. Then (i) of Figure 2 0-represents and 1-represents G of Figure 1, and (ii) 2-represents G.



(i) n= 0 or n=1 (ii) n=2

Figure 2

Theorem 7: If the input to the FCI algorithm is an n-oracle for G(**O**,**S**,**L**), the output π is a PAG that n-represents G(**O**,**S**,**L**).

Second,, if the input to the anytime FCI algorithm is an n-oracle for G(**O**,**S**,**L**) the output PAG represents all of the observable d-separation relations in the DAG G(**O**,**S**,**L**) whose conditioning sets are of size less than or equal to n.

Theorem 8: If π is the PAG output by the FCI algorithm with an n-oracle for G(**O**,**S**,**L**), and |**Z**| ≤ n, then **X** and **Y** are definitely d-connected given **Z** in π if and only if **X** and **Y** are d-connected given **Z**∪**S** in G(**O**,**S**,**L**).

In addition to stopping the FCI algorithm after any iteration in step B), it is also possible to stop the FCI algorithm in any iteration in step G. Once an edge is oriented in the repeat loop of step G of the anytime FCI algorithm the algorithm never changes that orientation. Hence step G can be halted at any stage without affecting the correctness of the output; only the informativeness of the output will be affected.

6 APPENDIX

The proof of Lemma 1 is in Spirtes, Glymour, and Scheines (1993).

Lemma 1: In a DAG G over $\mathbf{V}$, if X and Y are not in $\mathbf{Z}$, and there is a sequence H of distinct vertices in $\mathbf{V}$ from X to Y, and there is a set T of undirected paths such that

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

(ii) if a vertex Q in H is in $\mathbf{Z}$, then the paths in T that contain Q as an endpoint collide (are both into) at Q, and

(iii) if for three vertices V, W, Q occurring in that order in H the d-connecting paths in T between V and W, and W and Q collide at W then W has a descendant in $\mathbf{Z}$,

then there is a path U in G that d-connects X and Y given $\mathbf{Z}$.

Lemma 2: In a DAG G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), if X is not an ancestor of $\{Y\}\cup\mathbf{S}$, and for each $\mathbf{M} \subseteq \mathbf{O}$, such that $|\mathbf{M}| \le r$, X and Y are d-connected given $\mathbf{M}\cup\mathbf{S}$, then for each $\mathbf{R} \subseteq \mathbf{O}$, such that $|\mathbf{R}| \le r$, there is a path U that d-connects X and Y given $\mathbf{R}\cup\mathbf{S}$ that is into X.

Proof. Let $\mathbf{R}$ be an arbitrary subset of $\mathbf{O}$ such that $|\mathbf{R}| = n \le r$. I will construct a sequence of subsets $\mathbf{R_0} \subset \mathbf{R_1} \ldots \subset \mathbf{R_{n-1}} \subset \mathbf{R}$, such that for each $\mathbf{R_i}$, there is a path U into X that d-connects X and Y given $\mathbf{R_i}\cup\mathbf{S}$, and each member of $\mathbf{R_i}$ is an ancestor of $\mathbf{S}\cup\{X,Y\}$.

Let $\mathbf{R_0} = \emptyset$. By hypothesis, X and Y are d-connected given $\mathbf{S}$. It follows that there is an undirected path T between X and Y such that every collider on T is an ancestor of $\mathbf{S}$. Because X is not an ancestor of Y, either T is a directed path from Y to X that does not contain $\mathbf{S}$, or T contains a collider. If T is a directed path from Y to X that does not contain $\mathbf{S}$ then T d-connects X and Y given $\mathbf{R_0}\cup\mathbf{S}$, and is into **X**. If T contains a collider, let C be the closest collider on T to X. C is an ancestor of $\mathbf{S}$. If T is out of X, then X is an ancestor of C, and hence of $\mathbf{S}$. Hence T is into X. Trivially, every member of $\mathbf{R_0}$ is an ancestor of $\{X,Y\}\cup\mathbf{S}$.

Suppose for $\mathbf{R_m} \subset \mathbf{R}$, every member of $\mathbf{R_m}$ is an ancestor of $\mathbf{S}\cup\{X,Y\}$, and there is a path U into X that d-connects X and Y given $\mathbf{R}\cup\mathbf{S}$. I will now show that there is a set $\mathbf{R_{m+1}} = \mathbf{R_m}\cup\{W\}$, where $W \in \mathbf{R}\backslash\mathbf{R_m}$, and a path U' that d-connects X and Y given $\mathbf{R_{m+1}}$ that is into X, and every member of $\mathbf{R_{m+1}}$ is an ancestor of $\{X,Y\}\cup\mathbf{S}$.

If U does not contain some member of $\mathbf{R}\backslash\mathbf{R_m}$ as a non-collider, then U d-connects X and Y given $\mathbf{R}\cup\mathbf{S}$, and is into X. Suppose then that U does contain some member W of $\mathbf{R}\backslash\mathbf{R_m}$ as a non-collider. Because U d-connects X and Y given $\mathbf{R_m}\cup\mathbf{S}$, it follows that every vertex on U is an ancestor of $\mathbf{R_m}\cup\mathbf{S}\cup\{X,Y\}$. By the induction hypothesis, every member of $\mathbf{R_m}$ is an ancestor of $\mathbf{S}\cup\{X,Y\}$, so W is an ancestor of $\mathbf{S}\cup\{X,Y\}$. Let $\mathbf{R_{m+1}} = \mathbf{R_m}\cup\{W\}$. It follows that every member of $\mathbf{R_{m+1}}$ is an ancestor of $\mathbf{S}\cup\{X,Y\}$. By hypothesis, there is a path U' that d-connects X and Y given $\mathbf{R_{m+1}}$. Every vertex on U' is an ancestor of $\mathbf{S}\cup\{X,Y\}$, because every collider on U' is an ancestor of $\mathbf{R_{m+1}}$, and every member of $\mathbf{R_{m+1}}$ is an ancestor of $\mathbf{S}\cup\{X,Y\}$. If U' is not into X, and there are no colliders on U', then X is an ancestor of Y, contrary to the hypothesis. If U' does contain a collider, let C be the collider on U' closest to X. If U' is out of X, then X is an ancestor of C. C is not an ancestor of X, because otherwise G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) contains a cycle. Hence C is an ancestor of $\mathbf{S}\cup\{Y\}$. It follows that X is an ancestor of $\mathbf{S}\cup\{Y\}$, contrary to the hypothesis. Hence U' is into X.

It follows by induction that there is a path U that d-connects X and Y given $\mathbf{R}$ that is into X. Q.E.D.

Lemma 3: In a DAG G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), if X is an ancestor of $\{Y\}\cup\mathbf{S}$, and for each $\mathbf{M} \subseteq \mathbf{O}$, such that $|\mathbf{M}| \le r$, X and Y are d-connected given $\mathbf{M}\cup\mathbf{S}$, then for each $\mathbf{R} \subseteq \mathbf{O}$ such that $|\mathbf{R}| \le r$, there is a path U that d-connects X and Y given $(\mathbf{R}\cup\mathbf{S})\backslash\{X,Y\}$ that is out of X, or X is an ancestor of $\mathbf{R}\cup\mathbf{S}$.

Proof. Let $\mathbf{R} \subseteq \mathbf{O}$ such that $|\mathbf{R}| \le r$. By hypothesis, X is an ancestor of $\{Y\}\cup\mathbf{S}$. If X is an ancestor of $\mathbf{S}$ the proof is done. Suppose then that X is an ancestor of Y. There is a directed path D from X to Y that does not contain any member of $\mathbf{S}$. If D contains a member of $\mathbf{R}$, then X is an ancestor of $\mathbf{R}$. If D does not contain a member of $\mathbf{R}$, then D is a path that d-connects X and Y given $(\mathbf{R}\cup\mathbf{S})\backslash\{X,Y\}$ that is out of X. Q.E.D

Lemma 4: In G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), if $\{A,B\} \subseteq \mathbf{O}$, A is an ancestor of $\{B\}\cup\mathbf{S}$, and B is an ancestor of $\{A\}\cup\mathbf{S}$, then A and B are ancestors of $\mathbf{S}$.

Proof. Because G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) is acyclic, either A is not an ancestor of B, or B is not an ancestor of A. Suppose without loss of generality that A is not an ancestor of B. It follows that A is an ancestor of $\mathbf{S}$. Either B is an ancestor of A, in which case it is an ancestor of $\mathbf{S}$, or it is an ancestor of $\mathbf{S}$. Q.E.D.

For a DAG G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), let G_n($\mathbf{O}$,$\mathbf{S'}$,$\mathbf{L'}$) be formed in the following way:

- Initialize $\mathbf{S'} = \mathbf{S}$ and $\mathbf{L'} = \mathbf{L}$.
- If there is an edge from X to Y in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), there is a corresponding edge from X to Y in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$.
- If there is no edge between X and Y in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), but in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), X and Y are d-connected given every subset $\mathbf{R}\cup\mathbf{S}$ such that $|\mathbf{R}| \leq n$, then
    - if X is an ancestor of $\{Y\}\cup\mathbf{S}$, and Y is an ancestor of $\{X\}\cup\mathbf{S}$, add S_{XY} to $\mathbf{S'}$, and add edges $X \rightarrow S_{XY} \leftarrow Y$ to $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$,
    - if X is an ancestor of $\{Y\}\cup\mathbf{S}$, and Y is not an ancestor of $\{X\}\cup\mathbf{S}$, add an edge $X \rightarrow Y$ to $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$,
    - if X is not an ancestor of $\{Y\}\cup\mathbf{S}$, and Y is an ancestor of $\{X\}\cup\mathbf{S}$, add an edge $X \leftarrow Y$ to $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$,
    - else if X is not an ancestor of $\{Y\}\cup\mathbf{S}$, and Y is not an ancestor of $\{X\}\cup\mathbf{S}$, add L_{XY} to $\mathbf{L'}$, and add edges $X \leftarrow L_{XY} \rightarrow Y$ to $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$.

Lemma 5: If $\mathbf{X}$, $\mathbf{Y} \subseteq \mathbf{O}$ in G($\mathbf{O}$,S,$\mathbf{L}$), $\mathbf{X}$ is an ancestor or $\mathbf{Y}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) if and only if $\mathbf{X}$ is an ancestor or $\mathbf{Y}\cup\mathbf{S'}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$.

Proof. Because $\mathbf{S} \subseteq \mathbf{S'}$, and the edges in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) are a subset of the edges in $G_n(\mathbf{O},\mathbf{S},\mathbf{L})$, if X is an ancestor of $\mathbf{Y}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) then X is an ancestor or $\mathbf{Y}\cup\mathbf{S'}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. Suppose that X is an ancestor or $\mathbf{Y}\cup\mathbf{S'}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. If $\{A,B\} \subseteq \mathbf{O}$, there is an edge from A to B in $G_n(\mathbf{O},\mathbf{S},\mathbf{L})$ but not in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) only if A is an ancestor of $\{B\}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$). If $A \in \mathbf{O}$, and $S \in \mathbf{S'}$, there is an edge from A to S in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$ but not in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) only if for some $B \in \mathbf{O}$, A is an ancestor of $\{B\}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), and B is an ancestor of $\{A\}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$). It follows from Lemma 4 that A and B are ancestors of $\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$). Q.E.D.

Lemma 6: In $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$, if $\mathbf{R} \subseteq \mathbf{O}$, $|\mathbf{R}| \leq n$, and X_1 and X_m are d-connected given $\mathbf{R}\cup\mathbf{S'}$ in $G_n(\mathbf{O},\mathbf{S},\mathbf{L})$, then X_1 and X_m are d-connected given $\mathbf{R}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$).

Proof. Suppose $\mathbf{R} \subseteq \mathbf{O}$, $|\mathbf{R}| \leq n$, and X_1 and X_m are d-connected given $\mathbf{R}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$ by some undirected path U = $<X_1, \ldots , X_m>$. Let U_S be the subsequence of vertices $<X_{j(1)}, \ldots , X_{j(m)}>$ on U that are in $\mathbf{O}\cup\mathbf{S}\cup\mathbf{L}$. By the definition of d-connection, each pair of vertices $X_{j(i)}$ and $X_{j(i+1)}$ that are adjacent on U_S are d-connected conditional on $(\mathbf{R}\cup\mathbf{S'})\backslash\{X_{j(i)},X_{j(i+1)}\}$ by the subpath $U(X_{j(i)},X_{j(i+1)})$, and similarly for the subpath $U(X_{j(i-1)},X_{j(i)})$.

Suppose that $U(X_{j(i-1)},X_{j(i)})$ and $U(X_{j(i)},X_{j(i+1)})$ collide at $X_{j(i)}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. It follows that either $U(X_{j(i-1)},X_{j(i)})$ is an edge $X_{j(i-1)} \rightarrow X_{j(i)}$ or a path $X_{j(i)} \leftarrow L \rightarrow X_{j(i+1)}$. If $U(X_{j(i-1)},X_{j(i)})$ is an edge $X_{j(i-1)} \rightarrow X_{j(i)}$ and there is a corresponding edge $X_{j(i-1)} \rightarrow X_{j(i)}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), then let $U'(X_{j(i-1)},X_{j(i)})$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) be the edge $X_{j(i-1)} \rightarrow X_{j(i)}$, in which case $U'(X_{j(i-1)},X_{j(i)})$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) d-connects $X_{j(i-1)}$ and $X_{j(i)}$ given $(\mathbf{R}\cup\mathbf{S})\backslash\{X_{j(i-1)},X_{j(i)}\}$ and is into $X_{j(i)}$. On the other hand, if the subpath $U(X_{j(i-1)},X_{j(i)})$ is not an edge in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$, or there is no corresponding edge $X_{j(i-1)} \rightarrow X_{j(i)}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), by the definition of $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$ $X_{j(i)}$ is not an ancestor of $X_{j(i-1)}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$). It follows from Lemma 2 that there is a path $U'(X_{j(i-1)},X_{j(i)})$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) that d-connects $X_{j(i-1)}$ and $X_{j(i)}$ given $(\mathbf{R}\cup\mathbf{S})\backslash\{X_{j(i-1)},X_{j(i)}\}$ that is into $X_{j(i)}$. Hence, there is a path $U'(X_{j(i-1)},X_{j(i)})$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) that d-connects $X_{j(i-1)}$ and $X_{j(i)}$ given $(\mathbf{R}\cup\mathbf{S})\backslash\{X_{j(i-1)},X_{j(i)}\}$ that is into $X_{j(i)}$, and similarly in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) there is a path $U'(X_{j(i+1)},X_{j(i)})$ that d-connects $X_{j(i+1)}$ and $X_{j(i)}$ given $(\mathbf{R}\cup\mathbf{S})\backslash\{X_{j(i)},X_{j(i+1)}\}$ that is into $X_{j(i)}$. Because the subpaths $U(X_{j(i-1)},X_{j(i)})$ and $U(X_{j(i)},X_{j(i+1)})$ collide at $X_{j(i)}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$, $X_{j(i)}$ is an ancestor of $\mathbf{R}\cup\mathbf{S'}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. By Lemma 5, $X_{j(i)}$ is an ancestor of $\mathbf{R}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$).

Suppose that $U(X_{j(i-1)},X_{j(i)})$ and $U(X_{j(i)},X_{j(i+1)})$ do not collide at $X_{j(i)}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. $X_{j(i)}$ is not in $\mathbf{R}\cup\mathbf{S'}$, and hence by Lemma 5, $X_{j(i)}$ is not in $\mathbf{R}\cup\mathbf{S}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. At least one of the two subpaths $U(X_{j(i-1)},X_{j(i)})$ and $U(X_{j(i)},X_{j(i+1)})$ is out of $X_{j(i)}$. Suppose without loss of generality that $U(X_{j(i-1)},X_{j(i)})$ is out of $X_{j(i)}$. It follows that either $U(X_{j(i-1)},X_{j(i)})$ is $X_{j(i-1)} \leftarrow X_{j(i)}$ or a path $X_{j(i)} \rightarrow S \leftarrow X_{j(i+1)}$. If $U(X_{j(i-1)},X_{j(i)})$ is an edge $X_{j(i-1)} \leftarrow X_{j(i)}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$, and there is a corresponding edge in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), then let $U'(X_{j(i-1)},X_{j(i)})$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) be $X_{j(i-1)} \leftarrow X_{j(i)}$. Otherwise, by the definition of $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$, $X_{j(i)}$ is an ancestor of $\mathbf{S}\cup\{X_{j(i-1)}\}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$). By Lemma 3, either $X_{j(i)}$ is an ancestor of $\mathbf{R}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) or in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) there is a path $U'(X_{j(i-1)},X_{j(i)})$ that d-connects $X_{j(i-1)}$ and $X_{j(i)}$ given $(\mathbf{R}\cup\mathbf{S})\backslash\{X_{j(i-1)},X_{j(i)}\}$ that is out of $X_{j(i)}$.. It follows that either $U'(X_{j(i-1)},X_{j(i)})$ d-connects $X_{j(i-1)}$ and $X_{j(i)}$ given the set $(\mathbf{R}\cup\mathbf{S})\backslash\{X_{j(i-1)},X_{j(i)}\}$ and $U'(X_{j(i)},X_{j(i+1)})$ d-connects $X_{j(i)}$ and $X_{j(i+1)}$ given the set $(\mathbf{R}\cup\mathbf{S})\backslash\{X_{j(i)},X_{j(i+1)}\}$in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), and they are not both into $X_{j(i)}$, or $X_{j(i)}$ is an ancestor of $\mathbf{R}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$). It follows from Lemma 1 that X_1 and X_m are d-connected given $\mathbf{R}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$). Q.E.D.

Theorem 7: If the input to the FCI algorithm is an n-oracle for G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), the output π is a PAG that n-represents G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$).

Proof. By Theorem 5 the output of the FCI algorithm represents $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$ if the input is an oracle for $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. Because every edge in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) is also in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$, $\mathbf{S} \subseteq \mathbf{S'}$, and each member of $\mathbf{S'}\backslash\mathbf{S}$ has no edges out of it, for $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z} \subseteq \mathbf{O}$, if $\mathbf{X}$ and $\mathbf{Y}$ are d-connected given $\mathbf{Z}\cup\mathbf{S}$ in G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$), $\mathbf{X}$ and $\mathbf{Y}$ are d-connected given $\mathbf{Z} \cup \mathbf{S'}$ in $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$. Thus by Lemma 6, an oracle for $G_n(\mathbf{O},\mathbf{S'},\mathbf{L'})$ is an n-oracle for G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$).
In order to show that π **n-represents** G($\mathbf{O}$,$\mathbf{S}$,$\mathbf{L}$) I will show that it satisfies the 7 clauses of the definition. Parts 1 through 3 follow immediately.

Suppose that A and B are d-connected given $\mathbf{Z}\cup\mathbf{S}$ in G(**O**,**S**,**L**) for all $|\mathbf{Z}| < n$. It follows that there is a path U that d-connects A and B conditional on $\mathbf{Z}\cup\mathbf{S}$ in G(**O**,**S**,**L**). The corresponding path U' with the same edges exists in G_n(**O**,**S'**,**L'**). Every collider on U is an ancestor of $\mathbf{Z}\cup\mathbf{S}$ in G(**O**,**S**,**L**) and hence by Lemma 5 every collider on U' is an ancestor of $\mathbf{Z}\cup\mathbf{S'}$ in G_n(**O**,**S'**,**L'**). Every non-collider on U is not in $\mathbf{Z}\cup\mathbf{S}$, and hence every non-collider on U' is not in $\mathbf{Z}\cup\mathbf{S'}$. It follows that A and B are d-connected conditional on $\mathbf{Z}\cup\mathbf{S'}$ in G_n(**O**,**S'**,**L'**). Since the FCI algorithm only removes an edge from π when the answer to a d-connection question from the n-oracle is "no", it follows that there is an edge between A and B in π.

Suppose that there is an edge between A and B in π. It follows that A and B are definitely d-connected given every subset **Z**, and hence are d-connected given $\mathbf{Z}\cup\mathbf{S'}$ in G_n(**O**,**S'**,**L'**) By Lemma 6 if $\mathbf{Z} \subseteq \mathbf{O}$, $|\mathbf{Z}| \le n$, and A and B are d-connected given $\mathbf{Z}\cup\mathbf{S'}$ in G_n(**O**,**S'**,**L'**), then A and B are d-connected given $\mathbf{Z} \cup \mathbf{S}$ in G(**O**,**S**,**L**).

Suppose there is an edge in π oriented as $A \rightarrow B$. By Theorem 5 A is an ancestor of B but not **S'** in G_n(**O**,**S'**,**L'**). By Lemma 5, A is an ancestor of$\{B\}\cup\mathbf{S}$ in G(**O**,**S**,**L**). If A is an ancestor of **S** in G(**O**,**S**,**L**) it is an ancestor of **S** in G_n(**O**,**S'**,**L'**), and because $\mathbf{S} \subseteq \mathbf{S'}$, A is an ancestor of **S'** in G_n(**O**,**S'**,**L'**). But because A is not an ancestor of **S'** in G_n(**O**,**S'**,**L'**), it is not an ancestor of **S** in G(**O**,**S**,**L**). Hence A is an ancestor of B but not **S** in G(**O**,**S**,**L**). Because the output of the FCI algorithm is a function of the n-oracle for G(**O**,**S**,**L**), it produces the same output for any DAG G'(**O**,**S'**,**L'**) in **n-O-Equiv**(G). Hence A is an ancestor of B but not **S'** for any DAG G(**O,S'**,**L'**) in **n-O-Equiv**(G).

Suppose there is an edge oriented as $A \ast\!\!\rightarrow B$ in π. Then by Theorem 5, B is not an ancestor of $\{A\}\cup\mathbf{S'}$ in G_n(**O**,**S'**,**L'**). By Lemma 5 B is not an ancestor of $\{A\}\cup\mathbf{S}$ in G(**O**,**S**,**L**). Because the output of the FCI algorithm is a function of the n-oracle for G(**O**,**S**,**L**), it produces the same output for any DAG G'(**O**,**S'**,**L'**) in **n-O-Equiv**(G). Hence B is not an ancestor of $\{A\}\cup\mathbf{S'}$ for any DAG G(**O,S'**,**L'**) in **n-O-Equiv**(G).

Suppose that there are edges $A \ast\!\!-\!\!\underline{\ast\ B \ast}\!\!-\!\!\ast\ C$ in π. By Theorem 5 it follows that B is ancestor of $\{A\}\cup\{C\}\cup\mathbf{S'}$ in G_n(**O**,**S'**,**L'**). By Lemma 5, B is an ancestor of $\{A\}\cup\{C\}\cup\mathbf{S}$ in G(**O**,**S**,**L**). Because the output of the FCI algorithm is a function of the n-oracle for G(**O**,**S**,**L**), it produces the same output for any DAG G'(**O**,**S'**,**L'**) in **n-O-Equiv**(G). Hence B is ancestor of $\{A\}\cup\{C\}\cup\mathbf{S'}$ for any DAG G(**O,S'**,**L'**) in **n-O-Equiv**(G). Q.E.D.

Theorem 8: If π is the PAG output by the FCI algorithm with an n-oracle for G(**O**,**S**,**L**), and $|\mathbf{Z}| \le n$, then **X** and **Y** are definitely d-connected given **Z** in P if and only if **X** and **Y** are d-connected given $\mathbf{Z} \cup \mathbf{S}$ in G(**O**,**S**,**L**).

Proof. By Lemma 6, if π is the output of the FCI algorithm with an oracle for G_n(**O**,**S'**,**L'**), then π is also the output of the FCI algorithm with an n-oracle for G(**O**,**S**,**L**). By Theorem 6, **X** and **Y** are definitely d-connected given **Z** in π if and only if **X** and **Y** are d-connected given $\mathbf{Z}\cup\mathbf{S'}$ in G_n(**O**,**S**,**L**). By Lemma 6, if $|\mathbf{Z}| \le n$, **X** and **Y** are d-connected given $\mathbf{Z}\cup\mathbf{S'}$ in G_n(**O**,**S**,**L**) if and only if **X** and **Y** are d-connected given $\mathbf{Z} \cup \mathbf{S}$ in G(**O**,**S**,**L**). So **X** and **Y** are d-connected given $\mathbf{Z}\cup\mathbf{S}$ in G(**O**,**S**,**L**) if and only if **X** and **Y** are definitely d-connected given **Z** in π.

Acknowlegements

This research was partially supported by NSG grant DMS-9873442.

References

Cooper, G. (1995). Causal discovery from data in the presence of selection bias. *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL.

Heckerman, D., Meek, C., and Cooper, G. (1999) A Bayesian approach to causal discovery. *Computation, Causation and Discovery*. C. Glymour and G. Cooper. Cambridge, MA, MIT Press.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Mateo, Morgan Kaufmann.

Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, **82**, 669-709.

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

Spirtes, P., Meek, C., and Richardson, T. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. *Computation, Causation and Discovery*. C.Glymour and G. Cooper. Cambridge, MA.