Complete Identification Methods for Causal Inference

Ilya Shpitser
April 2008

Technical Report R-341
Cognitive Systems Laboratory
Department of Computer Science
University of California
Los Angeles, CA 90095-1596, USA

This report reproduces a dissertation submitted to UCLA in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science. This work was supported in part by AFOSR grant #F49620-01-1-0055, NSF grant #IIS-0535223, MURI grant #N00014-00-1-0617, and NLM grant #NIH EB000216.

The dissertation of Ilya Shpitser is approved.

Sheldon Smith

Eleazar Eskin

Adnan Darwiche

Judea Pearl, Committee Chair

University of California, Los Angeles

2008

To hms, my muse

Table of Contents

List of Figures

List of Tables

Acknowledgments

I thank whoever is responsible for constructing this beautiful world, and giving human beings the drive to explore and understand its mysteries. I thank my family for giving me the opportunity to pursue my dreams. I thank my friends at the Berkeley CSUA for getting me interested in computer science when I was a tyro, and for their continuing help and support over the years. I thank Stuart Russell and his group for giving me my first taste of research. I thank my friends, lab mates, and collaborators at UCLA: Gunes Ercal, Chen Avin, Carlos Brito, and Mark Hopkins for helping me make UCLA my home, for teaching me the ropes, and for sharing the joys and pains of graduate student life. I thank Verra Morgan for her faith in me. I thank Manabu Kuroki for his encouragement and interest in my work. I thank Kaoru Mulvihill for making my life easier tenfold at least. I thank everyone at the UCLA medical imaging informatics lab for their kind support, advice, and help. I thank Jin Tian for blazing a trail which made this thesis possible. I thank `hms` for the gift of inspiration.

Finally, I thank Judea Pearl. For asking beautiful questions. For his humor, his kindness, his wisdom. For meeting hatred with tolerance, blindness with knowledge, and for showing how the human spirit can triumph over the worst personal tragedy.

Vita

1976	Born, Dzhankoy, Ukraine.
1999	B.A., Computer Science and Mathematics, University of California, Berkeley.
1999	Software Engineer, SHAI, San Mateo, California.
1999-2001	Senior Software Engineer, Black Pearl Inc., San Francisco, California.
2002-2005	Teaching Assistant, Computer Science Department, University of California, Los Angeles.
2005	M.S., Computer Science, University of California, Los Angeles.
2005-2008	Research Assistant, Computer Science Department, University of California, Los Angeles.

Publications

Avin, C., Shpitser, I., and Pearl, J. (2005). Identifiability of Path-Specific Effects. Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI).

Pasula, H., Marthi, B., Milch, B., Russell, S., and Shpitser, S. (2002). Identity Uncertainty and Citation Matching. In Advances in Neural Information Processing Systems (NIPS).

Shpitser, I., and Pearl, J. (2006). Identification of Joint Interventional Distributions in Recursive Semi-Markovian Causal Models. Proceedings of the Twenty First Conference on Artificial Intelligence (AAAI).

—, and Pearl, J. (2006). Identification of Conditional Interventional Distributions. Proceedings of the Twenty Second Conference on Uncertainty in Artificial Intelligence (UAI).

—, and Pearl, J. (2007). What Counterfactuals Can Be Tested. Proceedings of the Twenty Third Conference on Uncertainty in Artificial Intelligence (UAI).

—, and Pearl, J. (2008). Complete Identification Methods for the Causal Hierarchy. To Appear in Journal of Machine Learning Research.

—, and Pearl, J. (2008). Dormant Independence. To Appear in Proceedings of the Twenty Third Conference on Artificial Intelligence (AAAI).

Abstract of the Dissertation

Complete Identification Methods for Causal Inference

by

Ilya Shpitser
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2008
Professor Judea Pearl, Chair

Human beings organize their intuitive understanding of the world in terms of causes and effects. Primitive humanity posited gods and spirits as invisible causes of phenomena they did not comprehend. As our attempts to understand the world began to be formalized and codified as empirical science, the emphasis on discerning cause-effect relationships remained. Though we, the modern humanity, are armed with powerful computers, sophisticated technology, and highly developed mathematics and statistics, our fundamental questions remain the same as those of our cave dwelling ancestors – we seek to understand the causes of windfalls and misfortunes that befall us, what effects our actions have, and what would happen if the past were different from what it is. This thesis will address these ancient questions with the rigor and generality of modern mathematics.

Using the framework of graphical causal models which formalizes a variety of causal queries, such as causal effects, counterfactuals and path-specific effects as certain types of probability distributions, I will develop algorithms which will evaluate these probability distributions from available information; prove that whenever these algorithms fail to evaluate a query, no other method could succeed; provide characterizations based on directed graphs for cases where these algorithms do succeed; and finally show how a class of constraints placed on

the causal model by its directed graph are due to conditional independence in these probability distributions, and how these conditional independencies can be exploited for testing causal theories.

CHAPTER 1

Introduction

Causality is fundamental to our understanding of the natural world. Causal questions and claims are a part of everyday speech, as well as legal, scientific and philosophical vocabulary. In discussing causal questions, just as in discussing questions of arithmetic or geometry, human beings seem to reach consensus on meaning. That isn't to say that all causal notions are unambiguous and crystal clear, but there is broad agreement on what claims such as "smoking causes cancer," or "carbon dioxide emissions contribute to global warming" mean. However, unlike arithmetic or geometry, there isn't a universally agreed upon formalization of causality. Instead, the consensus on causal issues seems to be driven largely by intuition. Even the most honed intuition can fail or lead astray, so formal, mathematical approaches to causality are preferable. Fortunately, the existence of consensus suggests that some formal structure for representing and reasoning about causality is present in the human brain. Though the exact way in which we reason about causality is not known, there are a number of formalization attempts which can claim to lead to reasonable conclusions which generally agree with human intuition [Wri21], [Ney23], [Tin37], [Lew73], [Rub74], [Rob87], [Pea00]. In this thesis, I will represent causality using *graphical causal models*, a representation method based on directed graphs and probability theory which was independently discovered multiple times during the 20th century, with various degrees of rigor [Wri21], [Pea95].

1.1 Causality and Graphs

People generally distinguish causes from effects because the former influence the latter, but not vice versa. Certainly in some cases involving dynamic equilibrium, like economic or physical systems, mutual causation is possible. [1] Yet even in such cases human beings tend to untangle the influences involved in causal loops by considering distinct causes and effects. Causality thus implies *directionality* of influence. In addition to directionality, people assume that causal influence is

[1] For instance, it's well known that supply affects demand and vice versa. Similarly, it's possible to contrive physical systems with mutual causation, like two boards forming a "tent" propping each other up. I am grateful to Sheldon Smith for this example.

modular, which means that full knowledge of all direct causes of a given effect is sufficient for concluding the effect regardless of the state of the rest of the world. Of course, when considering causal questions, human beings don't have access to the world "as it is." Instead, they typically have in mind some model of causal interactions of some part of the world, at a particular level of granularity. In reality, no model, with the possible exception of extremely detailed models of quantum interactions, will truly contain all direct causes of a given observable effect. Instead, whenever a given cause explicitly named in a model is fixed, an untold number of intermediate causes and effects omitted from the model operate, according to natural laws, to bring about the explicitly named effect. Neither the notion of "direct cause," nor the intuitive notion of modularity of causal influence, is absolute but dependent on the model. Nevertheless, the notion of modularity is meaningful when applied to a particular model, since it implies a much weaker claim, namely that the knowledge of all causes considered direct for a particular effect in the model implies no other variable in the model can influence that effect.

These properties of directionality and modularity can be naturally expressed using directed graphs. Perhaps due to the intuitiveness of such a visual representation, the use of directed graphs to represent causality is an idea that arose multiple times, in genetics [Wri21], econometrics [Haa43], and artificial intelligence [Pea88], [SGS93], [Pea00]. In each case, variables of interest were represented as nodes in the graph, while an arrow from parent to child node stood for a direct cause-effect relationship between the corresponding variables. Associated with each node is an autonomous causal mechanism, independent of other such mechanisms, which determined the value of that node depending on the values of its parents in the graph. Directed graphs with this kind of interpretation are called *causal diagrams*, and the causal domains they represent are called *graphical causal models* [Pea00].

1.2 The Causal Hierarchy

An example of a graphical causal model is an electronic circuit. In a circuit, causal mechanisms correspond to logic gates, while variables are input and output wires, along with intermediate values computed by logic gates. Circuits and propositional logic in general have been applied to a wide variety of problems. Nevertheless, our knowledge of many interesting domains such as medicine, law, social interactions, economics, and so on is incomplete. Our ignorance manifests in two ways. Firstly, we rarely understand specific causal mechanisms so well that we can describe them in terms of a function. Secondly, we rarely observe all causes which help determine observable effects in our models. In order to construct causal models faithful to the realities of our ignorance, we need to handle uncertainty; the mathematical framework used for this purpose is probability theory. Fortunately, the framework of graphical causal models can be easily extended to handle uncertainty. We model unobserved causes with observable effects by considering certain root nodes in the graph as unobservable, while ignorance of functional mechanisms can be represented by only exposing coarser features of the model than causal mechanisms themselves, for example, conditional probabilities of observing particular values given some input values. To handle our uncertainty in a principled way, we endow unobservable nodes with a probability distribution. This unobservable distribution, together with unknown causal mechanisms specified in the model induce a probability distribution over observable variables. This distribution is generally accessible, since we are free to collect statistics pertaining to the observable parts of our domains.

A wide variety of causal queries, such as those concerning effects of actions, or counterfactual situations are represented as probability distributions ultimately derived from unobserved variables and causal mechanisms. I will consider a hierarchy consisting of three kinds of causal queries in graphical causal models. The lowest level in the hierarchy consists of what I call *associational questions.* A typical question of this sort is "I have taken an aspirin an hour ago. How likely am I to get a headache?" Such questions are represented as marginal or conditional distributions over observable quantities (e.g., $P(headache|aspirin)$), and can be computed from the joint distribution over all variables in the domain. Much research in statistics and artificial intelligence is devoted to finding answers to these sorts of questions when the knowledge of the joint distribution is constrained by missing or limited information. It is well-known that association does not imply causation, and associational queries are therefore not strictly speaking causal. Nevertheless, I place such queries at the base of the hierarchy because techniques developed for answering them will be invaluable for computing answers to more intricate questions, and because associational statements form an easily available base from which such computations can begin.

Placed above associational questions in the hierarchy are questions about effects of interventions imposed on the causal model from the outside. Interventions disrupt the normal flow of influence from causes to effects by setting some set of variables to specific values, regardless of what the normal causes of that set dictate. An example of a question that involves effects of interventions is "I am about to take an aspirin. Will it help my headache?" Here I model a decision to take medicine as disrupting the normal schedule of daily food intake. I denote interventions using the $do(.)$ notation used by [Pea00], where $do(\mathbf{x})$ means that a set of variables $\mathbf{X}$ is set to values $\mathbf{x}$. The effects of interventions will be represented using *interventional distributions* denoted with either the $do(.)$ operator past the conditioning bar or a subscript denoting a set of intervened values (e.g.,

$P(\mathbf{y}|do(\mathbf{x}))$, or $P_{\mathbf{x}}(\mathbf{y})$). The effect of intervention $do(\mathbf{x})$ on a variable set $\mathbf{Y}$ is often called the *causal effect* of $do(\mathbf{x})$ on $\mathbf{Y}$.

The final set of questions, placed above both interventional and associational queries in the hierarchy, involves hypothetical, "what-if" situations. An example of such a *counterfactual* question would be "I took an aspirin and my headache is gone; would I have a headache had I not taken an aspirin?" As their name implies, counterfactuals often involve conflicts between the true state of affairs and the hypothetical situation involved in the question. Despite these conflicts, human beings frequently invoke and evaluate counterfactuals both in everyday situations, and in technical domains. Nevertheless, it is not obvious how to answer counterfactual questions correctly without complete knowledge of all aspects of a causal model. Since some aspects of a causal model may not be experimentally testable, the use of counterfactuals has been the subject of some criticism [Daw00]. I will represent counterfactuals as joint or conditional distributions over sets of events resulting from multiple, possibly conflicting interventions.

I also consider a special class of effect queries known as path-specific effects. Such queries arise in situations where we want to know the effect of a given intervention $do(x)$ on the outcome Y, but only along certain causal paths. These sorts of effects come up often in policy analysis [Pea01], and in legal cases. For instance, gender discrimination occurs if a person's gender has a direct effect on the hiring decision. However, it is permissible that gender influence certain factors which themselves have a strong influence on a person's suitability for the job. For example, women may, on average, be more affable than men in customer-facing situations. In evaluating claims of discrimination, we are interested in determining whether gender had no direct effect hiring, while possibly having an indirect effect. Despite calling these kinds of queries path-specific effects, I will show later that they can be computed from counterfactuals, and so properly belong in the third level of the causal hierarchy.

1.3 Identification

This thesis is concerned with answering questions in the causal hierarchy. The answering strategies available to us naturally depend on the complexity of the question. Associational questions involving certain observable variables, such as *headache*, and *aspirin*, can be computed from the joint probability distribution over all observables in the domain, using basic probability theory. In practice the joint probability distribution is generally not available, and must instead be estimated, using techniques developed in statistics and artificial intelligence. However, for the purposes of this thesis, I simplify the task by assuming that we are given the true probability distribution representing the domain, rather than an approximation obtained from some estimation procedure using a finite set of samples. Given this assumption, it is a simple matter to compute an arbitrary associational question from the corresponding joint distribution.

Computing causal effects is a more difficult task because interventions change probability distributions. The stochastic behavior of the original domain, summarized by the joint distribution over the observable variables, cannot be translated in a straightforward way to the stochastic behavior of the post-intervention domain, represented by the interventional distribution.

There are two main approaches to computing causal effects. The first is the direct approach: implement the intervention $do(\mathbf{x})$ directly in an individual, circuit, living cell, etc. and observe the consequences. More generally, if we want to compute the effect of an intervention in a population, we can perform a *randomized experiment* [Fis26] where every member of the population in question is randomly assigned either to the group subjected to the manipulation, or the control group where no manipulation is performed. Needless to say, in most situations of interest, direct manipulation is not possible (e.g., no way to manipulate gender), too expensive (e.g., public policy changes), or unethical (e.g., manipulation of human bodies in medicine). It is desirable, then, to use a less direct approach to inferring causal effects.

The second approach involves finding a way to link the effect of an intervention with the probability distribution associated with the original, unmanipulated model. If such a link can be found, it becomes possible to compute causal effects from observational studies alone, without performing randomized experiments or manipulations of any kind. This approach to causal inference bears a striking resemblance to logical inference: we have some premises, in this case an observational distribution, and we are interested in computing conclusions of interest, or more generally as many conclusions as possible. However, unlike conventional logical inference, we are not operating over sentences in a particular logic, but instead over probability distributions, using axioms of probability and perhaps additional rules specific to graphical causal models. Causal inference of this sort is called *identification* [Pea95], [Pea00].

Though identification was the framework used in the literature to compute causal effects from observations, it is a more general notion which can be applied any time we wish to deduce conclusions from premises in some set of models. I will use this generality to answer not only questions involving causal effects, but also counterfactuals. In this thesis, I view counterfactuals as distributions which span multiple hypothetical worlds, often with contradictory features (e.g., in one world aspirin was taken, in another it was not). We could consider the version of the identification problem analogous with causal effects, where we try to determine which counterfactuals can be computed from observational distributions. However, even if we permit ourselves to perform arbitrary experiments, it's unclear how we could evaluate counterfactual questions with such conflicts, since, for example, no experimental setup exists which both gives and doesn't give someone aspirin. To simplify, I will consider the following identification problem: assuming we allow ourselves any experiment in a given causal model, represented by the set of all possible interventional distributions in this model, can we infer a given counterfactual? Of course, if I can express a counterfactual in terms of some set of interventional distributions, those distributions may, in turn, be expressible in terms of observational distributions. In this case I will be able to identify a counterfactual from observations. I will consider a similar identification problem for path-specific effects, which are a particular kind of counterfactual.

1.4 Dormant Independence

Answering causal questions from observational studies using graphical causal assumptions is an important problem in itself, however advances in this area also have useful applications for inducing and testing causal theories expressed as causal graphs. A given causal graph constrains probability distributions in any model consistent with this graph in two ways. Firstly, such distributions all contain certain conditional independencies which can be read off from the graph using the notion of d-separation [Pea86], [Ver86], [Pea88], which I will discuss in Chapter 3. Secondly, such distributions also obey certain algebraic constraints, noted by Verma [VP90].

Conditional independence constraints are relatively well-understood and frequently used by causal induction algorithms, such as **IC** [VP90], [Pea00], and **FCI** [SGS93]. For instance, such algorithms are able to conclude in certain classes of models that two nodes X and Y are not connected by an edge in a causal diagram if the corresponding random variables are conditionally independent in the observed distribution. On the other hand, algebraic constraints are still relatively poorly understood and seldom used for induction and testing.

I will consider a special subset of algebraic constraints which is easy to understand and apply, and which arises from "dormant independencies," in other words independencies that prevail in post-intervention distributions. I will develop a complete algorithm for determining if a conditional independence exists between two sets of variables in an interventional distribution which is also identifiable, and show how this algorithm can be used to test certain features of causal diagrams which ordinary conditional independence cannot test.

1.5 Thesis Outline

This thesis is organized as follows. Chapter 2 discusses related work in graphical models and causal inference which lead to the questions considered in this thesis. Chapter 3 precisely defines graphical causal models, the hierarchy of causal queries I consider, the notion of identification which I will use to answer these queries, and other mathematical machinery needed to obtain my results. Chapter 4 considers the problem of identifying causal effects from observational studies. Chapter 5 considers the problem of identifying counterfactuals from experimental studies. Chapter 6 generalizes the notion of causal effect to the situation where we are interested only in certain paths, and considers the problem of identifying such path-specific effects. Chapter 7 considers the problem of determining if an identifiable dormant independence exists between two sets of variables, and how to use dormant independencies to test features of the causal graph. Chapter 8 is the conclusion.

CHAPTER 2

Related Work

In this chapter, I overview the conceptual developments over the last century that culminated in the modern understanding of causal inference.

2.1 Graphical Models

Causal modeling using directed graphs started with the seminal work of Sewall Wright on path analysis [Wri21]. Linear models considered by Wright became the subject of study in the statistics community under the name of *Structural Equation Models* [Wri21], [Haa43], [Kli05]. More recently, the use of graphs to represent uncertainty became popular in the fields of artificial intelligence and statistics with the introduction of *Bayesian Networks* [Pea85], [Pea88], [LS88], [Lau96].

It soon became apparent that the use of graphs to represent uncertainty is a powerful idea which arose multiple times and the emerging formalism of *Graphical Models* [JW02] subsumed many special cases developed in separate disciplines, such as Kalman filters in engineering [Kal60], Markov random fields in physics [Bes74], and statistical mechanics [Bax92], hidden Markov models in signal processing [Rab89], and many others [RG99]. Common to these approaches is the decomposition of the joint probability distribution representing the domain of interest into tractable pieces, and the use of graphs to mirror this decomposition via various Markov properties. Most graphical models serve as a compact representation of the underlying distribution, and do not make any causal claims, though causal knowledge is often used in their construction.

2.2 Causal Inference

More recent work [VP90], [Pea93a], [SGS93], [Pea95], [Pea00] has added a causal interpretation to graphical models, with directed arrows in the graph being interpreted as causal influence between variables. This interpretation allowed formalization of *causal inference*, posing and answering an additional class of causal questions, such as interventional and counterfactual queries I discussed in the introduction. Interventional queries $P(\mathbf{y}|do(\mathbf{x}))$ represent the notion of *causal effects*, which is ubiquitous in both informal and professional discourse, and forms an important building block from which our understanding of the world is built. While randomized experiments can often be used to estimate causal effects, in practice such experiments can be expensive to conduct. Furthermore, certain forms of experimentation (e.g., drug testing, surgical alteration, etc.) may be illegal or unethical to conduct on human subjects. It is desirable, therefore, to determine conditions under which a given causal effect can be computed from observational studies, which are generally less expensive to conduct, less objectionable on human subjects, and therefore more common. The formal problem of characterizing models where queries of interest may be computable from limited information is known as the *identification problem* [Pea95], [Pea00]. Identification of causal effects has received considerable attention in the literature, with two approaches being dominant. The first approach deals not with causal models themselves, but with causal diagrams, and attempts to derive graphical conditions a model must satisfy before a given causal effect can be computed. A number of such graphical conditions are known, for example the Back-Door Criterion [Pea93b], and the Front-Door Criterion [Pea95]. While these two conditions are intuitive and easy to state, their suffer from the problem of limited applicability. The second approach views causal inference as a special case of logical inference, and attempts to derive axioms to codify behavior of quantities derived from causal models, and rules of inference to reason about such quantities appropriately. [GP98], [Hal00] proposed a complete set of axioms for causal inference, while [Pea93c] proposed a set of three rules of do-calculus for reasoning about interventional distributions. While the resulting reasoning systems are more general, the constructed proofs can be difficult for the unaided mind to follow. Moreover such systems suffer from standard difficulties of theorem proving: large search spaces of possible proofs, and lack of termination guarantees. The algorithms in this thesis, which can be viewed as simplifications and elaborations of Jin Tian's original algorithms for causal effect identification [TP02], [Tia04], [Tia02], combine the strengths of both approaches – we can derive intuitive graphical conditions while at the same time retaining the generality, in fact completeness, of the logical methods. A number of interesting corollaries follow from the completeness of these algorithms. For instance, my results imply that do-calculus is complete for identifying all causal effect queries. Some of these results and corollaries were derived independently elsewhere [HV06b], [HV06a].

2.3 Potential Outcomes and Counterfactuals

Another strand of work on causal modeling did not employ graphs and dealt with the so called *potential response variables* [Ney23], [Rub74], written as $Y_x(u)$ or $Y(x, u)$. This notation is taken to mean "the value attained by Y in unit u under intervention $do(x)$." If the domain is not observable at the unit level, we can average over possible units to attain random variables Y_x which I will call *counterfactual variables*, since they can be viewed as responses to hypothetical interventions. Research in the potential response framework has sought to establish rules governing such variables, and the way these variables relate to those actually observed. Important causal assumptions such as *exogeneity* can be expressed in terms of probabilistic independence among certain counterfactual variables [Pea00], while evaluation of causal effects based on *g-estimation* [Rob87] assumes that such counterfactual independencies hold. Recent work on axiomatizing causal reasoning [GP98], [Pea00], [Hal00] has shown that the framework of potential outcomes and the framework of graphical causal models both describe the same mathematical objects, probability distributions over counterfactual variables. This unification allowed the expression of counterfactual independence in terms of graphs, and evaluation of counterfactual queries themselves if all parameters in a causal model are known [BP94a], [BP94b]. I provide a generalization of this approach by providing a graphical representation of independence among counterfactual variables in an arbitrary number of hypothetical worlds, and provide complete algorithms for evaluating counterfactuals from experimental studies. The results of such studies are more likely to be available than complete knowledge of all model parameters as required by previous work [BP94b].

2.4 Natural and Path-specific Effects

[RG92] and [Pea01] introduced the notion of direct and indirect effects, meant to represent cases where we are interested in the effect of an intervention $do(x)$ on an outcome variable Y, but only along certain causal paths. Such cases arise, for instance, when discussing discrimination, where the question is whether a given characteristic, say gender, has a direct effect on the decision (e.g., hiring, admission, lease, etc.) I say direct effect because indirect effect of gender on hiring does not constitute discrimination. For instance, an employer may hire a greater percentage of women, if women are more qualified on average than men, and this would not necessarily be considered discriminatory. Formalizing the notion of direct effect, where indirect effects are "forbidden," or an indirect effect where direct effects are "forbidden," requires probabilities over nested counterfactual variables [Pea01]. [Pea01] further provides some conditions where such effects can be identified from the causal graph and observational studies. Subsequently, [Pea01] and [ASP05] consider a generalization of natural effects to cases where arbitrary sets of edges are "forbidden." These generalized natural effects are termed *path-specific effects*. In this thesis, I will provide a complete method for identifying such effects in causal diagrams without latent variables, along with a simple graphical characterization of such identifiable path-specific effects. [1] Furthermore, I will use the results on identifying counterfactual distributions to provide identification criteria for path-specific effects in semi-Markovian causal diagrams.

2.5 Algebraic Constraints and Causal Induction

One of the most important problems in causal inference is the problem of causal induction, namely inferring aspects of the causal model, such as the graph, from observations. Inferring the structure of graphical models has a long history in Artificial Intelligence, with two approaches being dominant. The score-based approach [Suz93], [LB94] assigns a score to each possible causal structure, where "small" structures, and structures likely given the observed data are given high scores. Score-based algorithms perform a search for high scoring structures. The constraint-based approach rules out causal structures which are inconsistent with various constraints imposed by the observed data. Well-known constraint-based algorithms are the **IC** algorithm [VP90], [Pea00] and the **FCI** algorithm [SGS93]. These algorithms return a set of all causal graphs which have the same set of d-separation statements (and the corresponding independencies) as the graph of the model which generated the observed distribution.

Constraint-based induction algorithms generally only make use of constraints implied by conditional independencies, although causal graphs entail a wider class of algebraic constraints, first noted in [VP90]. I extend the identification results in this thesis to show that a special subset of such algebraic constraints is obtained from conditional independence in interventional distributions, which I call "dormant independence." Although full use of dormant independence for causal induction remains an open problem, I show how this kind of independence can be used for model testing by giving an algorithm which uses dormant independence to rule out extraneous edges from causal graphs.

[1] Some of these results were derived as a joint work with Chen Avin

CHAPTER 3

Notation and Definitions

In this chapter I go over the definitions and mathematical machinery used in causal inference.

3.1 Causal Models and Causal Diagrams

The primary object of causal inquiry is a probabilistic causal model. I will denote variables by uppercase letters, and their values by lowercase letters. Similarly, sets of variables will be denoted by bold uppercase, and sets of values by bold lowercase.

Definition 1 *A probabilistic causal model (PCM) is a tuple* $M = \langle \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{F}, P(\boldsymbol{u}) \rangle$*, where*

- $\boldsymbol{U}$ *is a set of background or exogenous variables, which cannot be observed or experimented on, but which affect the rest of the model.*
- $\boldsymbol{V}$ *is a set* $\{V_1, ..., V_n\}$ *of observable or endogenous variables. These variables are functionally dependent on some subset of* $\boldsymbol{U} \cup \boldsymbol{V}$*.*
- $\boldsymbol{F}$ *is a set of functions* $\{f_1, ..., f_n\}$ *such that each* f_i *is a mapping from a subset of* $\boldsymbol{U} \cup \boldsymbol{V} \setminus \{V_i\}$ *to* V_i*, and such that* $\bigcup \boldsymbol{F}$ *is a function from* $\boldsymbol{U}$ *to* $\boldsymbol{V}$*.*
- $P(\boldsymbol{u})$ *is a joint probability distribution over* $\boldsymbol{U}$*.*

The set of variables $\mathbf{V}$ in this definition represents the part of the causal domain we can see and experiment on, the set of functions $\mathbf{F}$ corresponds to the causal mechanisms which determine the values of $\mathbf{V}$, while $\mathbf{U}$ represents the background context that influences $\mathbf{V}$, yet remains outside it. Our ignorance of the background context is represented by a distribution $P(\mathbf{u})$. This distribution, together with the mechanisms in $\mathbf{F}$, induces a distribution $P(\mathbf{v})$ over the observable domain.

The causal diagram, our vehicle for expressing causal assumptions, contains two kinds of edges: directed edges which represent direct causal relationships, and

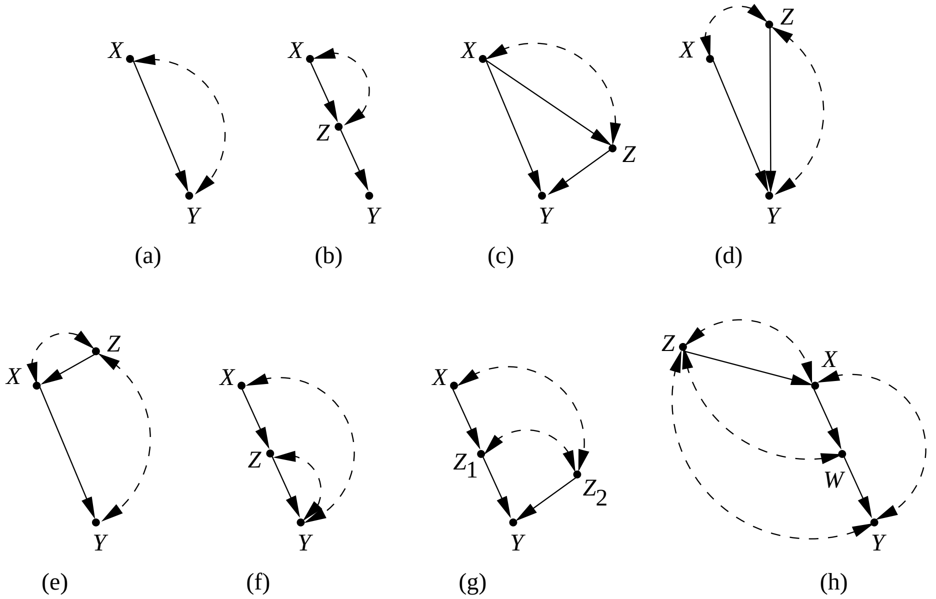


Figure 3.1: Causal graphs where $P(y|do(\mathbf{x}))$ is not identifiable

bidirected edges which represent "non-causal dependence," or confounding. A causal diagram is defined by the causal model as follows. Each observable variable $V_i \in \mathbf{V}$ corresponds to a vertex in the graph. Any two variables $X \in \mathbf{U} \cup \mathbf{V}$, $V_j \in \mathbf{V}$ such that X appears in the description of f_j are connected by a directed arrow from X to V_j. In this thesis, we assume that all U variables are mutually independent, in other words $P(\mathbf{u}) = \prod_i P(u_i)$, and that each $U_i \in \mathbf{U}$ appears in at most two functions in $\mathbf{F}$. [1] If there is some $U_k \in \mathbf{U}$ which appears in the functions f_i and f_j of two observable nodes V_i, V_j, instead of drawing two directed arcs from U_k to V_i and V_j, we can draw a bidirected arc between V_i and V_j and omit U_k from the graph entirely. Similarly, U variables with a single child can be omitted from the graph. The graph defined in this way from a causal model M is said to be *induced* by M. Fig. 3.1 and Fig. 3.2 show some examples of causal diagrams. I will only consider *recursive* causal models, those models which induce acyclic directed graphs.

In the remainder of this thesis, I will make heavy use of standard graph-theoretic "family relations." Specifically, $Pa(\mathbf{X})_G, Ch(\mathbf{X})_G, De(\mathbf{X})_G, An(\mathbf{X})_G$ stands for the set of parents, children, descendants and ancestors (respectively) of the node set $\mathbf{X}$ in the graph G. We view $De(.)$ and $An(.)$ as inclusive relations, in other words, $X \in De(\mathbf{X})$ and $X \in An(\mathbf{X})$, for any $X \in \mathbf{X}$.

[1] Most of the results in this thesis do not depend on this, and can easily be extended to the general case of the same U variable influencing multiple functions. Similarly, if some U variables are dependent, this dependence can be represented by bidirected arcs

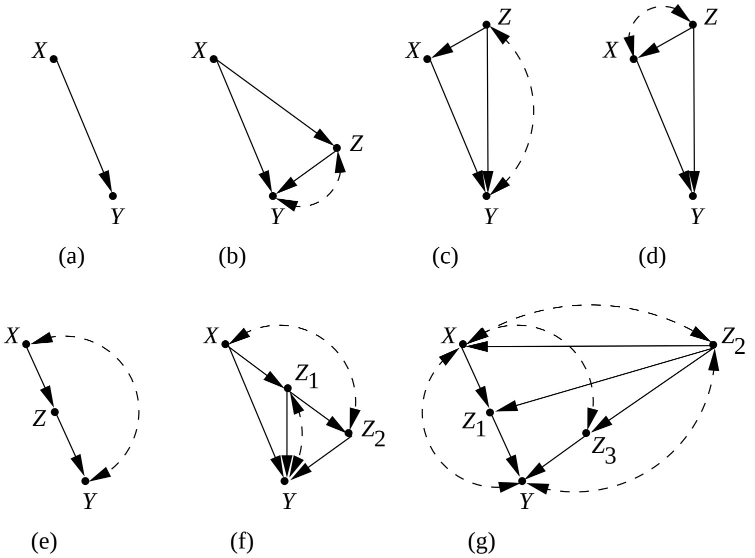


Figure 3.2: Causal graphs where $P(y|do(\mathbf{x}))$ is identifiable

3.2 Interventions and Intervention-based Queries

The functions in $\mathbf{F}$ are assumed to be *modular* in a sense that changes to one function do not affect any other. [2] This assumption allows us to model effectively how a PCM would react to changes imposed from the outside. The simplest change that is possible for causal mechanisms of a variable set $\mathbf{X}$ would be one that removes the mechanisms entirely and sets $\mathbf{X}$ to specific values $\mathbf{x}$. This change, denoted by $do(\mathbf{x})$ [Pea00], is called an *intervention.* [3] An intervention $do(\mathbf{x})$ applied to a model M results in a *submodel* $M_{\mathbf{x}}$. The effects of interventions will be formulated in several ways. For any given $\mathbf{u}$, the effect of $do(\mathbf{x})$ on a set of variables $\mathbf{Y}$ will be represented by *counterfactual variables* $Y_{\mathbf{x}}(\mathbf{u})$, where $Y \in \mathbf{Y}$. Sometimes we will write a set of counterfactual variables $Y^1_{\mathbf{x}}, ...Y^k_{\mathbf{x}}$ with the same subscript as $\mathbf{Y}_{\mathbf{x}}$, where $\mathbf{Y} = \{Y^1, ..., Y^k\}$. As $\mathbf{U}$ varies, the counterfactuals $Y_{\mathbf{x}}(\mathbf{u})$ will vary as well, and their *interventional distribution*, denoted by $P(\mathbf{y}|do(\mathbf{x}))$ or $P_{\mathbf{x}}(\mathbf{y})$ will be used to specify the effect of $\mathbf{x}$ on $\mathbf{Y}$. I will denote the proposition "variable Y attains value y in $M_{\mathbf{x}}$" by the shorthand $y_{\mathbf{x}}$.

Interventional distributions are a mathematical formalization of an intuitive

[2] This does not preclude functions from sharing parameters. The only requirement is that external manipulation of arguments of one function does not affect other functions, except through the output of the function being manipulated.

[3] The simplicity and determinism of the $do(.)$ operator sometimes draw criticism. In reality, it is no simpler to develop complex accounts of change and causation without dealing with something like the $do(.)$ operator, than it is to understand the richness of chemistry without understanding the "simple" elements of the periodic table

notion of "effect of action." I now define joint probabilities on counterfactuals, in multiple worlds, which will serve as the formalization of counterfactual queries. Consider a conjunction of events $\gamma = y^1_{\mathbf{x}^1} \wedge ... \wedge y^k_{\mathbf{x}^k}$. If all the subscripts $\mathbf{x}^i$ are the same and equal to $\mathbf{x}$, γ is simply the set of values that variables take on in $M_{\mathbf{x}}$, and $P(\gamma) = P_{\mathbf{x}}(y^1, ..., y^k)$. However, if the actions $do(\mathbf{x}^i)$ are not the same, and potentially contradictory, a single submodel is no longer sufficient. Instead, γ is invoking multiple causal worlds, each represented by a submodel $M_{\mathbf{x}^i}$. I assume each submodel shares the same set of exogenous variables $\mathbf{U}$, corresponding to the shared "causal context" or background history of the hypothetical worlds. Because the submodels are linked by common context, they can really be considered as one large causal model, with its own induced graph, and joint distribution over observable variables. $P(\gamma)$ can then be defined as a marginal distribution in this causal model. Formally, $P(\gamma) = \sum_{\{\mathbf{u}|\mathbf{u} \models \gamma\}} P(\mathbf{u})$, where $\mathbf{u} \models \gamma$ is taken to mean that each variable assignment in γ holds true in the corresponding submodel of M when the exogenous variables $\mathbf{U}$ assume values $\mathbf{u}$. In this way, $P(\mathbf{u})$ induces a distribution on all possible counterfactual variables in M. I will represent counterfactual utterances by joint distributions such as $P(\gamma)$ or conditional distributions such as $P(\gamma|\delta)$, where γ and δ are conjunctions of counterfactual events. [Pea00] (chapter 7) discusses counterfactuals, and their probabilistic representation in greater depth.

Finally, I define path-specific effects, which represent situations where we are interested in the effect of $do(\mathbf{x})$ on $\mathbf{Y}$ along only certain causal paths. Graphically, we can represent path-specific effects in some causal model M by considering the causal diagram G of M, where certain edges are marked as forbidden. Intuitively, we would like the "flow of influence" to proceed "downward" along causal paths from $do(\mathbf{x})$ to $\mathbf{Y}$, just as in regular causal effects, but not along forbidden edges. How can we prevent flow along a particular edge? We can remove forbidden edges from the graph, but causal diagrams aren't just arbitrary graphs, the edges represent the participation of the parent in the causal mechanism of the child. The removal of the edge must correspond to a well-defined of change of the corresponding function.

Following [Pea01], I define this change as follows. For each variable W, let $Pa(W)$ be divided into two sets, $Pa^+(W)$ is the set of parents connected to W by "allowed" edges, and $Pa^-(W)$ is the set of parents connected to W by "forbidden" edges. Let $\mathbf{x}^*$ be the reference values of $\mathbf{X}$. For the purposes of determining the value of W, we want $Pa^-(W)$ to behave as if $\mathbf{X}$ was set to $\mathbf{x}^*$. The follow formal definition is a generalization of the one found in [Pea01], which was applicable to a single effect variable X and single outcome variable Y.

Definition 2 (path-specific effect) *Let G be a causal diagram induced from a model M, $\boldsymbol{Y}, \boldsymbol{X}$ sets of variables, $\boldsymbol{x}, \boldsymbol{x}^*$ values of $\boldsymbol{X}$. Let g be the subset "allowed"*

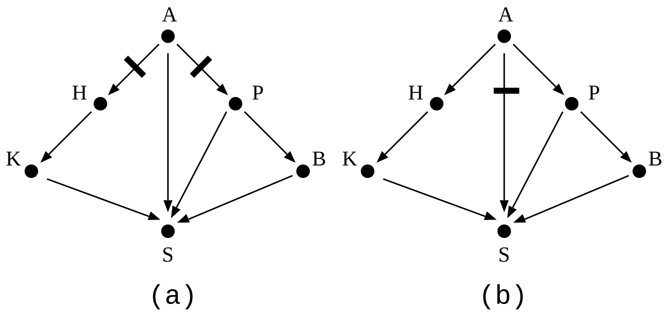


Figure 3.3: Path-specific effects of $do(a)$ on S

edges for the flow of effect from $do(\boldsymbol{x})$ *to* $\boldsymbol{Y}$*. Let* M_g *be defined as follows. For each observable* W*, if* $W \in \boldsymbol{X}$*, replaced* f_W *by a constant function which returns the corresponding value of* W *in* $\boldsymbol{x}$*. Otherwise, replace* f_W *by another function* f^g_W *which maps* $Pa^+(W)$ *to* W *as follows:* $f^g_W(pa^+(w), \boldsymbol{u}) = f_W(pa^+(w), pa^-(w)^*, \boldsymbol{u})$*, where* $pa^-(w)^*$ *are the values obtained by* $Pa^-(W)$ *under intervention* $do(\boldsymbol{x}^*)$*. The path-specific effect* $PSE_g(\boldsymbol{x}, \boldsymbol{x}^*; \boldsymbol{Y}, \boldsymbol{u})$ *is defined to equal* $\boldsymbol{Y}_{\boldsymbol{x}}(\boldsymbol{u}) - \boldsymbol{Y}_{\boldsymbol{x}^*}(\boldsymbol{u})$*, where both counterfactual value sets are from* M_g*.*

If we wish to summarize the path-specific effect over all settings of $\mathbf{u}$, we should resort to the expectation of the above difference, or the expected path-specific effect. To identify this effect, we need to identify $P(\mathbf{y}_{\mathbf{x}})$ and $P(\mathbf{y}_{\mathbf{x}^*})$ in M_g. For our purposes we can restrict our attention to $P(\mathbf{y}_{\mathbf{x}})$, as the second term corresponds to the quantity $P(\mathbf{y}_{\mathbf{x}^*})$ in the original model M, which corresponds to an ordinary causal effect expression $P(\mathbf{y}|do(\mathbf{x}^*))$.

Path-specific effects, despite their name, are more akin to counterfactuals than causal effects. This is because the same variable can behave as if the intervention $do(\mathbf{x})$ was performed with respect to some edges, and at the same time behave as if the intervention $do(\mathbf{x}^*)$ was performed with respect to other edges. For instance the variable A in Fig. 6.3 behaves like this. In this way a single path-specific effect involves random variables from different submodels which disagree on variable settings; the same is true of counterfactual distributions.

3.3 Identification

A fundamental question in causal inference is whether a given causal query, either interventional or counterfactual in nature, can be uniquely specified by the assumptions embodied in the causal diagram, and easily available information, usually observational, associated with the causal model. To get a handle on this question, I introduce the important notion of *identifiability* [Pea95], [Pea00].

Definition 3 (identifiability) *Consider a class of models $\boldsymbol{M}$ with a description T, and two objects ϕ and θ computable from each model. I say that ϕ is θ-identified in T if ϕ is uniquely computable from θ in any $M \in \boldsymbol{M}$. In other words all models in $\boldsymbol{M}$ which agree on θ will also agree on ϕ.*

If ϕ is θ-identifiable in T, I write $T, \theta \vdash_{id} \phi$. Otherwise, I write $T, \theta \nvdash_{id} \phi$. The above definition leads immediately to the following corollary which we will use to prove non-identifiability results.

Corollary 1 *Let T be a description of a class of models $\boldsymbol{M}$. Assume there exist $M^1, M^2 \in \boldsymbol{M}$ that share objects θ, while ϕ in M^1 is different from ϕ in M^2. Then $T, \theta \nvdash_{id} \phi$.*

In our context, the objects ϕ, θ are probability distributions derived from the PCM, where θ represents available information, while ϕ represents the quantity of interest. The description T is a specification of the properties shared by all causal models under consideration, in other words, the set of assumptions we wish to impose on those models. Since I chose causal graphs as a language for specifying assumptions, T would correspond to a given graph.

3.4 D-separation

Next, I briefly review the standard results which link directed graphs with independencies in probability distributions. Graphs earn their ubiquity as a specification language because they reflect in many ways the way people store experiential knowledge, especially cause-effect relationships. The ease with which people embrace graphical metaphors for causal and probabilistic notions – ancestry, neighborhood, flow, and so on – are proof of this affinity, and help ensure that the assumptions specified are meaningful and reliable. A consequence of this is that probabilistic dependencies among variables can be verified by checking if the "flow of influence" is *blocked* along paths linking the variables. By a path I mean a sequence of distinct nodes where each node is connected to the next in the sequence by an edge. The precise way in which the flow of dependence can be blocked is defined by the notion of d-separation [Pea86], [Pea88].

Definition 4 (d-separation) *A path p in G is said to be d-separated by a set $\boldsymbol{Z}$ if and only if either*

1 p contains one of the following three patterns of edges: $I \rightarrow M \rightarrow J$, $I \leftrightarrow M \rightarrow J$, or $I \leftarrow M \rightarrow J$, such that $M \in \boldsymbol{Z}$, or

2 p contains one of the following three patterns of edges (called colliders): $I \rightarrow M \leftarrow J$, $I \leftrightarrow M \leftarrow J$, $I \leftrightarrow M \leftrightarrow J$, such that $De(M)_G \cap \boldsymbol{Z} = \emptyset$.

Two sets $\mathbf{X}, \mathbf{Y}$ are said to be d-separated given $\mathbf{Z}$ in G if all paths from $\mathbf{X}$ to $\mathbf{Y}$ in G are d-separated by $\mathbf{Z}$. Paths or sets which are not d-separated are said to be d-connected. What allows us to connect this notion of blocking of paths in a causal diagram to the notion of probabilistic independence among variables is that the probability distribution over $\mathbf{V}$ and $\mathbf{U}$ in a causal model can be represented as a product of factors each of which is a conditional distribution of a given node given the values of its parents in the graph. In other words, $P(\mathbf{v}, \mathbf{u}) = \prod_i P(x_i | pa(X_i)_G)$, where $pa(X_i)_G$ is the values of the set of parents of X_i in G. Whenever this property holds, it is said that G is an I-map [Pea88] of P. The following well known theorem [VP88] links d-separation of vertex sets in an I-map G with the independence of corresponding variable sets in P.

Theorem 1 *If sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$ in G, then $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ given $\boldsymbol{Z}$ in every P for which G is an I-map. Furthermore, the causal diagram induced by any PCM M is an I-map of the distribution $P(\boldsymbol{v}, \boldsymbol{u})$ induced by M.*

Proof: It is not difficult to see that if I restrict d-separation queries to a subset of variables $\mathbf{W}$ in some graph G, the corresponding independencies in $P(\mathbf{w})$ will only hold whenever the d-separation statements hold. Furthermore, if I replace G by a latent projection L [PV91], [Pea00], where I view variables $\mathbf{V} \setminus \mathbf{W}$ as hidden, independencies in $P(\mathbf{w})$ will only hold whenever the corresponding d-separation statement (extended to include bidirected arcs) holds in L. □

I will abbreviate the statement of d-separation as $(\mathbf{X} \perp \mathbf{Y} | \mathbf{Z})_G$, and corresponding independence as $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_P$, following the notation of [Daw79].

3.5 Axioms of Causal Inference

Finally I consider the axioms and inference rules that will be needed. Since PCMs contain probability distributions, the inference rules I would use to compute queries in PCMs would certainly include the standard axioms of probability. They also include a set of axioms which govern the behavior of counterfactuals, such as Effectiveness, Composition, etc. [GP98], [Hal00], [Pea00]. However, I will concentrate on a set of three identities applicable to interventional distributions known as do-calculus [Pea93c], [Pea00]:

- Rule 1: $P_{\mathbf{x}}(\mathbf{y} | \mathbf{z}, \mathbf{w}) = P_{\mathbf{x}}(\mathbf{y} | \mathbf{w})$ if $(\mathbf{Y} \perp \mathbf{Z} | \mathbf{X}, \mathbf{W})_{G_{\overline{\mathbf{X}}}}$

- Rule 2: $P_{\mathbf{x},\mathbf{z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{z},\mathbf{w})$ if $(\mathbf{Y} \perp \mathbf{Z}|\mathbf{X},\mathbf{W})_{G_{\overline{\mathbf{x}},\underline{\mathbf{z}}}}$
- Rule 3: $P_{\mathbf{x},\mathbf{z}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$ if $(\mathbf{Y} \perp \mathbf{Z}|\mathbf{X},\mathbf{W})_{G_{\overline{x},\overline{z(\mathbf{w})}}}$

where $Z(\mathbf{W}) = \mathbf{Z} \setminus An(\mathbf{W})_{G_{\overline{\mathbf{X}}}}$. $An(\mathbf{W})_G$ is the set of ancestors of the set $\mathbf{W}$ in G, $G_{\overline{\mathbf{x}},\underline{\mathbf{y}}}$ stands for a directed graph obtained from G by removing all incoming arrows to $\mathbf{X}$ and all outgoing arrows from $\mathbf{Y}$. The rules of do-calculus provide a way of linking ordinary statistical distributions with distributions resulting from various manipulations.

CHAPTER 4

Causal Effects

In this chapter, I consider the problem of identifying causal effects from statistical knowledge, represented by the observational distribution, and causal assumptions encoded in a causal diagram. Starting with simplest graphs, I develop an interpretation of causal effect as resulting from a specific kind of flow of probabilistic influence along edges in the graph. I introduce successively more complicated techniques which recover causal effects from observational distributions in successively more complicated graphs. At the same time, I show that in various classes of graphs certain causal effects cannot be identified by any means. These developments culminate in an algorithm which either identifies a given causal effect, or this causal effect cannot be identified by any means in the causal diagram given. Finally, I provide a simple extension to handle conditional interventional distributions, and provide some important corollaries of my results.

4.1 Identifying Simple Effects in Simple Graphs

Like probabilistic dependence, the notion of causal effect of X on Y has an interpretation in terms of flow. Intuitively, X has an effect on Y if changing X causes Y to change. Since intervening on X cuts off X from the normal causal influences of its parents in the graph, we can interpret the causal effect of X on Y as the flow of dependence which leaves X via outgoing arrows only.

Recall that the ultimate goal is to express distributions of the form $P(\mathbf{y}|do(\mathbf{x}))$ in terms of the joint distribution $P(\mathbf{v})$. The interpretation of effect as downward dependence immediately suggests a set of graphs where this is possible. Specifically, whenever all d-connected paths from $\mathbf{X}$ to $\mathbf{Y}$ are start with an outgoing arrow from $\mathbf{X}$ (following [Pea00], I call such paths front-door), the causal effect $P(\mathbf{y}|do(\mathbf{x}))$ is equal to $P(\mathbf{y}|\mathbf{x})$. In graphs shown in Fig. 3.2 (a) and (b) causal effect $P(y|do(x))$ has this property.

In general, we don't expect acting on $\mathbf{X}$ to produce the same effect as observing $\mathbf{X}$ due to the presence of paths which do not start with an outgoing arrow (I will call such paths back-door as in [Pea00]) between $\mathbf{X}$ and $\mathbf{Y}$. However, d-separation gives us a way to block undesirable paths by conditioning. If we can find a set $\mathbf{Z}$ that blocks all back-door paths from $\mathbf{X}$ to $\mathbf{Y}$, we obtain the following: $P(\mathbf{y}|do(\mathbf{x})) = \sum_{\mathbf{z}} P(\mathbf{y}|\mathbf{z}, do(\mathbf{x}))P(\mathbf{z}|do(\mathbf{x}))$. The term $P(\mathbf{y}|\mathbf{z}, do(\mathbf{x}))$ is reduced to $P(\mathbf{y}|\mathbf{z}, \mathbf{x})$ since the influence flow from $\mathbf{X}$ to $\mathbf{Y}$ is blocked by $\mathbf{Z}$. However, the act of adjusting for $\mathbf{Z}$ introduced a new effect we must compute, corresponding to the term $P(\mathbf{z}|do(\mathbf{x}))$. If it so happens that no variable in $\mathbf{Z}$ is a descendant of $\mathbf{X}$, we can reduce this term to $P(\mathbf{z})$ using the intuitive argument that acting on effects should not influence causes, or a more formal appeal to rule 3 of do-calculus. Computing effects in this way is always possible if we can find a set $\mathbf{Z}$ blocking all back-door paths which contains no descendants of $\mathbf{X}$. This is known as the *back-door criterion* [Pea93b], [Pea00]. Fig. 3.2 (c) and (d) shows some graphs where the node z satisfies the back-door criterion with respect to $P(y|do(x))$, which means $P(y|do(x))$ is identifiable.

The back-door criterion can fail – a common way involves a confounder that is unobserved, which prevents adjusting for it. Surprisingly, it is sometimes possible to identify the effect of $\mathbf{X}$ on $\mathbf{Y}$ even in the presence of such a confounder. To do so, we want to find a set $\mathbf{Z}$ located downstream of $\mathbf{X}$ but upstream of $\mathbf{Y}$, such that the downward flow of the effect of $\mathbf{X}$ on $\mathbf{Y}$ can be decomposed into the flow from $\mathbf{X}$ to $\mathbf{Z}$, and the flow from $\mathbf{Z}$ to $\mathbf{Y}$. Clearly, in order for this to happen $\mathbf{Z}$ must d-separate all front-door paths from $\mathbf{X}$ to $\mathbf{Y}$. However, in order to make sure that the component effects $P(\mathbf{z}|do(\mathbf{x}))$ and $P(\mathbf{y}|do(\mathbf{z}))$ are themselves identifiable, and combine appropriately to form $P(\mathbf{y}|do(\mathbf{x}))$, we need two additional assumptions: there are no back-door paths from $\mathbf{X}$ to $\mathbf{Z}$, and all back-door paths from $\mathbf{Z}$ to $\mathbf{Y}$ are blocked by $\mathbf{X}$. It turns out that these three conditions imply that $P(\mathbf{y}|do(\mathbf{x})) = \sum_{\mathbf{z}} P(\mathbf{y}|do(\mathbf{z}))P(\mathbf{z}|do(\mathbf{x}))$, and the latter two conditions further imply that the first term is identifiable by the back-door criterion and equal to $\sum_{\mathbf{z}} P(\mathbf{y}|\mathbf{z}, \mathbf{x})P(\mathbf{x})$, while the second term is equal to $P(\mathbf{z}|\mathbf{x})$. Whenever these three conditions hold, the effect of $\mathbf{X}$ on $\mathbf{Y}$ is identifiable. This is known as the *front-door criterion* [Pea95], [Pea00]. The front-door criterion holds in the graph shown in Fig. 3.2 (e).

4.2 C-components and General Identification

Unfortunately, in some graphs neither the front-door, nor the back-door criterion hold for an outcome of interest. Yet even in such graphs we can sometimes conclude that the effect is identifiable. Two examples of such graphs are shown in Fig. 3.2 (f) and (g). A general method for identifying effects in such graphs was developed in [TP02], [Tia02]. This method relies on a key graphical structure known as a C-component.

Definition 5 (C-component) *A set of nodes S is a C-component in a graph*

G if any two nodes in S are connected by a path consisting entirely of bidirected arrows in G.

Tian showed that if a given graph G is not a C-component, it can be uniquely partitioned into a set of maximal C-components. Moreover, the observable distribution $P(\mathbf{v})$ of any causal model inducing G can be expressed as a product of interventional distribution terms, where each term corresponds to a C-component, and all such terms are identifiable. This property is known as C-component factorization of causal models.

As an example, the graph in Fig. 3.2 (f) is partitioned into two C-components, the first is the set $\{X, Z_2\}$, and the second is the set $\{Z_1, Y\}$. Moreover, $P(\mathbf{v}) = P_{z_1,y}(x, z_2)P_{x,z_2}(z_1, y)$, and both $P_{z_1,y}(x, z_2)$ and $P_{x,z_2}(z_1, y)$ are identifiable. As we can see from this example, each term in the C-component factorization corresponds to the effect of fixing all variables outside some C-component, on all variables inside this C-component.

C-component factorization is a powerful idea, since it allows us to decompose a complicated identification problem into a set of simpler ones. [TP02] used C-components to give a general algorithm for identifying causal effects which generalizes both the back-door and the front-door criterion, and handles some graphs which fail both of these criteria. In the subsequent sections, I give a somewhat simplified version of Tian's algorithm, and prove it complete. In other words, I show that whenever the algorithm fails to identify an effect in some graph, that effect is not identifiable in every model inducing this graph.

4.3 Simple Non-identifiable Effects

In order to show completeness of causal effect identification, it is necessary to catalogue non-identifiable graphs. The simplest such graph, known as the bow arc graph due to its shape, is shown in Fig. 3.1 (a). The back-door criterion fails for this graph since the confounder node is unobservable, while the front-door criterion fails since no intermediate variables between X and Y exist in the graph. While the failure of these two criteria does not imply non-identification, a simple argument shows that $P(y|do(x))$ is not identifiable in the bow arc graph (see Appendix).

Theorem 2 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y|do(x))$ in G shown in Fig. 3.1 (a).*

Since we are interested in completely characterizing graphs where a given causal effect $P(\mathbf{y}|do(\mathbf{x}))$ is identifiable, it would be desirable to list difficult graphs like the bow arc graph which prevent identification of causal effects, in the hope of eventually making such a list complete and finding a way to identify effects in all graphs not on the list. I start constructing this list by considering graphs which generalize the bow arc graph since they can contain more than two nodes, but which also inherit its difficult structure. I call such graphs C-trees.

Definition 6 (C-tree) *A graph G where the set of all its nodes is a C-component, where each node has at most one child, and all nodes are ancestors of a single (root) node is called a C-tree.*

I call a C-tree with a root node Y Y-rooted. The graphs in Fig. 3.1 (a), (d), (e), (f), and (h) are Y-rooted C-trees. It turns out that in any Y-rooted C-tree, the effect of any subset of nodes, other than Y, on the root Y is not identifiable.

Theorem 3 *Let G be a Y-rooted C-tree. Let $\boldsymbol{X}$ be any subset of observable nodes in G which does not contain Y. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(y|do(\boldsymbol{x}))$.*

C-trees play a prominent role in the identification of *direct effects*. Intuitively, the direct effect of X on Y exists if there is an arrow from X to Y in the graph, and corresponds to the flow of influence along this arrow. However, simply considering changes in Y after fixing X is insufficient for isolating direct effect, since X can influence Y along other, longer front-door paths than the direct arrow. In order to disregard such influences, I also fix all other parents of Y (which as noted earlier removes all arrows incoming to these parents and thus to Y). The expression corresponding to the direct effect of X on Y is then $P(y|do(pa(y)))$. The following theorem links C-trees and direct effects.

Theorem 4 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y|do(pa(y)))$ if and only if there exists a subgraph of G which is a Y-rooted C-tree.*

This theorem might suggest that C-trees might play an equally strong role in identifying arbitrary effects on a single variable, not just direct effects. Unfortunately, this turns out not to be the case, due to the following lemma.

Lemma 1 (downward extension lemma) *Let $\boldsymbol{V}$ be the set of observable nodes in G. Assume $P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y}|do(\boldsymbol{x}))$. Let G' contain all the nodes and edges of G, and an additional node Z which is a child of all nodes in $\boldsymbol{Y}$. Then $P(\boldsymbol{v}, z), G' \not\vdash_{id} P(z|do(\boldsymbol{x}))$.*

Proof: Let $|Z| = \prod_{Y_i \in \mathbf{Y}} |Y_i| = n$. By construction, $P(z|do(\mathbf{x}))$ is equal to $\sum_{\mathbf{y}} P(z|\mathbf{y})P(\mathbf{y}|do(\mathbf{x}))$. Due to the way I set the arity of Z, $P(Z|\mathbf{Y})$ is an

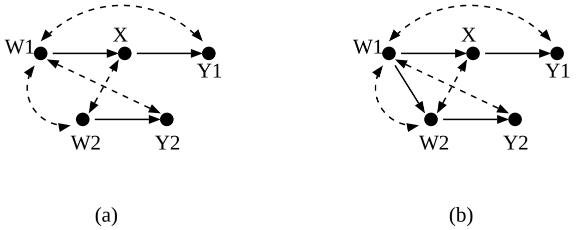


Figure 4.1: (a) a graph hedge-less for $P(y|do(x))$ (b) a graph containing a hedge for $P(y|do(x))$

n by n matrix which acts as a linear map which transforms $P(\mathbf{y}|do(\mathbf{x}))$ into $P(z|do(\mathbf{x}))$. Since I can arrange this linear map to be one to one, any proof of non-identifiability of $P(\mathbf{y}|do(\mathbf{x}))$ immediately extends to the proof of non-identifiability of $P(z|do(\mathbf{x}))$. □

What this lemma shows is that identification of effects on a singleton is not any simpler than the general problem of identification of effect on a set. In the next section, I consider this general problem.

4.4 C-Forests and Hedges

To find difficult graphs which prevent identification of effects on sets, I consider a multi-root generalization of C-trees.

Definition 7 (C-forest) *A graph G where the set of all its nodes is a C-component, and where each node has at most one child is called a C-forest.*

If a given C-forest has a set of root nodes (e.g., a set of nodes with no children) $\mathbf{R}$, I call it $\mathbf{R}$-rooted. Graphs in Fig. 4.1 (a), (b) are $\{Y1, Y2\}$-rooted C-forests. A naive way to generalize Theorem 3 would be to state that if G is an $\mathbf{R}$-rooted C-forest, then the effect of any set $\mathbf{X}$ that does not intersect $\mathbf{R}$ is not identifiable. However, as I later show, this is not true. Specifically, I later prove that $P(y1, y2|do(x))$ in the graph in Fig. 4.1 (a) is identifiable. To formulate the correct generalization of Theorem 3, we must understand what made C-trees difficult for the purposes of identifying effects on the root Y. It turned out that for particular function choices, the effects of ancestors of Y on Y precisely canceled themselves out so even though Y itself was dependent on its parents, it was observationally indistinguishable from a constant function. To get the same canceling of effects with C-forests, we must define a more complex graphical structure.

Definition 8 (hedge) *Let $\boldsymbol{X}$, $\boldsymbol{Y}$ be sets of variables in G. Let F, F' be $\boldsymbol{R}$-rooted*

C-forests in G such that F' is a subgraph of F, $\boldsymbol{X}$ only occur in F, and $\boldsymbol{R} \in An(\boldsymbol{Y})_{G_{\overline{\boldsymbol{x}}}}$. Then F and F' form a hedge for $P(\boldsymbol{y}|do(\boldsymbol{x}))$.

The graph in Fig. 4.1 (b) contains a hedge for $P(y1, y2|do(x))$. The mental picture for a hedge is as follows. We start with a C-forest F'. Then, F' grows new branches, while retaining the same root set, and becomes F. Finally, we "trim the hedge," by performing the action $do(\mathbf{x})$ which has the effect of removing some incoming arrows in $F \setminus F'$ (the subgraph of F consisting of vertices not a part of F'). Note that any Y-rooted C-tree and its root node Y form a hedge. The right generalization of Theorem 3 can be stated on hedges.

Theorem 5 *Let F, F' be subgraphs of G which form a hedge for $P(\boldsymbol{y}|do(\boldsymbol{x}))$. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y}|do(\boldsymbol{x}))$.*

Proof outline: As before, assume binary variables. I let the causal mechanisms of one of the models consists entirely of bit parity functions. The second model also computes bit parity for every mechanism, except those nodes in F' which have parents in F ignore the values of those parents. It turns out that these two models are observationally indistinguishable. Furthermore, any intervention in $F \setminus F'$ will break the bit parity circuits of the models. This break will be felt at the root set $\mathbf{R}$ of the first model, but not of the second, by construction. □

4.5 A Complete Identification Algorithm

Unlike the bow arc graph, and C-trees, hedges prevent identification of effects on multiple variables at once. Certainly a complete list of all possible difficult graphs must contain structures like hedges. But are there other kinds of structures that present problems? It turns out that the answer is "no," any time an effect is not identifiable in a causal model (if we make no restrictions on the type of function that can appear), there is a hedge structure involved. To prove that this is so, we need an algorithm which can identify any causal effect lacking a hedge. This algorithm, which I call **ID**, and which can be viewed as a simplified version of the identification algorithm due to [Tia02], appears in Fig. 4.2.

I will explain why each line of **ID** makes sense, and conclude by showing the operation of the algorithm on an example. The formal proof of soundness of **ID** can be found in the appendix. The first line merely asserts that if no action has been taken, the effect on $\mathbf{Y}$ is just the marginal of the observational distribution $P(\mathbf{v})$ on $\mathbf{Y}$. The second line states that if we are interested in the effect on $\mathbf{Y}$, it is sufficient to restrict our attention on the parts of the model ancestral to $\mathbf{Y}$. One intuitive argument for this is that descendants of $\mathbf{Y}$ can be viewed as "noisy

```
function ID(y, x, P, G)
INPUT: x,y value assignments, P a probability distribution,
G a causal diagram.
OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y})$ in terms of P or FAIL(F,F').

  1 if $\mathbf{x} = \emptyset$ return $\sum_{\mathbf{v} \setminus \mathbf{y}} P(\mathbf{v})$.

  2 if $\mathbf{V} \setminus An(\mathbf{Y})_G \neq \emptyset$
    return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cap An(\mathbf{Y})_G, \sum_{\mathbf{v} \setminus An(\mathbf{Y})_G} P, G_{An(\mathbf{Y})})$.

  3 let $\mathbf{W} = (\mathbf{V} \setminus \mathbf{X}) \setminus An(\mathbf{Y})_{G_{\overline{\mathbf{x}}}}$.
    if $\mathbf{W} \neq \emptyset$, return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cup \mathbf{w}, P, G)$.

  4 if $C(G \setminus \mathbf{X}) = \{S_1, ..., S_k\}$
    return $\sum_{\mathbf{v} \setminus (\mathbf{y} \cup \mathbf{x})} \prod_i \mathbf{ID}(s_i, \mathbf{v} \setminus s_i, P, G)$.

    if $C(G \setminus \mathbf{X}) = \{S\}$

      5 if $C(G) = \{G\}$, throw $\mathbf{FAIL}(G, G \cap S)$.
      6 if $S \in C(G)$ return $\sum_{s \setminus \mathbf{y}} \prod_{\{i | V_i \in S\}} P(v_i | v_\pi^{(i-1)})$.
      7 if $(\exists S') S \subset S' \in C(G)$ return $\mathbf{ID}(\mathbf{y}, \mathbf{x} \cap S',$
        $\prod_{\{i | V_i \in S'\}} P(V_i | V_\pi^{(i-1)} \cap S', v_\pi^{(i-1)} \setminus S'), G_{S'})$.
```

Figure 4.2: A complete identification algorithm. **FAIL** propagates through recursive calls like an exception, and returns the hedge which witnesses non-identifiability. $V_\pi^{(i-1)}$ is the set of nodes preceding V_i in some topological ordering π in G.

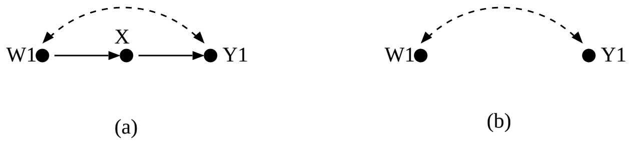


Figure 4.3: Subgraphs of G used for identifying $P_x(y_1, y_2)$.

versions" of $\mathbf{Y}$ and so any information they may impart which may be helpful for identification is already present in $\mathbf{Y}$. On the other hand, variables which are neither ancestors nor descendants of $\mathbf{Y}$ lie outside the relevant causal chain entirely, and have no useful information to contribute.

Line 3 forces an action on any node where such an action would have no effect on $\mathbf{Y}$ – assuming we already acted on $\mathbf{X}$. Since actions remove incoming arrows, we can view line 3 as simplifying the causal graph we consider by removing certain arcs from the graph, without affecting the overall answer. Line 4 is the key line of the algorithm, it decomposes the problem into a set of smaller problems using the key property of C-component factorization of causal models. If the entire graph is a single C-component already, further problem decomposition is impossible, and we must provide base cases. **ID** has three base cases. Line 5 fails because it finds two C-components, the graph G itself, and a subgraph S that does not contain any $\mathbf{X}$ nodes. But that is exactly one of the properties of C-forests that make up a hedge. In fact, it turns out that it is always possible to recover a hedge from these two c-components.

Line 6 asserts that if there are no bidirected arcs from $\mathbf{X}$ to the other nodes in the current subproblem under consideration, then we can replace acting on $\mathbf{X}$ by conditioning, and thus solve the subproblem. Line 7 is the most complex case where $\mathbf{X}$ is partitioned into two sets, $\mathbf{W}$ which contain bidirected arcs into other nodes in the subproblem, and $\mathbf{Z}$ which do not. In this situation, identifying $P(\mathbf{y}|do(\mathbf{x}))$ from $P(\mathbf{v})$ is equivalent to identifying $P(\mathbf{y}|do(\mathbf{w}))$ from $P(\mathbf{V}|do(\mathbf{z}))$, since $P(\mathbf{y}|do(\mathbf{x})) = P(\mathbf{y}|do(\mathbf{w}), do(\mathbf{z}))$. But the term $P(\mathbf{V}|do(\mathbf{z}))$ is identifiable using the previous base case, so we can consider the subproblem of identifying $P(\mathbf{y}|do(\mathbf{w}))$.

I give an example of the operation of the algorithm by identifying $P_x(y_1, y_2)$ from $P(\mathbf{v})$ in the graph shown in in Fig. 4.1 (a). Since $G = G_{An(\{Y_1,Y_2\})}, C(G \setminus \{X\}) = \{G\}$, and $\mathbf{W} = \{W_1\}$, I invoke line 3 and attempt to identify $P_{x,w}(y_1, y_2)$. Now $C(G \setminus \{X, W\}) = \{Y_1, W_2 \to Y_2\}$, so I invoke line 4. Thus the original problem reduces to identifying $\sum_{w_2} P_{x,w_1,w_2,y_2}(y_1)P_{w,x,y_1}(w_2, y_2)$. Solving for the second expression, I trigger line 2, noting that we can ignore nodes which are not ancestors of W_2 and Y_2, which means $P_{w,x,y_1}(w_2, y_2) = P(w_2, y_2)$. Solving for the first expression, I first trigger line 2 also, obtaining $P_{x,w_1,w_2,y_2}(y_1) =$

$P_{x,w}(y_1)$. The corresponding G is shown in Fig. 4.3 (a). Next, I trigger line 7, reducing the problem to computing $P_w(y_1)$ from $P(Y_1|X, W_1)P(W_1)$. The corresponding G is shown in Fig. 4.3 (b). Finally, I trigger line 2, obtaining $P_w(y_1) = \sum_{w_1} P(y_1|x, w_1)P(w_1)$. Putting everything together, I obtain: $P_x(y_1, y_2) = \sum_{w_2} P(y_1, w_2) \sum_{w_1} P(y_1|x, w_1)P(w_1)$.

As mentioned earlier, whenever the algorithm fails at line 5, it is possible to recover a hedge from the C-components S and G considered for the subproblem where the failure occurs. In fact, it can be shown that this hedge implies the non-identifiability of the original query with which the algorithm was invoked, which implies the following result.

Theorem 6 ***ID*** *is complete.*

The completeness of **ID** implies that hedges can be used to characterize all cases where effects of the form $P(\mathbf{y}|do(\mathbf{x}))$ cannot be identified from the observational distribution $P(\mathbf{v})$.

Theorem 7 (hedge criterion) $P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y}|do(\boldsymbol{x}))$ *if and only if* G *contains a hedge for some* $P(\boldsymbol{y'}|do(\boldsymbol{x'}))$*, where* $\boldsymbol{y'} \subseteq \boldsymbol{y}$*,* $\boldsymbol{x'} \subseteq \boldsymbol{x}$*.*

4.6 Conditional Effects

I close this chapter by considering identification of *conditional effects* of the form $P(\mathbf{y}|do(\mathbf{x}), \mathbf{z})$ which are defined to be equal to $P(\mathbf{y}, \mathbf{z}|do(\mathbf{x}))/P(\mathbf{z}|do(\mathbf{x}))$. Such expressions are a formalization of an intuitive notion of "effect of action in the presence of non-contradictory evidence," for instance the effect of smoking on lung cancer incidence rates in a particular age group (as opposed to the effect of smoking on cancer in the general population). I say that evidence $\mathbf{z}$ is non-contradictory since it is conceivable to consider questions where the evidence $\mathbf{z}$ stands in logical contradiction to the proposed hypothetical action $do(\mathbf{x})$: for instance what is the effect of smoking on cancer among the non-smokers. Such counterfactual questions will be considered in the next chapter. Conditioning can both help and hinder identifiability. $P(y|do(x))$ is not identifiable in the graph shown in Fig. 4.4 (a), while it is identifiable in the graph shown in Fig. 4.4 (b). Conditioning reverses the situation. In Fig. 4.4 (a), conditioning on Z renders Y independent of any changes to X, making $P_x(y|z)$ equal to $P(y|z)$. On the other hand, in Fig. 4.4 (b), conditioning on Z makes X and Y dependent, resulting in $P_x(y|z)$ becoming non-identifiable.

I would like to reduce the problem of identifying conditional effects to the familiar problem of identifying causal effects without evidence for which I already

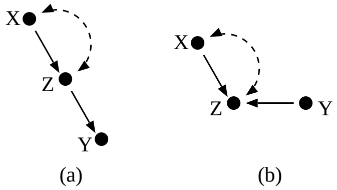


Figure 4.4: (a) Causal graph with an identifiable conditional effect $P(y|do(x), z)$ (b) Causal graph with a non-identifiable conditional effect $P(y|do(x), z)$

have a complete algorithm. Fortunately, rule 2 of do-calculus provides me with a convenient way of converting the unwanted evidence $\mathbf{z}$ into actions $do(\mathbf{x})$ which I know how to handle. The following convenient lemma allows me to remove as many evidence variables as possible from a conditional effect.

Theorem 8 *For any G and any conditional effect $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$ there exists a unique maximal set $\boldsymbol{Z} = \{Z \in \boldsymbol{W} | P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w}) = P_{\boldsymbol{x},z}(\boldsymbol{y}|\boldsymbol{w} \setminus \{z\})\}$ such that rule 2 applies to $\boldsymbol{Z}$ in G for $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$. In other words, $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w}) = P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y}|\boldsymbol{w} \setminus \boldsymbol{z})$.*

Of course Theorem 8 does not guarantee that the entire set $\mathbf{z}$ can be handled in this way. In many cases, even after rule 2 is applied, some set of evidence will remain in the expression. Fortunately, the following result implies that identification of unconditional causal effects is all we need.

Theorem 9 *Let $\boldsymbol{Z} \subseteq \boldsymbol{W}$ be the maximal set such that $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w}) = P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y}|\boldsymbol{w} \setminus \boldsymbol{z})$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$ is identifiable in G if and only if $P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y}, \boldsymbol{w} \setminus \boldsymbol{z})$ is identifiable in G.*

The previous two theorems suggest a simple addition to **ID**, which I call **IDC**, shown in Fig. 4.5, which handles identification of conditional causal effects.

Theorem 10 ***IDC** is sound and complete.*

Proof: This follows from Theorems 8 and 9. □

[Tia04] developed a significantly more complicated algorithm for identifying conditional effects. It can be shown, nevertheless, that Tian's algorithm is in some sense equivalent to **IDC** since it is complete [Shp07].

I conclude this section by noting that since the **IDC** algorithm uses d-separation tests to remove conditioning variables, and since the **ID** algorithm it uses as a subroutine has a graphical condition characterizing the input graphs on which it succeeds, it is possible to derive a complete graphical criterion for identifiable conditional effects.

```
function $\mathbf{IDC}(\mathbf{y}, \mathbf{x}, \mathbf{z}, P, G)$
INPUT: $\mathbf{x}$,$\mathbf{y}$,$\mathbf{z}$ value assignments, P a probability
distribution, G a causal diagram (an I-map of P).
OUTPUT: Expression for $P_{\mathbf{x}}(\mathbf{y}|\mathbf{z})$ in terms of P or $\mathbf{FAIL}$(F,F').

  1 if $(\exists Z \in \mathbf{Z})(\mathbf{Y} \perp Z|\mathbf{X}, \mathbf{Z} \setminus \{Z\})_{G_{\overline{\mathbf{x}},\underline{z}}}$,
    return $\mathbf{IDC}(\mathbf{y}, \mathbf{x} \cup \{z\}, \mathbf{z} \setminus \{z\}, P, G)$.

  2 else let $P' = \mathbf{ID}(\mathbf{y} \cup \mathbf{z}, \mathbf{x}, P, G)$.
    return $P' / \sum_{\mathbf{y}} P'$.
```

Figure 4.5: A complete identification algorithm for conditional effects.

Corollary 2 (back-door hedge criterion) *Let $\boldsymbol{Z} \subseteq \boldsymbol{W}$ be the unique maximal set such that $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w}) = P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y}|\boldsymbol{w} \setminus \boldsymbol{z})$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$ is identifiable from P if and only if there does not exist a hedge for $P_{\boldsymbol{x}'}(\boldsymbol{y}')$, for any $\boldsymbol{Y}' \subseteq (\boldsymbol{Y} \cup \boldsymbol{W}) \setminus \boldsymbol{Z}$, $\boldsymbol{X}' \subseteq \boldsymbol{X} \cup \boldsymbol{Z}$.*

The name 'back-door hedge' comes from the fact that both back-door paths and hedge structures are key for identifiability of conditional effects. In particular, $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$ is identifiable if and only if $P_{\mathbf{x},\mathbf{z}}(\mathbf{y}, \mathbf{w} \setminus \mathbf{z})$ does not contain any hedges and every $W \in \mathbf{W} \setminus \mathbf{Z}$ has a back-door path to some $Y \in \mathbf{Y}$ in the context of the effect.

4.7 Corollaries

I conclude this section by showing that the notion of a causal theory as a set of independencies embodied by the causal graph, together with rules of probability and do-calculus is complete for computing causal effects, if we also take statistical data embodied by $P(\mathbf{v})$ as axiomatic.

Theorem 11 *The rules of do-calculus are complete for identifying effects of the form $P(\boldsymbol{y}|do(\boldsymbol{x}), \boldsymbol{z})$, where $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ are arbitrary sets.*

Proof: The proofs of soundness of **ID** and **IDC** in the appendix use do-calculus. This implies every line of the algorithms I presented can be rephrased as a sequence of do-calculus manipulations. But **ID** and **IDC** are also complete, which implies the conclusion. $\square$

CHAPTER 5

Counterfactuals

In this chapter, I consider the problem of inferring distributions over atomic counterfactual events from the results of all possible experiments we can perform. I approach this problem in the same spirit I approached the problem of identifying causal effects from the previous chapter. First, I propose a graphical representation called *the counterfactual graph* for displaying causal assumptions involved in multiple hypothetical worlds mentioned in counterfactual queries. With such a representation, it's not a difficult matter to construct an identification algorithm along similar lines as the algorithm in the previous chapter. To prove completeness, I construct the set of difficult counterfactual graphs which imply non-identification of certain counterfactuals.

5.1 Counterfactuals and Multiple Worlds

While effects of actions have an intuitive interpretation as downward flow, the interpretation of counterfactuals, or what-if questions is more complex. An informal counterfactual statement in natural language such as "would I have a headache had I taken an aspirin" talks about multiple worlds: the actual world, and other, hypothetical worlds which differ in some small respect from the actual world (e.g., the aspirin was taken), while in most other respects are the same. In this chapter, I represent the actual world by a causal model in its natural state, devoid of any interventions, while the alternative worlds are represented by submodels $M_{\mathbf{x}}$ where the action $do(\mathbf{x})$ implements the hypothetical change from the actual state of affairs considered. People make sense of informal statements involving multiple, possibly conflicting worlds because they expect not only the causal rules to be invariant across these worlds (e.g., aspirin helps headaches in all worlds), but the worlds themselves to be similar enough where evidence in one world has ramifications in another. For instance, if I find myself with a headache, I expect the usual causes of my headache to also operate in the hypothetical world, interacting there with the preventative influence of aspirin. In the representation of counterfactuals used in this thesis, I model this interaction between worlds by assuming that the world histories or background contexts, represented by the unobserved $\mathbf{U}$ variables are shared across all hypothetical worlds.

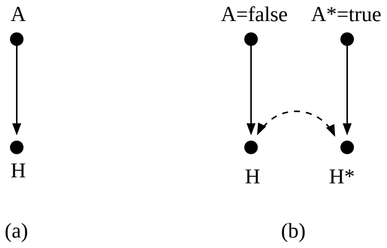


Figure 5.1: (a) A causal graph for the aspirin/headache domain (b) A corresponding twin network graph for the query $P(H^*_{a^*=true}|A = false)$.

I illustrate the representation method for counterfactuals I introduced in Section 2 by modeling the example question “would I have a headache had I taken an aspirin?” The actual world referenced by this query is represented by a causal model containing two variables, headache and aspirin, with aspirin being a parent of headache, see Fig. 5.1 (a). In this world, I observe that aspirin has value false. The hypothetical world is represented by a submodel where the action $do(aspirin = true)$ has been taken. To distinguish nodes in this world I augment their names with an asterisk. The two worlds share the background variables $\mathbf{U}$, and so can be represented by a single causal model with the graph shown in Fig. 5.1 (b). The query is represented by the distribution $P(H^*_{a^*=true}|A = false)$, where H is headache, and A is aspirin. Note that the nodes $A^* = true$ and $A = false$ in Fig. 5.1 (b) do not share a bidirected arc. This is because an intervention $do(a^* = true)$ removes all incoming arrows to A^*, which removes the bidirected arc between A^* and A.

5.2 Evaluating Counterfactuals

The graphs representing two hypothetical worlds invoked by a counterfactual query like the one shown in Fig. 5.1 (b) are called *twin network graphs*, and were first proposed as a way to represent counterfactuals by [BP94b], and [BP94a]. In addition, [BP94b] proposed a method for evaluating counterfactual expressions like $P(H^*_{a^*=true}|A = false)$ when all parameters of a causal model are known. This method can be explained as follows. If we forget the causal and counterfactual meaning behind the twin network graph, and simply view it as a Bayesian network, the query $P(H^*_{a^*=true}|A = false)$ can be evaluated using any of the standard inference algorithms available, provided we have access to all conditional probability tables generated by $\mathbf{F}$ and $\mathbf{U}$ of a causal model which gave rise to the twin network graph. In practice, however, complete knowledge of the model is too much to ask for; the functional relationships as well as the distribution $P(\mathbf{u})$ are not known exactly, though some of their aspects can be inferred from the observable distribution $P(\mathbf{v})$.

Instead, the typical state of knowledge of a causal domain is the statistical behavior of the observable variables in the domain, summarized by the distribution $P(\mathbf{v})$, together with knowledge of causal directionality, obtained either from expert judgment (e.g., we know that visiting the doctor does not make us sick, though disease and doctor visits are highly correlated), or direct experimentation (e.g., it's easy to imagine an experiment which establishes that wet grass does not cause sprinklers to turn on). I already used these two sources of knowledge in the previous chapter as a basis for computing causal effects. Nevertheless, there are reasons to consider computing counterfactual quantities from experimental, rather than observational studies. In general, a counterfactual can posit worlds with features contradictory to what has actually been observed. For instance, questions resembling the headache/aspirin question I used as an example are actually frequently asked in epidemiology in the more general form where we are interested in estimating the effect of a treatment x on the outcome variable Y for the patients that were not treated (x'). In my notation, this is just the familiar expression $P(Y_x|X = x')$. The problem with questions such as these is that no experimental setup exists in which someone is both given and not given treatment. Therefore, it makes sense to ask under what circumstances we can evaluate such questions even if we are given as input every experiment that is possible to perform in principle on a given causal model. In my framework the set of all experiments is denoted as P_*, and is formally defined as $\{P_{\mathbf{x}}|$ where $\mathbf{x}$ is any set of values of $\mathbf{X} \subseteq \mathbf{V}\}$. The question that I ask in this chapter, then, is whether it is possible to identify a query $P(\gamma|\delta)$, where γ, δ are conjunctions of counterfactual events (with δ possibly empty), from the graph G and the set of all experiments P_*. I can pose the problem in this way without loss of generality since I already developed complete methods for identifying members of P_* from G and $P(\mathbf{v})$. This means that if for some reason using P_* as input is not realistic I can combine the methods which I will develop in this chapter with those in the previous chapter to obtain identification results for $P(\gamma|\delta)$ from G and $P(\mathbf{v})$.

5.3 The Counterfactual Graph

Before tackling the problem of identifying counterfactual queries from experiments, I extend the example in Fig. 5.1 (b) to a general graphical representation for worlds invoked by a counterfactual query. The twin network graph is a good first attempt at such a representation. It is essentially a causal diagram for a model encompassing two potential worlds. Nevertheless, the twin network graph suffers from a number of problems. Firstly, it can easily come to pass that a coun-

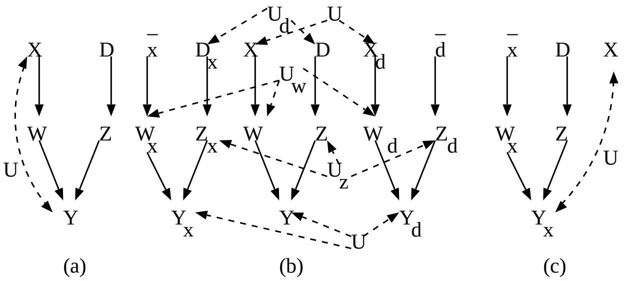


Figure 5.2: Nodes fixed by actions denoted with an overline, signifying that all incoming arrows are cut. (a) Original causal diagram (b) Parallel worlds graph for $P(y_x|x', z_d, d)$ (the two nodes denoted by U are the same). (c) Counterfactual graph for $P(y_x|x', z_d, d)$.

terfactual query of interest would involve three or more worlds. For instance, we might be interested in how likely the patient would be to have a symptom Y given a certain dose x of drug X, assuming we know that the patient has taken dose x' of drug X, dose d of drug D, and we know how an intermediate symptom Z responds to treatment d. This would correspond to the query $P(y_x|x', z_d, d)$, which mentions three worlds, the original model M, and the submodels M_d, M_x. This problem is easy to tackle – I simply add more than two submodel graphs, and have them all share the same $\mathbf{U}$ nodes. This simple generalization of the twin network model was considered by [ASP05], and was called there the parallel worlds graph. Fig. 5.2 shows the original causal graph and the parallel worlds graph for $\gamma = y_x \wedge x' \wedge z_d \wedge d$.

The other problematic feature of the twin network graph, which is inherited by the parallel worlds graph, is that multiple nodes can sometimes correspond to the same random variable. For example, in Fig. 5.2 (b), the variables Z and Z_x are represented by distinct nodes, although it's easy to show that since Z is not a descendant of X, $Z = Z_x$. These equality constraints among nodes can make the d-separation criterion misleading if not used carefully. For instance, $Y_x \perp D_x | Z$ even though using d-separation in the parallel worlds graph suggests the opposite. This sort of problem is fairly common in causal models which are not *faithful* [SGS93] or *stable* [PV91], [Pea00], in other words in models where d-separation statements in a causal diagram imply independence in a distribution, but not vice versa. However, lack of faithfulness usually arises due to "numeric coincidences" in the observable distribution. In this case, the lack of faithfulness is "structural," in a sense that it is possible to refine parallel worlds graphs in such a way that the node duplication disappears, and the attendant independencies not captured by d-separation are captured by d-separation in refined graphs.

This refinement has two additional beneficial side effects. The first is that by removing node duplication, we also determine which syntactically distinct counterfactual variables correspond to the same random variable. By identifying such equivalence classes of counterfactual variables, we guarantee that syntactically different variables are in fact different, and this makes it simpler to reason about counterfactuals in order to identify them. For instance, a counterfactual $P(y_x, y')$ may either be non-identifiable or inconsistent (and so identifiable to equal 0), depending on whether Y_x and Y are the same variable. The second benefit of this refinement is that resulting graphs are generally much smaller and less cluttered than parallel worlds graphs, and so are easier to understand. Compare, for instance, the graphs in Fig. 5.2 (b) and Fig. 5.2 (c). To rid ourselves of duplicates, we need a formal way of determining when variables from different submodels are in fact the same. The following lemma does this.

Lemma 2 *Let M be a model inducing G containing variables α, β with the following properties:*

- *α and β have the same domain of values.*
- *There is a bijection f from $Pa(\alpha)$ to $Pa(\beta)$ such that a parent γ and $f(\gamma)$ have the same domain of values.*
- *The functional mechanisms of α and β are the same (except whenever the function for α uses the parent γ, the corresponding function for β uses $f(\gamma)$).*

Assume an observable variable set $\boldsymbol{Z}$ was observed to attain values $\boldsymbol{z}$ in $M_{\boldsymbol{x}}$, the submodel obtained from M by forcing another observable variable set $\boldsymbol{X}$ to attain values $\boldsymbol{x}$. Assume further that for each $\gamma \in Pa(\alpha)$, either $f(\gamma) = \gamma$, or γ and $f(\gamma)$ attain the same values (whether by observation or intervention). Then α and β are the same random variable in $M_{\boldsymbol{x}}$ with observations $\boldsymbol{z}$.

Proof: This follows from the fact that variables in a causal model are functionally determined from their parents. □

If two distinct nodes in a causal diagram represent the same random variable, the diagram contains redundant information, and the nodes must be merged. If two nodes, say corresponding to $Y_{\mathbf{x}}, Y_{\mathbf{z}}$, are established to be the same in G, they are merged into a single node which inherits all the children of the original two. These two nodes either share their parents (by induction) or their parents attain the same values. If a given parent is shared, it becomes the parent of the new node. Otherwise, I pick one of the parents arbitrarily to become the parent of the new node. This operation is summarized by the following lemma.

Lemma 3 *Let $M_{\boldsymbol{x}}$ be a submodel derived from M with set $\boldsymbol{Z}$ observed to attain values $\boldsymbol{z}$, such that Lemma 2 holds for α, β. Let M' be a causal model obtained from M by merging α, β into a new node ω, which inherits all parents and the functional mechanism of α. All children of α, β in M' become children of ω. Then $M_{\boldsymbol{x}}, M'_{\boldsymbol{x}}$ agree on any distribution consistent with $\boldsymbol{z}$ being observed.*

Proof: This is a direct consequence of Lemma 2. □

The new node ω I obtain from Lemma 3 can be thought of as a new counterfactual variable. As mentioned in chapter 3, such variables take the form $Y_{\mathbf{x}}$ where Y is the variable in the original causal model, and $\mathbf{x}$ is a subscript specifying the action which distinguishes the counterfactual. Since I only merge two variables derived from the same original, specifying Y is simple. But what about the subscript? Intuitively, the subscript of ω contains those fixed variables which are ancestors of ω in the graph G' of M'. Formally the subscript is $\mathbf{w}$, where $\mathbf{W} = An(\omega)_{G'} \cap \mathbf{sub}(\gamma)$, where the $\mathbf{sub}(\gamma)$ corresponds to those nodes in G' which correspond to subscripts in γ. Since I replaced α, β by ω, I replace any mention of α, β in the given counterfactual query $P(\gamma)$ by ω. Note that since α, β are the *same*, their value assignments must be the same (say equal to y). The new counterfactual ω inherits this assignment.

5.4 Constructing Counterfactual Graphs

I summarize the inductive applications of Lemma 2, and 3 by the **make-cg** algorithm, which takes γ and G as arguments, and constructs a version of the parallel worlds graph without duplicate nodes. I call the resulting structure the $counterfactual\ graph$ of γ, and denote it by G_γ. The algorithm is shown in Fig. 5.3.

There are three additional subtleties in **make-cg**. The first is that if variables $Y_{\mathbf{x}}, Y_{\mathbf{z}}$ were judged to be the same by Lemma 2, but γ assigns them different values, this implies that the original set of counterfactual events γ is inconsistent, and so $P(\gamma) = 0$. The second is that if we are interested in identifiability of $P(\gamma)$, we can restrict ourselves to the ancestors of γ in G'. I can justify this using the same intuitive argument I used in Section 3 to justify Line 2 in **ID**. The formal proof for line 2 I provide in the Appendix applies with little change to **make-cg**. Finally, because the algorithm can make an arbitrary choice picking a parent of ω each time Lemma 3 is applied, both the counterfactual graph G', and the corresponding modified counterfactual γ' are not unique. This does not present a problem, however, as any such graph is acceptable for our purposes.

I illustrate the operation of **make-cg** by showing how the graph in Fig. 5.2 (c) is derived from the graph in Fig. 5.2 (b). I start the application of Lemma

function **make-cg**(G, γ)
INPUT: G a causal diagram, γ a conjunction of counterfactual events
OUTPUT: A counterfactual graph G_{γ}, and either a set of events γ' s.t. $P(\gamma') = P(\gamma)$ or **INCONSISTENT**

- Construct a submodel graph $G_{\mathbf{x}_i}$ for each action $do(\mathbf{x}_i)$ mentioned in γ. Construct the parallel worlds graph G' by having all such submodel graphs share their corresponding U nodes.
- Let π be a topological ordering of nodes in G', let $\gamma' := \gamma$.
- Apply Lemmas 2 and 3, in order π, to each observable node pair α, β derived from the same variable in G. For each α, β that are the same, do:
    - Let G' be modified as specified in Lemma 3.
    - Modify γ' by renaming all occurrences of β to α.
    - If $\mathbf{val}(\alpha) \neq \mathbf{val}(\beta)$, return G', **INCONSISTENT**.
- return $(G'_{An(\gamma')}, \gamma')$, where $An(\gamma')$ is the set of nodes in G' ancestral to nodes corresponding to variables mentioned in γ'.

Figure 5.3: An algorithm for constructing counterfactual graphs

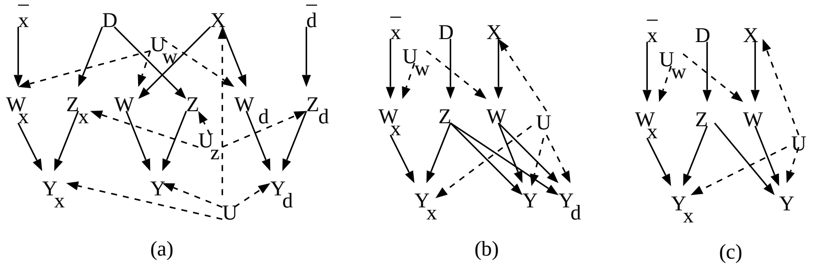


Figure 5.4: Intermediate graphs used by **make-cg** in constructing the counterfactual graph for $P(y_x|x', z_d, d)$ from Fig. 5.2 (b).

2 from the topmost observable nodes, and conclude that the node pairs D_x, D, and X_d, X have the same functional mechanisms, and the same parent set (in this case the parents are unobservable nodes U_d for the first pair, and U_x for the second). I then use Lemma 3 to obtain the graph shown in Fig. 5.4 (a). Since the node pairs are the same, we pick the name of one of the nodes of the pair to serve as the name of the new node. In this case, I picked D and X. Note that for this graph, and all subsequent intermediate graphs I generate, I use the convention that if a merge creates a situation where an unobservable variable has a single parent, that variable is omitted from the graph. For instance, in Fig. 5.4 (a), the variable U_d, and its corresponding arrow to D omitted.

Next, I apply Lemma 2 for the node pair W_d, W. In this case, the functional mechanisms are once again the same, while the parents of W_d, W are X and U_w. I can also apply Lemma 2 twice to conclude that Z, Z_x and Z_d are in fact the same node, and so can be merged. The functional mechanisms of these three nodes are the same, and they share the parent U_z. As far as the parents of this triplet, the U_z parent is shared by all three, while Z, Z_x share the parent D, and Z_d has a separate parent d, fixed by intervention. However, in the counterfactual query in question, which is $P(y_x|x', z_d, d)$, the variable D happens to be observed to attain the value d, the same as the intervention value for the parent of Z_d. This implies that for the purposes of the Z, Z_x, Z_d triplet, their D-derived parents share the same value, which allows us to conclude they are the same random variable. The intuition here is that while intervention and observation are not the same operation, they have the same effect if the relevant U variables happen to react in the same way to both the given intervention, and the given observation (this is the essence of the Axiom of Composition [Pea00].) In this case, U variables react the same way because the parallel worlds share all unobserved variables.

There is one additional subtlety in performing the merge of the triplet Z, Z_x, Z_d. If we examine the query $P(y_x|x', z_d, d)$, we notice that Z_d, or more precisely its value, appears in it. When I merge nodes, only one name out of the original two is used. It's possible that some of the old names appear in the query, which means I must replace all references to the old, pre-merge nodes to the new post-merge name I picked. Since I picked the name Z for the newly merged node, I replace the reference to Z_d in the query by the reference to Z, so the modified query is $P(y_x|x', z, d)$. Since the variables were established to be the same, this is a safe syntactic transformation.

After W_d, W, and the Z, Z_x, Z_d triplet are merged, the resulting graph appears in Fig. 5.4 (b). Finally, I apply Lemma 2 one more time to conclude Y and Y_d are the same variable, using the same reasoning as before. After performing this final merge, I obtain the graph in Fig. 5.4 (c). It's easy to see that Lemma 2 no longer applies to any node pair: W and W_x differ in their X-derived parent, and Y, and Y_x differ on their W-derived parent, which was established inductively. The final operation which **make-cg** performs is restricting the graph in Fig. 5.4 (b) to variables actually relevant for computing the (potentially syntactically modified) query it was given as input, namely $P(y_x|x', z, d)$, in other words those variables which are ancestral to variables in the query in the final intermediate graph I obtained. In this case, I remove nodes W and Y (and their adjacent edges) from consideration, to finally obtain the graph in Fig. 5.2 (c), which is a counterfactual graph for the original query.

5.5 Counterfactual Identification Algorithms

Having constructed a graphical representation of worlds mentioned in counterfactual queries, I can turn to identification. I construct two algorithms for this task, the first is called **ID*** and works for unconditional queries, while the second, **IDC***, works on queries with counterfactual evidence and calls the first as a subroutine. These are shown in Figs. 5.5 and 5.6.

These algorithms make use of the following notation: **sub**(.) returns the set of subscripts, **var**(.) the set of variables, and **ev**(.) the set of values (either set or observed) appearing in a given counterfactual, while **val**(.) is the value assigned to a given counterfactual variable. As before, $C(G')$ is the set of maximal C-components of G', except I don't count nodes in G' fixed by interventions as part of any C-component. $V(G')$ is the set of observable nodes of G'. Following [Pea00], $G'_{\underline{y_{\mathbf{x}}}}$ is the graph obtained from G' by removing all outgoing arcs from $Y_{\mathbf{x}}$; $\gamma'_{\underline{y_{\mathbf{x}}}}$ is obtained from γ' by replacing all descendant variables $W_{\mathbf{z}}$ of $Y_{\mathbf{x}}$ in γ' by $W_{\mathbf{z},y}$. A counterfactual $\mathbf{s}_{\mathbf{r}}$, where $\mathbf{s}, \mathbf{r}$ are value assignments to sets of nodes, represents the event "the node set $\mathbf{S}$ attains values $\mathbf{s}$ under intervention $do(\mathbf{r})$." Finally, I take $x_{x..}$ to mean some counterfactual variable derived from X where x appears in the subscript (the rest of the subscript can be arbitrary), which also

```
function $\mathbf{ID*}(G, \gamma)$
INPUT: $G$ a causal diagram, $\gamma$ a conjunction of counterfactual events
OUTPUT: an expression for $P(\gamma)$ in terms of $P_*$ or $\mathbf{FAIL}$

    if $\gamma = \emptyset$, return 1

    if $(\exists x_{x'..} \in \gamma)$, return 0

    if $(\exists x_{x..} \in \gamma)$, return $\mathbf{ID*}(G, \gamma \setminus \{x_{x..}\})$

    $(G', \gamma') = \mathbf{make\text{-}cg}(G, \gamma)$

    if $\gamma' = \mathbf{INCONSISTENT}$, return 0

    if $C(G') = \{S^1, ..., S^k\}$,
    return $\sum_{\mathbf{v}(G') \setminus \gamma'} \prod_i \mathbf{ID*}(G, s^i_{\mathbf{v}(G') \setminus s^i})$

    if $C(G') = \{S\}$ then,

        if $(\exists \mathbf{x}, \mathbf{x}')$ s.t. $\mathbf{x} \neq \mathbf{x}', \mathbf{x} \in \mathbf{sub}(S), \mathbf{x}' \in \mathbf{ev}(S)$,
        throw $\mathbf{FAIL}$

        else, let $\mathbf{x} = \bigcup \mathbf{sub}(S)$
        return $P_{\mathbf{x}}(\mathbf{var}(S))$
```

Figure 5.5: An identification algorithm for joint counterfactual distributions.

```
function $\mathbf{IDC*}(G, \gamma, \delta)$
INPUT: $G$ a causal diagram, $\gamma, \delta$ conjunctions of counterfactual events
OUTPUT: an expression for $P(\gamma|\delta)$ in terms of $P_*$, $\mathbf{FAIL}$, or $\mathbf{UNDEFINED}$

    if $\mathbf{ID*}(G, \delta) = 0$, return $\mathbf{UNDEFINED}$

    $(G', \gamma' \wedge \delta') = \mathbf{make\text{-}cg}(G, \gamma \wedge \delta)$

    if $\gamma' \wedge \delta' = \mathbf{INCONSISTENT}$, return 0

    if $(\exists y_{\mathbf{x}} \in \delta')$ s.t. $(Y_{\mathbf{x}} \perp \gamma')G'_{\underline{y_{\mathbf{x}}}}$,
    return $\mathbf{IDC*}(G, \gamma'_{y_{\mathbf{x}}}, \delta' \setminus \{y_{\mathbf{x}}\})$

    else, let $P' = \mathbf{ID*}(G, \gamma' \wedge \delta')$. return $P'/P'(\delta)$
```

Figure 5.6: An identification algorithm for conditional counterfactual distributions.

attains value x.

The notation used in these algorithms is somewhat intricate, so I give an intuitive description of each line. I start with **ID***. The first line states that if γ is an empty conjunction, then its probability is 1, by convention. The second line states that if γ contains a counterfactual which violates the Axiom of Effectiveness [Pea00], then γ is inconsistent, and I return probability 0. The third line states that if a counterfactual contains its own value in the subscript, then it is a tautological event, and it can be removed from γ without affecting its probability. Line 4 invokes **make-cg** to construct a counterfactual graph G', and the corresponding relabeled counterfactual γ'. Line 5 returns probability 0 if an inconsistency was found during the construction of the counterfactual graph, e.g., if two variables found to be the same in γ had different value assignments. Line 6 is analogous to Line 4 in the **ID** algorithm, it decomposes the problem into a set of subproblems, one for each C-component in the counterfactual graph. In the **ID** algorithm, the term corresponding to a given C-component S_i of the causal diagram was the effect of all variables not in S_i on variables in S_i, in other words $P_{\mathbf{v} \setminus s_i}(s_i)$, and the outermost summation on line 4 was over values of variables not in $\mathbf{Y}, \mathbf{X}$. Here, the term corresponding to a given C-component S^i of the counterfactual graph G' is the conjunction of counterfactual variables where each variable contains in its subscript all variables not in the C-component S^i, in other words $\mathbf{v}(G') \setminus s^i$, and the outermost summation is over variables not in γ'. Line 7 is the base case, where the counterfactual graph has a single C-component. There are two cases, corresponding to line 8 and line 9. Line 8 says that if γ' contains a "conflict," that is an inconsistent value assignment where at least one value is in the subscript, then I fail. Line 9 says if there are no conflicts, then its safe to take the union of all subscripts in γ', and return the effect of the subscripts in γ' on the variables in γ'.

The **IDC***, like its counterpart **IDC** is shorter. The first line fails if δ is inconsistent. **IDC** did not have an equivalent line, since I can assume $P(\mathbf{v})$ is positive. The problem with counterfactual distributions is there is no simple way to prevent non-positive distributions spanning multiple worlds from arising, even if the original $P(\mathbf{v})$ was positive – hence the explicit check. The second line constructs the counterfactual graph, except since **make-cg** can only take conjunctions, I provide it with a joint counterfactual $\gamma \wedge \delta$. Line 3 returns 0 if an inconsistency was detected. Line 4 is the central line of the algorithm and is analogous to line 1 of **IDC**. In **IDC**, I moved a value assignment $Z = z$ from being observed to being fixed if there were no back-door paths from Z to the outcome variables $\mathbf{Y}$ given the context of the effect of $do(\mathbf{x})$. Here, I move a counterfactual value assignment $Y_{\mathbf{x}} = y$ from being observed (that is being a part of δ), to being fixed (that is appearing in every subscript of γ') if there are no back-door paths from Y_z to the counterfactual of interest γ'. Finally, line 5 is the analogue of line 2 of **IDC**, we attempt to identify a joint counterfactual probability, and then obtain a conditional counterfactual probability from the result.

I illustrate the operation of these algorithms by considering the identification of the query $P(y_x|x', z_d, d)$ I mentioned earlier. Since $P(x', z_d, d)$ is not inconsistent, I proceed to construct the counterfactual graph on line 2. Suppose I produce the graph in Fig. 5.2 (c), where the corresponding modified query is $P(y_x|x', z, d)$. Since $P(y_x, x', z, d)$ is not inconsistent I proceed to the next line, which moves z, d (with d being redundant due to graph structure) to the subscript of y_x, to obtain $P(y_{x,z}|x')$. Finally, I call **ID*** with the query $P(y_{x,z}, x')$. The first interesting line is 6, where the query is expressed as $\sum_w P(y_{x,z,w}, x')P(w_x)$. Note that x is redundant in the first term, so a recursive call reaches line 9 with $P(y_{z,w}, x')$, which is identifiable as $P_{z,w}(y, x')$ from P_*. The second term is trivially identifiable as $P_x(w)$, which means the query $P(y_x, x', z, d)$ is identifiable as $P' = \sum_w P_{z,w}(y, x')P_x(w)$, and the conditional query is equal to $P'/P'(x')$.

5.6 Soundness and Completeness

The definitions of **ID***, and **IDC*** reveal their close similarity to algorithms **ID** and **IDC** in the previous section. The major differences lie in the failure and success base cases, and slightly different subscript notation. This is not a coincidence, since a counterfactual graph can be thought of as a causal graph for a particular large causal model which happens to have some distinct nodes have the same causal mechanisms. This means that all the theorems and definitions used in the previous sections for causal diagrams transfer over without change to counterfactual graphs. Using this fact, I will show that **ID***, and **IDC*** are sound and complete for identifying $P(\gamma)$, and $P(\gamma|\delta)$ respectively.

Theorem 12 (soundness) *If **ID*** succeeds, the expression it returns is equal to $P(\gamma)$ in a given causal graph. Furthermore, if **IDC*** does not output **FAIL**, the expression it returns is equal to $P(\gamma|\delta)$ in a given causal graph, if that expression is defined, and **UNDEFINED** otherwise.*

Proof outline: The first line merely states that the probability of an empty conjunction is 1, which is true by convention. Lines 2 and 3 follow by the Axiom of Effectiveness [GP98]. The soundness of **make-cg** has already been established, which implies the soundness of line 4. Line 6 decomposes the problem using c-component factorization. The soundness proof for this decomposition, also used in the previous section, is in the appendix. Line 9 asserts that if a set of counterfactual events does not contain conflicting value assignments to any variable, obtained either by observation or intervention, then taking the union of all actions of the events results in a consistent action. The probability of the set of events can then be computed from a submodel where this consistent action has taken place. Full proof of this is in the appendix. □

To show completeness, I follow the same strategy I used in the previous section. I catalogue all difficult counterfactual graphs which arise from queries which cannot be identified from P_*. I then show these graphs arise whenever **ID*** and **IDC*** fail. This, together with the soundness theorem I already proved, implies that these algorithms are complete.

The simplest difficult counterfactual graph arises from the query $P(y_x, y'_{x'})$ named "probability of necessity and sufficiency" by [Pea00]. This graph, shown in Fig. 5.1 (b) with variable relabeling, is called the "w-graph" due to its shape [ASP05]. This query is so named because if $P(y_x, y'_{x'})$ is high, this implies that if the variable X is forced to x, variable Y is likely to be y, while if X is forced to some other value, Y is likely to not be y. This means that the action $do(x)$ is likely a necessary and sufficient cause of Y assuming value y, up to noise. The w-graph starts the catalogue of bad graphs with good reason, as the following lemma shows.

Lemma 4 *Assume X is a parent of Y in G. Then $P_*, G \not\vdash_{id} P(y_x, y'_{x'}), P(y_x, y')$ for any value pair y, y'.*

Proof: I construct two causal models M^1, M^2 that agree on P_* but disagree on the counterfactual distributions in question. In fact, I only need two variables. The two models agree on the following: X is the parent of Y, U_X, X and Y are binary variables, U_Y be a ternary variable, $f_X = U_X$, and $P(u_X)$, and $P(u_Y)$ are uniform. The two models only differ on the functions f_Y, which are given by Table 5.6. It's easy to verify the claim holds for the two models for any values $x^* \neq x$ of X. □

The intuitive explanation for this result is that $P(y_x, y'_{x'})$ is derived from the joint distribution over the counterfactual variables in the w-graph, while if I restrict myself to P_*, I only have access to marginal distributions – one marginal for each possible world. Because counterfactual variables Y_x and $Y_{x'}$ share an unobserved parent U, they are dependent, and their joint distribution cannot be decomposed into a product of marginals. This means that the information encoded in the marginals is insufficient to uniquely determine the joint we are interested in. This intuitive argument can be generalized to a counterfactual graph with more than two nodes, the so-called "zig-zag graphs" an example of which is shown in Fig. 5.7 (b).

Table 5.1: The functions f^1_Y and f^2_Y

X	U_Y	$Y = f^1_Y(x, u_Y)$	$Y = f^2_Y(x, u_Y)$
0	1	0	1
0	2	1	1
0	3	1	0
1	1	1	1
1	2	0	0
1	3	0	0

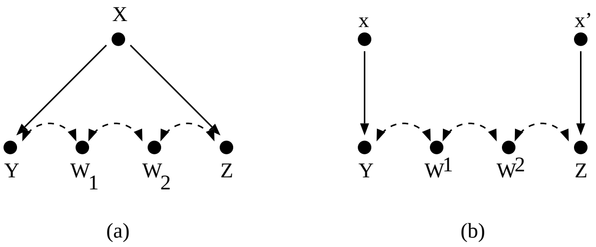


Figure 5.7: (a) Causal diagram (b) Corresponding counterfactual graph for the non-identifiable query $P(Y_x, W^1, W^2, Z_{x'})$.

Lemma 5 *Assume G is such that X is a parent of Y and Z, and Y and Z are connected by a bidirected path with observable nodes $W^1, ..., W^k$ on the path. Then $P_*, G \not\vdash_{id} P(y_x, w^1, ..., w^k, z_{x'}), P(y_x, w^1, ..., w^k, z)$ for any value assignments $y, w^1, ..., w^k, z$.*

The w-graph in Fig. 5.1 (b) and the zig-zag graph in Fig. 5.7 (b) have very special structure, so I don't expect my characterization to be complete with just these graphs. In order to continue, I must provide two lemmas which allow me to transform difficult graphs in various ways by adding nodes and edges, while retaining the non-identifiability of the underlying counterfactual from P_*.

Lemma 6 (downward extension lemma) *Assume $P_*, G \not\vdash_{id} P(\gamma)$.*
Let $\{y^1_{\boldsymbol{x}^1}, ..., y^n_{\boldsymbol{x}^m}\}$ be a subset of counterfactual events in γ. Let G' be a graph obtained from G by adding a new child W of $Y^1, ..., Y^n$. Let $\gamma' = (\gamma \backslash \{y^1_{\boldsymbol{x}^1}, ..., y^n_{\boldsymbol{x}^m}\}) \cup \{w_{\boldsymbol{x}^1}, ..., w_{\boldsymbol{x}^m}\}$, where w is an arbitrary value of W. Then $P_*, G' \not\vdash_{id} P(\gamma')$.

The first result states that non-identification on a set of parents (causes) translates into non-identification on children (effects). The intuitive explanation for this is that it is possible to construct a one-to-one function from the space of distributions on causes to the space of distributions on effects. If a given $P(\gamma)$

cannot be identified from P_*, this implies that there exist two models which agree on P_*, but disagree on $P(\gamma)$, where γ is a set of counterfactual causes. It is then possible to augment these models using the one-to-one function in question to obtain disagreement on $P(\delta)$, where δ is a set of counterfactual effects of γ. A more detailed argument is found in the appendix.

Lemma 7 (contraction lemma) *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let G' be obtained from G by merging some two nodes X, Y into a new node Z where Z inherits all the parents and children of X, Y, subject to the following restrictions:*

- *The merge does not create cycles.*
- *If $(\exists w_{\boldsymbol{s}} \in \gamma)$ where $x \in \boldsymbol{s}$, $y \notin \boldsymbol{s}$, and $X \in An(W)_G$, then $Y \notin An(W)_G$.*
- *If $(\exists y_{\boldsymbol{s}} \in \gamma)$ where $x \in \boldsymbol{s}$, then $An(X)_G = \emptyset$.*
- *If $(Y_{\boldsymbol{w}}, X_{\boldsymbol{s}} \in \gamma)$, then $\boldsymbol{w}$ and $\boldsymbol{s}$ agree on all variable settings.*

Assume $|X| \times |Y| = |Z|$ and there's some isomorphism f assigning value pairs x, y to a value $f(x, y) = z$. Let γ' be obtained from γ as follows. For any $w_{\boldsymbol{s}} \in \gamma$:

- *If $W \notin \{X, Y\}$, and values x, y occur in $\boldsymbol{s}$, replace them by $f(x, y)$.*
- *If $W \notin \{X, Y\}$, and the value of one of X, Y occur in $\boldsymbol{s}$, replace it by some z consistent with the value of X or Y.*
- *If X, Y do not occur in γ, leave γ as is.*
- *If $W = Y$ and $x \in \boldsymbol{s}$, replace $w_{\boldsymbol{s}}$ by $f(x, y)_{\boldsymbol{s} \setminus \{x\}}$.*
- *otherwise, replace every variable pair of the form $Y_{\boldsymbol{r}} = y, X_{\boldsymbol{s}} = x$ by $Z_{\boldsymbol{r},\boldsymbol{s}} = f(x, y)$.*

Then $P_*, G' \not\vdash_{id} P(\gamma')$.

This lemma has a rather complicated statement, but the basic idea is very simple. If I have a causal model with a graph G where some counterfactual $P(\gamma)$ is not identifiable, then a coarser, more "near-sighted" view of G which merges two distinct variables with their own mechanisms into a single variable with a single mechanism will not render $P(\gamma)$ identifiable. This is because merging nodes in the graph does not alter the model, but only our state of knowledge of the model. Therefore, whatever model pair was used to prove $P(\gamma)$ non-identifiable will remain the same in the new, coarser graph. The complicated statement of the lemma is due to the fact that I cannot allow arbitrary node merges, I must satisfy certain coherence conditions. For instance, the merge cannot create directed cycles in the graph.

It turns out that whenever **ID*** fails on $P(\gamma)$, the corresponding counterfactual graph contains a subgraph which can be obtained by a set of applications of the previous two lemmas to the w-graph and the zig-zag graphs. This allows an argument that shows $P(\gamma)$ cannot be identified from P_*.

Theorem 13 (completeness) *If* ***ID**** *or* ***IDC**** *fail, then the corresponding query is not identifiable from* P_*.

5.7 Corollaries

Since **ID*** is complete for $P(\gamma)$ queries, I can give a graphical characterization of counterfactual graphs where $P(\gamma)$ cannot be identified from P_*.

Theorem 14 *Let* G_γ, γ' *be obtained from* ***make-cg***(G, γ)*. Then* $P_*, G \not\vdash_{id} P(\gamma)$ *iff there exists a C-component* $S \subseteq An(\gamma')_{G_\gamma}$ *where some* $X \in Pa(S)$ *is set to* x *while at the same time either* X *is also a parent of another node in* S *and is set to another value* x'*, or* S *contains a variable derived from* X *which is observed to be* x'*.*

Proof: This follows from Theorem 13 and the construction of **ID***. □

CHAPTER 6

Path-specific Effects

In this chapter, I consider the problem of identifying path-specific effects. I show how path-specific effects, though understood to be causal effects along a subset of causal paths nevertheless can be represented using nested counterfactual variables. I will use this representation to express every path-specific effect in terms of counterfactual distributions considered in Chapter 5, and give complete graphical conditions for identifying these distributions in graphs without bidirected arcs. Furthermore, I will use the results on counterfactual identification found in Chapter 5 to give a powerful identification condition for path-specific effects in graphs with bidirected arcs as well. [1]

6.1 Natural Effects

Consider the study of UC Berkeley's alleged gender bias in admissions, as described in [PJ75], and Chapter 4 of [Pea00]. This case was interesting since the data "paradoxically" showed males were more likely to be admitted overall, while each department was more likely to admit females. Let's assume the causal diagram in Fig. 6.1 (a) is a coarse (but correct) representation of the admission situation: the applicants' gender influences their life goals, these goals along with their gender shape their decisions to apply at particular departments, while each department has its own admission procedure which incorporates the applicant competence (an unmeasured confounder between goals and admission), and possibly gender itself. To exonerate the university, we must show that the link between gender and admission is in some sense vacuous, in which case admission decisions are not based directly on gender. In other words, we must show that the admission decision would have stayed the same had gender been different, but everything else stayed the same.

[Pea01] introduces a special subscript notation to represent such hypothetical questions. Specifically, $Y_{x,Z_{x^*}}(\mathbf{u})$ is taken to mean the value achieved by Y when the background variables achieve values $\mathbf{u}$, we fix X to x, and Z to whatever value it would have attained when X is fixed to x^*. If we are uncertain about

[1] Some of the results in this chapter were derived as a joint work with Chen Avin.

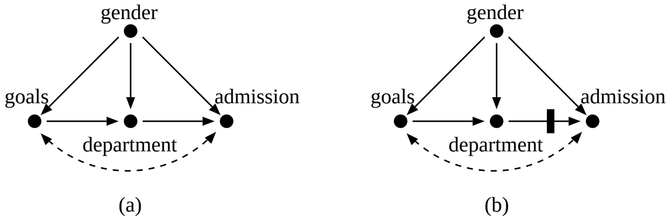


Figure 6.1: Causal diagram for the Berkeley discrimination domain (adopted from [Pea00]).

the values of $\mathbf{u}$, we have to deal with $Y_{x,Z_{x^*}}$ as a random variable. In such cases, there is no unique value z in the subscript. Instead, we must average over all possible value assignments to Z. In other words, $P(Y_{x,Z_{x^*}})$ is a shorthand for $\sum_z P(Y_{x,z}, Z_{x^*} = z)$.

In the model, we are interested in the probability $\sum_d P(admission_{gender=male,department=d}, department_{gender=female} = d)$, which is the probability of admission of a male given that all other known causes of admission assumed values consistent with being female. One way to describe this probability is as a direct effect of gender on admissions. In Chapter 4, I defined the direct effect of X on Y by considering how $do(x)$ affects Y, when all other parents $\mathbf{W}$ of Y are fixed to specific values $\mathbf{w}$. The sort of direct effect I discuss here, where we average over possible parent settings under a setting of X to a default value x^* is called *natural direct effect* in [Pea01]. Aside from being a more faithful formalization of the intuitive quantity relevant to discrimination cases, natural direct effects have another advantage over conventional direct effects – they allow a symmetric definition of an intuitive definition of "indirect effects." In the discrimination case, an indirect effect would correspond to all ways gender can influence admission – except any direct influence. The conventional direct effect definition cannot be extended to handle indirect effects, however natural effects easily express indirect effects by merely changing reference values. For instance the indirect effect of being male on admission would be represented by the expression $\sum_d P(admission_{gender=female,department=d}, department_{gender=male} = d)$.

I can represent natural effects graphically by marking "forbidden" edges whose parents behave as if the control variable was set to a reference value. For instance, Fig. 6.1 (b) represents the natural direct effect of gender on admission, so the edge from department to admission is crossed out. Being able to "forbid" arbitrary paths when considering causal effects is a powerful notion, which comes up in situations other than discrimination.

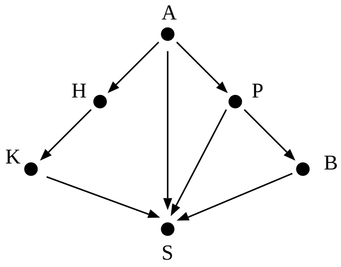


Figure 6.2: Causal model for the AZT domain.

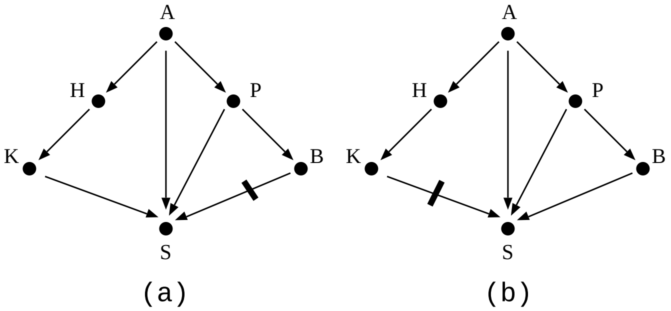


Figure 6.3: Path-specific effects in the AZT domain

6.2 An Example of Path-specific Effect

Consider the following example, inspired by [Rob97]. A study is performed on the effects of the AZT drug on AIDS patients. AZT is a harsh drug known to cause a variety of complications. For the purposes of the model, I restrict my attention to two – pneumonia and severe headaches. In turn, pneumonia can be treated with antibiotics, and severe headache sufferers can take painkillers. Ultimately, all the above variables, except headache, are assumed to have a direct effect on the survival chances of the patient. The graphical causal model for this situation is shown in Fig. 6.2.

Say we are interested in the interactions between antibiotics and AZT that negatively affect survival. To study such interactions, we might consider the effect of administering AZT on survival in the idealized situation where the antibiotics variable behaved as if AZT was not administered, and compare this to the effect of AZT on survival (where side effects are present). Graphically this amounts to "forbidding" the direct edge between antibiotics and survival. This is shown graphically in Fig. 6.3 (a). Similarly, the path-specific effect in Fig. 6.3 (b) represents the idealized situation where AZT has no side-effects on painkiller medication.

6.3 Counterfactual Definition of Path-Specific Effects

Path-specific effects in a model M as they were defined in Chapter 3, and in [Pea01], are really total effects in a causal model M^* modified from the original by replacing certain causal mechanisms. It is awkward to use this definition directly if we are interested in identifying path-specific effects, since my arguments must rest on the bedrock of algebraic manipulations. Therefore, I provide a generalization of Pearl's subscript notation for natural effects, which I show will be sufficient to represent arbitrary path-specific effects in terms of counterfactual distributions of the original causal model M.

Definition 9 (nested counterfactual variable) *Let M be a causal model. A nested counterfactual variable is defined inductively as either a counterfactual variable $Y_{\boldsymbol{x}}(\boldsymbol{u})$, (where Y is a variable, and $\boldsymbol{X}$ is a variable set in M), or a variable $Y_{\boldsymbol{x},z^1,...,z^k}(\boldsymbol{u})$, where $z^1,...,z^k$ are values attained by nested counterfactual variables $Z^1(\boldsymbol{u}),...,Z^k(\boldsymbol{u})$.*

Note that the domain of a nested counterfactual variable always corresponds to a domain of some variable in the original causal model. Thus, the index notation I use is meaningful. The difference between nested counterfactual variables and ordinary counterfactual variables defined in Chapter 3, is that the values which occur in the subscripts of the former are not given constants, but are attained inductively from other nested counterfactual variables. I will avoid deep subscript nesting by referring to nested counterfactual variables by a single name such as $Z^i_{..}$ and summarize the nesting in the subscript by the ellipsis, rather than by listing its entire expression.

If we are uncertain about the values $\mathbf{u}$ of background nodes, nested counterfactual variables, like their ordinary counterparts, become random variables. Since writing $Y_{\mathbf{x},z^1,...,z^k}(\mathbf{u})$ is equivalent to writing $Y_{\mathbf{x},Z^1_{..}(\mathbf{u}),...,Z^k_{..}(\mathbf{u})}(\mathbf{u})$, by definition, I will use the notation $P(Y_{\mathbf{x},Z^1_{..},...,Z^k_{..}} = y)$ (with nested variables in the subscript) as a shorthand for $\sum_{\{\mathbf{u}|Y_{\mathbf{x},Z^1_{..}(\mathbf{u}),...,Z^k_{..}(\mathbf{u})}(\mathbf{u})=y\}} P(\mathbf{u})$. Note that variables $Z^i_{..}$ may themselves involve nested subscripts, so the overall expression may be quite difficult to write.

The following lemma shows how nested counterfactual random variables can be expressed in terms of distributions over counterfactual events.

Lemma 8 $P(Y_{\boldsymbol{x},Z^1_{..},...,Z^k_{..}}) = \sum_{z^1,...,z^k} P(Y_{\boldsymbol{x},z^1,...,z^k}, Z^1_{..} = z^1,...,Z^k_{..} = z^k)$, *where $Z^i_{..} = z^i$ stands for the event "nested counterfactual variable $Z^i_{..}$ assumes values z^i."*

I can use Lemma 8 to express every nested counterfactual in terms of joint probability distributions over ordinary counterfactual variables.

Theorem 15 *Let $Y_{\boldsymbol{x},Z^1_{..},...,Z^k_{..}}$ be a nested counterfactual variable (with $Z^1_{..},...,Z^k_{..}$ nested counterfactual variables as well). For every nested counterfactual variable $W_{\boldsymbol{m},S^1_{..},...,S^k_{..}}$ used in the inductive definition of $Y_{\boldsymbol{x},Z^1_{..},...,Z^k_{..}}$, let $W_{\boldsymbol{m},s^1,...,s^k}$ be the corresponding "unrolled" ordinary counterfactual ($s^1,...,s^k$ are values attained by $S^1_{..},...,S^k_{..}$).*

Then $P(Y_{\boldsymbol{x},Z^1_{..},...,Z^k_{..}}) = \sum_{\boldsymbol{s}} P(\bigwedge_i W^i_{\boldsymbol{m},s^1,...,s^k})$, where the index i ranges over all "unrolled" ordinary counterfactuals attained from nested counterfactuals which occur in $Y_{\boldsymbol{x},Z^1_{..},...,Z^k_{..}}$, and $\boldsymbol{s}$ is the set of values attained by all nested counterfactuals in $Y_{\boldsymbol{x},Z^1_{..},...,Z^k_{..}}$, except $Y_{\boldsymbol{x},Z^1_{..},...,Z^k_{..}}$ itself.

This result shows that nested counterfactuals are quantities obtainable from joint distributions over ordinary counterfactual variables. What I now show is that every path-specific effect of a single variable X on another single variable Y is expressible as a nested counterfactual, and thus as a counterfactual distribution.

Theorem 16 *Let g be a subset of "allowed edges." Let $\boldsymbol{Y}_{\boldsymbol{x}}(\boldsymbol{u}) - \boldsymbol{Y}_{\boldsymbol{x}*}(\boldsymbol{u})$ be a path-specific effect in M_g. Then both (sets of) random variables $\boldsymbol{Y}_{\boldsymbol{x}}$, $\boldsymbol{Y}_{\boldsymbol{x}^*}$ can be expressed in terms of a nested counterfactual in the original model M.*

Corollary 3 *Let g be a subset of "allowed edges." Let $\boldsymbol{Y}_{\boldsymbol{x}}(\boldsymbol{u}) - \boldsymbol{Y}_{\boldsymbol{x}*}(\boldsymbol{u})$ be a path-specific effect in M_g. Then both (sets of) random variables $\boldsymbol{Y}_{\boldsymbol{x}}$, $\boldsymbol{Y}_{\boldsymbol{x}^*}$ can be expressed in terms of counterfactual distributions in the original model M.*

6.4 Effect-invariant Transformations

Path-specific effects have two complementary representations, as quantities derived from counterfactual distributions, and as marked graphs. The marked graph representation is by far the more intuitive, so it would be preferable to operate on graphs rather than distributions. In this section, I introduce three rules which allow us to make changes to the marked graphs without affecting either the value or the identifiability of the corresponding path-specific effect. Systematic application of these three rules will allow me to derive a complete identification condition for path-specific effects of a single variable X on a single outcome Y in Markovian graphs (that is, graphs without bidirected arcs).

Definition 10 (rule 1) *Rule 1 applies to a marked graph G_g at V if all arrows outgoing from V which start directed paths from V to Y are forbidden. The* *marked graph $G_{R_1^v(g)}$ obtained from G_g by the application of rule 1 forbids all incoming arrows to V and allows all previously marked outgoing arrows from V, leaving the status of other edges unchanged. See Fig. 6.4.*

The important invariant with path-specific effects is the set of all allowed paths, that is paths consisting only of allowed edges, from X to Y, and this set is not changed by the application of rule 1, since any path which contains a newly forbidden edge incoming to V must have had a forbidden edge leaving V.

Definition 11 (rule 2) *Rule 2 applies to a marked graph G_g at V if there is a forbidden edge e leaving V, and all directed paths from X to V contain forbidden edges. The marked graph $G_{R_2^v(g)}$ obtained from G_g by the application of rule 2 allows the formerly forbidden edge e, leaving the status of other edges unchanged. See Fig. 6.5.*

Rule 2 also preserves the set of all allowed paths since any path containing the newly allowed edge e cannot be an allowed path.

Definition 12 (rule 3) *Rule 3 applies to a marked graph G_g at V if there is a forbidden edge e entering V, and $V \notin An(Y)$, or there is a forbidden edge e leaving V, and $V \notin De(X)$. The marked graph $G_{R_3^v(g)}$ obtained from G_g by the application of rule 3 allows the formerly forbidden edge e, leaving the status of other edges unchanged. See Fig. 6.6.*

I want to prove a result which will lets us conclude that arbitrary changes of the marked graph using rules 1, 2, and 3 do not change the underlying path-specific effect. To prove this, I need one utility lemma.

Lemma 9 *Let $V_{..}$ be a nested counterfactual where all constant subscripts are the same and equal to $\boldsymbol{x}$. Then $V_{..} = V_{\boldsymbol{x}}$.*

Proof: This follows by definition of nested counterfactuals. □

Theorem 17 *If rule 1 applies to G_g at V, then the path-specific effect in G_g is equal to the path-specific effect in $G_{R_1^v(g)}$. If rule 2 applies to G_g at V, then the path-specific effect in G_g is equal to the path-specific effect in $G_{R_2^v(g)}$. If rule 3 applies to G_g at V, then the path-specific effect in G_g is equal to the path-specific effect in $G_{R_3^v(g)}$.*

Since R_1 moves forbidden edges closer to the manipulated variables and R_2, R_3 remove redundant forbidden edges, it is not surprising that these two rules cannot be applied forever in a marked graph.

Lemma 10 *Let G_g be a marked graph. Then rules 1, 2 and 3 can only be applied finitely many times.*

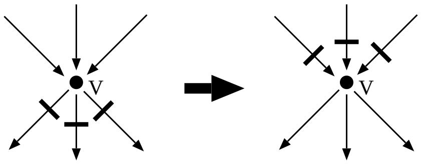


Figure 6.4: Rule 1

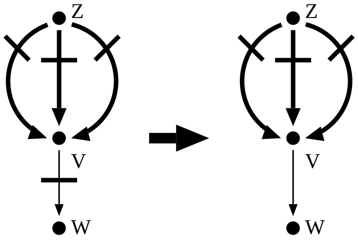


Figure 6.5: Rule 2 (marked thick arrows correspond to forbidden directed paths).

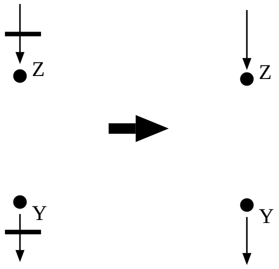


Figure 6.6: Rule 3

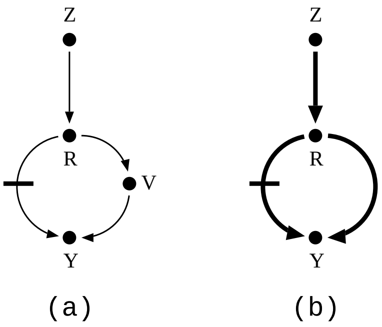


Figure 6.7: (a) The simplest non-identifiable path-specific effect (b) The kite graph (thick arrows correspond to directed paths)

6.5 Completeness for Single-Source Single-Outcome Path-specific Effects

I will use the two rules defined in the previous section to obtain a completeness result for identification of path-specific effects from a single variable X to a single outcome Y in Markovian graphs. The general strategy will be similar to that used in the previous chapters. I will show that a particular, simple kind of counterfactual distribution is not identifiable, and then show that this distribution arises in all marked graphs of a certain form. I will then repeatedly use the two rules to reduce a given marked graph to a form where identification becomes simple to establish.

I start with a non-identifiable counterfactual distribution which already made an appearance in Chapter 5.

Lemma 4 *Assume X is a parent of Y in G. Then $P_*, G \not\vdash_{id} P(y_x, y'_{x'}), P(y_x, y')$ for any value pair y, y'.*

The next theorem shows how a particular path-specific effect leads to problematic counterfactuals from the previous lemma.

Theorem 18 *The g-specific effect of Z on Y as described in Fig. 6.7 (a) is not P_*-identifiable.*

It turns out that anytime a path-specific effect of X on Y is not identifiable, the corresponding marked graph looks similar to the graph in Fig. 6.7 (a), in fact it looks like the graph in Fig. 6.7 (b), where thick arrows are interpreted as directed paths. Whenever the graph has this “kite” structure, I say it satisfies the recanting witness criterion.

Definition 13 (recanting witness criterion) *Let $R \neq Z$ be a node in G, such that there exists a directed path in g from Z to R, a directed path from R to Y in g, and a direct path from R to Y in G but not g. Then Z, Y, and g satisfy the recanting witness criterion with R as a witness*

The name "recanting witness" comes from the behavior of the variable R in the center of the "kite." This variable, in some sense, "tries to have it both ways." Along one path from R to Y, R behaves as if the variable Z was set to one value, but along another path, R behaves as if Z was set to another value. This "changing of the story" of R is what causes the problem, and as I will show it essentially leads to the the existence of a non P_*-identifiable counterfactual in Theorem 4.

I now show that repeated applications of rules 1, 2, and 3 to a marked graph with a single source X and a single outcome Y result in either the "kite" graph, or a marked graph where all marked arrows leave X.

Theorem 19 *Assume G_g is a marked graph with a single source X and a single outcome Y, such that rules 1,2, and 3 do not apply. Then either G_g satisfies the recanting witness criterion, or all marked edges emanate from X.*

What I have left to show is that the kite graph always results in a non-identifiable path-specific effect, and a graph where all marked nodes leave X results in an identifiable path-specific effect.

Theorem 20 *Assume rules 1, 2, and 3 do not apply to G_g, and G_g satisfies the recanting witness criterion. Then the g-specific effect of X on Y is not P_*-identifiable.*

Theorem 21 *If rules 1, 2, and 3 do not apply to G_g and all marked arrows emanate from X, then the path-specific effect of X on Y along g is identifiable in Markovian models.*

6.6 General Path-specific Effects

In the previous section, I developed a complete characterization of identifiable path-specific effects from a single source X to a single outcome Y in terms of marked Markovian graphs. It turns out that it is possible to generalize the graphical condition developed in the previous section for the case of multiple sources and multiple outcomes. Unfortunately, if the marked graph is semi-Markovian, there is no longer a straightforward graphical representation of identifiable path-specific effects, since individual counterfactuals in the counterfactual distribution representation of path-specific effects are no longer independent. However, I can use the results I developed in Chapter 5 to give identification conditions in this more general setting as well, although such conditions are not necessarily complete.

First, I need to generalize distributions over a single nested counterfactual to range over multiple such counterfactuals.

Definition 14 (nested counterfactual distributions) *Let* $Y^1_{..}, ...Y^k_{..}$ *be a set of nested counterfactual variables. Then I define* $P(Y^1_{..} = y^1, ..., Y^k_{..} = y^k)$ *as a shorthand for* $\sum_{\{\boldsymbol{u}|Y^1_{..}(\boldsymbol{u})=y^1,...,Y^k_{..}(\boldsymbol{u})=y^k\}} P(\boldsymbol{u})$*.*

It turns out that I can generalize Theorem 15 to show that every nested counterfactual distribution can be expressed in terms of distributions over ordinary counterfactual variables.

Theorem 22 $P(Y^1_{..} = y^1, ..., Y^k_{..} = y^k) = \sum_{\boldsymbol{s}} P(\bigwedge_i W^i_{..})$*, where the index* i *ranges over all "unrolled" ordinary counterfactuals attained from nested counterfactuals which occur in* $Y^1_{..}, ..., Y^k_{..}$*, and* $\boldsymbol{s}$ *is the set of values attained by all nested counterfactuals in* $Y^1_{..}, ..., Y^k_{..}$*, except those in the set* $\{Y^1_{..}, ..., Y^k_{..}\}$*.*

Proof: The proof is a straightforward generalization of the proof of Theorem 15. □

If I restrict myself to Markovian graphs, I need not reason on the level of counterfactual distributions, but can deal instead with marked graphs, as in the previous section. However, I need to generalize the three graph transformation rules I used to work in the multi-source multi-outcome setting. It turns out that rule 1 carries over to this setting without changes, while rules 2 and 3 merge into a new rule.

Definition 15 (unmarking rule) *The unmarking rule applies to a marked graph* G_g *at a marked edge* e *emanating from node* V *if either there are no allowed directed paths from* $\boldsymbol{X}$ *to* V*, or* $V \notin An(\boldsymbol{Y})$*. The marked graph* $G_{R_4^e(g)}$ *obtained from* G_g *by the application of the unmarking rule allows the formerly forbidden edge* e*, leaving the status of other edges unchanged.*

As with the other rules, applications of the unmarking rule are "safe," in the sense that the path-specific effect is preserved.

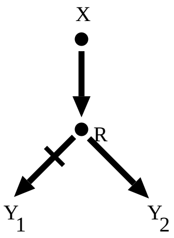


Figure 6.8: The generalized kite graph (Y_1, Y_2 may be the same node). Thick arrows correspond to directed paths.

Theorem 23 *If the unmarking rule applies to G_g at e, then path-specific effect in G_g is equal to the path-specific effect in $G_{R^e_4(g)}$.*

As before, rule 1, and the unmarking rule can only be applied finitely many times in a given marked graph, and if they can no longer be applied, the resulting graph will be in one of two forms. The first form will generalize the "kite graph" from the previous section, while in the second form all marked edges emanate from **X**.

Theorem 24 *Assume G_g is a marked graph, we are interested in a g-specific effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$, and neither rule 1, nor the unmarking rule are applicable to G_g. Then either all marked edges emanate from nodes in $\boldsymbol{X}$, or there is a node R such that there is an allowed directed path from $\boldsymbol{X}$ to R, an allowed directed path from R to $\boldsymbol{Y}$, and a forbidden directed path from R to $\boldsymbol{Y}$. See Fig. 6.8.*

What remains to show is that the first form, corresponding to the generalized kite graph always results in a non-identifiable path-specific effect, while the second form results in identifiable path-specific effects in Markovian graphs.

Theorem 25 *Assume G_g contains the patterns shown in Fig. 6.8. Then the g-specific effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$ is not P_*-identifiable.*

Theorem 26 *Assume all marked arrows emanate from $\boldsymbol{X}$ in G_g. Then the path-specific effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$ is identifiable in Markovian models.*

Having established a complete condition for identification of path-specific effects with multiple sources and multiple outcomes in Markovian graphs, we turn to the semi-Markovian case. Unfortunately, while most of the reasoning carries over without change, I can no longer establish independence of each counterfactual term, as in the proof of the Theorem 26. This means that there is no longer a complete condition for identification which can be expressed in a straightforward way using the marked graph. However, I can use the results developed in Chapter 5 to obtain a condition for identification using the **ID*** algorithm.

Corollary 4 *Let G_g be a marked graph, $\boldsymbol{X}$ the set of sources, $\boldsymbol{Y}$ the set of outcomes. Let P' be the counterfactual distribution corresponding to a path-specific effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$ due to Corollary 3. Then the path-specific effect is identifiable if P' is identifiable by* ***ID*****.*

CHAPTER 7

Dormant Independence

In this chapter, I consider dormant independencies, in other words conditional independencies in interventional distributions. I develop an algorithm which, given two arbitrary sets of variables, determines in polynomial time if there is an identifiable dormant independence between them. I show that this algorithm is complete in a sense that if it fails, there is no "good graphical reason" for there to be a dormant independence (although it might still exist in some models). I also show how dormant independencies can be used for model testing and induction, in a way similar to conditional independencies, by giving an algorithm which tests for the presence of extraneous edges in causal diagrams.

7.1 An Example of Dormant Independence

Consider the causal graph in Fig. 7.1 (a). Any model which induces this graph is subject to certain constraints on its observable distribution. Some of these constraints are due to conditional independence. For instance, in any such model $X \perp\!\!\!\perp Z|W$, which means $P(x|w)$ must equal $P(x|w, z)$. However, there is an additional constraint implied by this graph which cannot be expressed in terms of conditional independence in the observable distribution. This constraint, noted in [VP90], states that the distribution $\sum_w P(y|z, w, x)P(w|x)$ is a function of only y and z, but not x. The key insight that motivates this chapter is that this constraint does emanate from conditional independencies, albeit not the original observable distribution, but rather in an interventional distribution.

Consider a model M inducing the graph in Fig. 7.1 (a). If we intervene on

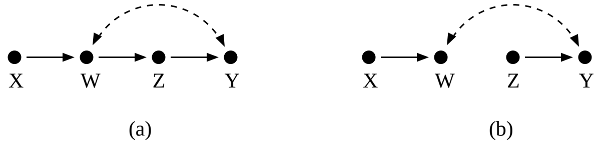


Figure 7.1: (a) The "P" graph. (b) The graph of the submodel M_z derived from the "P" graph.

Z, we obtain the submodel M_z inducing the graph in Fig. 7.1 (b). Moreover, the distribution of the unfixed observables in this submodel, $P_z(x, w, y)$, is identifiable and equals to $P(y|z, w, x)P(w|x)P(x)$. It's not difficult to establish by inspecting the graph in Fig. 7.1 (b) that X is d-separated from Y, and so $X \perp\!\!\!\perp Y$ in $P_z(x, w, y)$. This implies that $P_z(y|x) = P_z(y)$. But it's not hard to show that $P_z(y|x)$ is equal to $\sum_w P(y|z, w, x)P(w|x)$, which means this expression depends only on z and y. Thus, the identifiability of $P_z(x, w, y)$ leads to a constraint on observational distributions in the original, unmutilated model M.

Enumerating constraints of this type can be used to infer features of the causal graphs, just as conditional independencies are used for this purpose by causal induction algorithms. For example, establishing that X is independent of Y in $P_z(x, w, y)$ allows us to conclude that the causal graph lacks an edge between X and Y, assuming that the submodel M_z is stable [PV91], [Pea00], or faithful [SGS93]. Moreover, since $P_z(x, w, y)$ is identifiable from $P(\mathbf{v})$ in the graph in question, we can conclude the edge absence without relying on interventions.

In the remainder of this chapter, I show how to achieve a full enumeration of conditional independencies in identifiable interventional distributions entailed by the structure of the graph, and how to use these independencies to infer features of the graph.

7.2 Dormant Independence and d*-separation

I call a conditional independence *dormant* if it exists in an interventional distribution.

Definition 16 (dormant independence) *A dormant independence exists between variable sets* $\boldsymbol{X}, \boldsymbol{Y}$ *in* $P(\boldsymbol{v})$ *obtained from the causal graph* G *if there exist variable sets* $\boldsymbol{Z}, \boldsymbol{W}$ *such that* $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, do(\boldsymbol{w})) = P(\boldsymbol{y}|\boldsymbol{z}, do(\boldsymbol{w}))$. *Furthermore, if* $P(\boldsymbol{v}), G \vdash_{id} P(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{z}, do(\boldsymbol{w}))$, *the dormant independence is identifiable and I denote this as* $\boldsymbol{X} \perp\!\!\!\perp_{\boldsymbol{w}} \boldsymbol{Y}|\boldsymbol{Z}$. *If an identifiable dormant independence does not exist between* $\boldsymbol{X}, \boldsymbol{Y}$ *I write* $\boldsymbol{X} \not\perp\!\!\!\perp_{*} \boldsymbol{Y}$.

I would like to represent dormant independence using graphs. Fortunately, every concept I used in the definition of dormant independence has a graphical interpretation: ordinary conditional independence can be represented using d-separation, the effect of interventions on a graph can be represented by cutting incoming arrows to intervened nodes, and complete graphical conditions for identification of interventions has been developed in Chapter 4. Using these interpretations together allows us to generalize d-separation in appropriate way to mirror dormant independence. I call the resulting notion d*-separation.

Definition 17 (d*-separation) *Let G be a causal diagram. Variable sets $\boldsymbol{X}$, $\boldsymbol{Y}$ are d*-separated in G given $\boldsymbol{Z}$, $\boldsymbol{W}$ (written $\boldsymbol{X} \perp_{\boldsymbol{w}} \boldsymbol{Y}|\boldsymbol{Z}$), if we can find sets $\boldsymbol{Z}$, $\boldsymbol{W}$, such that $\boldsymbol{X} \perp \boldsymbol{Y}|\boldsymbol{Z}$ in $G_{\overline{\mathbf{w}}}$, and $P(\boldsymbol{v}), G \vdash_{id} P(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{z}, do(\boldsymbol{w}))$. If $\boldsymbol{X}$, $\boldsymbol{Y}$ are not d*-separable, we write $\boldsymbol{X} \not\perp_{*} \boldsymbol{Y}$.*

Note that despite the presence of probability notation in the definition, this is a purely graphical notion, since identification can be determined using only the graph by the back-door hedge criterion. Consequently, I can prove a theorem analogous to Theorem 1 for identifiable dormant independencies, which allows us to reason about such independencies graphically.

Theorem 27 *Let G be a causal diagram. Then in any model M inducing G, if $\boldsymbol{X} \perp_{\boldsymbol{w}} \boldsymbol{Y}|\boldsymbol{Z}$, then $\boldsymbol{X} \perp\!\!\!\perp_{\boldsymbol{w}} \boldsymbol{Y}|\boldsymbol{Z}$.*

Proof: This follows from the fact that $G_{\overline{\mathbf{w}}}$ is the graph induced by the submodel $M_{\mathbf{w}}$, and any submodel is just an ordinary causal model where Theorem 1 holds. $\square$

In the following two sections I will develop a complete condition for d*-separation of two disjoint sets of variables $\mathbf{X}$ and $\mathbf{Y}$, and a corresponding algorithm which returns the conditioning set $\mathbf{Z}$ and intervention set $\mathbf{W}$ which witness this d*-separation. In this way I capture all identifiable dormant independencies which have a "graphical reason" to exist.

7.3 D*-separation Among Singletons

In this section, I consider a simpler problem of determining if variables X and Y can be rendered conditionally independent in some identifiable interventional distribution. To characterize identifiable dormant independence between X and Y, it makes sense to consider the "difficult" neighborhoods of X, Y, in a sense that no intervention on those neighborhoods is identifiable. I call such neighborhoods ancestral confounding sets.

Definition 18 *Let Y be a variable in G. A set S is ancestral confounded (ACS) for Y if $S = An(Y)_{G_S} = C(Y)_{G_S}$.*

Ancestral confounded sets are "difficult" because they can be used to form a Y-rooted C-tree, and I know from Chapter 4 that the effect of any intervention in this structure on Y is not identifiable.

Theorem 28 *Let S be ancestral confounded for Y. Then for any $S' \subseteq S \setminus \{Y\}$, $P(\boldsymbol{v}), G \not\vdash_{id} P(y|do(s'))$.*

```
function **Find-MACS**($G, Y$)
INPUT: $G$, a causal diagram, $Y$ a node in $G$.
OUTPUT: $T_y$, the MACS for $Y$ in $G$.

1 If $(\exists X \notin An(Y)_G)$,
    return **Find-MACS**($G_{An(Y)}, Y$).
2 If $(\exists X \notin C(Y)_G)$,
    return **Find-MACS**($G_{C(Y)}, Y$).
3 Else, return $G$.
```

Figure 7.2: An algorithm for computing the MACS of a node.

Proof: It's trivial to construct a Y-rooted C-tree T from S. But it is known from Theorem 3 that for any set S' of nodes in T that does not contain Y, $P(\mathbf{v}), G \nvdash_{id} P(y|do(s'))$. □

In my search for suitable variables to intervene on, in order to separate X and Y, I can exclude ancestral confounded sets for X and Y. But there can be potentially many such sets. It would be preferable to exclude all such sets at once. Fortunately, the following results allows us to accomplish just that.

Theorem 29 *For any variable Y in G, there exists a unique maximum ancestral confounded set (MACS) T_y.*

T_y contains all ancestral confounded sets for Y, which means if I can find an efficient procedure for computing T_y, I could rule out all "difficult" sets from consideration at once. Such an algorithm exists, and is given in Fig. 7.2.

Theorem 30 ***Find-MACS***(G, Y) *outputs the MACS of Y in polynomial time.*

In the effort to d*-separate X and Y no interventions on nodes in in T_x and T_y can be made, since these interventions are not identifiable. Furthermore, conditioning on T_y or T_x does not d-separate paths from Y out of T_y which consist entirely of colliders, although all paths with a non-collider are blocked. In order to block some all-collider paths out of T_x, T_y we can attempt to intervene on the set $Pa(T_x \cup T_y) \setminus (T_x \cup T_y)$. It turns out these interventions are sufficient to create identifiable dormant independence among singletons, if one exists.

Figure 7.3: (a) A graph where $X \perp_z Y|W, K, L, N$. (b) A graph where $X \perp_z Y$, $X \perp_k L$, but $X \not\perp_* \{Y, L\}$.

Theorem 31 *Let T_x, T_y be the MACSs of X, Y. Let $I_{x,y} = Pa(T_x \cup T_y) \setminus (T_x \cup T_y)$. Then if either X is a parent of T_y, Y is a parent of T_x or there is a bidirected arc between T_x an T_y, then X, Y are not d*-separable. Otherwise, $X \perp_{i_{x,y}} Y|T_x \cup T_y \setminus \{X, Y\}$.*

To illustrate this theorem, consider the graph in Fig. 7.3. Here, $T_y = \{K, L, N, Y\}$, and $T_x = \{W, X\}$. By Theorem 31, $X \perp_z Y|W, K, L, N$.

Thus, the MACSs turn out to be key structures for determining d*-separation between two variables. In the next section, we generalize my results to handle d*-separation among sets of variables.

7.4 D*-separation Among Sets

To determine if two arbitrary disjoint sets can be d*-separated I consider a multi-node generalization of MACS. Unfortunately a MACS, as it is defined in the previous section, is not guaranteed to exist for sets of nodes (consider for instance a set consisting of two nodes with no path connecting them). In order to generalize the notion of a MACS appropriately, I must consider a partition of an arbitrary set where a MACS can be defined for each element in the partition. I start with a straightforward generalization of ancestral confounded sets for sets of variables.

Definition 19 *Let $\boldsymbol{Y}$ be a variable set in G. A set S is ancestral confounded for $\boldsymbol{Y}$ if for every $Y \in \boldsymbol{Y}, S = An(\boldsymbol{Y})_{G_S} = C(Y)_{G_S}$.*

I want to define an appropriate partition of an arbitrary set, where each element of the partition has an ACS. I will show the following definition will work for this purpose.

Definition 20 (AC-component) *A set $\boldsymbol{Y}$ of nodes in G is an ancestral confounded component (AC-component) if*

- $\boldsymbol{Y} = \{Y\}$, *e.g.,* $\boldsymbol{Y}$ *is a singleton set, or*
- $\boldsymbol{Y}$ *is a union of two distinct AC-components* $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$ *which have ancestral confounded sets* S_1, S_2*, respectively, and* S_1, S_2 *are connected by a bidirected arc*

Lemma 11 *Every AC-component has an ancestral confounded set.*

AC-components behave just as singleton sets do with respect to ACS. In fact, there is a unique MACS for every AC-component, and the algorithm to find it is the familiar **Find-MACS** with set inputs.

Theorem 32 *Let* $\boldsymbol{Y}$ *be an AC-component. Then there exists a unique MACS* $T_{\boldsymbol{y}}$ *for* $\boldsymbol{Y}$*, and* ***Find-MACS*** *(shown in Fig. 7.4) finds it in polynomial time.*

Proof: The proof is a straightforward generalization of the proof of Theorems 30 and 29. □

What I have shown is that certain special sets of nodes have a MACS, just as singletons do. While I cannot show the same for arbitrary sets, I can show the next best thing, namely that there exists a unique partition of any set into AC-components.

Lemma 12 *Let* $\boldsymbol{Y}$ *be a variable set,* $Y \in \boldsymbol{Y}$*. Then there is a unique maximum AC-component which both contains* Y *and is a subset of* $\boldsymbol{Y}$*.*

Theorem 33 *Any variable set* $\boldsymbol{Y}$ *has a unique partition* p*, called the AC-partition, where each element* S *in* p *is a maximal AC-component in a sense that no superset of* S *which is also a subset of* $\boldsymbol{Y}$ *is an AC-component.*

There is a simple algorithm, shown in Fig. 7.4, which, given an arbitrary set $\mathbf{Y}$, finds the unique AC-partition p of $\mathbf{Y}$, and finds the MACS for each AC-component in p.

Theorem 34 ***Find-AC-Partition***$(G, \boldsymbol{Y})$ *outputs the unique AC-partition of* $\boldsymbol{Y}$*, and the set of MACSs for each element in the partition.*

I want to prove a result analogous to Theorem 31 for sets. To do so, I must generalize the notion of an inducing path to sets.

Definition 21 (inducing paths for sets) *Let* $\boldsymbol{X}$, $\boldsymbol{Y}$ *be sets of variables in* G*. A path* p *between* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *is called an inducing path if the following two conditions hold*

function **Find-AC-Partition**$(G, \mathbf{Y})$
INPUT: G, a causal diagram, $\mathbf{Y}$ a set of nodes in G.
OUTPUT: p, the unique partition of $\mathbf{Y}$ into AC-components, and the unique MACS $T_{\mathbf{s}}$ for each $\mathbf{S} \in P$.

```
1 Let $p$ be the partition of $\mathbf{Y}$ containing all singleton subsets of $\mathbf{Y}$.
2 For each $Y \in \mathbf{Y}$, let $T_y =$ Find-MACS$(G, \{Y\})$.
3 Repeat until no merges are possible: If $\exists \mathbf{Y}_1, \mathbf{Y}_2 \in p$ such that $T_{\mathbf{y}_1}, T_{\mathbf{y}_2}$
  share a bidirected arc, merge $\mathbf{Y}_1, \mathbf{Y}_2$ into $\mathbf{Y}'$ in $p$, and let $T_{\mathbf{y}'} =$ Find-MACS$(G, \mathbf{Y}')$.
4 return $p$, and the set of MACSs for each element in $p$.
```

function **Find-MACS**$(G, \mathbf{Y})$
INPUT: G, a causal diagram, $\mathbf{Y}$ an AC-component in G.
OUTPUT: $T_{\mathbf{y}}$, the MACS for $\mathbf{Y}$ in G.

```
1 If $(\exists X \not\in An(\mathbf{Y})_G)$,
  return Find-MACS$(G_{An(\mathbf{Y})}, \mathbf{Y})$.
2 If $(\exists X \not\in C(Y)_G)$,
  return Find-MACS$(G_{C(\mathbf{Y})}, \mathbf{Y})$.
3 Else, return $G$.
```

Figure 7.4: An algorithm for computing the AC-partition (and the corresponding sets of MACSs) of $\mathbf{Y}$.

- *The path forms a collider for every non-terminal node*
- *Every non-terminal node is an ancestor of* $\boldsymbol{X}$ *or* $\boldsymbol{Y}$.

Not surprisingly, inducing paths characterize d-separability for sets just as they do for singleton variables.

Theorem 35 $\boldsymbol{X}$ *cannot be d-separated from* $\boldsymbol{Y}$ *in* G *if and only if there exists an inducing path from* $\boldsymbol{X}$ *to* $\boldsymbol{Y}$ *in* G*,*

I can now prove the generalization of Theorem 31 for sets. The idea is to find the AC-partition of $\mathbf{X} \cup \mathbf{Y}$, and generalize the two conditions for d*-separability in Theorem 31 for this AC-partition.

Theorem 36 *Let* $\boldsymbol{X}$, $\boldsymbol{Y}$ *be arbitrary sets of variables. Let* p *be the AC-partition of* $\boldsymbol{X} \cup \boldsymbol{Y}$*. Then if either elements of both* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *share a single AC-component in* p*, or some element of* $\boldsymbol{X}$ *is a parent of the MACS of some AC-component containing elements of* $\boldsymbol{Y}$ *(or vice versa), then* $\boldsymbol{X}$ *cannot be d*-separated from* $\boldsymbol{Y}$*. Otherwise, let* T_p *be the union of all MACSs of elements in* p*, and let* $I_p = Pa(T_p) \setminus T_p$*. Then,* $\boldsymbol{X} \perp_{i_p} \boldsymbol{Y} | T_p \setminus (\boldsymbol{X} \cup \boldsymbol{Y})$*.*

I conclude this section by noting that just as was the case with conditional independence, identifiable dormant independence among subsets does not entail dormant independence on sets. For example, in the graph shown in Fig. 7.3 (b), $X \perp_z Y$, $X \perp_k L$, but $X \not\perp_* \{Y, L\}$.

Having given a complete solution to the problem of determining if arbitrary sets can be d*-separated, I show in the next section how to use dormant independence to test aspects of the causal diagram.

7.5 Testing Causal Structure

To illustrate the usefulness of identifiable dormant independencies for induction and testing of causal structures, I consider the problem of detecting if certain edges in a particular causal graph are extraneous. I call graphs where every edge is either correct or extraneous valid.

Definition 22 (valid graph) *A causal graph* G *is valid for a model* M *if every edge in the graph induced by* M *is present in* G*.*

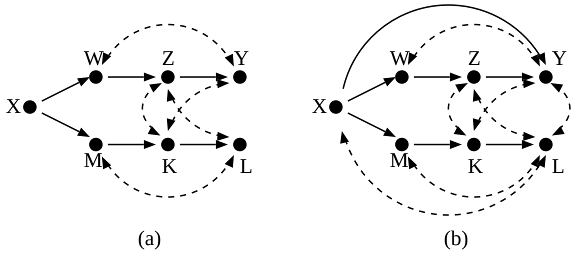


Figure 7.5: (a) The true causal graph. (b) A possible valid graph for the same domain.

It is possible to rule out out the presence of certain extraneous edges using conditional independence tests. In order to do so, an additional property of *stability* [PV91], [Pea00], or *faithfulness* [SGS93] is assumed. In faithful models, lack of d-separation implies dependence. In other words, $\mathbf{X} \perp \mathbf{Y}|\mathbf{Z}$ iff $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$. This property allows us to reach graphical conclusions from probabilistic premises. For instance, the presence of a conditioning set $\mathbf{Z}$ such that $X \perp\!\!\!\perp Y|\mathbf{Z}$ implies X and Y cannot share an edge. Systematic use of conditional independence tests to rule out adjacencies in this way is an important part of causal inference algorithms such as **IC** [VP90], [Pea00] and **FCI** [SGS93].

The advantage of dormant independencies is their ability to rule out edges even if all conditional independence tests fail. For instance, it is possible to rule out the edge from X to Y in Fig. 7.5 (b) as extraneous since $X \perp_z Y$, though no conditional independence test can succeed in doing the same, since there is an inducing path from X to Y.

However, in order to reach graphical conclusions from dormant independencies, I need to extend the faithfulness property to hold in interventional settings.

Definition 23 (experimental faithfulness) *A model M is experimentally faithful, or P_*-faithful if every submodel $M_{\boldsymbol{x}}$ of M is faithful (that is d-connectedness in $G_{\overline{\boldsymbol{x}}}$ implies dependence).*

Experimental faithfulness states that no "numerically coincidental independencies" are introduced by interventions. I use dormant independence tests to rule out extraneous edges in valid graphs of experimentally faithful models. To test if an edge between X and Y is extraneous, I must find sets $\mathbf{Z}, \mathbf{W}$ such that $X \perp\!\!\!\perp_{\mathbf{w}} Y|\mathbf{Z}$. A naive brute-force approach to this problem is intractable since I must try all subsets $\mathbf{Z}, \mathbf{W}$. However, if I assume the edge I am testing is absent in the graph, I can use the **Find-MACS** algorithm to propose a dormant independence to test in polynomial time. Since this independence is guaranteed to be identifiable, the test can be performed on the observational distribution alone.

```
function Test-Edges($G, P(\mathbf{v})$)
INPUT: $G$, a valid graph of an experimentally faithful model $M$, $P(\mathbf{v})$, a corresponding probability distribution.
OUTPUT $G'$, a valid graph with some extraneous edges removed.

    • Let $\pi$ be a topological order of edges in $G$, where $(X, Y) \prec_\pi (W, Z)$ if $X, Y \in An(\{W, Z\})_G$. Let $G'$ equal $G$.
    • For every edge $(X, Y)$ in $\pi$, if we can find sets $\mathbf{Z}, \mathbf{W}$ using Theorem 31 such that $X \perp_{\mathbf{w}} Y | \mathbf{Z}$ in $G' \setminus (X, Y)$, and
      $X \perp\!\!\!\perp_{\mathbf{w}} Y | \mathbf{Z}$ in $P(\mathbf{v}), G'$, remove $(X, Y)$ from $G'$.
    • return $G'$.
```

Figure 7.6: An algorithm for testing edges in valid graphs.

There is an additional complication, namely that certain edges ancestral to X and Y may themselves be extraneous. This may result in a situation where $X \not\perp_* Y$ if the ancestral extraneous edges are present, while a dormant independence can be established if they are removed. Fortunately, since I restrict myself to acyclic graphs, I can establish a topological order among edges based on ancestry, and test for extraneous edges using this order. The resulting algorithm is shown in Fig. 7.6

It is not difficult to establish that **Test-Edges** is sound.

Theorem 37 ***Test-Edges*** *terminates in polynomial time, and any edge it removes from G', valid for an experimentally faithful model M, is extraneous.*

To illustrate the operation of the algorithm, consider the valid graph G' in Fig. 7.5 (b). If the graph G in Fig. 7.5 (a) represents the true causal model, **Test-Edges** will be able to remove the edges (X, Y) and (X, L), but not the edge (L, Y). In the case of (X, Y), $X \perp_z Y$ in $G' \setminus (X, Y)$ and the corresponding dormant independence holds since the true model induces G. Similarly, for (X, L), $X \perp_k L$ in $G' \setminus (X, L)$ and the corresponding dormant independence holds. On the other hand, even though (Y, L) is an extraneous edge, **Test-Edges** cannot remove it, since the algorithm cannot establish dormant independence between Y and L, even though $P(y, l|do(z, k))$ is identifiable in the true model. The intuition here is that this identification relies on the absence of the very edge we are trying to test (since $P(y, l|do(z, k))$ is not identifiable in G').

Similarly, if the graph G shown in Fig. 7.3 (a) is the true causal graph, and the valid graph contains an extra edge from X to Y, **Test-Edges** will be able to remove this edge since $X \perp_z Y | W, K, L, N$ in G, and $P(\mathbf{v}), G' \vdash_{id} P_z(\mathbf{v} \setminus z)$, where G' is G plus any edge from X to Y.

CHAPTER 8

Conclusions

In this thesis, I have considered the problem of evaluating a variety of causal queries (causal effects, counterfactuals and path-specific effects) from available information, represented as observational or interventional distributions, and causal assumptions, represented in the form of a graph. I have presented complete algorithms for all identification problems I considered, and used these algorithms to derive graphical characterizations of identifiable and non-identifiable queries.

Furthermore, I considered the notion of dormant independence, namely conditional independence in interventional distributions. I showed how certain algebraic constraints induced on the observable distribution by the causal graph arise due to identifiable dormant independencies. I have provided a graphical notion of d*-separation which mirrors identifiable dormant independence, and given a complete algorithm which determines if two disjoint sets of variables can be d*-separated. Finally, I have used dormant independence to construct another algorithm which tests for the presence of extraneous arcs in a causal graph.

APPENDIX A

Proofs for Chapter 4 (Causal Effects)

Theorem 2 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y|do(x))$ in G shown in Fig. 3.1 (a).*
Proof: I construct two causal models M^1 and M^2 such that $P^1(X,Y) = P^2(X,Y)$, and $P^1_x(Y) \neq P^2_x(Y)$. The two models agree on the following: all 3 variables are boolean, U is a fair coin, and $f_X(u) = u$. Let $\oplus$ denote the exclusive or (XOR) function. Then the value of Y is determined by the function $u \oplus x$ in M^1, while Y is set to 0 in M^2. Then $P^1(Y = 0) = P^2(Y = 0) = 1$, $P^1(X = 0) = P^2(X = 0) = 0.5$. Therefore, $P^1(X,Y) = P^2(X,Y)$, while $P^2_x(Y = 0) = 1 \neq P^1_x(Y = 0) = 0.5$. Note that while P is non-positive, it is straightforward to modify the proof for the positive case by letting f_Y functions in both models return 1 half the time, and the values outlined above half the time. □

Theorem 3 *Let G be a Y-rooted C-tree. Let $\boldsymbol{X}$ be any subset of observable nodes in G which does not contain Y. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(y|do(\boldsymbol{x}))$.*
Proof: I generalize the proof for the bow arc graph. I can assume without loss of generality that each unobservable U in G has exactly two observable children. I construct two models with binary nodes. In the first model, the value of all observable nodes is set to the bit parity (sum modulo 2) of the parent values. In the second model, the same is true for all nodes except Y, with the latter being set to 0 explicitly. All $\mathbf{U}$ nodes in both models are fair coins. Since G is a tree, and since every $U \in \mathbf{U}$ has exactly two children in G, every $U \in \mathbf{U}$ has exactly two distinct downward paths to Y in G. It's then easy to establish that Y counts the bit parity of every node in $\mathbf{U}$ twice in the first model. But this implies $P^1(Y = 1) = 0$.

Because bidirected arcs form a spanning tree over observable nodes in G, for any set of nodes $\mathbf{X}$ such that $Y \not\in \mathbf{X}$, there exists $U \in \mathbf{U}$ with one child in $An(\mathbf{X})_G$ and one child in $G \setminus An(\mathbf{X})_G$. Thus $P^1_{\mathbf{x}}(Y = 1) > 0$, but $P^2_{\mathbf{x}}(Y = 1) = 0$. It is straightforward to generalize this proof for the positive $P(\mathbf{v})$ in the same way as in Theorem 2. □

Theorem 4 *$P(\boldsymbol{v}), G \not\vdash_{id} P(y|do(pa(y)))$ if and only if there exists a subgraph of G which is a Y-rooted C-tree.*
Proof: From [Tia02], I know that whenever there is no subgraph G' of G, such that all nodes in G' are ancestors of Y, and G' is a C-component, $P_{pa(Y)}(Y)$ is

identifiable. From Theorem 3, I know that if there is a Y-rooted C-tree containing a non-empty subset S of parents of Y, then $P_s(Y)$ is not identifiable. But it is always possible to extend the counterexamples which prove non-identification of $P_s(Y)$ with additional variables which are independent. $\square$

Theorem 5 *Let F, F' be subgraphs of G which form a hedge for $P(\boldsymbol{y}|do(\boldsymbol{x}))$. Then $P(\boldsymbol{v}), G \not\vdash_{id} P(\boldsymbol{y}|do(\boldsymbol{x}))$.*

Proof: I first show $P_{\mathbf{x}}(\mathbf{r})$ is not identifiable in F. As before, I assume each U has two observable children. I construct two models with binary nodes. In M^1 every variable in F is equal to the bit parity of its parents. In M^2 the same is true, except all nodes in F' disregard the parent values in $F \setminus F'$. All $\mathbf{U}$ are fair coins in both models.

As was the case with C-trees, for any C-forest F, every $U \in \mathbf{U} \cap F$ has exactly two downward paths to $\mathbf{R}$. It is now easy to establish that in M^1, $\mathbf{R}$ counts the bit parity of every node in $\mathbf{U}^1$ twice, while in M^2, $\mathbf{R}$ counts the bit parity of every node in $\mathbf{U}^2 \cap F'$ twice. Thus, in both models with no interventions, the bit parity of $\mathbf{R}$ is even.

Next, fix two distinct instantiations of $\mathbf{U}$ that differ by values of $\mathbf{U}^*$. Consider the topmost node $W \in F$ with an odd number of parents in $\mathbf{U}^*$ (which exists because bidirected edges in F form a spanning tree). Then flipping the values of $\mathbf{U}^*$ once will flip the value W once. Thus the function from $\mathbf{U}$ to $\mathbf{V}$ induced by a C-forest F in M^1 and M^2 is one to one.

The above results, coupled with the fact that in a C-forest, $|\mathbf{U}| + 1 = |\mathbf{V}|$ implies that any assignment where $\sum \mathbf{r} \pmod 2 = 0$ is equally likely, and all other node assignments are impossible in both F and F'. Since the two models agree on all functions and distributions in $F \setminus F'$, $\sum_{f'} P^1 = \sum_{f'} P^2$. It follows that the observational distributions are the same in both models.

As before, I can find $U \in \mathbf{U}$ with one child in $An(\mathbf{X})_F$, and one child in $F \setminus An(\mathbf{X})_F$, which implies the probability of odd bit parity of $\mathbf{R}$ is 0.5 in M^1, and 0 in M^2.

Next, I note that the construction so far results in a non-positive distribution P. To rid this proof of non-positivity, I "soften" the two models with new unobservable binary U_R for every $R \in \mathbf{R}$ which assumes value 1 with very small probability p. Whenever U_R is 1, the node R flips its value, otherwise it keeps the value as defined above. Note that $P(\mathbf{v})$ will remain the same in both models because the augmentation is the same, and the previous unsoftened models agreed on $P(\mathbf{v})$. It's easy to see that the bit parity of R in both models will be odd only when an odd number of U_R assume values of 1. Because p is arbitrarily small, the probability of an odd parity is far smaller than the probability of even parity. Now consider what happens after $do(\mathbf{x})$. In M^2, the probability of odd bit parity stays the same. In M^1 before the addition of U_R, the probability was 0.5. But it's easy to see that U_R nodes change the bit parity of $\mathbf{R}$ in a completely symmetric way, so the probability of even parity remains 0.5.

This implies $P_{\mathbf{x}}(\mathbf{r})$ is not identifiable. Finally, to see that $P_{\mathbf{x}}(\mathbf{y})$ is not identifiable, augment the counterexample by nodes in $\mathbf{I} = An(\mathbf{Y}) \cap De(\mathbf{R})$. Without loss of generality, assume every node in $\mathbf{I}$ has at most one child. Let each node I in $\mathbf{I}$ be equal to the bit parity of its parents. Moreover, each I has an exogenous parent U_I independent of the rest of $\mathbf{U}$ which, with small probability p causes it to flip it's value. Then the bit parity of $\mathbf{Y}$ is even if and only if an odd number of $\mathbf{U_I}$ turn on. Moreover, it's easy to see $P(\mathbf{I}|\mathbf{R})$ is positive by construction. I can now repeat the previous argument. □

Next, I provide the proof of soundness of **ID** and **IDC** using do-calculus. This both simplifies the proofs and allows us to infer do-calculus is complete from completeness of these algorithms. I will invoke do-calculus rules by just using their number, for instance "by rule 2." First, I prove that a joint distribution in a causal model can be represented as a product of interventional distributions corresponding to the set of c-component in the graph induced by the model.

Lemma 13 (c-component factorization) *Let M be a causal model with graph G. Let $\boldsymbol{y}, \boldsymbol{x}$ be value assignments. Let $C(G \setminus \boldsymbol{X}) = \{S_1, ..., S_k\}$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}) = \sum_{\boldsymbol{v} \setminus (\boldsymbol{y} \cup \boldsymbol{x})} \prod_i P_{\boldsymbol{v} \setminus s_i}(s_i)$.*

Proof: A proof of this was derived by [Tia02]. Nevertheless, I reprove this result using do-calculus to help with the subsequent completeness results. Assume $\mathbf{X} = \emptyset$, $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$, $C(G) = \{S_1, ..., S_k\}$, and let $A_i = An(S_i)_G \setminus S_i$. Then

$$\prod_i P_{\mathbf{v} \setminus s_i}(s_i) = \prod_i P_{a_i}(s_i) = \prod_i \prod_{V_j \in S_i} P_{a_i}(v_j | v_\pi^{(j-1)} \setminus a_i)$$

$$= \prod_i \prod_{V_j \in S_i} P(v_j | v_\pi^{(j-1)}) = \prod_i P(v_i | v_\pi^{(i-1)}) = P(\mathbf{v})$$

The first identity is by rule 3, the second is by chain rule of probability. To prove the third identity, I consider two cases. If $A \in A_i \setminus V_\pi^{(j-1)}$, I can eliminate the intervention on A from the expression $P_{a_i}(v_j | v_\pi^{(j-1)})$ by rule 3, since $(V_j \perp A | V_\pi^{(j-1)})_{G_{\overline{a_i}}}$.

If $A \in A_i \cap V_\pi^{(j-1)}$, consider any back-door path from A_i to V_j. Any such path with a node not in $V_\pi^{(j-1)}$ will be d-separated because, due to recursiveness, it must contain a blocked collider. Further, this path must contain bidirected arcs only, since all nodes on this path are conditioned or fixed. Because $A_i \cap S_i = \emptyset$, all such paths are d-separated. The identity now follows from rule 2.

The last two identities are just grouping of terms, and application of chain rule. Having proven that c-component factorization holds for $P(\mathbf{v})$, I want to extend the result to $P_{\mathbf{x}}(\mathbf{y})$. First, let's consider $P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x})$. This is just the distribution of the submodel $M_{\mathbf{x}}$. But $M_{\mathbf{x}}$ is just an ordinary causal model inducing $G \setminus \mathbf{X}$, so I can apply the same reasoning to obtain $P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x}) = \prod_i P_{\mathbf{v}\setminus s_i}(s_i)$, where $C(G \setminus \mathbf{X}) = \{S_1, ..., S_k\}$. As a last step, it's easy to verify that $P_{\mathbf{x}}(\mathbf{y}) = \sum_{\mathbf{v}\setminus(\mathbf{x}\cup\mathbf{y})} P_{\mathbf{x}}(\mathbf{v} \setminus \mathbf{x})$. □

Lemma 14 *Let $\boldsymbol{X}' = \boldsymbol{X} \cap An(\boldsymbol{Y})_G$. Then $P_{\boldsymbol{x}}(\boldsymbol{y})$ obtained from P in G is equal to $P'_{\boldsymbol{x}'}(\boldsymbol{y})$ obtained from $P' = P(An(\boldsymbol{Y}))$ in $An(\boldsymbol{Y})_G$.*

Proof: Let $\mathbf{W} = \mathbf{V} \setminus An(\mathbf{Y})_G$. Then the submodel $M_{\mathbf{w}}$ induces the graph $G \setminus \mathbf{W} = An(\mathbf{Y})_G$, and its distribution is $P' = P_{\mathbf{w}}(An(\mathbf{Y})) = P(An(\mathbf{Y}))$ by rule 3. Now $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{x}'}(\mathbf{y}) = P_{\mathbf{x}',\mathbf{w}}(\mathbf{y}) = P'_{\mathbf{x}'}(\mathbf{y})$ by rule 3. □

Lemma 15 *Let $\boldsymbol{W} = (\boldsymbol{V} \setminus \boldsymbol{X}) \setminus An(\boldsymbol{Y})_{G_{\overline{\boldsymbol{x}}}}$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}) = P_{\boldsymbol{x},\boldsymbol{w}}(\boldsymbol{y})$, where $\boldsymbol{w}$ are arbitrary values of $\boldsymbol{W}$.*

Proof: Note that by assumption, $\mathbf{Y} \perp \mathbf{W} | \mathbf{X}$ in $G_{\overline{\mathbf{X}},\overline{\mathbf{W}}}$. The conclusion follows by rule 3. □

Lemma 16 *When the conditions of line 6 are satisfied, $P_{\boldsymbol{x}}(\boldsymbol{y}) = \sum_{s\setminus \boldsymbol{y}} \prod_{V_i \in S} P(v_i | v_\pi^{(i-1)})$.*

Proof: If line 6 preconditions are met, then G local to that recursive call is partitioned into S and $\mathbf{X}$, and there are no bidirected arcs from $\mathbf{X}$ to S. The conclusion now follows from the proof of Lemma 13. □

Lemma 17 *Whenever the conditions of the last recursive call of **ID** are satisfied, $P_{\boldsymbol{x}}$ obtained from P in the graph G is equal to $P'_{\boldsymbol{x}\cap S'}$ obtained from $P' = \prod_{V_i \in S'} P(V_i | V_\pi^{(i-1)} \cap S', v_\pi^{(i-1)} \setminus S')$ in the graph S'.*

Proof: It is easy to see that when the last recursive call executes, $\mathbf{X}$ and S partition G, and $\mathbf{X} \subset An(S)_G$. This implies that the submodel $M_{\mathbf{x}\setminus S'}$ induces the graph $G \setminus (\mathbf{X} \setminus S') = S'$. The distribution $P_{\mathbf{x}\setminus S'}$ of $M_{\mathbf{x}\setminus S'}$ is equal to P' by the proof of Lemma 13. It now follows that $P_{\mathbf{x}} = P_{\mathbf{x}\cap S', \mathbf{x}\setminus S'} = P'_{\mathbf{x}\cap S'}$. □

Theorem 38 (soundness) *Whenever **ID** returns an expression for $P_{\boldsymbol{x}}(\boldsymbol{y})$, it is correct.*

Proof: If $\mathbf{x} = \emptyset$, the desired effect can be obtained from P by marginalization, thus this base case is clearly correct. The soundness of all other lines except the failing line 5 has already been established. □

Having established soundness, I show that whenever **ID** fails, we can recover a hedge for an effect involving a subset of variables involved in the original effect expression $P(\mathbf{y}|do(\mathbf{x}))$. This in turn implies completeness.

Theorem 39 *Assume **ID** fails to identify $P_{\boldsymbol{x}}(\boldsymbol{y})$ (executes line 5). Then there exist $\boldsymbol{X}' \subseteq \boldsymbol{X}$, $\boldsymbol{Y}' \subseteq \boldsymbol{Y}$ such that the graph pair G, S returned by the fail condition of **ID** contain as edge subgraphs C-forests F, F' that form a hedge for $P_{\boldsymbol{x}'}(\boldsymbol{y}')$.*

Proof: Consider line 5, and G and $\mathbf{y}$ local to that recursive call. Let $\mathbf{R}$ be the root set of G. Since G is a single C-component, it is possible to remove a set of directed arrows from G while preserving the root set $\mathbf{R}$ such that the resulting graph F is an $\mathbf{R}$-rooted C-forest.

Moreover, since $F' = F \cap S$ is closed under descendants, and since only single directed arrows were removed from S to obtain F', F' is also a C-forest. $F' \cap \mathbf{X} = \emptyset$, and $F \cap \mathbf{X} \neq \emptyset$ by construction. $\mathbf{R} \subseteq An(\mathbf{Y})_{G_{\overline{\mathbf{x}}}}$ by lines 2 and 3 of the algorithm. It's also clear that $\mathbf{y}, \mathbf{x}$ local to the recursive call in question are subsets of the original input. □

Theorem 6 ***ID** is complete.*

Proof: By the previous theorem, if **ID** fails, then $P_{\mathbf{x}'}(\mathbf{y}')$ is not identifiable in a subgraph $H = G_{An(\mathbf{Y}) \cap De(F)}$ of G. Moreover, $\mathbf{X} \cap H = \mathbf{X}'$, by construction of H. As such, it is easy to extend the counterexamples in Theorem 39 with variables independent of H, with the resulting models inducing G, and witnessing the unidentifiability of $P_{\mathbf{x}}(\mathbf{y})$. □

Next, I prove the results necessary to establish completeness of **IDC**.

Lemma 18 *If rule 2 of do-calculus applies to a set $\boldsymbol{Z}$ in G for $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$ then there are no d-connected paths to $\boldsymbol{Y}$ that pass through $\boldsymbol{Z}$ in neither $G_1 = G \setminus \boldsymbol{X}$ given $\boldsymbol{Z}$, $\boldsymbol{W}$ nor in $G_2 = G \setminus (\boldsymbol{X} \cup \boldsymbol{Z})$ given $\boldsymbol{W}$.*

Proof: Clearly, there are no d-connected paths through $\mathbf{Z}$ in G_2 given $\mathbf{W}$. Consider a d-connected path through $Z \in \mathbf{Z}$ to $\mathbf{Y}$ in G_1, given $\mathbf{Z}, \mathbf{W}$. Note that this path must either form a collider at Z or a collider which is an ancestor of Z. But this must mean there is a back-door path from $\mathbf{Z}$ to $\mathbf{Y}$, which is impossible, since rule 2 is applicable to $\mathbf{Z}$ in G for $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$. Contradiction. □

Theorem 8 *For any G and any conditional effect $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$ there exists a unique maximal set $\boldsymbol{Z} = \{Z \in \boldsymbol{W} | P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w}) = P_{\boldsymbol{x},z}(\boldsymbol{y}|\boldsymbol{w} \setminus \{z\})\}$ such that rule 2 applies to*

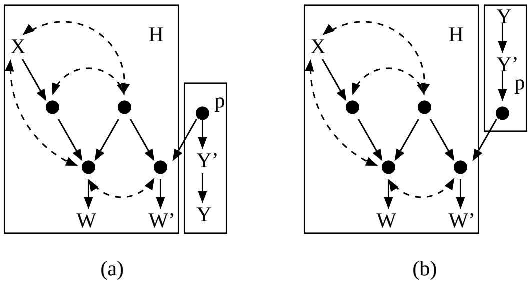


Figure A.1: Inductive cases for proving non-identifiability of $P_x(y|w, w')$.

$\boldsymbol{Z}$ in G for $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$. In other words, $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w}) = P_{\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{y}|\boldsymbol{w} \setminus \boldsymbol{z})$.
Proof: Fix two maximal sets $\mathbf{Z}_1, \mathbf{Z}_2 \subseteq \mathbf{W}$ such that rule 2 applies to $\mathbf{Z}_1, \mathbf{Z}_2$ in G for $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$. If $\mathbf{Z}_1 \neq \mathbf{Z}_2$, fix $Z \in \mathbf{Z}_1 \setminus \mathbf{Z}_2$. By Lemma 18, rule 2 applies for $\{Z\} \cup \mathbf{Z}_2$ in G for $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$, contradicting the original assumption.

Thus if I fix G and $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$, any set to which rule 2 applies must be a subset of the unique maximal set $\mathbf{Z}$. It follows that $\mathbf{Z} = \{Z \in \mathbf{W} | P_{\mathbf{x}}(\mathbf{y}|\mathbf{w}) = P_{\mathbf{x},z}(\mathbf{y}|\mathbf{w} \setminus \{z\})\}$. □

Lemma 19 *Let F, F' form a hedge for $P_{\boldsymbol{x}}(\boldsymbol{y})$. Then $F \subseteq F' \cup \boldsymbol{X}$.*

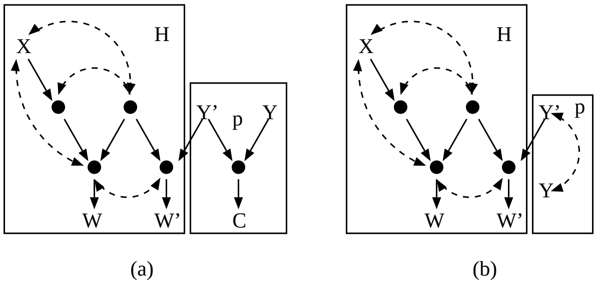
Proof: It has been shown that **ID** fails on $P_{\mathbf{x}}(\mathbf{y})$ in G and returns a hedge if and only if $P_{\mathbf{x}}(\mathbf{y})$ is not identifiable in G. In particular, edge subgraphs of the graphs G and S returned by line 5 of **ID** form the C-forests of the hedge in question. It is easy to check that a subset of $\mathbf{X}$ and S partition G. □

I rephrase the statement of Theorem 9 somewhat, to reduce "algebraic clutter."

Theorem 9 *Let $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$ be such that every $W \in \boldsymbol{W}$ has a back-door path to $\boldsymbol{Y}$ in $G \setminus \boldsymbol{X}$ given $\boldsymbol{W} \setminus \{W\}$. Then $P_{\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{w})$ is identifiable in G if and only if $P_{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{w})$ is identifiable in G.*
Proof: If $P_{\mathbf{x}}(\mathbf{y}, \mathbf{w})$ is identifiable in G, then we can certainly identify $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$ by marginalization and division. The difficult part is to prove that if $P_{\mathbf{x}}(\mathbf{y}, \mathbf{w})$ is not identifiable then neither is $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$.

Assume $P_{\mathbf{x}}(\mathbf{w})$ is identifiable. Then if $P_{\mathbf{x}}(\mathbf{y}|\mathbf{w})$ were identifiable, I would be able to compute $P_{\mathbf{x}}(\mathbf{y}, \mathbf{w})$ by the chain rule. Thus the conclusion follows.

Assume $P_{\mathbf{x}}(\mathbf{w})$ is not identifiable. I also know that every $W \in \mathbf{W}$ contains a back-door path to some $Y \in \mathbf{Y}$ in $G \setminus \mathbf{X}$ given $\mathbf{W} \setminus \{W\}$. Fix such W and Y, along with a subgraph p of G which forms the witnessing back-door path.



(a)

(b)

Figure A.2: Inductive cases for proving non-identifiability of $P_x(y|w, w')$.

Consider also the hedge F, F' which witnesses the non-identifiability of $P_{\mathbf{x}'}(\mathbf{w}')$, where $\mathbf{X}' \subseteq \mathbf{X}, \mathbf{W}' \subseteq \mathbf{W}$.

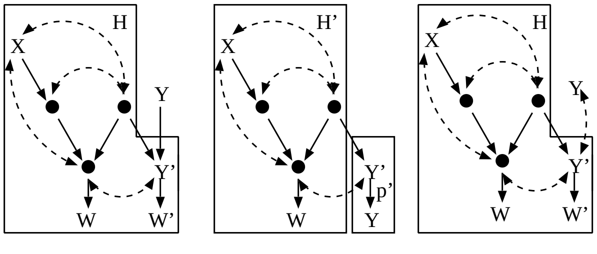
Let $H = G_{De(F)\cup An(\mathbf{W}')_{G_{\overline{\mathbf{x}'}}}}$. I will attempt to show that $P_{\mathbf{x}'}(Y|\mathbf{w})$ is not identifiable in $H \cup p$. Without loss of generality, I make the following three assumptions. First, I restrict my attention to $\mathbf{W}'' \subseteq \mathbf{W}$ that occurs in $H \cup p$. Second, I assume p is a path segment which starts at H and ends at Y, and does not intersect H. Third, I assume all observable nodes in H have at most one child.

Consider the models M^1, M^2 from the proof of Theorem 5 which induce H. I extend the models by adding to them binary variables in p. Each variable $X \in p$ is equal to the bit parity of its parents, if it has any. If not, X behaves as a fair coin. If $Y \in H$ has a parent $X \in p$, the value of X is added to the bit parity computation Y makes.

Call the resulting models M^1_*, M^2_*. Because M^1, M^2 agreed on $P(H)$, and variables and functions in p are the same in both models, $P^1_* = P^2_*$. I will assume $\mathbf{w}''$ assigns 0 to every variable in $\mathbf{W}''$. What remains to be shown is that $P^1_{*\mathbf{x}}(y|\mathbf{w}'') \neq P^2_{*\mathbf{x}}(y|\mathbf{w}'')$. I will prove this by induction on the path structure of p. I handle the inductive cases first. In all these cases, I fix a node Y' that is between Y and H on the path p, and prove that if $P_{\mathbf{x}'}(y'|\mathbf{w}'')$ is not identifiable, then neither is $P_{\mathbf{x}'}(y|\mathbf{w}'')$.

Assume neither Y nor Y' have descendants in $\mathbf{W}''$. If Y' is a parent of Y as in Fig. A.1 (a), then $P_{\mathbf{x}'}(y|\mathbf{w}'') = \sum_{y'} P(y|y')P_{\mathbf{x}'}(y'|\mathbf{w}'')$. If Y is a parent of Y', as in Fig. A.1 (b) then the next node in p must be a child of Y'. Therefore, $P_{\mathbf{x}'}(y|\mathbf{w}'') = \sum_{y'} P(y|y')P_{\mathbf{x}'}(y'|\mathbf{w}'')$. In either case, by construction $P(Y|Y')$ is a 2 by 2 identity matrix. This implies that the mapping from $P_{\mathbf{x}'}(y'|\mathbf{w}'')$ to $P_{\mathbf{x}'}(y|\mathbf{w}'')$ is one to one. If Y' and Y share a hidden common parent U as in Fig. A.2 (b), then the result follows by combining the previous two cases.

The next case is if Y and Y have a common child C which is either in $\mathbf{W}''$ or has a descendant in $\mathbf{W}''$, as in Fig. A.2 (a). Now $P_{\mathbf{x}'}(y|\mathbf{w}'') = \sum_{y'} P(y|y', c)P_{\mathbf{x}'}(y'|\mathbf{w}'')$.



(a) (b) (c)

Figure A.3: Base cases for proving non-identifiability of $P_x(y|w, w')$.

Because all nodes in $\mathbf{W}''$ were observed to be 0, $P(y|y', c)$ is again a 2 by 2 identity matrix.

Finally, I handle the base cases of the induction. In all such cases, Y is the first node not in H on the path p. Let Y' be the last node in H on the path p.

Assume Y is a parent of Y', as shown in Fig. A.3 (a). By Lemma 19, I can assume $Y \not\in An(F \setminus F')_H$. By construction, ($\sum \mathbf{W}'' = Y + 2 * \sum \mathbf{U}$) (mod 2) in M^1_*, and ($\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F')$) (mod 2) in M^2_*. If every variable in $\mathbf{W}''$ is observed to be 0, then $Y = (2 * \sum \mathbf{U})$ (mod 2) in M^1_*, and $Y = (2 * \sum(\mathbf{U} \cap F'))$ (mod 2) in M^2_*. If an intervention $do(\mathbf{x})$ is performed, ($\sum \mathbf{W}'' = Y + 2 * \sum(\mathbf{U} \cap F')$) (mod 2) in $M^2_{*\mathbf{x}}$, by construction. Thus if $\mathbf{W}''$ are all observed to be zero, $Y = 0$ with probability 1. Note that in $M^1_{\mathbf{x}}$ as constructed in the proof of Theorem 5, ($\sum \mathbf{w}'' = \mathbf{x} + \sum \mathbf{U}'$) (mod 2), where $\mathbf{U}' \subseteq \mathbf{U}$ consists of unobservable nodes with one child in $An(\mathbf{X})_F$ and one child in $F \setminus An(\mathbf{X})_F$.

Because $Y \not\in An(F \setminus F')_H$, I can conclude that if $\mathbf{W}''$ are observed to be 0, $Y = (\mathbf{x} + \sum \mathbf{U}')$ (mod 2) in $M^1_{*\mathbf{x}'}$. Thus, $Y = 0$ with probability 0.5. Therefore, $P^1_{*\mathbf{x}'}(y|\mathbf{w}'') \neq P^2_{*\mathbf{x}'}(y|\mathbf{w}'')$ in this case.

Assume Y is a child of Y'. Now consider a graph G' which is obtained from $H \cup p$ by removing the (unique) outgoing arrow from Y' in H. If $P_{\mathbf{x}'}(Y|\mathbf{w}'')$ is not identifiable in G', I am done. Assume $P_{\mathbf{x}'}(Y|\mathbf{w}'')$ is identifiable in G'. If $Y' \in F$, and $\mathbf{R}$ is the root set of F, then removing the Y'-outgoing directed arrow from F results in a new C-forest, with a root set $\mathbf{R} \cup \{Y'\}$. Because Y is a child of Y', the new C-forests form a hedge for $P_{\mathbf{x}'}(y, \mathbf{w}'')$. If $Y' \in H \setminus F$, then removing the Y'-outgoing directed arrow results in substituting Y for $W \in \mathbf{W}'' \cap De(Y')_H$. Thus in G', F, F' form a hedge for $P_{\mathbf{x}'}(y, \mathbf{w}'' \setminus \{w\})$. In either case, $P_{\mathbf{x}'}(y, \mathbf{w}'')$ is not identifiable in G'.

If $P_{\mathbf{x}'}(\mathbf{w}'')$ is identifiable in G', I am done. If not, consider a smaller hedge $H' \subset H$ witnessing this fact. Now consider the segment p' of p between Y and H'. I can repeat the inductive argument for H', p' and Y. See Fig. A.3 (b).

If $P_{\mathbf{x}'}(\mathbf{w}'')$ is identifiable in G', I am done. If not, consider a smaller hedge

$H' \subset H$ witnessing this fact. Now consider the segment p' of p between Y and H'. I can repeat the inductive argument for H', p' and Y. See Fig. A.3 (b). If Y and Y' have a hidden common parent, as is the case in Fig. A.3 (c), I can combine the first inductive case, and the first base case to prove the result.

I conclude the proof by introducing a slight change to rid us of non-positivity in the distributions P^1, P^2 in the counterexamples. Specifically, for every node I in $p \cup (De(\mathbf{R}) \cap An(\mathbf{Y}))$, add a new binary exogenous parent U_I which is independent of other nodes in $\mathbf{U}$, and has an arbitrarily small probability of assuming the value 1, and causing its child to flip its current value. I let P_{odd} be the probability an odd number of U_I nodes assume the value 1. Because $P(U_I = 1)$ is vanishingly small for every I, P_{odd} is much smaller than 0.5. It's easy to see that P is positive in counterexamples augmented in this way. In the base case when Y is a parent of Y', I modify my equations to account for the addition of U_I. Specifically, $(\sum \mathbf{W}'' = Y + 2 * \sum \mathbf{U} + \sum \mathbf{U_I}) \pmod 2$ in M^1_*, and $(\sum \mathbf{W}'' = Y + 2 * \sum (\mathbf{U} \cap F') + \sum \mathbf{U_I}) \pmod 2$ in M^2_*, where $U_{\mathbf{U}}$ is the set of nodes added. If every variable in $\mathbf{W}''$ is observed to be 0, then $Y = (2 * \sum \mathbf{U} + \sum \mathbf{U_I}) \pmod 2$ in M^1_*, and $Y = (2 * \sum (\mathbf{U} \cap F') + \sum \mathbf{U_I}) \pmod 2$ in M^2_*. So prior to the intervention, $P(Y = 1 | \mathbf{w}'') = P_{odd}$. But because $P^1_{\mathbf{x}'}(Y = 1 | \mathbf{w}'') = 0.5$, adding U_I nodes to the model does not change this probability. Because $P^2(Y = 1 | \mathbf{w}'') = P^2_{\mathbf{x}}(Y = 1 | \mathbf{w}'')$, the conclusion follows.

In the inductive cases above, I showed that $P_{\mathbf{x}}(Y' = Y | \mathbf{W}'') = 1$ in our counterexamples. It's easy to see that with the addition of U_I, $P_{\mathbf{x}}(Y' = Y | \mathbf{W}'') = P_{odd}$. This implies that if $P^1_{\mathbf{x}}(Y' | \mathbf{W}'') \neq P^2_{\mathbf{x}}(Y' | \mathbf{W}'')$, then $P^1_{\mathbf{x}}(Y | \mathbf{W}'') \neq P^2_{\mathbf{x}}(Y | \mathbf{W}'')$.

This completes the proof. □

APPENDIX B

Proofs for Chapter 5 (Counterfactuals)

Lemma 16 *If the preconditions of line 7 are met, $P(S) = P_{\boldsymbol{x}}(\boldsymbol{var}(S))$, where $\boldsymbol{x} = \bigcup \boldsymbol{sub}(S)$.*
Proof: Let $\mathbf{x} = \bigcup \mathbf{sub}(S)$. Since the preconditions are met, $\mathbf{x}$ does not contain conflicting assignments to the same variable, which means $do(\mathbf{x})$ is a sound action in the original causal model. Note that for any variable $Y_{\mathbf{w}}$ in S, any variable in $(Pa(S) \setminus S) \cap An(Y_{\mathbf{w}})_S$ is already in $\mathbf{w}$, while any variable in $(Pa(S) \setminus S) \setminus An(Y_{\mathbf{w}})_S$ can be added to the subscript of $Y_{\mathbf{w}}$ without changing the variable. Since $Y \cap \mathbf{X} = \emptyset$ by assumption, $Y_{\mathbf{w}} = Y_{\mathbf{x}}$. Since $Y_{\mathbf{w}}$ was arbitrary, The result follows. □

For convenience, I show the soundness of **ID*** and **IDC*** asserted in Theorem 12 separately.

Theorem 12 a *If **ID*** succeeds, the expression it returns is equal to $P(\gamma)$ in a given causal graph.*
Proof: The proof outline in section 3 is sufficient for everything except the base cases. In particular, line 6 follows by Lemma 13. For soundness, we only need to handle the positive base case, which follows from Lemma 16. □

The soundness of **IDC*** is also fairly straightforward.

Theorem 12 b *If **IDC*** does not output **FAIL**, the expression it returns is equal to $P(\gamma|\delta)$ in a given causal graph, if that expression is defined, and **UNDEFINED** otherwise.*
Proof: Theorem 8 shows how an operation similar to line 4 is sound by rule 2 of do-calculus [Pea95] when applied in a causal diagram. But I know that the counterfactual graph is just a causal diagram for a model where some nodes share functions, so the same reasoning applies. The rest is straightforward. □

To show completeness of **ID*** and **IDC***, I first prove a utility lemma which will make it easier to construct counterexamples which agree on P_* but disagree on a given counterfactual query.

Lemma 20 *Let G be a causal graph partitioned into a set $\{S_1, ..., S_k\}$ of C-components. Then two models M_1, M_2 which induce G agree on P_* if and only if their submodels $M^1_{\boldsymbol{v} \setminus s_i}$,*

$M^2_{\boldsymbol{v}\setminus s_i}$ agree on P_* for every C-component S_i, and value assignment $\boldsymbol{v} \setminus s_i$.

Proof: This follows from C-component factorization: $P(\mathbf{v}) = \prod_i P_{\mathbf{v}\setminus s_i}(s_i)$. This implies that for every $do(\mathbf{x})$, $P_{\mathbf{x}}(\mathbf{v})$ can be expressed as a product of terms $P_{\mathbf{v}\setminus(s_i\setminus\mathbf{x})}(s_i \setminus \mathbf{x})$, which implies the result. □

The next result generalizes Lemma 4 to a wider set of counterfactual graphs which result from non-identifiable queries.

Lemma 5 *Assume G is such that X is a parent of Y and Z, and Y and Z are connected by a bidirected path with observable nodes $W^1, ..., W^k$ on the path. Then $P_*, G \not\vdash_{id} P(y_x, w^1, ..., w^k, z_{x'}), P(y_x, w^1, ..., w^k, z)$ for any value assignments $y, w^1, ..., w^k, z$.*

Proof: I construct two models with graph G as follows. In both models, all variables are binary, and $P(\mathbf{u})$ is uniform. In M^1, each variable is set to the bit parity of its parents. In M^2, the same is true except Y and Z ignore the values of X. To prove that the two models agree on P_*, I use Lemma 20. Clearly the two models agree on $P(X)$. To show that the models also agree on $P_x(\mathbf{V} \setminus \mathbf{X})$ for all values of x, note that in M_2 each value assignment over $\mathbf{V} \setminus \mathbf{X}$ with even bit parity is equally likely, while no assignment with odd bit parity is possible. But the same is true in M^1 because any value of x contributes to the bit parity of $\mathbf{V} \setminus \mathbf{X}$ exactly twice. The agreement of M^1_x, M^2_x on P_* follows by the graph structure of G.

To see that the result is true, I note firstly that $P(\Sigma_i W^i + Y_x + Z_{x'} \pmod 2 = 1) = P(\Sigma_i W^i + Y_x + Z \pmod 2 = 1) = 0$ in M^2, while the same probabilities are positive in M^1, and secondly that in both models distributions $P(y_x, w^1, ..., w^k, z_{x'})$ and $P(y_x, w^1, .., w^k, z)$ are uniform. Note that the proof is easy to generalize for positive P_* by adding a small probability for Y to flip its normal value. □

To obtain a full characterization of non-identifiable counterfactual graphs, I augment the difficult graphs I obtained from the previous two results using certain graph transformation rules which preserve non-identifiability. These rules are given in the following two lemmas.

Lemma 6 *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let $\{y^1_{\boldsymbol{x}^1}, ..., y^n_{\boldsymbol{x}^m}\}$ be a subset of counterfactual events in γ. Let G' be a graph obtained from G by adding a new child W of $Y^1, ..., Y^n$. Let $\gamma' = (\gamma \setminus \{y^1_{\boldsymbol{x}^1}, ..., y^n_{\boldsymbol{x}^m}\}) \cup \{w_{\boldsymbol{x}^1}, ..., w_{\boldsymbol{x}^m}\}$, where w is an arbitrary value of W. Then $P_*, G' \not\vdash_{id} P(\gamma')$.*

Proof: Let M^1, M^2 witness $P_*, G \not\vdash_{id} P(\gamma)$. I will extend these models to witness $P_*, G' \not\vdash_{id} P(\gamma')$. Since the function of a newly added W will be shared, and M^1, M^2 agree on P_* in G, the extensions will agree on P_* by Lemma 20. I have two cases.

Assume there is a variable Y^i such that $y^i_{\mathbf{x}^j}, y^i_{\mathbf{x}^k}$ are in γ. By Lemma 4, $P_*, G \not\vdash_{id} P(y^i_{\mathbf{x}^j}, y^i_{\mathbf{x}^k})$. Then let W be a child of just Y^i, and assume $|W| = |Y^i| =$

c. Let W be set to the value of Y^i with probability $1-\epsilon$, and otherwise it is set to a uniformly chosen random value of Y^i among the other $c-1$ values. Since ϵ is arbitrarily small, and since $W_{\mathbf{x}^j}$ and $W_{\mathbf{x}^k}$ pay attention to the same U variable, it is possible to set ϵ in such a way that if $P^1(Y^i_{\mathbf{x}^j}, Y^i_{\mathbf{x}^k}) \neq P^2(Y^i_{\mathbf{x}^j}, Y^i_{\mathbf{x}^k})$, however minutely, then $P^1(W_{\mathbf{x}^j}, W_{\mathbf{x}^k}) \neq P^2(W_{\mathbf{x}^j}, W_{\mathbf{x}^k})$.

Otherwise, let $|W| = \prod_i |Y^i|$, and let $P(W|Y^1, ..., Y^n)$ be an invertible stochastic matrix. The result follows. $\square$

Lemma 7 *Assume $P_*, G \not\vdash_{id} P(\gamma)$. Let G' be obtained from G by merging some two nodes X, Y into a new node Z where Z inherits all the parents and children of X, Y, subject to the following restrictions:*

- *The merge does not create cycles.*
- *If $(\exists w_{\boldsymbol{s}} \in \gamma)$ where $x \in \boldsymbol{s}$, $y \notin \boldsymbol{s}$, and $X \in An(W)_G$, then $Y \notin An(W)_G$.*
- *If $(\exists y_{\boldsymbol{s}} \in \gamma)$ where $x \in \boldsymbol{s}$, then $An(X)_G = \emptyset$.*
- *If $(Y_{\boldsymbol{w}}, X_{\boldsymbol{s}} \in \gamma)$, then $\boldsymbol{w}$ and $\boldsymbol{s}$ agree on all variable settings.*

Assume $|X| \times |Y| = |Z|$ and there's some isomorphism f assigning value pairs x, y to a value $f(x, y) = z$. Let γ' be obtained from γ as follows. For any $w_{\boldsymbol{s}} \in \gamma$:

- *If $W \notin \{X, Y\}$, and values x, y occur in $\boldsymbol{s}$, replace them by $f(x, y)$.*
- *If $W \notin \{X, Y\}$, and the value of one of X, Y occur in $\boldsymbol{s}$, replace it by some z consistent with the value of X or Y.*
- *If X, Y do not occur in γ, leave γ as is.*
- *If $W = Y$ and $x \in \boldsymbol{s}$, replace $w_{\boldsymbol{s}}$ by $f(x, y)_{\boldsymbol{s}\setminus\{x\}}$.*
- *otherwise, replace every variable pair of the form $Y_{\boldsymbol{r}} = y, X_{\boldsymbol{s}} = x$ by $Z_{\boldsymbol{r},\boldsymbol{s}} = f(x, y)$.*

Then $P_*, G' \not\vdash_{id} P(\gamma')$.

Proof: Let Z be the Cartesian product of X, Y, and fix f. I want to show that the proof of non-identification of $P(\gamma)$ in G carries over to $P(\gamma')$ in G'.

I have four types of modifications to variables in γ. The first clearly results in the same counterfactual variable. For the second, due to the restrictions I imposed, $w_{\mathbf{z}} = w_{\mathbf{z},y,x}$, which means I can apply the first modification.

For the third, I have $P(\gamma) = P(\delta, y_{x,\mathbf{z}})$. By my restrictions, and rule 2 of do-calculus [Pea95], this is equal to $P(\delta, y_{\mathbf{z}}|x_{\mathbf{z}})$. Since this is not identifiable, then neither is $P(\delta, y_{\mathbf{z}}, x_{\mathbf{z}})$. Now it's clear that this modification is equivalent to the fourth.

The fourth modification is simply a merge of events consistent with a single causal world into a conjunctive event, which does not change the overall expression. □

I am now ready to show the main completeness results for counterfactual identification algorithms. Again, I prove this results separately for **ID*** and **IDC*** for convenience.

Theorem 13 a ***ID**** *is complete.*
Proof: I want to show that if line 8 fails, the original $P(\gamma)$ cannot be identified. There are two broad cases to consider. If G_γ contains the w-graph, the result follows by Lemmas 4 and 6. If not, I argue as follows.

Fix some X which witnesses the precondition on line 8. I can assume X is a parent of some nodes in S. Assume no other node in $\mathbf{sub}(S)$ affects S (effectively I delete all edges from parents of S to S except from X). Because the w-graph is not a part of G_γ, this has no ramifications on edges in S. Further, I assume X has two values in S.

If $X \not\in S$, fix $Y, W \in S \cap Ch(X)$. Assume S has no directed edges at all. Then $P_*, G \not\vdash_{id} P(S)$ by Lemma 5. The result now follows by Lemma 6, and by construction of G_γ, which implies all nodes in S have some descendant in γ.

If S has directed edges, I want to show $P_*, G \not\vdash_{id} P(R(S))$, where $R(S)$ is the subset of S with no children in S. I can recover this from the previous case as follows. Assume S has no edges as before. For a node $Y \in S$, fix a set of childless nodes $\mathbf{X} \in S$ which are to be their parents. Add a virtual node Y' which is a child of all nodes in $\mathbf{X}$. Then $P_*, G \not\vdash_{id} P((S \setminus \mathbf{X}) \cup Y')$ by Lemma 6. Then $P_*, G \not\vdash_{id} P(R(S'))$, where S' is obtained from S by adding edges from $\mathbf{X}$ to Y by Lemma 7, which applies because no w-graph exists in G_γ. I can apply this step inductively to obtain the desired forest (all nodes have at most one child) S while making sure $P_*, G \not\vdash_{id} P(R(S))$.

If S is not a forest, I can simply disregard extra edges so effectively it is a forest. Since the w-graph is not in G_γ this does not affect edges from X to S.

If $X \in S$, fix $Y \in S \cap Ch(X)$. If S has no directed edges at all, replace X by a new virtual node Y, and make X be the parent of Y. By Lemma 5, $P_*, G \not\vdash_{id} P((S \setminus x) \cup y_x)$. I now repeat the same steps as before, to obtain that $P_*, G \not\vdash_{id} P((R(S) \setminus x) \cup y_x)$ for general S. Now I use Lemma 7 to obtain $P_*, G \not\vdash_{id} P(R(S))$. Having shown $P_*, G \not\vdash_{id} P(R(S))$, I conclude the result by inductively applying Lemma 6. □

Theorem 13 b ***IDC**** *is complete.*
Proof: The difficult step is to show that after line 5 is reached, if $P_*, G \not\vdash_{id} P(\gamma, \delta)$ then $P_*, G \not\vdash_{id} P(\gamma|\delta)$. If $P_*, G \vdash_{id} P(\delta)$, this is obvious. Assume $P_*, G \not\vdash_{id} P(\delta)$. Fix the S which witnesses that for $\delta' \subseteq \delta$, $P_*, G \not\vdash_{id} P(\delta')$. Fix some Y such that a back-door, i.e. starting with an incoming arrow, path exists from δ' to Y in $G_{\gamma,\delta}$. I want to show that $P_*, G \not\vdash_{id} P(Y|\delta')$. Let $G' = G_{An(\delta') \cap De(S)}$.

Assume Y is a parent of a node $D \in \delta'$, and $D \in G'$. Augment the counterexample models which induce counterfactual graph G' with an additional binary node for Y, and let the value of D be set as the old value plus Y modulo $|D|$. Let Y attain value 1 with vanishing probability ϵ. That the new models agree on P_* is easy to establish. To see that $P_*, G \not\vdash_{id} P(\delta')$ in the new model, note that $P(\delta')$ in the new model is equal to $P(\delta' \setminus D, D = d) * (1 - \epsilon) + P(\delta' \setminus D, D = (d - 1) \pmod{|D|}) * \epsilon$. Because ϵ is arbitrarily small, this implies the result. To show that $P_*, G \not\vdash_{id} P(Y = 1|\delta')$, I must show that the models disagree on $P(\delta'|Y = 1)/P(\delta')$. But to do this, I must simply find two consecutive values of D, $d, d+1 \pmod{|D|}$ such that $P(\delta' \setminus D, d+1 \pmod{|D|})/P(\delta' \setminus D, d)$ is different in the two models. But this follows from non-identification of $P(\delta')$.

If Y is not a parent of $D \in G'$, then either it is further along on the back-door path or it's a child of some node in G'. In case 1, I must construct the distributions along the back-door path in such a way that if $P_*, G \not\vdash_{id} P(Y'|\delta')$ then $P_*, G \not\vdash_{id} P(Y|\delta')$, where Y' is a node preceding Y on the path. The proof follows closely the one in Theorem 9. In case 2, I duplicate the nodes in G' which lead from Y to δ', and note that I can show non-identification in the resulting graph using reasoning in case 1. I obtain the result by applying Lemma 7. □

APPENDIX C

Proofs for Chapter 6 (Path-specific Effects)

Lemma 8 *$P(Y_{\boldsymbol{x},Z^1,...,Z^k}) = \sum_{z^1,...,z^k} P(Y_{\boldsymbol{x},z^1,...,z^k}, Z^1 = z^1, ..., Z^k = z^k)$, where $Z^i = z^i$ stands for the event "nested counterfactual variable Z^i assumes values z^i."*

Proof: By definition, $P(Y_{\mathbf{x},Z^1,...,Z^k} = y) = \sum_{\{\mathbf{u}|Y_{\mathbf{x},Z^1(\mathbf{u}),...,Z^k(\mathbf{u})}(\mathbf{u})=y\}} P(\mathbf{u})$, and $P(Y_{\mathbf{x},z^1,...,z^k}, Z^1 = z^1, ..., Z^k = z^k) = \sum_{\{\mathbf{u}|Y_{\mathbf{x}},z^1,...,z^k(\mathbf{u})\wedge Z^1(\mathbf{u})=z^1\wedge...\wedge Z^k(\mathbf{u})=z^k\}} P(\mathbf{u})$.

But $Y_{\mathbf{x},Z^1(\mathbf{u}),...,Z^k(\mathbf{u})}(\mathbf{u}) = y$ is shorthand for $Y_{\mathbf{x},z^1,...,z^k}(\mathbf{u}) = y$, where $Z^1(\mathbf{u}) = z^1, ..., Z^k(\mathbf{u}) = z^k$. The conclusion follows. □

Theorem 15 *Let $Y_{\boldsymbol{x},Z^1,...,Z^k}$ be a nested counterfactual variable (with $Z^1, ..., Z^k$ nested counterfactual variables as well). For every nested counterfactual variable $W_{\boldsymbol{m},S^1,...,S^k}$ used in the inductive definition of $Y_{\boldsymbol{x},Z^1,...,Z^k}$, let $W_{\boldsymbol{m},s^1,...,s^k}$ be the corresponding "unrolled" ordinary counterfactual ($s^1, ..., s^k$ are values attained by $S^1, ..., S^k$).*

Then $P(Y_{\boldsymbol{x},Z^1,...,Z^k}) = \sum_{\boldsymbol{s}} P(\bigwedge_i W^i_{\boldsymbol{m},s^1,...,s^k})$, where the index i ranges over all "unrolled" ordinary counterfactuals attained from nested counterfactuals which occur in $Y_{\boldsymbol{x},Z^1,...,Z^k}$, and $\boldsymbol{s}$ is the set of values attained by all nested counterfactuals in $Y_{\boldsymbol{x},Z^1,...,Z^k}$, except $Y_{\boldsymbol{x},Z^1,...,Z^k}$ itself.

Proof: This result follows by inductive application of the argument used to establish Lemma 8. □

Theorem 16 *Let g be a subset of "allowed edges." Let $\boldsymbol{Y_x}(\boldsymbol{u}) - \boldsymbol{Y_{x*}}(\boldsymbol{u})$ be a path-specific effect in M_g. Then both random variables $\boldsymbol{Y_x}$, $\boldsymbol{Y_{x^*}}$ can be expressed in terms of a nested counterfactual in the original model M.*

Proof: It's not difficult to see that $P(\mathbf{Y}_{\mathbf{x}^*})$ in M_g corresponds to $P(\mathbf{Y}_{\mathbf{x}^*})$ in M.

The base case is if W has no observable parents in G. In this case, the distribution over W is just $P(W)$, a (trivial) counterfactual distribution, so W can be represented as a nested counterfactual.

In the inductive case, I partition the parent set of W into four sets. $Pa^+_{\mathbf{x}}(W)$ are the parents of W along "allowed" edges in G which are also in $\mathbf{X}$. Similarly, $Pa^-_{\mathbf{x}}(W)$ are the parents of W along "forbidden" edges in G which are in $\mathbf{X}$. $Pa^+_{\overline{\mathbf{x}}}(W)$ are the parents of W along "allowed" edges in G which are not in $\mathbf{X}$, and $Pa^-_{\overline{\mathbf{x}}}(W)$ are the parents of W along "forbidden" edges in G which are not in

$\mathbf{X}$. Let $\mathbf{x}^+$ be the values attained by $Pa^+_{\overline{\mathbf{x}}}(W)$ in $\mathbf{x}$, and $\mathbf{x}^-$ be the values attained by $Pa^-_{\overline{\mathbf{x}}}(W)$ in $\mathbf{x}$.

I claim that $P(W_{\mathbf{x}^+,\mathbf{x}^-,Z^1,...,Z^k})$ represents the effect of $\mathbf{x}$ on W in M_g. Here $Z^1, ..., Z^k$ are nested counterfactuals representing $Pa^+_{\overline{\mathbf{x}}}(W)$ and $Pa^-_{\overline{\mathbf{x}}}(W)$. Every $Z \in Pa^-_{\overline{\mathbf{x}}}(W)$ can be represented by a nested counterfactual since it just equals $Z_{\mathbf{x}^*}$ by definition. Similarly, every $Z \in Pa^+_{\overline{\mathbf{x}}}(W)$ is expressible by a nested counterfactual by the inductive hypothesis. The claim now follows by definition of M_g and by the inductive hypothesis. □

Corollary 3 *Let g be a subset of "allowed edges." Let $\boldsymbol{Y_x}(\boldsymbol{u}) - \boldsymbol{Y_{x^*}}(\boldsymbol{u})$ be a path-specific effect in M_g. Then both random variables $\boldsymbol{Y_x}$, $\boldsymbol{Y_{x^*}}$ can be expressed in terms of counterfactual distributions in the original model M.*
Proof: The result trivially follows for $P(\mathbf{Y}_{\mathbf{x}^*})$. It holds for $P(\mathbf{Y}_{\mathbf{x}})$ due to Theorems 15 and 16. □

Theorem 17 *If rule 1 applies to G_g at V, then the path-specific effect in G_g is equal to the path-specific effect in $G_{R^v_1(g)}$. If rule 2 applies to G_g at V, then the path-specific effect in G_g is equal to the path-specific effect in $G_{R^v_2(g)}$. If rule 3 applies to G_g at V, then the path-specific effect in G_g is equal to the path-specific effect in $G_{R^v_3(g)}$.*
Proof: I want to show that in either case, the nested counterfactuals corresponding to variables with incoming edges which changed status did not change after the rule was applied. Since no other nested counterfactual variables involved in the path-specific effect is affected by the marked graph modification, our result will follow.

This is easiest to show for rule 3. If $V \notin De(X)$, $V = V_{x^*} = V_x$, so the result follows. If $V \notin An(Y)$, then no nested counterfactual $V_{..}$ corresponding to V appears in any nested counterfactual corresponding to variables in Y, so the result follows.

Next, consider rule 2. The only variable to consider is the node W which is the child of V via the arrow e considered in that rule. The nested counterfactual $W_{..}$ for W has a single modification in its subscript, that of the nested counterfactual $V_{..}$ corresponding to V. But by Lemma 9, $V_{..} = V_{x^*}$, so our conclusion follows.

Next, consider rule 1. I want to show the the nested counterfactual $W_{..}$ corresponding to any Y-ancestral child W of V does not change after rule 1. Before rule 1, $W_{..} = W_{Z^1_{..},...,Z^k_{..},V_{x^*}}$, where $Z^1, ..., Z^k$ are other parents of W. After rule 2, $W_{..} = W_{Z^1_{..},...,Z^k_{..},V_{Y^1_{x^*},...Y^m_{x^*}}}$. But the nested counterfactuals $V_{..}$ in both expressions are equal by Lemma 9, which implies the result. □

Theorem 18 *The g-specific effect of Z on Y as described in Fig. 6.7 (a) is not P_*-identifiable.*
Proof: I extend models M^1 and M^2 from the previous proof with additional

variables V, Y, and U_Y. I assume $P(u_Y)$ is uniform, and both $P(V, Y|R)$ and the functions which determine V and Y are the same in both models.

Note that since all variables are discrete, the conditional probability distributions can be represented as tables. If I require $|R| = |V|$ and $|Y| = |V| * |R|$, then the conditional probabilities are representable as square matrices. I fix the functions f_V and f_Y, as well as the exogenous parents of V and Y such that the matrices corresponding to $P(y|v, r)$ and $P(v|r)$ matrices are invertible.

Call the extended models M^3 and M^4. Note that by construction, the two models are Markovian. Since M^1 and M^2 have the same P_*, and since the two extended models agree on all functions and distributions not in M^1 and M^2, they must also have the same P_*.

Consider the g-specific effect shown in Fig. 6.7 (a). From Theorem 3 I can express the path-specific effect in M^3_g in terms of M^3. In particular:

$$\begin{aligned} P(y_z)_{M^3_g} &= \sum_{rv} P(y_{rv} \wedge r_{z^*} \wedge v_z)_{M^3} \\ &= \sum_{r,v,r'} P(y_{rv} \wedge r_{z^*} \wedge v_{r'} \wedge r'_z)_{M^3} \\ &= \sum_{r,v,r'} P(y_{rv})_{M^3} P(v_{r'})_{M^3} P(r_{z^*}, r'_z)_{M^3} \end{aligned}$$

The last step is licensed by the independence assumptions encoded in the parallel worlds model of $y_{rv} \wedge r_{z^*} \wedge v_{r'} \wedge r'_z$. The same expression can be derived for $P(y_z)_{M^4_g}$. Note that since P_* is the same for both models they have the same values for the interventional distributions $P(y_{rv})$ and $P(v_{r'})$. Note that since $P(Y|R, V)$ and $P(V|R)$ are square matrices, the summing out of $P(Y|R, V)$ and $P(V|R)$ can be viewed as a *linear transformation*. Since the matrices are invertible, the transformations are one to one, and so if their composition. Since $P(y_{rv}) = P(y|r, v)$ and $P(v_{r'}) = P(v|r')$, and since $P(r_{z^*} \wedge r'_z)$ is different in the two models, I obtain that $P(y_z)_{M^3_g} \neq P(y_z)_{M^4_g}$. Since adding directed or bidirected edges to a graph cannot help identifiability, the result also holds in semi-Markovian models. □

Theorem 19 *Assume G_g is a marked graph with a single source X and a single outcome Y, such that rules 1,2, and 3 do not apply. Then either G_g satisfies the recanting witness criterion, or all marked edges emanate from X.*

Proof: Assume some marked edge does not leave X, and G_g does not satisfy the recanting witness criterion. Since rule 3 is not applicable, the marked edge must be in $An(Y) \cap De(X)$. Consider the nodes from which all marked edges in $An(Y) \cap De(X)$ emanate. Since the graph is acyclic, I can arrange these nodes in topological order. Pick the last node in the order, call it R. Since rule 1 is not applicable, there is an unmarked arrow leaving R in $An(Y) \cap De(X)$. By construction, there is a path from R to Y involving this arrow, and since R is the last node in the order, this path contains no marked edges. Since rule 2 is not applicable, there exists an allowable path from X to R. But this implies G_g satisfies the recanting witness criterion, which is a contradiction. □

Theorem 20 *Assume rules 1, 2, and 3 do not apply to G_g, and G_g satisfies the recanting witness criterion. Then the g-specific effect of X on Y is not P_*-identifiable.*

Proof: Consider the marked subgraph G'_G of G_g which just contains the paths which witness the recanting witness criterion. Let R be the "witness" node. Let $Y_{..}$ be a nested counterfactual corresponding to the path-specific effect of X on Y in G'_g. Since R has the only marked edge leaving it, $Y_{..}$ will contain two nested counterfactuals corresponding to R, the ordinary nested counterfactual $R_{..}$ which ultimately terminates with an x subscript, and R_{x^*}. Note that since the only value subscript in $R_{..}$ is x, $R_{..} = R_x$ by Lemma 9.

Let $Y'_{..}$ be the nested counterfactual where $R_{..}$ is replaced by R_x. By Theorem 3, $Y'_{..}$ can be expressed in terms of a counterfactual distribution. Moreover, by the method of construction used in the proof of Theorem 3, this distribution will contain a term for R_x and a term for R_{x^*}, and each term will have as subscripts all parents of the corresponding variable in G'_g. Since G'_g is Markovian, each term is thus independent of other terms. For every node W with parent Z on the path from R to Y, I can inductively apply the argument in Theorem 18 involving one-to-one linear maps. Specifically $P(W_z)$ is equal to $P(W|z)$. Moreover, since W is not Y, I am summing it out, which means I can arrange for $P(W|z)$ to be a one-to-one linear map. In this way, the conditional distributions of nodes on the two paths from R to Y compose with $P(Y_{pa(y)_{G'_g}})$ to construct a one-to-one map from $P(R_x, R_{x^*})$ to $P(Y_{..})$. But I know $P(R_x, R_{x^*})$ is not identifiable, so neither is $P(Y_{..})$.

To see that this translates into non-identification of $P(Y_{..})$ in G_g, note that I can arrange it so all nodes not in G'_g are independent of nodes in G'_g, and so do not affect my reasoning. □

Theorem 21 *If rules 1, 2, and 3 do not apply to G_g and all marked arrows emanate from X, then the path-specific effect of X on Y along g is identifiable in Markovian models.*

Proof: Let $\mathbf{W}$ be the set of children of X connected to X via a marked arrow, and $\mathbf{Z}$ the other children. Let $Y_{..}$ be a nested counterfactual corresponding to the path-specific effect in question. Since the only node with both marked and unmarked outgoing arrows is X (or possibly not even X), each variable in $De(X) \cap An(Y)$ gives rise to a single nested counterfactual in $Y_{..}$. Using Theorem 3, I can express $P(Y_{..})$ in terms of a counterfactual distribution. Moreover, since each counterfactual contains all parents as suffixes, and since the original graph is Markovian, all terms are independent of all other terms. But this means the expression is experimentally identifiable. □

Theorem 23 *If the unmarking rule applies to G_g at e, then path-specific effect in G_g is equal to the path-specific effect in $G_{R_4^e(g)}$.*
Proof: As before, I want to show that in either case, the nested counterfactuals corresponding to variables with incoming edges which changed status did not change after the rule was applied. Since no other nested counterfactual variables involved in the path-specific effect is affected by the marked graph modification, the result will follow.

If there is no marked directed path from $\mathbf{X}$ to V, then we can partition $\mathbf{X}$ into two subsets $\mathbf{X}_1, \mathbf{X}_2$, where e is not a descendant of nodes in $\mathbf{X}_1$, while all directed paths from $\mathbf{X}_2$ to e are blocked by a marked edge. Let V be the node from which e emanates. Then, $V = V_{\mathbf{x}_1^*}$. Furthermore, if I apply the unmarking rule to e, the nested counterfactual $W_{..}$, where W is the child of V via e, has a single modification in its subscript, that of the nested counterfactual $V_{..}$ corresponding to V. But since there are no allowed path from $\mathbf{X}_2$ to V, $V_{..} = V_{\mathbf{x}_2^*}$.

If $V \not\in An(\mathbf{Y})$, a nested counterfactual corresponding to V does not appear in any nested counterfactuals corresponding to nodes in $\mathbf{Y}$, so the result follows. □

Theorem 24 *Assume G_g is a marked graph, I am interested in a g-specific effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$, and neither rule 1, nor the unmarking rule are applicable to G_g. Then either all marked edges emanate from nodes in $\boldsymbol{X}$, or there is a node R such that there is an allowed directed path from $\boldsymbol{X}$ to R, an allowed directed path from R to $\boldsymbol{Y}$, and a forbidden directed path from R to $\boldsymbol{Y}$. See Fig. 6.8.*
Proof: Assume such an R does not exist, and some marked edge does not emanate from $\mathbf{X}$. Consider the nodes which all such marked edges emanate. Since the graph is acyclic, I can arrange these nodes in topological order. Pick the last node in the order, call it R. Since the unmarking rule is not applicable, R is both ancestral to $\mathbf{Y}$, and there is a directed path from $\mathbf{X}$ to R. Since rule 1 is not applicable, there is an unmarked arrow leaving R which is a part of a directed path from R to Y, and by construction this path contains no marked edges. By construction, there is a path from R to $\mathbf{Y}$ involving e, which means I have a contradiction. □

Theorem 25 *Assume G_g contains the patterns shown in Fig. 6.8. Then the g-specific effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$ is not P_*-identifiable.*
Proof: The proof is almost identical to that of Theorem 20. I first show that the counterfactual distribution representing the effect of interest must contain the terms R_x, R_{x^*}, for some X. I then use induction on the path of the generalized kite graph that this implies the path-specific effect of X on $\{Y_1, Y_2\}$ is not identifiable

from P_*. By making sure that all nodes in G_g outside the generalized kite are independent of nodes inside the generalized kite, I conclude the non-identifiability of the effect of $\mathbf{X}$ on $\mathbf{Y}$. □

Theorem 26 *Assume all marked arrows emanate from $\boldsymbol{X}$ in G_g. Then the path-specific effect of $\boldsymbol{X}$ on $\boldsymbol{Y}$ is identifiable in Markovian models.*

Proof: The proof is almost identical to that of Theorem 21. The only difference is that since there are multiple variables in $\mathbf{Y}$, a given node can give rise to multiple nested counterfactuals. However, since the only nodes with both marked and unmarked outgoing arrows are those in $\mathbf{X}$, and they do not give rise to nested counterfactuals, any node not in $\mathbf{X}$ will give rise to multiple nested counterfactuals which are syntactically identical, and so are duplicate events. Since the graph is Markovian, each counterfactual with its parents fixed is independent of all others. Thus, the whole expression is P_*-identifiable. □

APPENDIX D

Proofs for Chapter 7 (Dormant Independence)

Theorem 29 *For any variable Y in G, there exists a unique maximum ancestral confounded set (MACS) T_y.*
Proof: Maximal ancestral confounded sets exist for any Y since I only consider finite graphs. Assume there is Y with two distinct maximal ancestral confounded sets S_1, S_2. I claim that $S = S_1 \cup S_2$ is an ancestral confounded set, which is a contradiction. By construction, S is a C-component in G_S, since any node $X \in S_1$ and any node $Z \in S_2$ can be connected by a bidirected path constructed by appending the bidirected path from X to Y in G_{S_1} (guaranteed to exist since S_1 is a C-component in G_{S_1}) to the bidirected path from Z to Y in G_{S_2} (guaranteed to exist since S_2 is a C-component in G_{S_2}). Since $S_1 \in An(Y)_{G_{S_1}}$, and $S_2 \in An(Y)_{G_{S_2}}$, $S \in An(Y)_{G_S}$. □

Theorem 30 ***Find-MACS***(G, Y) *outputs the MACS of Y in polynomial time.*
Proof: The algorithm is polynomial since determining $An(.)$ and $C(.)$ sets can be done in polynomial time, and each recursive call eliminates at least one node from the graph. Since the MACS of Y is unique, all ancestral confounding sets of Y are contained in it (otherwise, I can repeat the argument in Theorem 29). First, I show that the output set S of **Find-MACS** is an ancestral confounding set of Y. If not, then either $S \neq An(Y)_{G_S}$ or $S \neq C(Y)$. But the algorithm only returns if there is no element in S outside $An(Y)_{G_S}$, and no element in S outside $C(Y)_{G_S}$. To show that S is maximum, assume this isn't the case, and let $\mathbf{Z} \subseteq T_y \setminus S$ be the first node set in T_y removed by **Find-MACS**. Let G' be the graph at the stage where $\mathbf{Z}$ is removed. By assumption, T_y is contained in G', and either $\mathbf{Z} \not\subset An(Y)_{G'}$ or $\mathbf{Z} \not\subset C(Y)_{G'}$. But $\mathbf{Z} \subset An(Y)_{G_{T_y}}$, and $\mathbf{Z} \subset C(Y)_{G_{T_y}}$ by definition of T_y. Contradiction. □

Theorem 31 *Let T_x, T_y be the MACSs of X, Y. Let $I_{x,y} = Pa(T_x \cup T_y) \setminus (T_x \cup T_y)$. Then if either X is a parent of T_y, Y is a parent of T_x or there is a bidirected arc between T_x an T_y, then X, Y are not d*-separable. Otherwise, $X \perp_{i_{x,y}} Y | T_x \cup T_y \setminus \{X, Y\}$.*
Proof: Assume either X is a parent of T_y or T_x, T_y are connected by a bidirected arc. It's easy to verify, by definition of T_y, that the the above imply the presence of an inducing path [VP90] from X to Y. Thus, no conditioning set can d-separate X and Y. I want to show that identifiable interventions don't help.

Consider disjoint subsets S, S' of T_y. A result in [SP06a] implies that $P(\mathbf{v}), G \not\vdash_{id} P(y|s', do(s))$ iff $P(\mathbf{v}), G \not\vdash_{id} P(y, t|do(s, t'))$, where T, T' is a certain partition of S'. By Theorem 28, $P(\mathbf{v}), G \not\vdash_{id} P(y|do(\mathbf{w}))$ for any subset $\mathbf{W}$ of T_y, which in turn implies $P(\mathbf{v}), G \not\vdash_{id} P(y, t|do(s, t))$. But if $P(\mathbf{v}), G \not\vdash_{id} P(y|s', do(s))$, then $P(\mathbf{v}), G \not\vdash_{id} P(y, x|s', do(s))$. It is not difficult to construct a model where for any superset $\mathbf{Z}$ of S', and superset $\mathbf{W}$ of S, $P(\mathbf{v}), G \not\vdash_{id} P(y, x|\mathbf{z}, do(\mathbf{w}))$ (by for instance letting nodes outside T_y be mutually independent). This implies the result.

To show the other direction, consider $G_{\overline{i_{x,y}}}$, and a possible d-connected path from X to Y. This path starts with an arrow leaving X or an arrow entering X. Assume the arrow is leaving X. X cannot have conditioned descendants in $G_{\overline{i_{x,y}}}$ unless X was a parent of T_y or $x \in T_y$, both of which are impossible by assumption. This means the path from X is just a set of directed arrows from X. But such a path must run into nodes fixed by $I_{x,y}$, unless X was a parent of T_y or in T_y, which is impossible. Thus, no path starting with an outgoing arrow from X can be d-connected to Y.

Assume the path starts with an incoming arrow into X. If the arrow is directed, the corresponding parent Z of X is either in T_x or in $I_{x,y}$ (and in neither case can Z be equal to Y). In either case, the path is not d-connected to Y. If the arrow is bidirected, I have two cases. Either the next node Z in the path is in T_y or outside both T_y and $I_{x,y}$ (Z cannot be in $I_{x,y}$ since then the path will not be d-connected). For the first case, I repeat the argument until I reach the second case. For the second case, Z cannot be in T_x, else there is a bidirected path from T_x to T_y, which is ruled out by assumption. Note that Z cannot have conditioned descendants in $G_{\overline{i_{x,y}}}$ unless Z was a parent of T_x or T_y or was in T_x or T_y. But I ruled all these cases out. Therefore, the subsequent arrows on the path are directed arrows away from Z. As before, these arrows must eventually reach $I_{x,y}$, which means the path is not d-connected. □

Lemma 11 *Every AC-component has an ancestral confounded set.*
Proof: If an AC-component is a singleton, this is obvious. Otherwise, $\mathbf{Y}$ is a union of AC-components $\mathbf{Y}_1, \mathbf{Y}_2$ with ancestral confounded sets S_1, S_2. Let $S = S_1 \cup S_2$. Since there is a bidirected arc from S_1 to S_2, for every node $X \in S, S = C(X)_S$. Moreover, by construction $S = An(\mathbf{Y})_S$. Thus, S is an ancestral confounded set for $\mathbf{Y}$. □

Lemma 12 *Let $\mathbf{Y}$ be a variable set, $Y \in \mathbf{Y}$. Then there is a unique maximum AC-component which both contains Y and is a subset of $\mathbf{Y}$.*
Proof: Some such AC-component exists, since Y itself is a trivial AC-component. Since $\mathbf{Y}$ is finite there is a maximal such AC-component. Assume there are two distinct maximal AC-components containing Y which are subsets of $\mathbf{Y}$, say $\mathbf{Y}_1, \mathbf{Y}_2$. Let S_1, S_2 be the corresponding MACSs. Since these AC-components

have the node Y in common, S_1 and S_2 have a node in common, and so are connected by a bidirected arc. This implies $\mathbf{Y}_1 \cup \mathbf{Y}_2$ is an AC-component, which is a contradiction. □

Theorem 33 *Any variable set* $\boldsymbol{Y}$ *has a unique partition* p*, called the AC-partition, where each element* S *in* p *is a maximal AC-component in a sense that no superset of* S *which is also a subset of* $\boldsymbol{Y}$ *is an AC-component.*
Proof: To see that there is a unique AC-partition p, start with some node $Y \in \mathbf{Y}$, find it's unique maximum AC-component which is still a subset of $\mathbf{Y}$, and repeat the process for the nodes which have not been made part of some AC-component. The set of AC-components obtained in this way is a partition where each element is a maximal AC-component. Since each AC-component is also maximum and unique, p is unique. □

Theorem 34 ***Find-AC-Partition***$(G, \boldsymbol{Y})$ *outputs the unique AC-partition of* $\boldsymbol{Y}$*, and the set of MACSs for each element in the partition.*
Proof: I first show that p, the output of **Find-AC-Partition**, consists of a partition of AC-components (not necessarily maximal). Clearly this is true at the initialization step, since a singleton is a trivial AC-component. It's also clear by definition that any merge of $\mathbf{Y}_1, \mathbf{Y}_2$ results in an AC-component $\mathbf{Y}'$. Furthermore, by Theorem 7, $T_{\mathbf{Y}'}$ is the MACS of $\mathbf{Y}'$.

Let p^* be the AC-partition of $\mathbf{Y}$. I claim that p^* must be coarser than p, in a sense that every element in p^* is a union of a set of elements in p. Note that this definition holds if p^* is equal to p. Assume not. Then there are some sets $S \in p, S' \in p^*$ such that some elements in S are in S' and some are not. Let $Z \in S \cap S'$. By Lemma 12, there is a unique maximum AC-component containing Z which is also a subset of $\mathbf{Y}$. By definition of p^*, S' is this AC-component. But if S is not contained in S', I can derive a contradiction by repeating the argument in the proof of Lemma 12.

Finally, I want to show p^* is equal to p. Assume this isn't the case, and fix some element S' in p^* which is a union of two or more elements in p. Since each AC-component is either a singleton, or constructed from two smaller AC-components, I can construct a binary tree T, where each leaf is a node in S', and each non-leaf represents an AC-component obtained from the AC-component corresponding to the left subtree of the non-leaf and the AC-component corresponding to the right subtree of the non-leaf.

I want to find an AC-component A in T with the property that its left subtree corresponds to a subset of some element S_1 in p, and its right subtree corresponds to a subset of another element S_2 in p. This AC-component must exist, since leaves in T are singletons, and the root of T corresponds to S', which spans multiple elements in p. This implies that the MACS of a subset of S_1 is connected to the MACS of a subset of S_2 by a bidirected arc. But the MACS of S_1 and the MACS of S_2 are supersets of these connected MACS, so they are themselves connected by a bidirected arc. But then p could not have been the output of **Find-AC-Partition**. □

Theorem 35 *$\mathbf{X}$ cannot be d-separated from $\mathbf{Y}$ in G if and only if there exists an inducing path from $\mathbf{X}$ to $\mathbf{Y}$ in G,*
Proof: Assume there is no inducing path from $\mathbf{X}$ to $\mathbf{Y}$. Let $\mathbf{A} = An(\mathbf{X} \cup \mathbf{Y}) \setminus (\mathbf{X} \cup \mathbf{Y})$. I claim that $\mathbf{X} \perp \mathbf{Y} | \mathbf{A}$. It's not hard to see that if there is a d-connected path from $\mathbf{X}$ to $\mathbf{Y}$, then it does not have any nodes not in $\mathbf{A}$. Assume otherwise. Then some node on this path not in $\mathbf{A}$ must contain a collider. But this implies the path is not d-connected, since this node does not have descendants in $\mathbf{A}$.

Since I condition on $\mathbf{A}$, the d-connected path must consist exclusively of colliders. Moreover, by definition every node on the path is an ancestor of either $\mathbf{X}$ or $\mathbf{Y}$. But this means the path is inducing. Contradiction.

Assume the inducing path from X to Y. I want to show I cannot d-separate $\mathbf{X}$ from $\mathbf{Y}$. First, I show that $\mathbf{X} \not\perp \mathbf{Y}$.

I have three cases. The inducing path contains either entirely bidirected arcs, or one directed arc following by zero or more bidirected arcs, or one directed arc, following by zero or more bidirected arcs, followed by a directed arc.

Let A be the first node on the inducing path after X, B be the first node on the inducing path after Y. If all nodes on the inducing path are ancestors of $\mathbf{X}$, then B is an ancestor of $\mathbf{X}$. But the edge between Y and B is either bidirected, or directed from Y to B. In either case, the ancestral path from $\mathbf{X}$ to B plus this edge forms a d-connected path from $\mathbf{X}$ to Y. The same argument applies if all nodes on the inducing path are ancestors of $\mathbf{Y}$. Otherwise, find two neighboring nodes C, D on the inducing path where C is an ancestor of $\mathbf{X}$, and D is an ancestor of $\mathbf{Y}$. Then the ancestral path from $\mathbf{X}$ to C, along with the edge along the inducing path from C to D, along with the ancestral path from $\mathbf{Y}$ to D form a d-connected path from $\mathbf{X}$ to $\mathbf{Y}$.

What I have to show is that regardless of which sets of nodes I condition on, some d-connected path between $\mathbf{X}$ and $\mathbf{Y}$ remains. Let p' the subpath of p such that nodes on p' are either conditioned on themselves, or their descendants are conditioned on. If $p' = p$, I am done since p is d-connected. Otherwise, consider every pair of nodes A, B on $p \setminus p'$ such that all nodes on p between A and B are in p'. By construction, the fragment of p between A and B is a d-connected path, terminating with arrowheads on both ends. To show that there is a d-connected path between $\mathbf{X}$ and $\mathbf{Y}$, we repeat the above d-connection argument, except rather than considering the path p, I consider the path $p \setminus p'$, and instead of the d-connected paths between every node pair A, B as above, I consider a bidirected arc. □

Theorem 36 *Let $\boldsymbol{X}$, $\boldsymbol{Y}$ be arbitrary sets of variables. Let p be the AC-partition of $\boldsymbol{X}\cup$ $\boldsymbol{Y}$. Then if either elements of both $\boldsymbol{X}$ and $\boldsymbol{Y}$ share a single AC-component in p, or some element of $\boldsymbol{X}$ is a parent of the MACS of some AC-component containing elements of $\boldsymbol{Y}$ (or vice versa), then $\boldsymbol{X}$ cannot be d*-separated from $\boldsymbol{Y}$. Otherwise, let T_p be the union of all MACSs of elements in p, and let $I_p = Pa(T_p) \setminus T_p$. Then, $\boldsymbol{X} \perp_{i_p} \boldsymbol{Y} | T_p \setminus (\boldsymbol{X} \cup \boldsymbol{Y})$.*
Proof: What I want to show is that the conditions for the absence of d*-separation of sets $\mathbf{X}, \mathbf{Y}$ imply that there is an inducing path between $\mathbf{X}$ and $\mathbf{Y}$, and that no interventions on nodes in that inducing path are identifiable, at least if either $\mathbf{X}$ or $\mathbf{Y}$ are the effect variables.

I first want to show that if $\mathbf{Z}$ is an AC-component, then for any disjoint subsets S, S' of the MACS $T_{\mathbf{z}}$, $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{z}|s', do(s))$. By a result from [SP06a], $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{z}|s', do(s))$ iff $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{z}, t|do(s, t'))$, where T, T' is a particular partition of S'. But if $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{z}|do(s, t'))$, then $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{z}, t|do(s, t'))$. Without loss of generality, then, I will prove that $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{z}|do(s))$. By Theorem 28, this is true if $\mathbf{Z} = \{Z\}$. Assume this is true for AC-components $\mathbf{Z}_1, \mathbf{Z}_2$. I want to show this also holds for the AC-component $\mathbf{Z}$ obtained from these two AC-components. Clearly, the result also holds for $T = T_{\mathbf{z}_1} \cup T_{\mathbf{z}_2}$. I want to show the same is true for $T_{\mathbf{z}}$. By construction, $T_{\mathbf{z}}$ can be used to construct a C-forest [SP06b] for $\mathbf{Z}$. The same is true for T. Then $T, T_{\mathbf{z}}$ form a hedge [SP06b] for $P(\mathbf{z}|do(s'))$, for any set $S' \subseteq T_{\mathbf{z}} \setminus T$, which means the result holds for $T_{\mathbf{z}}$.

If there is an AC-component containing both elements of $\mathbf{X}$ and $\mathbf{Y}$, then an inducing path between $\mathbf{X}$ and $\mathbf{Y}$ exists by the definition of AC-component. Similarly, if some element of $\mathbf{X}$ is a parent of the MACS of some AC-component which is a subset of $\mathbf{Y}$, then an inducing path between $\mathbf{X}$ and $\mathbf{Y}$ exists by the definition of AC-component.

If there is an AC-component $\mathbf{C}$ containing both elements of $\mathbf{X}$ and $\mathbf{Y}$, then by above reasoning for any disjoint subsets S, S' of $T_{\mathbf{z}}$, $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{c}|s', do(s))$. Similarly, if there is an element of $\mathbf{X}$ which is a parent of the MACS of some AC-component $\mathbf{Y}'$ which is a subset of $\mathbf{Y}$, then by above reasoning for any disjoint subsets S, S', $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{y}'|s', do(s))$. As before, it is not difficult to construct a model where for any superset $\mathbf{Z}$ of S' and superset $\mathbf{W}$ of S, $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{c}|\mathbf{z}, do(\mathbf{w}))$ (in the first case), or $P(\mathbf{v}), G \not\vdash_{id} P(\mathbf{y}'|\mathbf{z}, do(\mathbf{w}))$, (in the second case). In either case, no combination of fixing and conditioning can get rid of the inducing path, and the result follows.

To prove the other direction, consider a d-connected path in $G_{\overline{i_p}}$ from $X \in \mathbf{X}$ to $Y \in \mathbf{Y}$. Without loss of generality, assume no elements in $\mathbf{X}, \mathbf{Y}$, other than the end points are on this path.

The path either starts with an outgoing arrow, an incoming arrow, or a bidirected arrow. Assume it starts with an outgoing arrow into a node Z. If Z is inside some MACS, the next edge on the path can be assumed to be bidirected. This is because this MACS cannot contain any nodes in $\mathbf{Y}$, and because the next node Z is conditioned on by assumption. Since the arrow is bidirected, I handle this case in the "bidirected arrow" situation. If Z is outside any MACS, it is either in I_p, in which case the path is not d-connected, or it does not have any conditioned descendants, since the parents of every MACS are fixed. This means the segment of the path from Z is just a set of directed arrows pointing away from Z. But such a path must run into nodes fixed by I_p, which is impossible. Thus there are no d-connected path starting with an outgoing arrow from X.

Assume the path starts with an incoming arrow into X. If the arrow is directed, the corresponding parent Z of X is either in T_p. or in I_p. If it is in I_p, the path is not d-connected, since no element of Y can be a parent of the MACS of an AC-component containing X by assumption. If it is in T_p, it is conditioned on, and the path is not d-connected.

If the arrow is bidirected, I have two cases. Either the next node Z in the path is in the MACS of an AC-component containing X, or outside both this AC-component, and I_p. For the first case, I repeat the argument until I reach the second case. For the second case, Z cannot be in any other MACS. Otherwise, there is a bidirected arc between distinct MACSs returned by **Find-AC-Partition** which is impossible by Theorem 34. Note that Z cannot have conditioned descendants in $G_{\overline{i_p}}$ unless Z was in I_p, which is impossible. Therefore, the subsequent arrows on the path are directed arrows away from Z. As before, these arrows must eventually reach I_p, which means the path is not d-connected. □

Theorem 37 ***Test-Edges*** *terminates in polynomial time, and any edge it removes from G', valid for an experimentally faithful model M, is extraneous.*
Proof: The first claim is simple to establish since all input graphs are acyclic, and using Theorem 32. Let G be the true causal graph. Assume an edge (X, Y) is not extraneous but is removed from G' by **Test-Edges**. Assume sets $\mathbf{Z}, \mathbf{W}$ witness the removal. But $X \perp\!\!\!\perp_{\mathbf{w}} Y|\mathbf{Z}$, and since the submodel $M_{\mathbf{w}}$ of M is faithful, this implies (X, Y) must be extraneous. □

References

[ASP05] Chen Avin, Ilya Shpitser, and Judea Pearl. "Identifiability of Path-Specific Effects." In *International Joint Conference on Artificial Intelligence*, volume 19, pp. 357–363, 2005.

[Bax92] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, London, 1992.

[Bes74] J. Besag. "Spatial Interaction and the Statistical Analysis of Lattice Systems." *Journal of the Royal Statistical Society*, **36**:192–236, 1974.

[BP94a] Alexander Balke and Judea Pearl. "Counterfactual Probabilities: Computational Methods, Bounds and Applications." In *Proceedings of UAI-94*, pp. 46–54, 1994.

[BP94b] Alexander Balke and Judea Pearl. "Probabilistic Evaluation of Counterfactual Queries." In *Proceedings of AAAI-94*, pp. 230–237, 1994.

[Daw79] A. Philip Dawid. "Conditional Independence in Statistical Theory." *Journal of the Royal Statistical Society*, **41**:1–31, 1979.

[Daw00] A Philip Dawid. "Causal Inference without Counterfactuals." *Journal of the American Statistical Association*, **95**:407–448, 2000.

[Fis26] R. A. Fisher. *The Design of Experiments.* 6th edition. Edinburgh, U.K.: Oliver and Boyd, 1926.

[GP98] David Galles and Judea Pearl. "An axiomatic characterization of causal counterfactuals." *Foundation of Science*, **3**:151–182, 1998.

[Haa43] Trygve Haavelmo. "The statistical implications of a system of simultaneous equations." *Econometrica*, **11**:1–12, 1943.

[Hal00] Joseph Halpern. "Axiomatizing Causal Reasoning." *Journal of A.I. Research*, pp. 317–337, 2000.

[HV06a] Yimin Huang and Marco Valtorta. "Identifiability in Causal Bayesian Networks: A Sound and Complete Algorithm." In *Twenty-First National Conference on Artificial Intelligence*, 2006.

[HV06b] Yimin Huang and Marco Valtorta. "Pearl's Calculus of Interventions is Complete." In *Twenty Second Conference On Uncertainty in Artificial Intelligence*, 2006.

[JW02] M. I. Jordan and Y. Weiss. "Graphical Models: Probabilistic Inference." In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks, 2nd edition.* Cambridge, MA: MIT Press, 2002.

[Kal60] R. E. Kalman. "A New Approach to Linear Filter and Prediction Problems." *Transactions of the ASME - Journal of Basic Engineering*, **82**:35–45, 1960.

[Kli05] R. B. Kline. *Principles and Practice of Structural Equation Modeling.* The Guilford Press, 2005.

[Lau96] S.L. Lauritzen. *Graphical Models.* Oxford, U.K.: Clarendon, 1996.

[LB94] Wai Lam and Fahiem Bacchus. "Learning Bayesian Belief Networks: An Approach Based on the MDL Principle." *Computational Intelligence*, **10**(4), 1994.

[Lew73] D. Lewis. *Counterfactuals.* Cambridge, MA: Harvard University Press, 1973.

[LS88] S. L. Lauritzen and D.J SPiegelhalter. "Local computations with probabilities on graphical structures and their application to expert systems." *Journal of the Royal Statistical Society*, **Ser. B 50**:157–224, 1988.

[Ney23] J. Neyman. "Sur les applications de la thar des probabilities aux experiences Agaricales: Essay des principle. Excerpts reprinted (1990) in English." *Statistical Science*, **5**:463–472, 1923.

[Pea85] Judea Pearl. "A Constraint-Propagation Approach to Probabilistic Reasoning." In *Uncertainty in Artificial Intelligence (UAI)*, pp. 31–42, 1985.

[Pea86] Judea Pearl. "Fusion, propagation, and structuring in belief networks." *Artificial Intelligence*, **29**:241–288, 1986.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan and Kaufmann, San Mateo, 1988.

[Pea93a] Judea Pearl. "Belief Networks Revisited." *Artificial Intelligence*, **59**:49–56, 1993.

[Pea93b] Judea Pearl. "Graphical Models, Causality, and Intervention." *Statistical Science*, **8**:266–9, 1993.

[Pea93c] Judea Pearl. "A probabilistic calculus of actions." In *Uncertainty in Artificial Intelligence (UAI)*, volume 10, pp. 454–462, 1993.

[Pea95] Judea Pearl. "Causal Diagrams for Empirical Research." *Biometrika*, **82**(4):669–709, 1995.

[Pea00] Judea Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

[Pea01] Judea Pearl. "Direct and Indirect Effects." In *Proceedings of UAI-01*, pp. 411–420, 2001.

[PJ75] J.W. O'Connell P.J. Bickel, E.A Hammel. "Sex bias in graduate admissions: Data from Berkeley." *Science*, **187**:398–404, 1975.

[PV91] Judea Pearl and T. S. Verma. "A Theory of Inferred Causation." In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pp. 441–452, 1991.

[Rab89] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." In *Proceedings of the IEEE*, volume 77, pp. 257–286, 1989.

[RG92] James M. Robins and Sander Greenland. "Identifiability and Exchangeability of Direct and Indirect Effects." *Epidemiology*, **3**:143–155, 1992.

[RG99] Sam Roweis and Zoubin Ghahramani. "A Unifying Review of Linear Gaussian Models." *Neural Computation*, **11**:305–345, 1999.

[Rob87] J. M. Robins. "A Graphical Approach to the Identification and Estimation of Causal Parameters in Mortality Studies with Sustained Exposure Periods." *Journal of Chronic Disease*, **2**:139–161, 1987.

[Rob97] James M. Robins. "Causal Inference from Complex Longitudinal Data." In *Latent Variable Modeling and Applications to Causality*, volume 120, pp. 69–117, 1997.

[Rub74] D. B. Rubin. "Estimating causal effects of treatments in randomized and non-randomized studies." *Journal of Educational Psychology*, **66**:688–701, 1974.

[SGS93] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* Springer Verlag, New York, 1993.

[Shp07] Ilya Shpitser. "Appendum to Identification of Conditional Interventional Distributions." Technical Report R-329-APPENDUM, Cognitive Systems Laboratory, University of California, Los Angeles, 2007.

[SP06a] Ilya Shpitser and Judea Pearl. "Identification of Conditional Interventional Distributions." In *Uncertainty in Artificial Intelligence*, volume 22, 2006.

[SP06b] Ilya Shpitser and Judea Pearl. "Identification of Joint Interventional Distributions in Recursive Semi-Markovian Causal Models." In *Twenty-First National Conference on Artificial Intelligence*, 2006.

[Suz93] J. Suzuki. "A construction of Bayesian networks from databases based on an MDL scheme." In *UAI 93*, pp. 266–273, 1993.

[Tia02] Jin Tian. *Studies in Causal Reasoning and Learning*. PhD thesis, Department of Computer Science, University of California, Los Angeles, 2002.

[Tia04] Jin Tian. "Identifying Conditional Causal Effects." In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.

[Tin37] J. Tinbergen. *An Econometric Approach to Business Cycle Problems*. Hermann, Paris, 1937.

[TP02] Jin Tian and Judea Pearl. "A General Identification Condition for Causal Effects." In *Eighteenth National Conference on Artificial Intelligence*, pp. 567–573, 2002.

[Ver86] T. S. Verma. "Causal networks: semantics and expressiveness." Technical Report R-65, Cognitive Systems Laboratory, University of California, Los Angeles, 1986.

[VP88] T. Verma and Judea Pearl. "Influence Diagrams and d-Separation." Technical Report R-101, Cognitive Systems Laboratory, University of California, Los Angeles, 1988.

[VP90] T. S. Verma and Judea Pearl. "Equivalence and Synthesis of Causal Models." Technical Report R-150, Department of Computer Science, University of California, Los Angeles, 1990.

[Wri21] S. Wright. "Correlation and Causation." *Journal of Agricultural Research*, **20**:557–585, 1921.