

Correct Causal Inference in Probabilistic Logic Programming

Kilian Rückschloß, Felix Weitkämper

Ludwig-Maximilians-Universität München
Institut für Informatik
Lehr- und Forschungseinheit für Programmier- und Modellierungssprachen
Oettingenstraße 67
D-80538 München (Germany)

Abstract

In the current work we investigate the relationship between the classical distribution semantics and causal semantics for probabilistic logic programming. We first give an infinite family of programs yielding the same distribution. Therefore, we deduce that unlike in classical causal structure discovery, in the relational case it is not possible to build a finite search space of causal explanations for an observed distribution. Finally, this article develops a procedure to verify the causal content of a probabilistic logic program, whose structure is obtained from observations.

Keywords

Probabilistic Logic Programming, Causal Structure Discovery, Causal Semantics, Statistical Relational Artificial Intelligence

1. Introduction

To illustrate the difficulties of causal structure discovery in probabilistic logic programming we begin by constructing an infinite family of programs that all induce the same distribution:

Assume we are given a database that models a directed graph G , that is, it consists a fact $e(x, y)$ for every edge $x \rightarrow y$ in G . Moreover, each node x in G may satisfy a certain property p . In this case our database contains the fact $p(x)$.

Let $n \in \mathbb{N}$ be a natural number and let x be a node in G . We write $n_path(x)$ if there exists a path of length at most n from x to a node satisfying p . The predicate $n_path(_)$ can be defined by the following normal clause:

$$n_path(X) \leftarrow e(X, X_1), e(X_1, X_2), \dots, e(X_{n-1}, X_n), p(X_n)$$

Further, we write $no_n_path(x)$ if no such path exists, i.e.

$$no_n_path(X) \leftarrow \neg n_path(X).$$


Haifa'22: 15th Workshop on Answer Set Programming and Other Computing Paradigms


✉ kilian.rueckschloss@lmu.de (K. Rückschloß); felix.weitkaemper@lmu.de (F. Weitkämper)

🌐 <https://www.pms.ifi.lmu.de/mitarbeiter/derzeitige/kilian-rueckschloss/> (K. Rückschloß);

<https://www.pms.ifi.lmu.de/mitarbeiter/derzeitige/felix-weitkaemper/index.html> (F. Weitkämper)

🆔 0000-0002-7891-6030 (K. Rückschloß); 0000-0002-3895-8279 (F. Weitkämper)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Let $\alpha \in (0, 1]$ be an arbitrary real number. Further, assume that every node x in G gives rise to two Boolean random variables $r(x)$ and $s(x)$. We consider the following probabilistic logic program:

$$\mathbf{P}_n := \begin{cases} r(X) : \alpha \leftarrow n_path(X) \\ s(X) \leftarrow r(X), n_path(X) \\ s(X) : \alpha \leftarrow no_n_path(X) \\ r(X) \leftarrow s(X), no_n_path(X) \end{cases},$$

An easy calculation shows that all the programs \mathbf{P}_n yield the same distribution of the random variables $r(x)$ and $s(x)$, where x ranges over the nodes of the fixed directed graph G . However, for a fixed $n \in \mathbb{N}$ the program \mathbf{P}_n describes a causal mechanism, in which for every node x in G we find that $s(x)$ is a cause of $r(x)$ unless $n_path(x)$ holds. On the other hand $r(x)$ is a cause of $s(x)$ if we find $n_path(x)$ in G . Hence, all the \mathbf{P}_n represent different causal structures.

Furthermore, assume that we are given data sampled from the distribution of $r(_)$ and $s(_)$. Moreover, a structure learning algorithm like slipcover [1] yields a program \mathbf{P} that induces the same distribution as all the \mathbf{P}_n . Hence, we obtain infinitely many hypothesis for the underlying causal mechanism, even if we assume that \mathbf{P} represents the right distribution.

Therefore, unlike the situation in classical causal structure discovery in probabilistic logic programming it is in general impossible to compute the search space of all causal explanations for an observed distribution. Hence, given a probabilistic logic program, whose structure has been derived from observational data, the best one can do is to verify its causal content. The aim of this contribution is to propose a procedure for this verification task.

In Section 2 we begin by recalling Pearl’s functional causal models and their relations to causal Bayesian networks. Further, we give a brief overview over the theory of causal structure discovery for causal Bayesian networks. Finally, we introduce a semantics for probabilistic logic programs in terms of functional causal models, which allows us to ground a given program to a causal Bayesian network. The main idea of our work in Section 3 is then to lift the indistinguishability statement for causal Bayesian networks, Theorem 5, to probabilistic logic programming in order to accomplish our verification task.

2. Preliminaries

We begin with recalling Pearl’s theory of causality. Further, we establish a semantics for probabilistic logic programs that allows us to reason about causal relationships in those programs. Note that in the following we use π to denote probabilities of random variables.

2.1. Pearl’s Functional Causal Models

Let us recall the definition of a functional causal model [2, 1.4.1]:

Definition 1 (Functional Causal Model). *A **functional causal model** or **causal model** \mathcal{M} on a set of variables \mathbf{V} is a system of equations that consists of one **defining equation** of the form $X := f_X(\text{Pa}(X), \text{Error}(X))$ for each variable $X \in \mathbf{V}$. Here the **parents** $\text{Pa}(X) \subseteq \mathbf{V}$ of X form a subset of the set of variables \mathbf{V} , the **error term** $\text{Error}(X)$ of X is a vector of random variables*

and f_X is a function defining X in terms of the parents $\text{Pa}(X)$ and the error term $\text{Error}(X)$ of X .

To illustrate Definition 1 we introduce the following example:

Example 1. Assume we are given a database of countries. Further, we know which products p are produced in country c denoted by $\text{prod}(c, p)$. For each product and each country there is a probability α such that country c needs the product p , denoted by $\text{need}(c, p)$. This is reflected in the following equation:

$$\text{need}(c, p) := u_1(c, p) \text{ with } \pi(u_1(c, p)) := \alpha \quad (1)$$

Analogously, a country c is corrupt (denoted $\text{corrupt}(c)$) with probability β , i.e.

$$\text{corrupt}(c) := u_2(c) \text{ with } \pi(u_2(c)) := \beta. \quad (2)$$

If a country c_1 needs a product p and does not produce p , but c_2 does, then with probability γ the country c_1 imports from c_2 , denoted by $\text{imp}(c_1, c_2)$, i.e.

$$\text{imp}(c_1, c_2) := \bigvee_{\substack{\neg \text{prod}(c_1, p) \\ \text{prod}(c_2, p)}} \text{need}(c_1, p) \wedge u_3(c_1, c_2, p) \text{ with } \pi(u_3(c_1, c_2, p)) := \gamma. \quad (3)$$

Further, if a country c_1 imports from a country c_2 , we observe economic growth in c_2 , denoted $\text{grows}(c_2)$ with probability δ , i.e.

$$\text{grows}(c_2) := \bigvee_{c_1 \text{ country}} \text{imp}(c_1, c_2) \wedge u_4(c_1, c_2) \text{ with } \pi(u_4(c_1, c_2)) := \delta. \quad (4)$$

Finally, if we find economic growth in country c and c is not corrupt, the living standard rises in c , denoted by $\text{live}(c)$, with probability ϵ , i.e.

$$\text{live}(c) := \text{grows}(c) \wedge (\neg \text{corrupt}(c)) \wedge u_5(c) \text{ with } \pi(u_5(c)) := \epsilon. \quad (5)$$

The equations of the form (1), (2), (3), (4) and (5) yield a functional causal model \mathcal{M} .

Next, we note that causal models do not only support queries about conditional and unconditional probabilities. They support two other more general query types, namely determining the effect of interventions and counterfactuals. Here, we focus on the treatment of external interventions.

2.1.1. Predicting the Effect of Interventions

Fix a causal model \mathcal{M} on a set of variables \mathbf{V} . Assume we are given a subset of variables $\mathbf{X} := \{X_1, \dots, X_k\} \subseteq \mathbf{V}$ together with a vector of possible values $\mathbf{x} := (x_1, \dots, x_k)$. To calculate the effect of setting the variables in \mathbf{X} to the values specified by \mathbf{x} we replace the equations for X_i in \mathcal{M} by $X_i := x_i$ for all $1 \leq i \leq k$ and calculate the desired probabilities. The resulting probability distribution is then denoted by $\pi(_ | \text{do}(\mathbf{X} = \mathbf{x}))$ [2, §1.4.3].

Example 2. If a commission roots out corruption in country c , we can model this intervention by replacing equation (4) for c in the causal model \mathcal{M} of Example 1 with $\text{corrupt}(c) := \text{False}$.

2.2. Causal Bayesian Networks and d-Separation

We briefly recall the theory of Bayesian networks, a widespread representation for probability distributions. We compare them with the functional causal models of 2.1 and obtain a causal semantics for Bayesian networks, i.e. we discuss how to determine the effect of an intervention from a Bayesian networks structure.

We begin by recalling the definition of a Bayesian network:

Definition 2 (Bayesian Network). A **Bayesian network** on a set of random variables \mathbf{V} consists of a directed acyclic graph G on \mathbf{V} and the conditional probability distributions $\pi(X | \text{Pa}(X))$ of the random variables $X \in \mathbf{V}$ conditioned on the set $\text{Pa}(X)$ of their parents in the graph G . It gives rise to a probability distribution on $\mathbf{V} = \{X_1, \dots, X_k\}$ by setting

$$\pi(X_1 = x_1, \dots, X_k = x_k) = \prod_{i=1}^k \pi(X_i = x_i | \text{pa}(X_i)),$$

where $\text{pa}(X_i) := \{X_j = x_j | X_j \in \text{Pa}(X_i)\}$.

We proceed by recalling the relation between Bayesian networks and causal models.

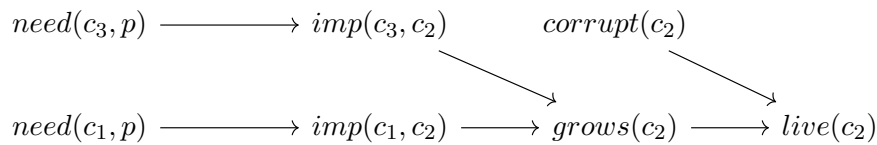
Definition 3 (Acyclic & Markovian Causal Models). For each causal model \mathcal{M} on \mathbf{V} we define the **causal diagram** $\text{graph}(\mathcal{M})$ to be the directed graph on \mathbf{V} , which is given by drawing an arrow $X \rightarrow Y$ if and only if $X \in \text{Pa}(Y)$. Further, we call \mathcal{M} an **acyclic** causal model if its causal diagram $\text{graph}(\mathcal{M})$ is a directed acyclic graph. Finally, an acyclic causal model is said to be **Markovian** if the error terms $\text{Error}(X)$ are mutually independent for all $X \in \mathbf{V}$.

The next theorem describes how causal models and Bayesian networks are related to each other [2, 1.4.2]:

Theorem 1. Every Markovian causal model \mathcal{M} gives rise to a probability distribution that can be stored in a Bayesian network $\mathcal{B}(\mathcal{M})$ on its causal diagram.

Theorem 1 yields a way to determine effects of interventions from a Bayesian network $\mathcal{B}(\mathcal{M})$ corresponding to a Markovian causal model \mathcal{M} . To calculate the effect of setting $\mathbf{X} := \{X_1, \dots, X_k\}$ to $\mathbf{x} := (x_1, \dots, x_k)$ one can modify $\mathcal{B}(\mathcal{M})$ by adjusting the conditional distributions $\pi(X_i | \text{Pa}(X_i))$ to $\pi(X_i | \text{pa}(X_i)) = \delta_{X_i, x_i}$, where δ_{X_i, x_i} denotes the Kronecker delta. The resulting structure is then called the **causal Bayesian network** $\mathcal{CB}(\mathcal{M})$ [2, §1.3]. In other words a causal Bayesian network is a Bayesian network that supports the calculation of effects of interventions.

Example 3. Assume that the error terms $u_1(-, -)$, $u_2(-)$, $u_3(-, -, -)$, $u_4(-, -)$ and $u_5(-)$ are mutually independent, i.e. the causal model \mathcal{M} of Example 1 is Markovian. Hence, for three countries c_1, c_2, c_3 and one product p with $\text{prod}(c_2, p)$ and not $\text{prod}(c_i, p)$ for $i = 1, 2$. We obtain by Theorem 1 that omitting isolated points the resulting distribution can be stored in a causal Bayesian network on the graph



Finally, we state a modularity result for causal models, whose proof is given in the Appendix:

Theorem 2 (Modularity of Causal Models). *Let \mathcal{M}_1 and \mathcal{M}_2 be functional causal models that induce the same distribution on the set of variables \mathbf{V} and let $X \in \mathbf{V}$ be a variable. Furthermore, let us assume that the set $\text{Pa}(X)$ of the parents of X coincide for both causal diagrams $\text{graph}(\mathcal{M}_1)$ and $\text{graph}(\mathcal{M}_2)$. If we construct the causal model \mathcal{M}_3 by exchanging the defining equation for X in \mathcal{M}_1 by the defining equation for X in \mathcal{M}_2 , we obtain that the causal Bayesian networks $\text{CB}(\mathcal{M}_1)$ and $\text{CB}(\mathcal{M}_3)$ coincide.*

2.2.1. d-Separation

The graph underlying a Bayesian network gives rise to a correct conditional independence oracle in the following way [3]:

Definition 4 (d-Connecting Path). *Let G be a directed acyclic graph. A subgraph of the form $A \rightarrow C \leftarrow B$ is called a **collider** at C . We say that a collider is **shielded** if there exists an edge $C - B$. Otherwise, a collider is said to be **unshielded**.*

*Further, let \mathbf{Z} be a set of nodes. A **d-connecting path** $P := R_1 - \dots - R_n$ is an undirected path in G such that each non-collider N of the path P does not lie in \mathbf{Z} and such that for every collider at C of the path P there exists a directed path from C to a node in \mathbf{Z} .*

*Moreover, we say that \mathbf{Z} **d-connects** two nodes A and B if there exists a d-connecting path between A and B with respect to \mathbf{Z} . Otherwise, we say that \mathbf{Z} **d-separates** A and B . Finally, two sets of nodes \mathbf{A} and \mathbf{B} are said to be **d-separated** by \mathbf{Z} if \mathbf{Z} d-separates A and B for every $A \in \mathbf{A}$ and every $B \in \mathbf{B}$. Otherwise, the sets \mathbf{A} and \mathbf{B} are **d-connected** by \mathbf{Z} .*

Remark 1. *The term “d-connected” is a shorthand for “directionally connected” [4].*

Theorem 3 (Correctness of d-Separation). *Let G be a directed acyclic graph on the set \mathbf{V} . Further, let $\mathbf{Z} \subseteq \mathbf{V}$ be a subset of nodes and let $A, B \in \mathbf{V}$ be nodes of G . If \mathbf{Z} d-separates A and B , we obtain that A and B are independent conditioned on \mathbf{Z} in every distribution that can be represented by a Bayesian network on G .*

Note that in general d-separation does not yield a complete independence oracle [5, 1.4]. This motivates the following definition:

Definition 5 (Causal Faithfulness). *Let π be a distribution on a set of random variables \mathbf{V} and let G be a directed acyclic graph on these random variables \mathbf{V} . We say that the distribution π is (**causally**) **faithful** to the graph G in the case where two random variables $A, B \in \mathbf{V}$ are conditionally independent with respect to a subset of random variables $\mathbf{Z} \subseteq \mathbf{V}$ if and only if A and B are d-separated by \mathbf{Z} in G .*

However, the following theorem shows that faithfulness holds for almost all discrete Bayesian networks [6]:

Theorem 4 (Obstruction to Causal Faithfulness). *Let G be a directed acyclic graph and let $\theta \in [0, 1]^n$ be a vector which determines the conditional distributions that turn G into a discrete Bayesian network \mathcal{B} representing the distribution π . In this case we obtain finitely many non-trivial polynomial equations such that π is faithful to G unless θ solves one of these equations.*

For an in depth discussion of causal faithfulness we refer the reader to [7].

2.2.2. Causal Structure Discovery for Bayesian Networks

At this place we want to discuss how we can learn a causal Bayesian network, i.e. a Bayesian network that allows us to predict the effect of interventions, from observational data. Since we have only observational data, the best we can do is to learn a Bayesian network that represents the right distribution. However, we want to find the Bayesian network with the right causal semantics!

More precisely, we want to find all possible causal explanations for the observed distribution. We consider this as an inverse problem and to tackle this inverse problem we assume that the causal Bayesian network that generated our data satisfies the causal faithfulness assumption of Definition 5. In the light of Theorem 4 this doesn't seem to be too restrictive. Furthermore, the following fact is known [8]:

Theorem 5. *Let G and G' be two directed acyclic graphs on a set of random variables V . In this case the following statements are equivalent:*

- i) *Each distribution π that is faithful to G is also faithful to G' .*
- ii) *The graphs G and G' share the same adjacencies and the same unshielded colliders.*

Theorem 5 forms the theoretical basis for causal structure discovery algorithms such as the PC-algorithm [9] and its generalizations.

Example 4. *It is easy to see that in Example 3 the causal relationship between $need(_, _)$ and $imp(_, _)$ cannot be identified.*

2.3. Probabilistic Logic Programming under the FCM-Semantics

Let us fix a **query language**, i.e. a language $\mathcal{Q} \supseteq \mathcal{L} \supseteq \mathcal{E}$ in three parts with an **extensional vocabulary** \mathcal{E} and a **logical vocabulary** \mathcal{L} . Here \mathcal{Q} is a finite multisorted relational vocabulary, i.e. it consists of a finite set of sorts, a finite set of relation symbols and a finite set of constants in those sorts, as well as a countably infinite set of variables in every sort. For every variable or constant x we will denote its sort by $s(x)$. Further, \mathcal{L} is a subvocabulary of \mathcal{Q} that contains all of the variables and constants of \mathcal{Q} as well as a (possibly empty) subset of the relation symbols of \mathcal{Q} . Moreover, \mathcal{E} is a subvocabulary of \mathcal{L} that satisfies the same properties with respect to \mathcal{L} as \mathcal{L} does with respect to \mathcal{Q} .

Example 5. *In order to model Example 1 we need two sorts: countries and products. Further, we model $corrupt(_)$, $need(_, _)$, $imp(_, _)$, $grows(_)$ and $live(_)$ as random predicates and $prod(_, _)$ as an external predicate. Later, we will also see an example for an internal predicate.*

As usual, an **atom** is an expression of the form $r(t_1, \dots, t_n)$, where r is a relation symbol and t_1 to t_n are constants or variables of the correct sorts and a **literal** is an expression of the form A or $\neg A$ for an atom A . It is called a **external atom** or **literal** if r is in \mathcal{E} , **logical atom** or **literal** if r is in \mathcal{L} , **internal atom** or **literal** if r lies in $\mathcal{L} \setminus \mathcal{E}$ and a **random atom** or **literal** if r is in $\mathcal{Q} \setminus \mathcal{L}$. A literal of the form A is called **positive** and a literal of the form $\neg A$ is called **negative**. Moreover, we will use the operator $\text{var}(_)$ to refer to the variables in an expression.

The logical vocabulary will be used to formulate constraints and conditions for our probabilistic logic program. The purpose of a probabilistic logic program, however, is to define distributions for the random variables, determined by the language \mathcal{L} . This is done by so called random clauses, i.e. LPAD-clauses with one random atom in the head.

Definition 6 (Random Clause). A **random clause** RC is an expression of the form

$$R : \pi(RC) \leftarrow R_1, \dots, R_m, L_1, \dots, L_n,$$

where $R := \text{effect}(RC)$ is a random atom, called the **effect** of RC , $\text{causes}(RC) := \{R_1, \dots, R_m\}$ is a finite set of random literals, called the **causes** of RC , $\text{cond}(RC) := \{L_1, \dots, L_n\}$ is a finite set of logical atoms, called the **condition** of RC and $\pi(RC) \in (0, 1]$ is a number called the probability of RC . In this case we also denote the random clause RC by

$$\text{effect}(RC) : \pi(RC) \leftarrow \text{causes}(RC) \cup \text{cond}(RC).$$

Having established the necessary syntax, we introduce the corresponding semantics. The semantics of logical expressions are given in a straightforward way; we just want to highlight the unique names assumption in our definition of a structure:

An \mathcal{L} -**structure** Λ consists of a sort universe \mathfrak{s}^Λ for every sort \mathfrak{s} , an element of the corresponding sort universe for every constant in \mathcal{L} , such that two different constants are mapped to different elements, and a relation among the corresponding sort universes for every relation in \mathcal{L} .

Whether a logical formula is **satisfied** by a given \mathcal{L} -structure (under an interpretation of its free variables) is determined by the usual rules of first-order logic. Finally, note that the semantics of external expressions is defined analogously.

Lifted Programs

We start with the definition of a lifted program:

Definition 7 (Lifted Program & Program Structure). A (**lifted**) **program** $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$ is a triple that consists of a finite set of normal clauses \mathbf{I} of the form $H \leftarrow B_1, \dots, B_m$ with logical literals B_1, \dots, B_m and an internal atom H , integrity constraints \mathbf{C} of the form $\perp \leftarrow L_1, \dots, L_k$ for logical literals L_i with $1 \leq i \leq k$, as well as a finite set of random clauses \mathbf{R} .

We call \mathbf{C} the **constraints**, \mathbf{I} the **internal part** and \mathbf{R} the **random part** of \mathbf{P} . Further, we say that \mathbf{P} is **stratified** if its internal part \mathbf{I} is a stratified set of normal clauses.

Example 6. We model the causal mechanism in Example 1 by a program \mathbf{P} , given by the random part:

$$\text{need}(C, P) : \alpha$$

$$\text{corrupt}(C) : \beta$$

$$\text{im}(C_1, C_2) : \gamma \leftarrow \text{need}(C_1, P), \text{neg_prod}(C_1, P), \text{prod}(C_2, P)$$

$$\text{grows}(C_2) : \delta \leftarrow \text{imp}(C_1), \text{imp}(C_2)$$

$$\text{live}(C) : \epsilon \leftarrow \text{grow}(C), \neg \text{corrupt}(C)$$

and the internal part: $\text{neg_prod}(C, P) \leftarrow \neg \text{prod}(C, P)$ for an internal predicate $\text{neg_prod}(_, _)$.

Note that the database \mathcal{E} described in Example 3 yields an external database for \mathbf{P} .

Notation 1. Let $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$ be a stratified program and let \mathcal{E} be an \mathfrak{E} -structure. In our setting we may w.l.o.g. assume that \mathcal{E} is a Herbrand model of an alphabet \mathfrak{E}^* , extending the external alphabet \mathfrak{E} by constants. Further, denote by \mathfrak{L}^* and \mathfrak{Q}^* the extension of the languages \mathfrak{L} and \mathfrak{Q} by the new constants in \mathfrak{E}^* . We write $\mathcal{E}^{\mathbf{I}} := \{L \text{ ground atom of } \mathfrak{L}^* : \mathbf{I} \cup \mathcal{E} \models L\}$ for the minimal Herbrand model of $\mathbf{I} \cup \mathcal{E}$, which is the result of applying the stratified datalog program \mathbf{I} to \mathcal{E} .

An \mathfrak{E} -structure \mathcal{E} is said to be an **external database** for the program \mathbf{P} if it satisfies all integrity constraints in \mathbf{C} , i.e. if $\mathcal{E}^{\mathbf{I}} \models \mathbf{C}$. A **ground variable** is a ground random atom $G := r(x_1, \dots, x_n)$ of \mathfrak{Q}^* . Finally, we write $\mathcal{G}(\mathcal{E})$ for the set of all ground variables.

The term ground variable indicates that we expect G to become a proper random variable under our semantics. From now on we will assume every program to be stratified.

Definition 8 (FCM-Semantics of lifted Programs). Let $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$ be a program and let \mathcal{E} be an external database for \mathbf{P} . The **grounding** $\mathbf{P}^{\mathcal{E}}$ of the program \mathbf{P} with respect to \mathcal{E} is the Boolean functional causal model on the set of ground variables $\mathcal{G}(\mathcal{E})$ given by the following defining equation for every ground variable $G \in \mathcal{G}(\mathcal{E})$:

$$G := \bigvee_{\substack{RC \in \mathbf{R}(\mathbf{P}) \\ \iota \text{ interpretation on } \text{var}(RC) \\ \text{effect}(RC)^\iota = G \\ (\mathcal{E}^{\mathbf{I}}, \iota) \models \text{cond}(RC)}} \left(\bigwedge_{C \in \text{causes}(RC)} C^\iota \wedge u(RC, \iota) \right)$$

Here, the **error term** $u(RC, \iota)$ is a distinct Boolean random variable with the distribution

$$\pi(u(RC, \iota)) = \pi(RC)$$

for every random clause $RC \in \mathbf{R}$ and every variable interpretation ι on $\text{var}(RC)$. Besides, the error terms $u(RC, \iota)$ are assumed to be mutually independent.

Example 7. As intended, the causal model \mathcal{M} of Example 1 yields the semantics of the program \mathbf{P} in Example 6 with respect to the described database \mathcal{E} .

Definition 9 (Ground Graph & Acyclicity). Let \mathcal{E} be an external database for the program $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$. Then the **ground graph** $\text{Graph}_{\mathcal{E}}(\mathbf{P})$ is the directed graph on the set of ground variables $\mathcal{G}(\mathcal{E})$ with an edge $G_1 \rightarrow G_2$ if and only if there exists a random clause $RC \in \mathbf{R}$, a cause $C \in \text{causes}(RC)$ and a variable interpretation ι on $\text{var}(RC)$ such that $(\mathcal{E}^{\mathbf{I}}, \iota) \models \text{cond}(RC)$, $C^\iota \in \{G_1, \neg G_1\}$ and such that $\text{effect}(RC)^\iota = G_2$. In this case we say that the edge $G_2 \leftarrow G_1$ is **induced** by RC . Moreover, we call \mathbf{P} an **acyclic** program if $\text{Graph}_{\mathcal{E}}(\mathbf{P})$ is a directed acyclic graph for every external database \mathcal{E} .

Example 8. The graph in Example 3 is the ground graph of the program \mathbf{P} in Example 6 with respect to the described database \mathcal{E} . Further, it is easy to see that \mathbf{P} yields an acyclic program.

From now on we assume every program $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$ to be acyclic. Then for every external database \mathcal{E} the grounding $\mathbf{P}^{\mathcal{E}}$ yields unique expressions for every ground variable $G \in \mathcal{G}(\mathcal{E})$ in terms of the error terms $u(RC, \kappa)$. Hence, it induces a unique probability distribution on $\mathcal{G}(\mathcal{E})$.

We will show in a forthcoming paper that this distribution coincides with the distribution induced by the LPAD-program $\mathbf{P} \cup \mathcal{E}$ via the distribution semantics [10, Chapter 1 §2.1]. All in all Definition 7 yields a generalization of the standard semantics for acyclic LPAD-programs with interesting properties regarding causation. Further, since we observe that $\text{Graph}_{\mathcal{E}}(\mathbf{P})$ is the causal diagram of $\mathbf{P}^{\mathcal{E}}$, we obtain:

Proposition 1. *Let $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$ be a program. For every external database \mathcal{E} the grounding $\mathbf{P}^{\mathcal{E}}$ yields a distribution, which can be stored in a Boolean Bayesian network on the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P})$. This is also the causal Bayesian network corresponding to the functional causal model $\mathbf{P}^{\mathcal{E}}$.*

We call the Bayesian network $\text{CB}(\mathbf{P}, \mathcal{E})$ of Proposition 1 the **causal Bayesian network semantics** of \mathbf{P} with respect to \mathcal{E} . In our future paper we will also show that our semantics is consistent with CP-logic [11]. Finally, note that Pearl’s do-calculus is implemented in cplint with respect to the CP-logic semantics [12].

3. Verifying the Causal Content of Lifted Programs

The proofs of all further statements can be found in the proof appendix. We now return to the task of verifying the causal content of a probabilistic logic program whose structure was derived from observational data. More formally, we investigate the following situation:

Problem 1. *Suppose we are given a query alphabet $\Omega \supseteq \mathcal{L} \supseteq \mathcal{E}$, a stratified datalog program \mathbf{I} , a set of integrity constraints \mathbf{C} and an \mathcal{E} -structure \mathcal{E} with $\mathcal{E}^{\mathbf{I}} \models \mathbf{C}$. Further, assume we are given data, which is sampled from a functional causal model $\tilde{\mathbf{P}}^{\mathcal{E}}$, where $\tilde{\mathbf{P}} = (\tilde{\mathbf{R}}, \mathbf{I}, \mathbf{C})$ is a program in the query alphabet Ω .*

From a structure learning algorithm like slipcover [1] we obtain a set of random clauses \mathbf{R} and hence another program $\mathbf{P} = (\mathbf{R}, \mathbf{I}, \mathbf{C})$ together with a functional causal model $\mathbf{P}^{\mathcal{E}}$. Can we use the program \mathbf{P} to predict the effect of external interventions if we assume that $\mathbf{P}^{\mathcal{E}}$ induces the same distribution as $\tilde{\mathbf{P}}^{\mathcal{E}}$?

As we are mainly interested in the causal Bayesian network semantics of the program $\tilde{\mathbf{P}}$ we introduce the following equivalence relation:

Definition 10. *Let \mathcal{E} be an external database for the programs $\mathbf{P}_1 = (\mathbf{R}_1, \mathbf{I}, \mathbf{C})$ and $\mathbf{P}_2 = (\mathbf{R}_2, \mathbf{I}, \mathbf{C})$. We call \mathbf{P}_1 and \mathbf{P}_2 **intervention equivalent** on \mathcal{E} if they induce the same causal Bayesian network semantics on \mathcal{E} , i.e. if the causal Bayesian networks $\text{CB}(\mathbf{P}_1, \mathcal{E})$ and $\text{CB}(\mathbf{P}_2, \mathcal{E})$ coincide. Note that this is the case if and only if the causal models $\mathbf{P}_1^{\mathcal{E}}$ and $\mathbf{P}_2^{\mathcal{E}}$ predict the same effects for every intervention.*

To apply Theorem 5, we need to make the following assumption.

Definition 11 (Faithful Programs). *Let $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$ be a program and let \mathcal{E} be an external database. We say that \mathbf{P} is **faithful** for \mathcal{E} if the distribution induced by $\mathbf{P}^{\mathcal{E}}$ is faithful to the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P})$. We say that two programs \mathbf{P}_1 and \mathbf{P}_2 are **faithfully indistinguishable** on \mathcal{E} if $\mathbf{P}_1^{\mathcal{E}}$ and $\mathbf{P}_2^{\mathcal{E}}$ induce the same distribution and if \mathbf{P}_1 and \mathbf{P}_2 are faithful for \mathcal{E} .*

Assumption 1 (Faithfulness). *In the situation of Problem 1 the programs $\tilde{\mathbf{P}}$ and \mathbf{P} are faithfully indistinguishable on the external database \mathcal{E} .*

With our notion of faithfulness in Definition 11, Theorem 5 translates to the following statement about faithfully indistinguishable programs:

Theorem 6. *Let \mathcal{E} be an external database for the programs $\mathbf{P}_1 = (\mathbf{R}_1, \mathbf{I}, \mathbf{C})$ and $\mathbf{P}_2 = (\mathbf{R}_2, \mathbf{I}, \mathbf{C})$. If \mathbf{P}_1 and \mathbf{P}_2 are faithfully indistinguishable on \mathcal{E} , the ground graphs $\text{Graph}_{\mathcal{E}}(\mathbf{P}_1)$ and $\text{Graph}_{\mathcal{E}}(\mathbf{P}_2)$ share the same adjacencies and the same unshielded colliders.*

Hence, Assumption 1 yields that the ground graphs $\text{Graph}_{\mathcal{E}}(\mathbf{P})$ and $\text{Graph}_{\mathcal{E}}(\tilde{\mathbf{P}})$ yield the same adjacencies and the same unshielded colliders. To reason about colliders instead of unshielded colliders we restrict ourselves to triangle free programs.

Definition 12 (Triangle Free Program Structures). *We call a program \mathbf{P} **triangle free** if for every external database \mathcal{E} the skeleton of the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P})$ does not contain any triangles.*

Proposition 2. *For every triangle free program $\mathbf{P} := (\mathbf{R}, \mathbf{I}, \mathbf{C})$ and every external database \mathcal{E} every collider in the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P})$ is unshielded. \square*

Assumption 2. *The programs \mathbf{P} and $\tilde{\mathbf{P}}$ in Problem 1 are triangle free.*

If in Problem 1 every edge of the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P})$ is involved in a collider, Theorem 6 and Proposition 2 imply that every program faithfully indistinguishable from \mathbf{P} yields the same ground graph on \mathcal{E} . Due to the following consequence of Theorem 2 the causal Bayesian network semantics also coincide in this case.

Lemma 1. *Let $\mathbf{P}_1 := (\mathbf{R}_1, \mathbf{I}, \mathbf{C})$ and $\mathbf{P}_2 := (\mathbf{R}_2, \mathbf{I}, \mathbf{C})$ be programs such that the functional causal models $\mathbf{P}_1^{\mathcal{E}}$ and $\mathbf{P}_2^{\mathcal{E}}$ induce the same distribution for an external database \mathcal{E} . In this case \mathbf{P}_1 and \mathbf{P}_2 are intervention equivalent on \mathcal{E} if the ground graphs $\text{Graph}_{\mathcal{E}}(\mathbf{P}_1)$ and $\text{Graph}_{\mathcal{E}}(\mathbf{P}_2)$ coincide. \square*

Assertions i)-iii) of the following lemma list the conditions under which a given random clause induces a collider in the ground graph of an external database \mathcal{E} . Moreover, the symmetry represented by the query alphabet \mathfrak{Q} implies that all edges induced by a random clause satisfying Assertion iv) also appear in every ground graph of a program faithfully indistinguishable from \mathbf{P} . Note that Assertion iv) yields an analogue of the relational orientation rule in the RPC-Algorithm [13].

Lemma 2. *Let $\mathbf{P}_1 := (\mathbf{R}_1, \mathbf{I}, \mathbf{C})$ and $\mathbf{P}_2 := (\mathbf{R}_2, \mathbf{I}, \mathbf{C})$ be triangle free programs that are faithfully indistinguishable on the external database \mathcal{E} . Let $RC \in \mathbf{R}_1$ be a random clause that induces an edge $C^{\iota} \rightarrow \text{effect}(RC)^{\iota}$ in the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P}_1)$ for an interpretation ι on $\text{var}(RC)$. Moreover, assume that the random clause $RC \in \mathbf{R}_1$ and the interpretation ι on $\text{var}(RC)$ satisfy one of the following assertions:*

- i) *We find two causes $C_1, C_2 \in \text{causes}(RC)$ such that $C_1^{\iota} \neq C_2^{\iota}$.*
- ii) *There exists a second random clause $RC' \in \mathbf{R}_1$ together with an interpretation ι' on $\text{var}(RC')$ satisfying $(\mathcal{E}^{\mathbf{I}}, \iota') \models \text{cond}(RC')$ such that we find $\text{effect}(RC)^{\iota} = \text{effect}(RC')^{\iota'}$ and $C^{\iota} \neq (C')^{\iota'}$ for two causes $C \in \text{causes}(RC)$ and $C' \in \text{causes}(RC')$.*

- iii) There exists a cause $C \in \text{causes}(RC)$ together with a variable $X \in \text{var}(C)$ such that $X \notin \text{var}(\text{effect}(RC))$. Moreover, we find another interpretation ι' on $\text{var}(RC)$ satisfying $(\mathcal{E}^I, \iota') \models \text{cond}(RC)$, $\text{effect}(RC)^\iota = \text{effect}(RC)^{\iota'}$ and $\iota(X) \neq \iota'(X)$.
- iv) There is only one cause $C \in \text{causes}(RC)$ in RC and there exists a variable $X \in \text{var}(\text{effect}(RC))$ such that $X \notin \text{var}(C)$. Moreover, there exists an interpretation ι' on $\text{var}(RC)$ of the same type in \mathfrak{L} as ι , i.e. ι makes an existential formula ϕ true if and only if ι' does, such that $C^\iota = C^{\iota'}$ and such that $\iota(X) \neq \iota'(X)$.

Then the edge $C^\iota \rightarrow \text{effect}(RC)^\iota$ appears in the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P}_2)$ as well.

Combining Theorem 6, Lemma 1 and Lemma 2 we get the following corollary:

Theorem 7. Let $\mathbf{P}_1 := (\mathbf{R}_1, \mathbf{I}, \mathbf{C})$ and $\mathbf{P}_2 := (\mathbf{R}_2, \mathbf{I}, \mathbf{C})$ be triangle free programs that are faithfully indistinguishable on the external database \mathcal{E} . Further, assume that every random clause $RC \in \mathbf{R}_1$ with $\text{causes}(RC) \neq \emptyset$ and every interpretation ι on $\text{var}(RC)$ with $(\mathcal{E}, \iota) \models \text{cond}(RC)$ satisfies at least one of assertions i)-iv) in Lemma 2. In this case \mathbf{P}_1 and \mathbf{P}_2 are intervention equivalent on the external database \mathcal{E} . \square

Example 9. Assume we obtained the program \mathbf{P} as described in Problem 1 and Assumption 1 from a structure learning algorithm on the database \mathcal{E} in Example 8. In this case we cannot verify the causal content of \mathbf{P} ! However, if we would add a country producing p or a country not producing p , the program \mathbf{P} would be guaranteed to be intervention equivalent to $\tilde{\mathbf{P}}$ as desired.

Theorem 7 yields a partial answer for Problem 1 if we want to predict effects of external interventions in the functional causal model $\tilde{\mathbf{P}}^{\mathcal{E}}$. Next, we want to generalize our result to other external databases. To this aim we introduce the following assumption:

Assumption 3 (Generalization). In Problem 1 we additionally assume that the query alphabet Ω contains no constants and that the programs $\tilde{\mathbf{P}}$ and \mathbf{P} are faithfully indistinguishable on every external database \mathcal{E}' .

Next, we modify Assertions i),iii) and iv) of Lemma 2 to guarantee that under Assumption 3 every edge in the ground graph $\text{Graph}_{\mathcal{E}'}(\mathbf{P})$ of an arbitrary external database \mathcal{E}' also occurs in the ground graph $\text{Graph}_{\mathcal{E}'}(\tilde{\mathbf{P}})$:

Lemma 3. Let $\mathbf{P}_1 := (\mathbf{R}_1, \mathbf{I}, \mathbf{C})$ and $\mathbf{P}_2 := (\mathbf{R}_2, \mathbf{I}, \mathbf{C})$ be triangle free programs that are faithfully indistinguishable on every external database \mathcal{E} . Further, assume that no variable appears twice in the head of a clause in \mathbf{I} .

Moreover, let $RC \in \mathbf{R}$ be a random clause that induces an edge $C^{\iota'} \rightarrow \text{effect}(RC)^{\iota'}$ in the ground graph $\text{Graph}_{\mathcal{E}'}(\mathbf{P}_1)$ of an external database \mathcal{E}' for an interpretation ι' on $\text{var}(RC)$ in \mathcal{E}' . Furthermore, assume that the random clause $RC \in \mathbf{R}_1$ satisfies at least one of the following assertions:

- i) We find two causes $C_1, C_2 \in \text{causes}(RC)$ that are not sharing the same underlying predicate.
- ii) We find a cause $C \in \text{causes}(RC)$ and a variable $X \in \text{var}(C)$ such that $X \notin \text{var}(\text{effect}(RC))$. Moreover, C and $\text{effect}(RC)$ do not share the same underlying predicate.

iii) We find a cause $C \in \text{causes}(RC)$ and variable $X \in \text{var}(\text{effect}(RC))$ such that $X \notin \text{var}(C)$.
 Moreover, C and $\text{effect}(RC)$ do not share the same underlying predicate.

Then we obtain that the edge $C' \rightarrow \text{effect}(RC)'$ is contained in the ground graph $\text{Graph}_{\mathcal{E}'}(\mathbf{P}_2)$ as well.

The proof of Lemma 3 requires the construction of external databases, in which causes are witnessed by more than one individual. Adding extra individuals is enabled by not having any kind of privileged equality, which is reflected in the following assumption:

Assumption 4. *In the situation of Problem 1 no variable appears twice in the head of a clause in I .*

Finally, we combine Theorem 6, Lemma 3 and Lemma 2:

Theorem 8. *In the situation of Lemma 3 we assume that every random clause in \mathbf{P}_1 satisfies one of the Assertions i)-iii). In this case the programs \mathbf{P}_1 and \mathbf{P}_2 are intervention equivalent on every external database \mathcal{E} . \square*

Example 10. *Assume we obtained the program \mathbf{P} as described in Problem 1 and Assumption 3 from a structure discovery algorithm on the database \mathcal{E} in Example 8. In this case we can invoke Theorem 8 to see that \mathbf{P} and $\tilde{\mathbf{P}}$ are intervention equivalent on every external database \mathcal{E}' .*

Based on our results we propose the following procedure to verify the causal content of a given probabilistic logic program.

Procedure 1. *In the situation of Problem 1 we first check that every random clause RC of program \mathbf{P} and every interpretation ι on $\text{var}(RC)$ in the external database \mathcal{E} satisfies one of the Assertions i)-iv) of Lemma 2. In this case we obtain that $\tilde{\mathbf{P}}$ and \mathbf{P} are intervention equivalent on \mathcal{E} under Assumption 2 and Assumption 1. Further, we check for every random clause RC of \mathbf{P}_1 whether there is an interpretation ι on $\text{var}(RC)$ in \mathcal{E} such that $(\mathcal{E}^I, \iota) \models \text{cond}(RC)$.*

Next, we use this result to justify Assumption 3, i.e. to justify that no random clause is verified on the basis of a collider not observed in the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P})$. Finally, we check whether every random clause in \mathbf{P} satisfies one of the Assertions i)-iii) of Lemma 3 and invoke Theorem 8 to conclude that under Assumption 4 the programs \mathbf{P} and $\tilde{\mathbf{P}}$ are intervention equivalent on every external database \mathcal{E}' as desired.

4. Conclusion

We first illustrated that in the lifted case, the search space of causal explanations for an observed distribution can be infinite. Hence, there is no straightforward generalization of causal structure discovery here. However, we developed a procedure to verify the causal content of an acyclic, stratified, triangle free probabilistic logic program, whose structure was determined by observational data. Intuitively, we argued that we can verify every clause that observably implies only colliders in the associated ground graphs.

In future we would like to remove the assumption of triangle freeness. Furthermore, we want to verify pairs of clauses that induce colliders in the ground graph for every external database. To obtain a complete result we also aim to incorporate the orientation rules of the PC-algorithm into our framework.

Acknowledgement

The research leading to this publication was supported by LMUexcellent, funded by the Federal Ministry of Education and Research (BMBF) and the Free State of Bavaria under the Excellence Strategy of the Federal Government and the Länder.

References

- [1] E. Bellodi, F. Riguzzi, Structure learning of probabilistic logic programs by searching the clause space, *Theory and Practice of Logic Programming* 15 (2014) 169–212. doi:10.1017/s1471068413000689.
- [2] J. Pearl, *Causality*, 2 ed., Cambridge University Press, Cambridge, UK, 2000. doi:10.1017/CBO9780511803161.
- [3] T. Verma, J. Pearl, Causal networks: Semantics and expressiveness, in: *Uncertainty in Artificial Intelligence*, volume 9 of *Machine Intelligence and Pattern Recognition*, North-Holland, 1990, pp. 69–76. doi:10.1016/B978-0-444-88650-7.50011-1.
- [4] D. Geiger, T. Verma, J. Pearl, Identifying independence in Bayesian networks, *Networks* 20 (1990) 507–534. doi:10.1002/net.3230200504.
- [5] M. McDermott, Redundant causation, *The British Journal for the Philosophy of Science* 46 (1995) 523–544. URL: <http://www.jstor.org/stable/687896>.
- [6] C. Meek, Strong completeness and faithfulness in Bayesian networks, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI'95*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, p. 411–418.
- [7] N. Weinberger, Faithfulness, coordination and causal coincidences, *Erkenntnis* 83 (2018) 113–133. doi:10.1007/s10670-017-9882-6.
- [8] T. Verma, J. Pearl, Equivalence and synthesis of causal models, in: *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence, UAI '90*, Elsevier Science Inc., USA, 1990, p. 255–270. doi:10.5555/647233.719736.
- [9] C. Meek, Causal inference and causal explanation with background knowledge, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI'95*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, p. 403–410. doi:10.5555/2074158.2074204.
- [10] F. Riguzzi, *Foundations of Probabilistic Logic Programming: Languages, Semantics, Inference and Learning*, River Publishers, 2020.
- [11] J. Vennekens, M. Denecker, M. Bruynooghe, CP-logic: A language of causal probabilistic events and its relation to logic programming, *Theory and Practice of Logic Programming* 9 (2009) 245–308.
- [12] F. Riguzzi, G. Cota, E. Bellodi, R. Zese, Causal inference in cplint, *International Journal of Approximate Reasoning* 91 (2017) 216–232. doi:10.1016/j.ijar.2017.09.007.
- [13] M. Maier, *Causal Discovery for Relational Domains: Representation, Reasoning, and Learning*, Doctoral dissertations. 279., University of Massachusetts - Amherst, 2014. URL: https://scholarworks.umass.edu/dissertations_2/279/.

Proof Appendix

Proof of Theorem 2. Recall Definition 3 to see that X has the same parents in the causal diagrams $\text{graph}(\mathcal{M}_2)$ and $\text{graph}(\mathcal{M}_3)$ and that every other node $Y \neq X$ has the same parents in $\text{graph}(\mathcal{M}_3)$ as in $\text{graph}(\mathcal{M}_1)$. Furthermore, apply the assumption to obtain that the causal diagram $\text{graph}(\mathcal{M}_3)$ coincides with $\text{graph}(\mathcal{M}_1)$.

Hence, we are left to show that the parameters in the causal Bayesian networks $\mathcal{CB}(\mathcal{M}_1)$ and $\mathcal{CB}(\mathcal{M}_3)$ coincide as well. As \mathcal{M}_1 and \mathcal{M}_2 induce the same distribution, we find that the parameters for X coincide in the causal Bayesian networks $\mathcal{CB}(\mathcal{M}_1)$ and $\mathcal{CB}(\mathcal{M}_2)$. To conclude we recall that the parameters of every node $Y \in \mathbf{V}$ in $\mathcal{CB}(\mathcal{M}_{1/2/3})$ can be computed from the corresponding defining equation by intervening on the set of parents $\text{Pa}(Y)$ [2, Theorem 3.3.2]. \square

Proof of Theorem 6. Assume that the functional causal models $\mathbf{P}_1^{\mathcal{E}}$ and $\mathbf{P}_2^{\mathcal{E}}$ induce the same distribution π . Further, assume that both programs \mathbf{P}_1 and \mathbf{P}_2 are faithful for \mathcal{E} . In this case by Definition 11 we see that the ground graphs $\text{Graph}_{\mathcal{E}}(\mathbf{P}_1)$ and $\text{Graph}_{\mathcal{E}}(\mathbf{P}_2)$ are both faithful to the distribution π . Hence, we obtain the desired result from Theorem 5. \square

Proof of Lemma 2. If RC satisfies one of the assertions i)-iii), it is easy to see that the edge $C^{\iota} \rightarrow \text{effect}(RC)^{\iota}$ is involved in a collider at $\text{effect}(RC)^{\iota}$ and the claim follows since by Theorem 6 and Proposition 2 the ground graphs $\text{Graph}_{\mathcal{E}}(\mathbf{P}_1)$ and $\text{Graph}_{\mathcal{E}}(\mathbf{P}_2)$ share the same colliders.

Next, assume that RC and ι satisfy iv). By assumption there exists an interpretation ι' on $\text{var}(RC)$ of the same type in \mathcal{L} as ι such that $C^{\iota} = C^{\iota'}$ and such that $\iota(X) \neq \iota'(X)$ for a variable $X \in \text{var}(\text{effect}(RC))$. As ι and ι' have the same type in \mathcal{L} we find that the edges $C^{\iota} - \text{effect}(RC)^{\iota}$ and $C^{\iota'} - \text{effect}(RC)^{\iota'}$ also share the same orientation in the ground graph $\text{Graph}_{\mathcal{E}}(\mathbf{P}_2)$. If they were both pointing to C^{ι} , we would observe a collider in $\text{Graph}_{\mathcal{E}}(\mathbf{P}_2)$ not contained $\text{Graph}_{\mathcal{E}}(\mathbf{P}_1)$. This is a contradiction to Theorem 6 and Proposition 2 and the claim follows. \square

To prove Lemma 3 we need the following lemma:

Lemma 4. *Let $\mathbf{P} = (\mathbf{R}, \mathbf{I}, \mathbf{C})$ be a lifted program such that no variable appears twice in the head of a clause in \mathbf{I} and let \mathcal{E} be an external database. Assume the following holds for every external atom $A = l(x_1, \dots, x_m)$:*

- (*) *Let I be a subset of $\{1, \dots, m\}$ and let ι be an interpretation on $\text{var}(A)$ with $\iota(x_i) = a$ if and only if $i \in I$. Moreover, let a' be an individual in \mathcal{E} and let ι' be the interpretation on $\text{var}(A)$, which is given by setting $\iota(y_i) = a'$ for $i \in I$ and $\iota'(x_j) = \iota(x_j)$ for $j \notin I$. In this case $(\mathcal{E}, \iota) \models A$ if and only if $(\mathcal{E}, \iota') \models A$.*

Then () also holds for every internal atom A .*

Proof. Let \mathcal{T} denote the fixpoint operator of the stratified datalog program \mathbf{I} . We use induction on the iteration of \mathcal{T} to show that for every $n \in \mathbb{N}$ we have $(\mathcal{T}^n(\mathcal{E}), \iota) \models A$ if and only if $(\mathcal{T}^n(\mathcal{E}), \iota') \models A$.

Note that the case $n = 0$ follows directly from the assumption of the lemma. So assume that $(\mathcal{T}^{n+1}(\mathcal{E}), \iota) \models A$. If we find $(\mathcal{T}^{n+1}(\mathcal{E}), \iota') \models A$, we are finished by the induction hypothesis. Hence, let us assume that $(\mathcal{T}^{n+1}(\mathcal{E}), \iota') \not\models A$.

In this case there exists a clause $C \in \mathbf{I}$ given by $H \leftarrow B_1, \dots, B_k$ and an interpretation ν on $\text{var}(C)$ such that $H^\nu = A^\iota$ and such that $(\mathcal{T}^n(\mathcal{E}), \nu) \models \{B_1, \dots, B_k\}$. By the assumption of the lemma, $H = l(w_1, \dots, w_m)$ where all w_1, \dots, w_m are distinct. Further, we know that $\nu(w_i) = \iota(x_i) = a$ for all $i \in I$. Define ν' by setting $\nu'(w_i) = a'$ for $i \in I$ and $\nu'(v) = \nu(v)$ for all other variables v on which ν is defined. Then the restrictions of ν and ν' to the variables in any B_i , $1 \leq i \leq k$, satisfies the requirements of the induction hypothesis. Hence, $(\mathcal{T}^n(\mathcal{E}), \nu') \models \{B_1, \dots, B_k\}$ and thus $(\mathcal{T}^{n+1}(\mathcal{E}), \nu') \models H$. Note that $H^{\nu'} = A^{\iota'}$ by construction. Therefore, $(\mathcal{T}^{n+1}(\mathcal{E}), \iota') \models A$ as required. \square

Proof of Lemma 3. We distinguish the following cases:

Case. *The clause $RC \in \mathbf{R}_1$ satisfies i).*

In this case the desired result follows immediately from Lemma 2.

Case. *The clause $RC \in \mathbf{R}_1$ satisfies ii).*

By Theorem 6 we find an edge $C^{\iota'} - \text{effect}(RC)^{\iota'}$ in the ground graph $\text{Graph}_{\mathcal{E}'}(\mathbf{P}_2)$. This edge needs now to be induced by a random clause $\tilde{RC} \in \mathbf{R}_2$, i.e. we obtain that

$$C^{\iota'} - \text{effect}(RC)^{\iota'} = \tilde{C}^{\tilde{\iota}} \rightarrow \text{effect}(\tilde{RC})^{\tilde{\iota}}.$$

Further, we may assume w.l.o.g. that the clauses RC and \tilde{RC} don't share common variables. Hence, we obtain an interpretation ι on $\text{var}(RC) \cup \text{var}(\tilde{RC})$ such that

$$((\mathcal{E}')^{\mathbf{I}}, \iota) \models \text{edge_eq} \cup \text{cond}(RC) \cup \text{cond}(\tilde{RC}),$$

where edge_eq denotes the set

$$\left\{ (\tilde{C} = \text{effect}(RC) \wedge C = \text{effect}(\tilde{RC})) \vee (\text{effect}(\tilde{RC}) = \text{effect}(RC) \wedge C = \tilde{C}) \right\}.$$

Moreover, we may assume w.l.o.g. that there exists a variable $X \in \text{var}(C)$ such that $X \notin \text{var}(\text{effect}(RC))$. Furthermore, choose a variable X' not contained in $\text{var}(RC) \cup \text{var}(\tilde{RC})$. Next, choose a constant $\iota(X')$ not contained in \mathcal{E}' of the sort $\mathfrak{s}(X)$ and add for each ground atom $p(c_1, \dots, \iota(X), \dots, c_n)$ in \mathcal{E}' the ground atom $p(c_1, \dots, \iota(X'), \dots, c_n)$. In this way we end up with an external database \mathcal{E}'' and an interpretation ι on $\text{var}(RC) \cup \text{var}(\tilde{RC}) \cup \{X'\}$. Further, Lemma 4 yields that the following set is satisfied with respect to ι and $(\mathcal{E}'')^{\mathbf{I}}$:

$$(\text{edge_eq} \cup \text{cond}(RC)) [X'/X] \cup \text{edge_eq} \cup \text{cond}(RC) \cup \text{cond}(\tilde{RC}) \cup \{X \neq X'\}$$

Hence, $RC \in \mathbf{R}_1$ satisfies the Assertion iii) of Lemma 2 with respect to the interpretation ι and the external database \mathcal{E}'' and the edge $C^{\iota'} - \text{effect}(RC)^{\iota'}$ is induced by \tilde{RC} in $\text{Graph}_{\mathcal{E}''}(\mathbf{P}_2)$ as well. From Lemma 2 we further obtain that the orientation of the edge $C^{\iota'} - \text{effect}(RC)^{\iota'}$ in $\text{Graph}_{\mathcal{E}''}(\mathbf{P}_2)$ is also given by $C^{\iota'} \rightarrow \text{effect}(RC)^{\iota'}$. Hence, we obtain from Definition 9 that the formula $\tilde{C} = \text{effect}(RC) \wedge C = \text{effect}(\tilde{RC})$ is not satisfiable with respect to $\mathbf{I} \cup \mathbf{C}$ as C and $\text{effect}(RC)$ do not share the same underlying random predicate. However, this means that $(\mathcal{E}^{\mathbf{I}}, \iota') \models \text{effect}(\tilde{RC}) = \text{effect}(RC) \wedge C = \tilde{C}$ and the desired result follows.

Case. *The clause $RC \in \mathbf{R}_1$ and the interpretation ι_2 satisfy iii) of Lemma 2.*

This can be proven analogously to the previous case.

□