International Journal of Approximate Reasoning 83 (2017) 298–319

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

On the complexity of propositional and relational credal networks ☆

Fabio Gagliardi Cozman [a], Denis Deratani Mauá [b,*]

[a] *Escola Politécnica, Universidade de São Paulo, Brazil*
[b] *Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil*

ARTICLE INFO

Article history:
Received 9 December 2015
Received in revised form 21 October 2016
Accepted 22 October 2016

Keywords:
Credal networks
Propositional logic
Function-free first-order logic
Complexity theory

ABSTRACT

A credal network associates a directed acyclic graph with a collection of sets of probability measures. Usually these probability measures are specified by tables containing probability values. Here we examine the complexity of inference in credal networks when probability measures are specified through formal languages. We focus on logical languages based on propositional logic and on the function-free fragment of first-order logic. We show that sub-Boolean and relational logics lead to interesting complexity results. In short, we explore the relationship between specification language and computational complexity in credal networks.

1. Introduction

A credal network represents a set of probability distributions through a directed acyclic graph and an associated set of "local" credal sets [2,10]. Usually these local credal sets are specified by tables containing probability values, possibly with some additional constraints between them.

In practice, any elicitation strategy must adopt some specification language in which to encode probability assessments and constraints. The simplest specification scheme is to allow inequalities such as $\mathbb{P}(A) \geq 1/2$. Interval-valued assessments such as $\mathbb{P}(A) \in [3/5, 7/10]$ are also common in the literature.

More complex modeling schemes may resort to Boolean operators, relations, and quantifiers. In that case the particular features of the specification language may greatly affect what can and what cannot be represented; also, these features may affect the computational complexity of queries we may wish to pose.

In this paper we study properties of credal networks as parameterized by specification languages, by extending a framework we have recently proposed in the context of Bayesian networks [15]. We look at the interplay between expressivity (of specification languages) and complexity (of inference). We focus on *logical* languages that are fragments of propositional and first-order logic. Additionally, we focus on *strong extensions* as semantics for credal networks. We show that complexity of inferences touches on several interesting complexity classes.

We start with some necessary background in Section 2. We discuss our framework in Section 3, in particular looking at propositional languages. Sections 4 and 5 examine relational languages. Final remarks and possible extensions to this work appear in Section 6.

☆ This paper is part of the virtual special issue on the Ninth International Conference on Scalable Uncertainty Management (SUM 2015), edited by Christoph Beierle and Alex Dekhtyar.

* Corresponding author.
E-mail addresses: fgcozman@usp.br (F. Gagliardi Cozman), denis.maua@usp.br (D. Deratani Mauá).

http://dx.doi.org/10.1016/j.ijar.2016.10.008
0888-613X/

2. Background: credal networks and their strong extensions, and some complexity theory

In this paper every possibility space Ω is finite; a random variable is simply a function from Ω into the reals. We use capital letters at the end of the alphabet (X, Y, Z) to denote random variables, and capital letters at the beginning of the alphabet (A, B) to denote events or propositions.

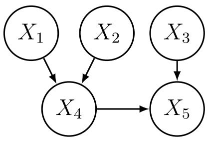
2.1. Credal sets

A set of probability measures is called a *credal set* [36]. When the possibility space Ω is finite, a set of probability distributions over random variables is completely characterized by a set of probability mass functions, each probability mass function corresponding to a distribution. We also use the term *credal set* to refer to sets of probability mass functions.

A set of probability distributions for a variable X is denoted by $\mathbb{K}(X)$. A set of probability mass functions for a variable X is also denoted by $\mathbb{K}(X)$. Given a non-empty credal set $\mathbb{K}$, for any event A we have its *lower* and *upper* probabilities, denoted by $\underline{\mathbb{P}}(A)$ and $\overline{\mathbb{P}}(A)$ respectively: $\underline{\mathbb{P}}(A) = \inf_{\mathbb{P}\in\mathbb{K}} \mathbb{P}(A)$ and $\overline{\mathbb{P}}(A) = \sup_{\mathbb{P}\in\mathbb{K}} \mathbb{P}(A)$. Conditional lower and upper probabilities are defined similarly: For events A and B we define $\underline{\mathbb{P}}(A|B) = \inf_{\mathbb{P}\in\mathbb{K}:\mathbb{P}(B)>0} \mathbb{P}(A|B)$ and $\overline{\mathbb{P}}(A|B) = \sup_{\mathbb{P}\in\mathbb{K}:\mathbb{P}(B)>0} \mathbb{P}(A|B)$ whenever $\overline{\mathbb{P}}(B) > 0$. We leave $\underline{\mathbb{P}}(A|B)$ and $\overline{\mathbb{P}}(A|B)$ undefined when $\overline{\mathbb{P}}(B) = 0$ [60]; in fact, the value of $\underline{\mathbb{P}}(A|B)$ and of $\overline{\mathbb{P}}(A|B)$ is never used in this paper when $\overline{\mathbb{P}}(B) = 0$. An alternative approach would be to consider inference problems under the convention that $\underline{\mathbb{P}}(A|B) = 0$ and $\overline{\mathbb{P}}(A|B) = 1$ whenever $\overline{\mathbb{P}}(B) = 0$ [59]. Another possible approach would be to resort to the theory of full conditional probabilities to obtain $\underline{\mathbb{P}}(A|B)$ and $\overline{\mathbb{P}}(A|B)$ even when $\overline{\mathbb{P}}(B) = 0$ [8]. We leave such possibilities to future work.

2.2. Credal networks

A credal network consists of a directed acyclic graph where each node is a random variable X_i, together with a set of constraints on probability values [2,9,10]. Such a structure is useful as a representation for beliefs, opinions, and statistical summaries that may be available when modeling a particular problem. For instance, suppose we have five variables organized as follows:



Here we have that X_1 and X_2 are *parents* of X_4; likewise, X_3 and X_4 are parents of X_5. The parents of X_i are denoted by $\mathrm{pa}(X_i)$. Intuitively, the graph is understood as indicating that a random variable X_i is directly affected by its parents, and that X_i is independent of its nondescendants nonparents when its parents are fixed (this is called the *Markov condition*).

Even though one is free to impose general constraints on probability values, such as $\mathbb{P}(X_3 = 1, X_2 = 0) \leq 1/2$, usually applications restrict assessments to a few simple forms [2,10]. Typically we have each variable X_i associated with a nonempty *local credal set* $\mathbb{K}_i^{\pi_i}(X_i)$ for each value π_i of the parents of X_i. We assume that each local credal set is closed and convex, as usual in the theory of credal sets [36,59]. Also we focus on local credal sets that are specified by finitely many extreme points,[1] as our specification languages only deal with such objects. In this paper the semantics of such a graph and associated local credal sets is their *strong extension*, defined as the convex hull of the set

$$\left\{ \mathbb{P} : \begin{array}{c} \text{there is } \mathbb{P}(\cdot|\mathrm{pa}(X_i) = \pi_i) \in \mathbb{K}_i^{\pi_i}(X_i) \text{ for each } X_i \text{ and } \pi_i, \text{ such that for every } x_1, \dots, x_n \\ \text{we have that } \mathbb{P}(X_1 = x_1, \dots, X_n = x_n) = \prod_i^n \mathbb{P}\left(X_i = x_i | \mathrm{pa}(X_i) = \pi_i'\right) \end{array} \right\}, \tag{1}$$

where π_i' denotes the projection of $\{X_1 = x_1, \dots, X_n = x_n\}$ on the parents of X_i. In this paper we focus solely on strong extensions, even though there are other ways to interpret the assessments in a credal network [11]. We refer to the set in Expression (1) as the *complete extension* of the credal network.

Note that every extreme point of the strong extension is a product measure consisting of extreme points selected from the local credal sets [21].

Given a credal network (graph and assessments) and its strong extension, we are interested in computing conditional upper probabilities such as $\overline{\mathbb{P}}(X_Q = 1|\mathbf{E})$, for some random variable X_Q and event $\mathbf{E}$.

Suppose we have $\overline{\mathbb{P}}(\mathbf{E}) > 0$ and we want to compute $\overline{\mathbb{P}}(X_Q = 1|\mathbf{E})$. It so happens that we need only look at (the finitely many) extreme points of the strong extension when searching for this latter value, and only at those points that assign positive probability to $\mathbf{E}$. To see this, note that every $\mathbb{P}$ in the strong extension can be written, due to closure and convexity, as a convex combination of extreme points of the strong extension. Hence

---

[1] An extreme element of a closed convex set is one that cannot be written as a convex combination of other elements in the set.

$$\overline{\mathbb{P}}\left(X_Q\,|\mathbf{E}\right) = \sup_{\mathbb{P}:\mathbb{P}(\mathbf{E})>0, \mathbb{P}=\sum_i \alpha_i \mathbb{P}_i, \alpha_i \geq 0, \sum_i \alpha_i=1} \frac{\sum_i \alpha_i \mathbb{P}_i(X_Q=1,\mathbf{E})}{\sum_j \alpha_j \mathbb{P}_i(\mathbf{E})},$$

where each $\mathbb{P}_i$ is an extreme point of the strong extension. However, since only $\mathbb{P}_i$ with $\mathbb{P}_i(\mathbf{E}) > 0$ can affect the ratio, we can write

$$\overline{\mathbb{P}}\left(X_Q\,|\mathbf{E}\right) = \sup_{\mathbb{P}:\mathbb{P}=\sum_i \alpha_i \mathbb{P}_i, \alpha_i > 0, \sum_i \alpha_i=1, \mathbb{P}_i(\mathbf{E})>0} \frac{\sum_i \alpha_i \mathbb{P}_i(X_Q=1,\mathbf{E})}{\sum_i \alpha_i \mathbb{P}_i(\mathbf{E})}.$$

Note that there is at least one $\mathbb{P}_i$ in each summation because $\overline{\mathbb{P}}(\mathbf{E}) > 0$. So we only need to consider the convex hull of those extreme points that assign positive probability to $\mathbf{E}$. Also, note that we can write the expression in the supremum as $\sum_i \beta_i \mathbb{P}_i(X_Q = 1|\mathbf{E})$ for $\beta_i = \alpha_i \mathbb{P}_i(\mathbf{E})/\sum_j \alpha_j \mathbb{P}_j(\mathbf{E})$, with $\beta_i > 0$ and $\sum_i \beta_i = 1$. So we always obtain $\overline{\mathbb{P}}\left(X_Q = 1|\mathbf{E}\right)$ by selecting some extreme point $\mathbb{P}$ of the strong extension such that $\mathbb{P}(\mathbf{E}) > 0$.

2.3. Complexity theory

We adopt basic terminology and notation from computational complexity [45]. A *language* is a set of bit-strings, and a *complexity class* is a set of languages. Whenever we have numbers as input to our problems, we assume them to be rational, to avoid dealing with non-computable reals and similar difficulties. A *decision problem* is to decide whether a given string belongs to a certain language (thus a decision problem defines a language). For some complexity class C, a decision problem D is C-*hard* if every problem D' in the complexity class C can be (many-one) reduced in polynomial time to D (that is, there is an algorithm that takes the input to D', modifies it with polynomial effort, calls D with the modified input, and then returns the output of D). If D is in C and is C-hard, then D is C-complete.

We use the well-known complexity class NP: a language $\mathcal{L}$ is in NP if $\mathcal{L}$ can be decided by a nondeterministic Turing machine within a polynomial bound on time (a language $\mathcal{L}$ is decided by a machine M if $\ell \in \mathcal{L}$ implies input ℓ is accepted by M, and $\ell \notin \mathcal{L}$ implies ℓ is rejected by M). The complexity class PSPACE is similar, but it uses a deterministic Turing machine with a polynomial bound on space. We will also use the class PP: a language $\mathcal{L}$ is in PP when there is a nondeterministic Turing machine M such that $\ell \in \mathcal{L}$ iff more than half of computation paths of M accept ℓ [45].

An oracle Turing machine $M^{\mathcal{L}}$ is a Turing machine with additional tapes, such that it can write a string ℓ to a tape and obtain from the oracle, in another tape and in unit time, the decision as to whether $\ell \in \mathcal{L}$ or not. If a class of languages A is defined by a set of Turing machines $\mathcal{M}$ (that is, the languages are decided by these machines), then define $\mathsf{A}^{\mathcal{L}}$ to consist of the languages defined by oracle machines in $\{M^{\mathcal{L}} : M \in \mathcal{M}\}$. If A and B are classes of languages, $\mathsf{A}^{\mathsf{B}} = \cup_{\mathcal{L}\in B}\mathsf{A}^{\mathcal{L}}$.

Hence $\mathsf{NP}^{\mathsf{PP}}$ consists of those problems that can be solved by a nondeterministic polynomially-bounded Turing machine with an oracle that solves PP decision problems [38]. In proofs we use reductions from E-MAJSAT, an $\mathsf{NP}^{\mathsf{PP}}$-complete decision problem [38]. The E-MAJSAT problem is: given a pair (ϕ, k) where ϕ is a Boolean sentence with n propositions, and $k \in [1, n]$ is an integer, is there a truth assignment to the first k propositions such that the majority of truth assignments to the remaining propositions satisfies ϕ? For instance, ϕ might be $(A_1 \vee \neg A_2 \vee \neg A_3) \wedge (\neg A_1 \vee A_3 \vee A_4)$, and then $(\phi, 1)$ yields YES (that is, it is a string in the corresponding language).

3. A framework for complexity analysis

We now extend a framework for complexity analysis that we have recently developed for Bayesian networks [15], so as to include probability intervals. The basic idea is to restrict assessments to two simple forms that are inspired by probabilistic rules [49,55] and structural models [48]. The framework lets one move down to sub-Boolean constructs and up to relations and quantifiers. The framework should also be useful in practical elicitation scenarios: even though it is quite flexible, it enforces some regularity by avoiding arbitrary constraints amongst probability values.

In this paper we focus on languages that are fragments of propositional and first-order logic. For now we focus on propositional logic; in Section 4 we discuss first-order constructs.

3.1. Propositions, events, variables

Suppose that we have atomic propositions $A_1, \ldots, A_n$ and their truth assignments [24]. The 2^n truth assignments for these atomic propositions are elements of the possibility space Ω. For any propositional formula B, write $\omega \models B$ to indicate that B is true in truth assignment ω. Note that each atomic proposition A_i is either true or false in ω, so we can associate with A_i a random variable X_i such that $X_i(\omega) = 1$ when $\omega \models A_i$ and $X_i(\omega) = 0$ otherwise. To simplify the presentation, we often use identical symbols for atomic propositions and their associated random variables.

Given a propositional formula B, we often refer to the event consisting of all truth assignments such that B is true; that is, the event $\{\omega \in \Omega : \omega \models B\}$. If B is an atomic proposition A_i, we can write the latter event either as $\{\omega \in \Omega : \omega \models A_i\}$ or as $\{\omega \in \Omega : A_i(\omega) = 1\}$. An *assignment* is either the event $\{\omega \in \Omega : A(\omega) = 0\}$ or $\{\omega \in \Omega : A(\omega) = 1\}$; we abbreviate these events respectively by $\{A = 0\}$ and $\{A = 1\}$.

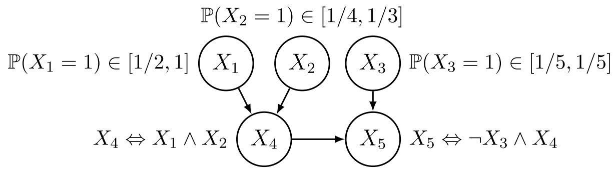


Fig. 1. A simple credal network, specified by definition axioms for non-root random variables and probabilistic assessments for root random variables.

Whenever clear from the context, we use the same symbols to denote a propositional formula and the event defined by the propositional formula. For instance, consider the formula $\neg A \wedge B$. We may write $\mathbb{P}(\neg A \wedge B)$ to mean the probability of the event $\{\omega \in \Omega : \omega \models \neg A \wedge B\}$. These conventions allow us to compactly refer to probabilities for propositional formulas.

To fix ideas, the following example introduces elements of the celebrated Three Prisoners Problem[2] that is discussed in more detail later:

Example 1. Consider three prisoners, P_1, P_2 and P_3, waiting for execution. Denote by E_i the proposition "prisoner P_i will be executed". With each prisoner P_i we associate a random variable that indicates whether E_i obtains; we denote this random variable by E_i as well. That is, $E_i(\omega) = 1$ when the truth assignment ω assigns true to E_i, and $E_i(\omega) = 0$ otherwise. Both $\mathbb{P}(E_i)$ and $\mathbb{P}(E_i = 1)$ mean the same thing: the probability that E_i obtains. Similarly, $\mathbb{P}(E_1 = 1|E_2 = 1, E_3 = 0)$ and $\mathbb{P}(E_1|E_2 \wedge \neg E_3)$ refer to the same probability value. □

The framework we are about to present can be extended to more sophisticated languages where random variables are not directly related to propositions, and have more than two values: we might allow formulas such as $(X = 3) \wedge \neg(Y = 2)$. This would require more notation; for the sake of simplicity, we stay with binary random variables throughout.

3.2. Specifying credal networks

We assume that a directed acyclic graph is given, where each node is a random variable X standing for an atomic proposition. Also, we assume, for each "node" X:

- that if X is a root node, then X is associated with a marginal (interval-valued) probability assessment
$$\mathbb{P}(X = 1) \in [\alpha, \beta], \tag{2}$$
where α and β are rational numbers such that $0 \leq \alpha \leq \beta \leq 1$; or
- that if X is not a root node, then X is associated with a logical equivalence
$$X \Leftrightarrow \phi(Y_1, \ldots, Y_m), \tag{3}$$
where ϕ is a formula on propositions $Y_1, \ldots, Y_m$ that are the parents pa(X) of X.

Inspired by description logics, we refer to Expression (3) as a *definition axiom* [3]. If one wishes to differentiate between a definition axiom and a logical equivalence that appears within a propositional formula, one might use $\leftrightarrow$ for the latter. For instance, one might write $X \Leftrightarrow (Y_1 \leftrightarrow Y_2)$ to mean that X is defined by the truth value of the formula $Y_1 \leftrightarrow Y_2$.

Note that we explicitly avoid direct assessments of conditional probability. Such an assessment can be used to essentially introduce negation (by imposing $\mathbb{P}(X = 1|Y = 1) = \mathbb{P}(X = 0|Y = 0) = 0$). We wish to explicitly control the use of negation. A secondary advantage of this framework is that a user is never asked to provide conditional probabilities, so she is never confronted with questions concerning the meaning of conditional probabilities [12].

We are interested in the strong extension of credal networks specified using such definition axioms and probabilistic assessments.

Example 2. To illustrate the framework, consider the specification in Fig. 1. One might interpret this network as follows: X_4 is a health condition that is identified with the conjunction of two risk factors, and X_5 is an illness that depends deterministically on X_4, with X_3 acting as a probabilistic "inhibitor".

The strong extension of this credal network is simply the convex hull of all extreme Bayesian networks obtained by taking extreme (upper or lower) probabilities for every root node [21,25]. Hence we have four distinct extreme joint probability distributions (the distribution of X_3 is actually a "precise" distribution). For instance, one such distribution assigns probability $1/2$ to $\{X_1 = 1\}$ and probability $1/4$ to $\{X_2 = 1\}$, while another distribution assigns probability 1 to $\{X_1 = 1\}$ and probability $1/4$ to $\{X_2 = 1\}$. By fixing the values of the root random variables we also fix the values of the non-root random

[2] It seems that the Three Prisoners Problem was first proposed by Gardner in 1959 [27]; the original problem and solution appear also in Gardner's compilation of problems [28]. This problem is quite popular and has been discussed in a number of books [30,42,47].

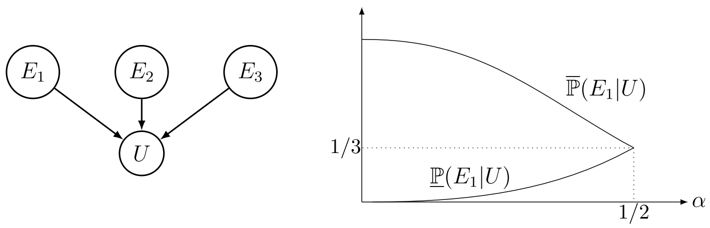


Fig. 2. Example 3: the Bayesian network for E_1, E_2, E_3 and U (left), and the lower and upper probability that P_1 is executed given U as a function of α (right).

variables; for instance, if X_1, X_2 and X_3 are set to true, then X_4 is true and X_5 is false. Hence the probabilities over root random variables induce probabilities over all random variables. □

Here is a longer example, continuing the description of the Three Prisoners Problem.

Example 3. Prisoners P_1, P_2 and P_3 are in prison waiting for their execution, to happen tomorrow. Tonight the governor can pardon each one of them; the decision to pardon each prisoner is taken by throwing three fair coins independently. During the night the honest guard learns the decisions made by the governor. The guard and prisoner P_1 are talking; the guard says that one and only one prisoner will be executed, and the others will be released. Prisoner P_1 concludes that his probability of execution is 1/3.

Denote by E_i the proposition "P_i will be executed". Clearly, E_1, E_2 and E_3 are independent of each other, and moreover

$$\mathbb{P}(E_1) = \mathbb{P}(E_2) = \mathbb{P}(E_3) = 1/2. \tag{4}$$

The following proposition U encodes the fact that one and only one prisoner will be executed:

$$U \Leftrightarrow (E_1 \wedge \neg E_2 \wedge \neg E_3) \vee (\neg E_1 \wedge E_2 \wedge \neg E_3) \vee (\neg E_1 \wedge \neg E_2 \wedge E_3). \tag{5}$$

Note that $\mathbb{P}(E_1|U) = \mathbb{P}(E_1 \wedge U)/\mathbb{P}(U) = (1/8)/(3/8) = 1/3$, as reasoned by Prisoner P_1.

So far this example specifies a Bayesian network; there is no uncertainty concerning probability values. This Bayesian network is depicted in Fig. 2 (left). Suppose instead there is some uncertainty about the three coins used by the governor. We still have the same directed acyclic graph, but now assume, for each E_i, that $\mathbb{P}(E_i)$ belongs to the interval $[\alpha, 1-\alpha]$, for some $\alpha \in [0, 1/2]$. To start, suppose $\alpha = 0$; that is, $\mathbb{P}(E_1) \in [0, 1]$, $\mathbb{P}(E_2) \in [0, 1]$, and $\mathbb{P}(E_3) \in [0, 1]$. Then $\underline{\mathbb{P}}(E_1|U) = 0$ and $\overline{\mathbb{P}}(E_1|U) = 1$. If instead $\alpha = 1/2$, then $\mathbb{P}(E_1) = \mathbb{P}(E_2) = \mathbb{P}(E_3) = 1/2$ as before, and $\underline{\mathbb{P}}(E_1|U) = \overline{\mathbb{P}}(E_1|U) = 1/3$. For other values of α, we obtain lower and upper probabilities as shown in Fig. 2 (right), using

$$\underline{\mathbb{P}}(E_1|U) = \left(1 + 2\left(\frac{1-\alpha}{\alpha}\right)^2\right)^{-1}, \quad \overline{\mathbb{P}}(E_1|U) = \left(1 + 2\left(\frac{\alpha}{1-\alpha}\right)^2\right)^{-1}.$$

For instance, if $\alpha = 19/40$, then $\underline{\mathbb{P}}(E_1|U) = 361/1243 \approx 0.2904$ and $\overline{\mathbb{P}}(E_1|U) = 441/1163 \approx 0.3792$.

Now return to the original problem, where the governor used fair coins.

During the conversation with the guard, Prisoner P_1 asks the guard to tell him the name of a prisoner, other than himself, who will be released (certainly at least one of the other prisoners will be released). The guard says P_2. Thinking a bit about it, P_1 gets desperate: before asking, his (unconditional) probability of execution was 1/3, but now it seems that, just by asking, his (conditional) probability of execution became 1/2, as there are only two prisoners who may be executed. Does this conclusion make sense?

Denote by G_i the proposition "guard says to P_1 that prisoner P_i, other than P_1, will be released". To obtain $\mathbb{P}(E_1|G_2 \wedge U)$, we do have to model the guard's response. There are two situations that cause the guard to say that P_2 will be released. If P_2 will be released, *and* P_3 will be executed, then the guard *must* say P_2. If instead both P_2 and P_3 will be released, then the guard can say P_2 *or* P_3. One way to express this reasoning is to introduce

$$G_2 \Leftrightarrow C_1 \vee C_2, \quad C_1 \Leftrightarrow \neg E_2 \wedge E_3, \quad C_2 \Leftrightarrow \neg E_2 \wedge \neg E_3 \wedge D, \tag{6}$$

where D captures the choice of the guard towards P_2 when the guard can say P_2 or P_3. Fig. 3 (left) depicts the corresponding network.

We have:

$$\mathbb{P}(G_2|E_1 \wedge U) = \frac{\mathbb{P}(G_2 \wedge E_1 \wedge \neg E_2 \wedge \neg E_3)}{\mathbb{P}(E_1 \wedge \neg E_2 \wedge \neg E_3)} = \frac{\mathbb{P}(D \wedge E_1 \wedge \neg E_2 \wedge \neg E_3)}{\mathbb{P}(E_1 \wedge \neg E_2 \wedge \neg E_3)} = \mathbb{P}(D),$$

and then, following Pearl [47, Eq. (2.57)]:

$$\mathbb{P}(E_1|G_2 \wedge U) = \frac{\mathbb{P}(G_2|E_1 \wedge U)\,\mathbb{P}(E_1|U)}{\mathbb{P}(G_2|U)} = \frac{\mathbb{P}(D)\,(1/3)}{(1+\mathbb{P}(D))(1/3)} = \frac{\mathbb{P}(D)}{1+\mathbb{P}(D)},$$

where we used $\mathbb{P}(G_2|U) = \sum_{i=1}^{3} \mathbb{P}(G_2|E_i \wedge U)\,\mathbb{P}(E_i|U)$, $\mathbb{P}(G_2|E_2 \wedge U) = 0$, and $\mathbb{P}(G_2|E_3 \wedge U) = 1$.

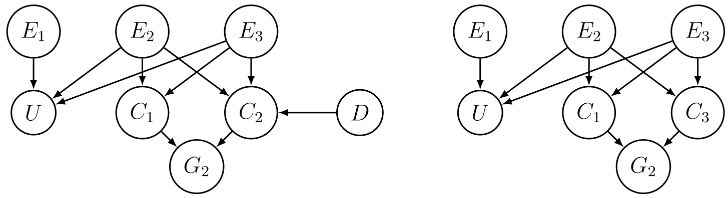


Fig. 3. The networks for the Three Prisoners Problem in Example 3. Left: using an auxiliary variable to model the guard's behavior. Right: using implication.

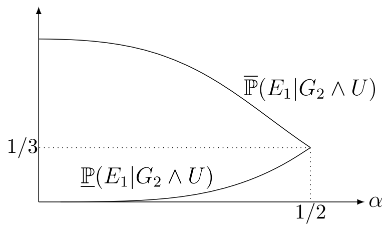


Fig. 4. Example 3: lower and upper probability that E_1 is executed given $G_2 \wedge U$ as a function of α (coins have interval-valued probabilities, guard's behavior is modeled by interval-probability).

Now if $\mathbb{P}(D) = 1/2$, then $\mathbb{P}(E_1|G_2 \wedge U) = 1/3$. So, when the guard decides by flipping a fair coin, the probability that P_1 is executed given $G_2 \wedge U$ is equal to the probability that P_1 is executed given U [27]. This is the usual "solution" offered for the Three Prisoners Problem, even in versions that do not explicitly indicate how the guard decides [42]. However, as also noted in the literature [22,47], if the value of $\mathbb{P}(D)$ is not given, then all that can be inferred is that $\mathbb{P}(D) \in [0, 1]$. In that case, all we can justifiably say is that $\mathbb{P}(E_1|G_2 \wedge U) \in [0, 1/2]$.

Suppose that, in addition to uncertainty concerning the guard's behavior, we also have uncertainty about the coins used by the governor (that is, we have four distinct coins, all associated with some uncertainty). Suppose we have $\mathbb{P}(D) \in [0, 1]$ and each $\mathbb{P}(E_i)$ belongs to a probability interval around 1/2, say $[\alpha, 1-\alpha]$ for $\alpha \in [0, 1/2]$. Regardless of α, we have $\underline{\mathbb{P}}(E_1|G_2 \wedge U) = 0$ (attained when $\mathbb{P}(D) = 0$). Moreover, $\overline{\mathbb{P}}(E_1|G_2 \wedge U) = \left(1 + (\alpha/(1-\alpha))^2\right)^{-1}$. For instance, if $\alpha = 1/2$, then $\overline{\mathbb{P}}(E_1|G_2 \wedge U) = 1/2$, and if $\alpha = 19/40$, then $\overline{\mathbb{P}}(E_1|G_2 \wedge U) = 441/802 \approx 0.5499$.

Finally, suppose that there is uncertainty concerning the governor's coins (as in the previous paragraph), but the guard also decides by flipping a coin such that $\mathbb{P}(D) \in [\alpha, 1-\alpha]$. We then have

$$\underline{\mathbb{P}}(E_1|G_2 \wedge U) = \left(1 + \left(\frac{1-\alpha}{\alpha}\right)^2 \frac{1}{\alpha}\right)^{-1},$$

$$\overline{\mathbb{P}}(E_1|G_2 \wedge U) = \left(1 + \left(\frac{\alpha}{1-\alpha}\right)^2 \frac{1}{1-\alpha}\right)^{-1}.$$

These lower and upper probabilities are depicted in Fig. 4. For instance, for $\alpha = 19/40$ we have $\underline{\mathbb{P}}(E_1|G_2 \wedge U) = 6859/24499 \approx 0.2800$ and $\overline{\mathbb{P}}(E_1|G_2 \wedge U) = 9261/23701 \approx 0.3907$. $\square$

3.3. Specification with conjunction and negation

We now argue that in fact *any* propositional strong extension can be specified using the framework described in the previous section, if the language allows for conjunction and negation.

First note, as a warm-up exercise, that conjunction and negation suffice to specify any joint distribution that can be specified by a *Bayesian* network over binary variables [15]. To see why, take a Bayesian network with random variables $X_1, \ldots, X_n$. Consider a random variable X_i, and suppose first that X_i has two parents denoted by Y_j and Y_k. Impose:

$$\begin{aligned} X_i \Leftrightarrow & \left(\neg Y_j \wedge \neg Y_k \wedge Z_{00}\right) \vee \left(\neg Y_j \wedge Y_k \wedge Z_{01}\right) \vee \\ & \left(Y_j \wedge \neg Y_k \wedge Z_{10}\right) \vee \left(Y_j \wedge Y_k \wedge Z_{11}\right), \\ \mathbb{P}(Z_{ab} = 1) = & \ \mathbb{P}\left(X_i = 1 | Y_j = a, Y_k = b\right) \quad \text{for all } a, b \in \{0, 1\}, \end{aligned}$$

where Z_{ab} are fresh binary random variables (that do not appear anywhere else). Obviously we can always produce disjunction using conjunction and negation, so $\vee$ appears as syntactic sugar in the definition axiom. Now for a random variable X_i with many parents, we just repeat this structure, by taking into account any possible configuration of parents. The marginal distribution of $X_1, \ldots, X_n$ is exactly the distribution specified by the original Bayesian network.

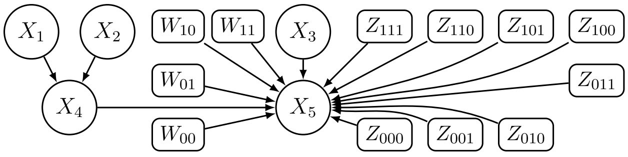


Fig. 5. A credal network specified with conjunction and negation (and twelve auxiliary random variables).

If we have interval-valued assessments in our language, we can go through a similar construction to build a strong extension. To see why, suppose we have a directed acyclic graph where nodes are binary random variables $X_1, \dots, X_n$, and we have all associated local credal sets. Start with a variable X_i with two parents Y_j and Y_k, and denote by π_{ab} the assignment $\{Y_j = a, Y_k = b\}$. Suppose each $\mathbb{K}_i^{\pi_{ab}}(X_i)$ has two extreme points, $p_0(X_i|Y_j = a, Y_k = b)$ and $p_1(X_i|Y_j = a, Y_k = b)$. Note that it is fair to write these extreme points as conditional distributions, because they are necessarily identical to conditional distributions computed with the strong extension. Now introduce fresh binary variables W_{ab} and Z_{abc},

$$X_i \Leftrightarrow \bigvee_{a \in \{0,1\}} \bigvee_{b \in \{0,1\}} \bigvee_{c \in \{0,1\}} (Y_j = a) \wedge (Y_k = b) \wedge (W_{ab} = c) \wedge (Z_{abc} = 1),$$

point-valued assessments $\mathbb{P}(Z_{abc} = 1) = p_c(X_i = 1|Y_j = a, Y_k = b)$, and interval-valued assessments $\mathbb{P}(W_{ab} = 1) \in [0, 1]$. This encodes the desired local credal sets. The idea is that a and b select a particular configuration of Y_j and Y_k, while c selects a particular extreme point of the corresponding local credal set (and then Z_{abc} carries the appropriate probability value).

Here is a simple example to illustrate the procedure. Suppose we have the graph depicted in Fig. 1, and the same probability intervals associated with X_1, X_2 and X_3. Suppose additionally that $X_4 \Leftrightarrow X_1 \wedge X_2$ and X_5 is associated with the following probability intervals (where $\pi_{ab} = \{X_3 = a, X_4 = b\}$):

$$\mathbb{P}(X_5 = 1|\pi_{00}) \in [1/9, 1/2], \qquad \mathbb{P}(X_5 = 1|\pi_{01}) \in [1/3, 1/3],$$
$$\mathbb{P}(X_5 = 1|\pi_{10}) \in [1/4, 1/4], \qquad \mathbb{P}(X_5 = 1|\pi_{11}) \in [4/7, 5/7].$$

The transformation in the previous paragraph, applied to X_5, takes us to the graph in Fig. 5. Each W_{ab} is associated with $\mathbb{P}(W_{ab}) \in [0, 1]$, and additionally:

$$\mathbb{P}(Z_{000} = 1) = 1/9, \quad \mathbb{P}(Z_{001} = 1) = 1/2, \quad \mathbb{P}(Z_{010} = 1) = \mathbb{P}(Z_{011} = 1) = 1/3,$$
$$\mathbb{P}(Z_{100} = 1) = \mathbb{P}(Z_{101} = 1) = 1/4, \quad \mathbb{P}(Z_{110} = 1) = 4/7, \quad \mathbb{P}(Z_{111} = 1) = 5/7.$$

Some of the auxiliary variables W_{ab} and Z_{abc} are not really needed in this example, given that some of the assessments in the original credal network are actually point-valued.

We have dealt with random variables with two parents. Now consider a variable X_i with parents $Y_1, \dots, Y_m$, associated with the local credal set $\mathbb{K}_i^{\pi}(X_i)$ for each configuration $\pi = \{Y_1 = y_1, \dots, Y_m = y_m\}$. Note that 2^m local credal sets must be specified, and each one of them is given by two extreme points, say $p_0(X_i|Y_1 = y_1, \dots, Y_m = y_m)$ and $p_1(X_i|Y_1 = y_1, \dots, Y_m = y_m)$. Introduce fresh binary variables $W_{y_1 \dots y_m}$ and $Z_{y_1 \dots y_m c}$ and the definition axiom

$$X_i \Leftrightarrow \bigvee_{y_1} \cdots \bigvee_{y_m} \bigvee_{c} \left(\bigwedge_{k} Y_k = y_k \right) \wedge (W_{y_1 \dots y_m} = c) \wedge (Z_{y_1 \dots y_m c} = 1), \tag{7}$$

where $y_1, \dots, y_m$ and c can be either 0 or 1, and

$$\mathbb{P}\left(Z_{y_1 \dots y_m c} = 1\right) = p_c(X_i = 1|Y_1 = y_1, \dots, Y_m = y_m),$$
$$\mathbb{P}\left(W_{y_1 \dots y_m} = 1\right) \in [0, 1].$$

We thus construct a joint credal set whose marginal is the strong extension of the original credal network. The number of auxiliary random variables is of order 2^m, thus polynomial in the size of the description of the original credal network.

3.4. Inferential complexity, parameterized by language

Fix a language $\mathcal{L}$. We can then define the set $\mathcal{C}(\mathcal{L})$ of all credal networks that are specified as in Section 3.2, and where each ϕ in Expression (3) is a formula in $\mathcal{L}$. That is, each $\mathbb{C} \in \mathcal{C}(\mathcal{L})$ is specified by a directed acyclic graph where each node is a random variable corresponding to a proposition, and by assessments given by Expressions (2) and (3), with the restriction that sentences ϕ in definition axioms use the language $\mathcal{L}$.

Now denote by $\mathsf{INF}(\mathcal{L})$ the set of decision problems that:

- have as input
  1. a credal network $\mathbb{C} \in \mathcal{C}(\mathcal{L})$,
  2. an assignment Q for a random variable in $\mathbb{C}$,
  3. a set of assignments $\mathbf{E}$ for random variables in $\mathbb{C}$,
  4. and a rational number γ;
- yield YES in case

$$\overline{\mathbb{P}}(\mathbf{E}) > 0 \quad \text{and} \quad \overline{\mathbb{P}}(Q|\mathbf{E}) > \gamma$$

  with respect to the strong extension, and NO otherwise.

To simplify the statement of some results, we denote by $\mathsf{INF}^+(\mathcal{L})$ the decision problems defined as $\mathsf{INF}(\mathcal{L})$, with the additional constraint that all assignments in $Q \cup \mathbf{E}$ are "positive" (that is, random variables are only set to 1).

3.5. The complexity of propositional languages

Denote by $\mathsf{Prop}(\wedge, \neg)$ the language of well-formed propositional sentences with conjunction and negation. Using the reasoning in Section 3.3, we see that $\mathsf{Prop}(\wedge, \neg)$ can specify any strong extension over binary variables. Moreover:

Theorem 4. $\mathsf{INF}(\mathsf{Prop}(\wedge, \neg))$ *is* $\mathsf{NP}^{\mathsf{PP}}$*-complete.*

Proof. To prove membership, consider that an inference can be produced by first guessing whether we must use the lower or upper end of the probability interval for each probability assessment, setting the probability to the selected value, and then running an inference in the resulting Bayesian network (thus overall we make as many nondeterministic guesses as there are intervals in the network description, followed by a call to a PP oracle). To prove hardness, note that any E-MAJSAT problem can be encoded as a MAP problem for a Bayesian network [46], where the query is a conjunction of assignments; the latter can be encoded as an inference in a credal network [14] that can then be expressed within $\mathcal{C}(\mathsf{Prop}(\wedge, \neg))$, with a query that is a conjunction of assignments. Because we have conjunction in our language, the conjunction of assignments can be encoded as an assignment for a single random variable X_Q. That is, by determining $\overline{\mathbb{P}}\left(X_Q = 1\right) > 1/2$ we solve the original E-MAJSAT problem, thus obtaining the desired $\mathsf{NP}^{\mathsf{PP}}$-hardness. □

Now consider a more restricted language: denote by $\mathsf{Prop}(\wedge, (\neg))$ the language that uses only conjunction and *atomic* negation (defined as negation that can appear only before a proposition associated with a probabilistic assessment). For instance, the credal network in Fig. 1 belongs to $\mathcal{C}(\mathsf{Prop}(\wedge, (\neg)))$. We know that inference for $\mathsf{Prop}(\wedge, (\neg))$ is polynomial for Bayesian networks as long as evidence is positive [15]. Surprisingly, this result applies to credal networks:

Theorem 5. $\mathsf{INF}^+(\mathsf{Prop}(\wedge, (\neg)))$ *can be solved in polynomial time.*

Proof. We must check whether $\overline{\mathbb{P}}(\mathbf{E}) > 0$, and if so, then proceed to verify whether $\overline{\mathbb{P}}(Q|\mathbf{E}) > \gamma$. We present the proof under the assumption that $\overline{\mathbb{P}}(\mathbf{E}) > 0$. One can use the same proof to check whether $\overline{\mathbb{P}}(\mathbf{E}) > 0$ with polynomial effort: to do so, introduce a random variable Y that is the conjunction of all random variables that appear in $\mathbf{E}$, and verify whether $\overline{\mathbb{P}}(Y = 1) > 0$ (note that no evidence is involved in this latter query).

So, consider first a network with just conjunction; we deal with atomic negation later. Note first that if a node X appears in Q or in $\mathbf{E}$, then its ascendants must all be set to true. So we first add to $\mathbf{E}$ all ascendants of nodes originally in $\mathbf{E}$; also, if a node has all parents set to true, then it must be true and this assignment can be added to $\mathbf{E}$, so we repeat this until no more assignments can be added to $\mathbf{E}$. Now if Q is included in $\mathbf{E}$, then $\underline{\mathbb{P}}(Q|\mathbf{E}) = \overline{\mathbb{P}}(Q|\mathbf{E}) = 1$.

Denote by X_Q the random variable that appears in Q. So, either we have that inference is immediate (last sentence of previous paragraph), or all descendants of X_Q are *barren* nodes in any extreme point of the strong extension, so they can be discarded (the probabilities of barren nodes are summed out in any Bayesian network [33]). So, discard these nodes and produce a sub-network where X_Q has no descendants.

Now collect all nodes that are ascendants of X_Q. Suppose one of these nodes, say W, points both to an ascendant of X_Q, and to a non-ascendant, say Y, of X_Q. Now if Y is not in $\mathbf{E}$, then it is a barren node that can be discarded. And if Y is in $\mathbf{E}$, then W itself must be in $\mathbf{E}$, hence Y is to be discarded. For instance, consider Fig. 6, and suppose Y is set to true. Then W, Z and W' are set to true, and we can discard them, as X_Q is independent of Z and W' given W for any extreme point of the strong extension. This is obtained by noting that such an extreme point is a Bayesian network, and d-separation yields the independence.[3]

[3] Recall that if $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ are three sets of nodes of a directed acyclic graph, then $\mathcal{X}$ and $\mathcal{Y}$ are d-separated by $\mathcal{Z}$ if along every (undirected) path between a node in $\mathcal{X}$ and a node in $\mathcal{Y}$ there is a node W satisfying: (i) W has converging arrows and none of W nor its descendants are in $\mathcal{Z}$, or (ii) W

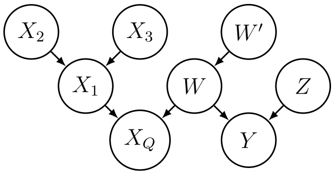


Fig. 6. Network in the proof of Theorem 5.

Once we have discarded all nodes that are not required for our computation, we are left with an "inverted" tree whose root is X_Q, and where each leaf is either a node set to true, or a node associated with a probability interval. Denote by $X_1, \ldots, X_m$ the leaves that are not set to true in this tree; we can then write $X_Q \Leftrightarrow X_1 \wedge \ldots \wedge X_m$. So we have $\overline{\mathbb{P}}(Q\,|\mathbf{E}) = \prod_{i=1}^m \overline{\mathbb{P}}(X_i)$; in fact, we also have that $\underline{\mathbb{P}}(Q\,|\mathbf{E}) = \prod_{i=1}^m \underline{\mathbb{P}}(X_i)$. To complete the proof, suppose that atomic negation is allowed, so some variables appear negated. We can run the same procedure already described, using $\overline{\mathbb{P}}(\neg X_i) = 1 - \underline{\mathbb{P}}(X_i)$ instead of $\overline{\mathbb{P}}(X_i)$ whenever $\overline{\mathbb{P}}(Q\,|\mathbf{E})$ depends on $\neg X_i$ instead of X_i, with the novelty that $X_i \wedge \neg X_i$ may appear in the definition of X_Q, and in this case $\overline{\mathbb{P}}(Q\,|\mathbf{E}) = 0$. In all cases the whole procedure reaches an output in time that is polynomial in the description of the network, because evidence can be propagated in linear time, and d-separation runs in linear time as well; the final products can be easily computed in polynomial time. □

It seems unlikely that polynomial-time inference can be obtained with other propositional languages, as several simple changes to $\mathsf{Prop}(\wedge, (\neg))$ move us into higher complexity.[4] Indeed, consider the following points. Even though $\mathsf{INF}^+(\mathsf{Prop}(\wedge, (\neg)))$ belongs to P, $\mathsf{INF}(\mathsf{Prop}(\wedge, (\neg)))$ does not, as it is PP-hard already when all probability intervals are singletons. The move from $\mathsf{INF}^+(\mathsf{Prop}(\wedge, (\neg)))$ to $\mathsf{INF}^+(\mathsf{Prop}(\wedge, \neg))$ leads to $\mathsf{NP}^{\mathsf{PP}}$-completeness using the proof of Theorem 4. Finally, if we add disjunction so as to focus on $\mathsf{INF}^+(\mathsf{Prop}(\wedge, \vee, (\neg)))$, again we move away from polynomial-time behavior, as the following result shows.

Theorem 6. $\mathsf{INF}^+(\mathsf{Prop}(\wedge, \vee, (\neg)))$ *is* $\mathsf{NP}^{\mathsf{PP}}$*-complete.*

Proof. Membership follows from Theorem 4, as $\mathsf{Prop}(\wedge, \vee, (\neg))$ is a sub-language of $\mathsf{Prop}(\wedge, \neg)$. Hardness: As in the proof of Theorem 4, encode an E-MAJSAT problem (that can be written using $\mathsf{Prop}(\wedge, \vee, (\neg))$) as an inference in a credal network, with a query that is a conjunction of assignments. By determining whether $\overline{\mathbb{P}}(Q) > 1/2$, we solve the E-MAJSAT problem, thus obtaining the desired $\mathsf{NP}^{\mathsf{PP}}$-hardness. □

3.6. Implication axioms

We now consider an additional specification scheme based on material implication. Suppose a directed acyclic graph is given, and each node is a random variable X standing for a proposition. Root nodes are still associated with probability assessments (Expression (2)), but non-root nodes are now associated *either* with a definition axiom (Expression (3)), *or* an expression

$$X \Leftarrow \phi(Y_1, \ldots, Y_m), \tag{8}$$

where as before $Y_1, \ldots, Y_m$ are the parents of X. We refer to Expression (8) as an *implication axiom*. As $\phi(Y_1, \ldots, Y_m)$ may itself contain material implications, a reasonable convention would be to use $\leftarrow$ to denote such "inner" implications. For instance, we might write $X_5 \Leftarrow \neg(\neg X_3 \leftarrow X_4)$ to mean $X_5 \Leftarrow X_3 \wedge X_4$.

To develop sensible semantics for implication axioms, it pays to revisit the semantics of definition axioms. When all nodes in the graph are associated with probabilistic assessments or definition axioms, then the value of all non-root nodes are fixed once we fix the root nodes. Hence probabilities over all random variables are induced by the probabilities over root random variables. Then the strong extension is, quite simply, the convex hull of all joint distributions induced by product measures over root variables (for all possible choices of their marginal probabilities).

We can easily adapt this scheme to accommodate implication axioms. Again, we have root random variables, associated with the same set of product measures over them as described in the previous paragraph. However, implication axioms do not uniquely fix the values of the non-root random variables when we fix the root random variables; we must consider the set of all possible joint distributions built this way. Additionally, we must require that any distribution in this set satisfies the Markov condition with respect to the directed acyclic graph, so as to respect the intended meaning of the graph. We call the resulting set of joint distributions the *complete extension* of the assessments and axioms. We then take the convex hull

---

does not have converging arrows and W is in $\mathcal{Z}$ [47]. In a Bayesian network, d-separation implies conditional independence: if $\mathcal{X}$ and $\mathcal{Y}$ are d-separated by $\mathcal{Z}$, then $\mathcal{X}$ and $\mathcal{Y}$ are conditionally independent given $\mathcal{Z}$ [19,33].

4 But note that polynomial complexity can be obtained by suitably constraining the network topology; e.g., the 2U algorithm for polytrees [25].

of the complete extension as the semantics. As before, joint distributions that maximize specific conditional probabilities are extreme points of the convex hull of the complete extension.

The next example illustrates this construction, by returning to the Three Prisoners Problem.

Example 7. Example 3 discussed the inner logical and probabilistic structure of the Three Prisoners Problem. Suppose the governor's coins are fair, so $\mathbb{P}(E_i) = 1/2$, and nothing is known about the decision behavior of the guard, so $\mathbb{P}(D) \in [0, 1]$. The insertion of the variable D, just to immediately say that its probability is vacuous, is far from an elegant device to express uncertainty about the guard's decision. A much sharper strategy is to adopt Expressions (4) and (5), and to replace Expression (6) by:

$$G_2 \Leftrightarrow C_1 \vee \neg C_3, \quad C_1 \Leftrightarrow \neg E_2 \wedge E_3, \quad C_3 \Leftarrow E_2 \vee E_3.$$

That is: if E_2 or E_3, then C_3 is true and therefore it does not affect G_2; but if both E_2 and E_3 are false, then C_3 can be true or false, depending on the guard's choice. The corresponding network is presented in Fig. 3 (right). We have eliminated the artificial random variable D.

Consider the set of all possible probability measures that are compatible with these assessments and logical expressions. Note that

$$\begin{aligned}\mathbb{P}(G_2|E_1 \wedge U) &= \mathbb{P}((\neg E_2 \wedge E_3) \vee \neg C_3|E_1 \wedge \neg E_2 \wedge \neg E_3)\\ &= \mathbb{P}(\neg C_3|E_1 \wedge \neg E_2 \wedge \neg E_3)\\ &= \mathbb{P}(\neg C_3|E_1 \wedge \neg(E_2 \vee E_3)).\end{aligned}$$

In principle, we might allow C_3 to depend on E_1 even when $\neg(E_2 \vee E_3)$ is given. However, our assumption that any distribution in the complete extension satisfies the Markov condition rules out this possibility; that is, we must have $\mathbb{P}(\neg C_3|E_1 \wedge \neg(E_2 \vee E_3)) = \mathbb{P}(\neg C_3|\neg(E_2 \vee E_3))$. Therefore,

$$\mathbb{P}(G_2|E_1 \wedge U) = \mathbb{P}(\neg C_3|\neg(E_2 \vee E_3)),$$

and then

$$\mathbb{P}(E_1|G_2 \wedge U) = \frac{\mathbb{P}(G_2|E_1 \wedge U)(1/3)}{(1 + \mathbb{P}(G_2|E_1 \wedge U))(1/3)} = \frac{\mathbb{P}(\neg C_3|\neg(E_2 \vee E_3))}{1 + \mathbb{P}(\neg C_3|\neg(E_2 \vee E_3))},$$

so we find, again, that $\mathbb{P}(E_1|G_2 \wedge U) \in [0, 1/2]$ as there is no constraint on $\mathbb{P}(C_3|\neg(E_2 \vee E_3))$. □

Even though the constraints induced by implication axioms may seem relatively weak, they suffice to specify any propositional credal network, *even* when all probabilistic assessments are point-valued. Suppose we have a network fragment consisting of X and its parents $Y_1, \dots, Y_m$, and we wish to specify credal sets $\mathbb{K}^\pi(X)$ for each assignment π of the parents. Note that there are 2^m local credal sets to be specified. We use Expression (7) to encode these credal sets, by introducing fresh variables $Z_{y_1 \dots y_m c}$ that are associated with point-valued assessments, and fresh variables $W_{y_1 \dots y_m}$ that are associated with interval-valued assessments $\mathbb{P}\left(W_{y_1 \dots y_m} = 1\right) \in [0, 1]$. To produce these latter interval-valued assessments, for each $W_{y_1 \dots y_m}$ introduce an additional fresh variable $W'_{y_1 \dots y_m}$. Now make $W'_{y_1 \dots y_m}$ the sole parent of $W_{y_1 \dots y_m}$, keep $W'_{y_1 \dots y_m}$ as a root variable, and add

$$W_{y_1 \dots y_m} \Leftarrow W'_{y_1 \dots y_m}, \qquad \mathbb{P}\left(W'_{y_1 \dots y_m} = 1\right) = 0. \tag{9}$$

Then the marginal probability of $W_{y_1 \dots y_m}$ is $\mathbb{P}\left(W_{y_1 \dots y_m} = 1\right) \in [0, 1]$, as desired. By repeating this for each variable in the credal network, we encode the desired local credal sets.

The discussion in the previous paragraph can be used to understand the complexity of inferences with implication axioms. To proceed, denote by $\mathsf{INF}^{\Leftarrow}(\mathcal{L})$ the set of decision problems that yield YES if $\overline{\mathbb{P}}(\mathbf{E}) > 0$ and $\overline{\mathbb{P}}(Q|\mathbf{E}) > \gamma$, and NO otherwise, for strong extensions specified using definition and implication axioms, probabilistic assessments, and constructs in the language $\mathcal{L}$. We have:

Theorem 8. $\mathsf{INF}^{\Leftarrow}(\mathsf{Prop}(\wedge, \neg))$ *is* $\mathsf{NP}^{\mathsf{PP}}$*-complete even when all probabilistic assessments are point-valued.*

Proof. Hardness follows from the fact that any strong extension can be specified using definition/implication axioms and point-valued assessments, using Expressions (7) and (9). To prove membership: Take each implication axiom $X \Leftarrow \phi(Y_1, \dots, Y_m)$ and determine (nondeterministically) whether X is true or false when $\phi(Y_1, \dots, Y_m)$ is false; in case there are interval-valued probability assessments, then determine (nondeterministically) whether we must use the lower or upper end of each probability interval; finally run inference in the resulting Bayesian network (a PP-complete problem). □

3.7. Implication, in the other direction

In the previous section we considered implication axioms that can be written as $X \Leftarrow \phi(Y_1, \ldots, Y_m)$. Another possibility is to allow implication axioms of the form $X \Rightarrow \phi(Y_1, \ldots, Y_m)$, while still $Y_1, \ldots, Y_m$ are parents in the directed acyclic graph, and where the same sort of semantics as before is adopted.

Example 9. Consider again the Three Prisoners Problem. In Example 3 we have employed a definition $G_2 \Leftrightarrow C_1 \vee \neg C_3$ and an implication $C_3 \Leftarrow E_2 \vee E_3$. Alternatively, we could use

$$G_2 \Leftrightarrow C_1 \vee C_4, \quad C_1 \Leftrightarrow \neg E_2 \wedge E_3, \quad C_4 \Rightarrow \neg E_2 \wedge \neg E_3.$$

That is: when E_2 and E_3 are both false, then C_4 can be either true or false, depending on the guard's preferences; otherwise, C_4 must be false (and then it does not affect G_2). Both descriptions produce the same inferences, even though they do have a distinct feel. □

In fact, as long as the graph is kept acyclic, one can mix implication axioms with both $\Leftarrow$ and $\Rightarrow$, thus moving closer to a general probabilistic logic [1,13].

3.8. Related literature: a bit of probabilistic logic programming

In the previous sections we have first introduced the implication axioms in the $\Leftarrow$ format; they lead to statements that are similar to the ones adopted in logic programming. Indeed, there are several formalisms that combine probabilities and logic programming; some have been designed to accommodate probability intervals from the outset [35,39,43,44], while others focus on point-valued probabilities [26,49,55]. The latter proposals have resorted to "independent choices" that are similar to our root nodes.

Poole's Probabilistic Horn Abduction (PHA) framework combines Prolog-style programming with independent choices to obtain a language that can represent a variety of probabilistic models [49]. There we have rules such as $A \Leftarrow A_1 \wedge \cdots \wedge A_n$, where each A_i is an atom, and where some such atoms are associated with precise probabilities. The semantics adopted for PHA's probabilistic Horn clauses uses their Clark's completion; that is, the Horn clause $A \Leftarrow A_1 \wedge \ldots \wedge A_n$ is in essence interpreted as the definition axiom $A \Leftrightarrow A_1 \wedge \ldots \wedge A_n$ [49, Definition A.2]. The same approach to semantics is adopted in Poole's Independent Choice Logic (ICL), where negation as failure is also included [50,51]. The point here is that our languages so far are closely related to propositional fragments of ICL; our results accordingly can be used in that setting.

4. Relational credal networks

Many phenomena in real life display repetitive patterns. For instance, social networks involve many individuals, several of which may share common characteristics. Epidemiological events may also bring together individuals that are geographically close, while temporal sequences modeled by hidden Markov models often capture similarities across time steps. In response to this, there are several formalisms that can represent repetition in Bayesian network fragments [29,31,34,52,53]. The simplest strategy is to allow random variables to be parameterized, as the next example illustrates.

Example 10. Consider Example 2. We might extend the specification to handle several patients, as follows:

$$\mathbb{P}(X_1(\mathfrak{x}) = 1) \geq 1/2, \tag{10}$$

$$\mathbb{P}(X_2(\mathfrak{x}) = 1) \in [1/4, 1/3], \tag{11}$$

$$\mathbb{P}(X_3(\mathfrak{x}) = 1) = 1/5, \tag{12}$$

$$X_4(\mathfrak{x}) \Leftrightarrow X_1(\mathfrak{x}) \wedge X_2(\mathfrak{x}), \tag{13}$$

$$X_5(\mathfrak{x}) \Leftrightarrow \neg X_3(\mathfrak{x}) \wedge X_4(\mathfrak{x}). \tag{14}$$

For instance, $X_5(\mathfrak{x})$ indicates whether a generic element of the domain, represented by the logical variable $\mathfrak{x}$, is suffering from a particular illness. Our intended interpretation for this specification is as follows. Given an element a of the domain, we instantiate the assessments to obtain $\mathbb{P}(X_1(a) = 1) \geq 1/2$, $\mathbb{P}(X_2(a) = 1) \in [1/4, 1/3]$, and so on. Thus a single element of the domain produces a Bayesian network similar to the network in Example 2. For a domain with N elements, we obtain N disconnected graphs with identical structure, one per element. □

We refer to $\mathfrak{x}, \mathfrak{y}, \ldots$ as *logical variables*. We say that $X(\mathfrak{x}_1, \ldots, \mathfrak{x}_k)$, where each $\mathfrak{x}_i$ is either a logical variable or an element of the domain, is an *atom*, and X is a *relation* of *arity* k. An atom with no logical variable is a *ground atom*.

We can then extend our previous specification framework as follows. We assume that a directed acyclic graph is given, where each node is a relation of some given arity, and that for each "node" X:

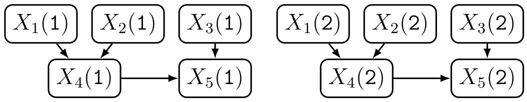


Fig. 7. Grounding assessments (10)–(14) with respect to domain $\mathcal{D} = \{1, 2\}$.

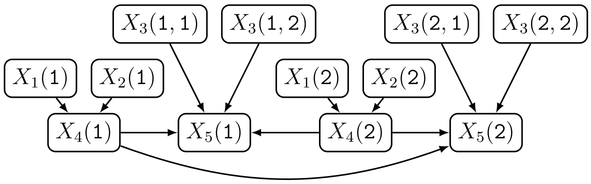


Fig. 8. Grounding assessments (10), (11), (13), (15), and (16) with respect to domain $\mathcal{D} = \{1, 2\}$.

- if X is a root node, then X is associated with a probabilistic assessment

$$\mathbb{P}(X(x_1, \dots, x_k) = 1) \in [\alpha, \beta],$$

  where α and β are rational numbers such that $0 \le \alpha \le \beta \le 1$; or
- if X is not a root node, then X is associated with a definition axiom

$$X(x_1, \dots, x_k) \Leftrightarrow \phi(x_1, \dots, x_k, Y_1, \dots, Y_m),$$

  where ϕ is a formula with free logical variables $x_1, \dots, x_k$, and possibly with other logical variables that are bound to quantifiers, and where $Y_1, \dots, Y_m$ are the parents of X.

We assume that our languages consist of formulas of function-free first-order logic with equality (referred to as FFFO). That is, we allow existential and universal quantifiers in our syntax, in addition to Boolean operators and equality, but we do not allow functions. The directed acyclic graph, its associated relations and associated probabilistic assessments and definition axioms are referred to as a *relational credal network*.

Concerning the semantics, we assume that we have a set $\mathcal{D}$, the *domain*. In this paper every domain is finite. By replacing logical variables by elements of the domain, we obtain ground atoms for the various relations. For instance, suppose we take assessments and axioms in Example 10, and a domain $\mathcal{D} = \{1, 2\}$. We can produce several ground atoms: $X_1(1)$, $X_1(2)$, $X_2(1)$, $X_2(2)$, $X_3(1)$, and so on. We can obviously attach a random variable to each ground atom, as we did before for propositions. Hence from now on we treat ground atoms as random variables and vice-versa, whenever appropriate.

Clearly we can ground every definition axiom, by instantiating all logical variables as usual in first-order logic. For instance, from Example 10 we can produce $X_5(1) \Leftrightarrow \neg X_3(1) \wedge X_4(1)$.

Now build a graph where each ground atom is a node, and where edges are added as indicated by grounded definition axioms. That is, if a ground atom is defined by a set of other ground atoms, then there are edges from the latter atoms to the former. This graph is the *grounded graph* of the relational credal network. For instance, the grounded graph for Example 10 with respect to domain $\{1, 2\}$ is the pair of disjoint identical graphs depicted in Fig. 7. Of course more involved specifications may lead to fully connected graphs. For instance, suppose X_3 is actually a binary relation with associated assessment

$$\mathbb{P}\left(X_3(x, y) = 1\right) = 1/5, \tag{15}$$

and X_5 is defined by

$$X_5(x) \Leftrightarrow \exists y : X_3(x, y) \wedge X_4(y). \tag{16}$$

Consider assessments (10), (11), (13), (15), and (16); then grounding takes us to Fig. 8. Note that a universal quantifier in a definition axiom yields a conjunction of ground atoms, while an existential quantifier in a definition axiom yields a disjunction of ground atoms; for instance, we have the following grounded definition:

$$X_5(1) \Leftrightarrow (X_3(1, 1) \wedge X_4(1)) \vee (X_3(1, 2) \wedge X_4(2)).$$

Now consider the probabilistic assessments. Take for instance

$$\mathbb{P}(X_1(x)) \in [1/2, 1].$$

What does it mean? The most direct interpretation seems to be that, for each $x \in \mathcal{D}$,

$$\mathbb{P}(X_1(x) = 1) \in [1/2, 1].$$

That is, we should see a universal quantifier before the assessment. So in Fig. 8 we would have

$$\mathbb{P}(X_1(1) = 1) \geq 1/2, \quad \mathbb{P}(X_1(2) = 1) \geq 1/2,$$
$$\mathbb{P}(X_2(1) = 1) \in [1/4, 1/3], \ldots, \mathbb{P}(X_3(2, 2) = 1) = 1/5,$$

and so on.

We thus obtain a grounded graph and grounded definitions and assessments; that is, we have a (grounded) credal network. We define the semantics of the relational credal network, with respect to a given domain, to be the strong extension of the grounded credal network. When this sort of semantics is adopted, we say that the relational credal network has *decoupled parameters*. To simplify the language, we often refer to a *decoupled credal network* to mean a relational credal network with decoupled parameters. Note that the strong extension of a decoupled credal network has extreme points obtained by combining the extreme points of probability intervals; for instance, Expressions (10), (11), (13), (15), and (16) produce a strong extension with 16 extreme Bayesian networks given $\mathcal{D} = \{1, 2\}$, as there are two ground atoms for X_1 and two for X_2 (thus we have four probability intervals to choose from, each with two distinct endpoints).

As often happens when one moves from sharp to interval probabilities, there is more than one way to interpret assessments. In our setting, there is an additional sensible semantics one can adopt for relational credal networks. The idea is to interpret an assessment

$$\mathbb{P}(X(\mathtt{x}_1, \ldots, \mathtt{x}_k) = 1) \in [\alpha, \beta], \tag{17}$$

as a constraint on a point-valued relational specification that in effect imposes a *coupling* amongst parameters. That is, we should consider the set of point-valued assessments

$$\forall \mathtt{x}_1 \in \mathcal{D}, \ldots, \mathtt{x}_k \in \mathcal{D} : \mathbb{P}(X(\mathtt{x}_1, \ldots, \mathtt{x}_k) = 1) = \gamma, \tag{18}$$

for each γ in $[\alpha, \beta]$.

We take the semantics of coupled parameters to be the set of all joint probability distributions produced as follows. First, generate the grounded graph and the grounded definition axioms as before. Second, generate the following set of Bayesian networks. For each root node associated with an assessment such as Expression (17), select a γ in $[\alpha, \beta]$, and interpret the assessment as Expression (18). That is, each grounding X_g of X is associated with $\mathbb{P}\left(X_g = 1\right) = \gamma$. Thus such a selection leads to a Bayesian network. By taking every possible Bayesian network generated this way, we obtain a set of Bayesian networks that we call the *complete extension* of the coupled credal network. Finally, take the convex hull of the complete extension: this is the semantics. We then say the relational credal network has *coupled parameters*; to simplify the language, we often refer to a *coupled credal network* to mean a relational credal network with coupled parameters.

The difference between the coupled and decoupled semantics just described is substantial, although both share the same grounded graph (for a given domain). Consider the following augmented version of the Three Prisoners Problem.

Example 11. Suppose we have a domain $\mathcal{D}$ containing prisoners $P_1, \ldots, P_N$. Prisoners are isolated, and again there is an honest guard. Introduce a relation E such that $E(\mathtt{x})$ denotes "prisoner $\mathtt{x}$ will be executed".

Suppose that again the governor will release some prisoners, based on throws of a *single* coin. Note that the governor is now using the *same* coin to make all decisions, so it makes sense to assume that

$$\mathbb{P}(E(P_1)) = \mathbb{P}(E(P_2)) = \ldots = \mathbb{P}(E(P_N)),$$

even if $\mathbb{P}(E(\mathtt{x}))$ is not precisely known. Such a constraint is not available in our previous propositional networks, and not available with decoupled parameters. But with the coupled semantics we easily have it, as the assessment $\mathbb{P}(E(\mathtt{x})) \in [\alpha, 1-\alpha]$ is interpreted as imposing the same probability value in the interval $[\alpha, 1-\alpha]$ to all prisoners.

So, suppose Prisoner P_1 learns that one and only one prisoner will be executed tomorrow. To encode this information, introduce

$$U \;\Leftrightarrow\; \exists \mathtt{x} : \Big(E(\mathtt{x}) \wedge \forall y \neq \mathtt{x} : \neg E(y)\Big).$$

Suppose we select $\gamma = \mathbb{P}(E(P_1))$; for any prisoner P_i, we have

$$\mathbb{P}(E(P_i)|U) = \frac{\mathbb{P}(E(P_i) \wedge U)}{\mathbb{P}(U)} = \frac{\gamma(1-\gamma)^{N-1}}{N\gamma(1-\gamma)^{N-1}} = \frac{1}{N}.$$

Consequently, $\underline{\mathbb{P}}(E(P_1)|U) = \overline{\mathbb{P}}(E(P_1)|U) = 1/N$.

Suppose prisoner P_1 asks the guard to tell him the name of $N-2$ prisoners (amongst the other $N-1$ prisoners) who will be released. The guard gives P_1 such a list of prisoners.[5] To be concrete, suppose the list does not contain P_2. Thinking a bit about it, P_1 gets desperate: before asking, his (unconditional) probability of execution was merely $1/N$; given the guard's list, the (conditional) probability of execution is $1/2$.

[5] In the solution of the original Three Prisoners Problem, Gardner in essence describes this variant of the problem [28]. Pearl discusses a similar problem with 1000 prisoners [47].

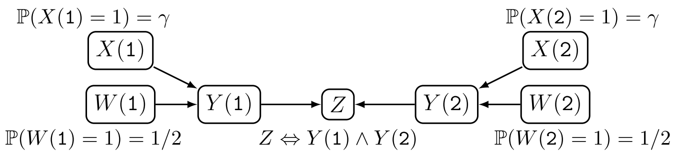


Fig. 9. The grounded network produced in Example 12, for a selection of $\gamma \in [1/4, 3/4]$. Note that $Y(1) \Leftrightarrow (X(1) \wedge W(1)) \vee (\neg X(1) \wedge \neg W(1))$, and likewise for $Y(2)$.

Denote by F_2 the fact that "prisoner P_2 fails to be in the guard's list". While in Example 3 it was convenient to model the information that P_2 was to be released, now it is more convenient to model the information that P_2 is *not* in the guard's list of released prisoners. We wish to compute $\mathbb{P}(E(P_1)|F_2 \wedge U)$.

To model the guard's behavior, introduce:

$$F_2 \Leftrightarrow (E(P_2) \wedge \forall \mathtt{x} \notin \{P_1, P_2\} : \neg E(\mathtt{x})) \vee (D \wedge \forall \mathtt{x} \neq P_1 : \neg E(\mathtt{x})),$$

where D is an auxiliary variable, independent of all $E(\mathtt{x})$.

As in Example 3, we have:

$$\begin{aligned}\mathbb{P}(E(P_1)|F_2 \wedge U) &= \frac{\mathbb{P}(F_2|E(P_1) \wedge U)\,\mathbb{P}(E(P_1)|U)}{\sum_{\mathtt{x}} \mathbb{P}(F_2|E(\mathtt{x}) \wedge U)\,\mathbb{P}(E(\mathtt{x})|U)} \\ &= \frac{\mathbb{P}(F_2|E(P_1) \wedge U)}{1 + \mathbb{P}(F_2|E(P_1) \wedge U)},\end{aligned}$$

where we used $\mathbb{P}(F_2|E(P_2) \wedge U) = 1$, and $\mathbb{P}(F_2|E(\mathtt{x}) \wedge U) = 0$ for $\mathtt{x} \in \{P_3, \ldots, P_N\}$. Moreover, for any element of the complete extension,

$$\begin{aligned}\mathbb{P}(F_2|E(P_1) \wedge U) &= \frac{\mathbb{P}(F_2 \wedge E(P_1) \wedge \forall \mathtt{x} \neq P_1 : \neg E(\mathtt{x}))}{\mathbb{P}(E(P_1) \wedge \forall \mathtt{x} \neq P_1 : \neg E(\mathtt{x}))} \\ &= \frac{\mathbb{P}(D \wedge E(P_1) \wedge \forall \mathtt{x} \neq P_1 : \neg E(\mathtt{x}))}{\mathbb{P}(E(P_1) \wedge \forall \mathtt{x} \neq P_1 : \neg E(\mathtt{x}))} \\ &= \mathbb{P}(D).\end{aligned}$$

Consequently,

$$\mathbb{P}(E(P_1)|F_2 \wedge U) = \frac{\mathbb{P}(D)}{1 + \mathbb{P}(D)}.$$

Suppose $\mathbb{P}(D) = 1/(N-1)$; then $\mathbb{P}(E(P_1)|F_2 \wedge U) = 1/N$. That is, if the guard, when confronted with $N-1$ options, chooses any of them with probability $1/(N-1)$, then the information given to P_1 does not change his probability of execution. Other values of $\mathbb{P}(D)$ do change the probability of execution, as the guard's preferences do yield clues about the execution. For instance, if $\mathbb{P}(D) = 1/2$, then $\mathbb{P}(E(P_1)|F_2 \wedge U) = 1/3$; and if all we know is $\mathbb{P}(D) \in [0, 1]$, then all we can justifiably say is $\mathbb{P}(E(P_1)|F_2 \wedge U) \in [0, 1/2]$. □

While the endpoints of probabilistic assessments in a decoupled credal network can be used to compute lower and upper probabilities, matters are more complicated for coupled credal networks[6]:

Example 12. Consider the following relational credal network:

$$\begin{aligned}\mathbb{P}(W(\mathtt{x})=1) &= 1/2, \\ \mathbb{P}(X(\mathtt{x})=1) &\in [1/4, 3/4], \\ Y(\mathtt{x}) &\Leftrightarrow (X(\mathtt{x}) \wedge W(\mathtt{x})) \vee (\neg X(\mathtt{x}) \wedge \neg W(\mathtt{x})), \\ Z &\Leftrightarrow \forall \mathtt{x} : Y(\mathtt{x}),\end{aligned}$$

and domain $\mathcal{D} = \{1, 2\}$. For any selection $\gamma \in [1/4, 3/4]$, we obtain a Bayesian network as depicted in Fig. 9. Take evidence $\mathbf{E} = \{W(1) = 1, W(2) = 0\}$; then $\mathbb{P}(Z = 1|\mathbf{E}) = \gamma(1-\gamma)$ for any selection of $\gamma \in [1/4, 3/4]$. Consequently, $\overline{\mathbb{P}}(Z = 1|\mathbf{E}) = 1/4$, attained at $\gamma = 1/2$. Note that such a value of γ is not an extreme point of the interval $[1/4, 3/4]$. □

[6] We are indebted to a reviewer who alerted us to the special properties of coupled credal networks (including the fact that upper probabilities may not be attained at endpoints of interval-valued probabilistic assessments). Those remarks prompted us to completely rethink our previous work on the subject [16].

To conclude this section, we note that as before we might consider implication axioms such as

$$X(\mathbb{x}_1,\ldots,\mathbb{x}_k) \Leftarrow \phi(\mathbb{x}_1,\ldots,\mathbb{x}_k,Y_1,\ldots,Y_m),$$

where ϕ, $\mathbb{x}_i$ and Y_j again denote a sentence, logical variables, and parent relations. Syntactically, implication axioms are similar to rules employed in probabilistic logic programming [49–51]. The easiest way to assign semantics to such axioms is to consider the grounded graph and the grounding of every implication axiom. If the probabilistic assessments are given a decoupled semantics, we obtain a credal network whose strong extension can be defined as before. (Of course, it is also possible to concoct a coupled semantics for the probabilistic assessments, but this does not seem as natural when one employs implication axioms.)

We might also allow implication axioms in the "reverse" form

$$X(\mathbb{x}_1,\ldots,\mathbb{x}_k) \Rightarrow \phi(\mathbb{x}_1,\ldots,\mathbb{x}_k,Y_1,\ldots,Y_m).$$

Syntactically, these implication axioms are similar to the *inclusion axioms* found in description logics [3]. Indeed most description logics handle *concepts* (unary relations) and *roles* (binary relations), and allow both for *definitions* and *inclusions*. In the usual notation employed in description logics, a definition such as

$$\mathsf{Woman} \equiv \mathsf{Person} \sqcap \mathsf{Female},$$

states that a woman is a person and a female (in our framework: $\mathsf{Woman}(\mathbb{x}) \Leftrightarrow \mathsf{Person}(\mathbb{x}) \wedge \mathsf{Female}(\mathbb{x})$). An inclusion axiom is typically stated as

$$\mathsf{Concept} \sqsubseteq \mathsf{Description},$$

with the interpretation that an element of the domain that is a Concept must also satisfy Description (in our framework: $\mathsf{Concept}(\mathbb{x}) \Rightarrow \mathsf{Description}(\mathbb{x})$). For instance [56],

$$\mathsf{Shark} \sqsubseteq \mathsf{Fish} \sqcap \mathsf{CanBite} \sqcap \mathsf{IsDangerous}$$

states a few facts about sharks. It should be noted that the combination of description logics and probabilities has received significant attention [41], and many different mixtures have been tried, including ones based on probabilistic choices [17,18,40,54]. We later examine briefly the complexity of relational languages that employ constructs from description logics.

5. The complexity of relational languages

We can now consider inference problems for selected relational languages. The input to our inference problems is a relational credal network, query (an assignment to a ground atom), evidence (a conjunction of assignments to ground atoms), and a list containing the elements of the domain. We assume that the arity of all relations is bounded. We also assume that the ground atoms in the query and in the evidence only mention elements of the domain.

Without loss of generality, the domain could be identified with integers; that is, $\mathcal{D} = \{1, 2, \ldots, \mathtt{N}\}$. Using this convention, a domain could be specified by its size $\mathtt{N}$ in binary notation. For instance, if $\mathtt{N} = 6$, then 110 is a possible binary encoding. In computational terms, binary encoding for $\mathtt{N}$ implies that input is of size $\log \mathtt{N}$, hence calculations should be expected to immediately require exponential effort (as there may be exponentially long numbers in the output) [15]. For this reason, it makes sense to assume that the domain is specified as an explicit list of elements (this is equivalent to specifying $\mathtt{N}$ in unary notation; for instance, if $\mathtt{N} = 6$, then 111111 is a possible unary encoding). In fact we do so in this paper.

We denote by $\mathsf{INF}(\mathcal{L})$ the decision problems for language $\mathcal{L}$, with bounded arity imposed on relations, and with a domain specified by a list, that yield YES in case $\overline{\mathbb{P}}(\mathbf{E}) > 0$ and $\overline{\mathbb{P}}(Q|\mathbf{E}) > \gamma$, and NO otherwise (for rational γ). And we denote by $\mathsf{INF}^+(\mathcal{L})$ the same problems where query and evidence are "positive" (that is, they contain no negation).

Consider then FFFO, the language of function-free first-order logic with equality. We have:

Theorem 13. $\mathsf{INF}(\mathsf{FFFO})$ *is* PSPACE*-complete for decoupled credal networks.*

Proof. To prove membership: Note that there are only a polynomial number of grounded root random variables, because relations have bounded arity. For each root node, we must select either its lower or its upper probability. Run an outer loop over every combination of lower and upper probability for these root nodes, keeping track (using polynomial space) of the result of each iteration. In each iteration, go through every truth assignment of the root nodes (using polynomial space); for each one of them, verify whether $\mathbf{E}$ can be satisfied: there is a polynomial space algorithm to do this, as we basically need to do model checking in first-order logic [37, Section 6.2]. While cycling through truth assignments, keep adding the probabilities of the truth assignments that satisfy $\mathbf{E}$. If the resulting probability for $\mathbf{E}$ is zero, move to the next combination of probability endpoints; otherwise, again go through every truth assignment of the root nodes, now keeping track of how many of them satisfy $\{Q, \mathbf{E}\}$, and adding the probabilities for these assignments. Then divide the probability of $\{Q, \mathbf{E}\}$ by the probability of $\mathbf{E}$, and compare the result with the rational number γ.

To show hardness, note that with a simple network we can encode a PSPACE-complete version of model checking in first-order logic [37, Section 6.5]. To do so, consider the definition $Y \Leftrightarrow Q_1\mathbb{x}_1 : \ldots Q_n\mathbb{x}_n : \phi(\mathbb{x}_1,\ldots,\mathbb{x}_n)$, where each Q_i is a quantifier (either ∀ or ∃) and ϕ is a quantifier-free formula containing only Boolean operators, a unary relation X and $\mathbb{x}_1, \dots, \mathbb{x}_n$. The relation X is associated with assessment $\mathbb{P}(X(\mathbb{x}) = 1) = 1/2$ (note that there is no need for interval-valued assessments). Take domain $\mathcal{D} = \{0, 1\}$ and evidence $\mathbf{E} = \{X(0) = 0, X(1) = 1\}$. Then $\mathbb{P}(Y = 1|\mathbf{E}) > 1/2$ iff $Q_1\mathbb{x}_1 : \dots Q_n\mathbb{x}_n : \phi(\mathbb{x}_1, \dots, \mathbb{x}_n)$ is satisfiable. Determining such a satisfiability question is a PSPACE-complete problem. □

To obtain more insightful results concerning complexity, we have previously proposed an analysis with respect to *query complexity*[7] and to *domain complexity* [15]. We now present similar results for relational credal networks.

We refer to the complexity of computing a conditional probability, given a relational credal network, query, evidence, and domain, as the *inferential complexity*. Theorem 13 deals with inferential complexity. We refer to the complexity of computing a conditional probability, for a fixed relational credal network, when query, evidence and domain are inputs, as the *query complexity*. And we refer to the complexity of computing a conditional probability, for a fixed relational credal network and fixed query and evidence, when the domain is the input, as the *domain complexity*.[8]

We use QINF($\mathcal{L}$) to indicate the query complexity of relational credal networks specified through language $\mathcal{L}$. We have:

Theorem 14. QINF(FFFO) *is* $\mathsf{NP}^{\mathsf{PP}}$*-complete for decoupled credal networks.*

Proof. To prove membership: Take a "base" polynomial time nondeterministic Turing machine that (nondeterministically) selects either the lower or the upper probability for each one of the (polynomially many) grounded root nodes. Now the base machine resorts to an auxiliary Turing machine (the oracle) that receives the resulting probabilistic assessments and logical formulas and decides whether $\mathbb{P}(Q|\mathbf{E}) > \gamma$. To do so, we adapt a proof by Park on the complexity of Bayesian networks [19, Theorem 11.5]. The proof relies on the following numbers, whose purpose will be clarified later:

$$a = \begin{cases} 1 & \text{if } \gamma < 1/2, \\ 1/(2\gamma) & \text{otherwise,} \end{cases} \qquad b = \begin{cases} (1-2\gamma)/(2-2\gamma) & \text{if } \gamma < 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

The base machine computes the smallest integers a_1, a_2, b_1, b_2 such that $a = a_1/a_2$ and $b = b_1/b_2$, and sends these values to the auxiliary machine as additional input.

The auxiliary machine then in sequence guesses a truth assignment for each grounded root node (and writes the guess in the working tape). Note that each grounded root node X is associated with an assessment $\mathbb{P}(X = 1) = c/d$, where c and d are the smallest such integers. If $c/d = 1/2$, then the auxiliary machine has simply two computation paths immediately after guessing the value of X: one in which it writes one possible value in the tape and proceeds, the other in which it writes the other value and proceeds. Now suppose c/d is some other rational. Then the auxiliary machine replicates its computation paths out of the guess on X: there will be d paths, and c of them proceed as established after guessing $\{X = 1\}$, while the other $d - c$ paths proceed as established after guessing $\{X = 0\}$.

The auxiliary machine verifies whether the guessed truth assignment satisfies $\mathbf{E}$ (there is a logarithmic space, hence polynomial time, algorithm that can verify whether $\mathbf{E}$ holds once the root nodes are set [37, Section 6.2]). If the truth assignment for root nodes does not satisfy $\mathbf{E}$, then branch into a_2b_2 computation paths that immediately stop and output YES, and a_2b_2 computation paths that immediately stop and output NO. If instead the truth assignment does satisfy $\mathbf{E}$, then verify whether the truth assignment satisfies $\{Q, \mathbf{E}\}$. If the truth assignment does not satisfy $\{Q, \mathbf{E}\}$, it satisfies $\{\neg Q, \mathbf{E}\}$; in this case branch into $2a_2b_1$ paths that immediately stop and output YES, and $2(b_2 - b_1)a_2$ paths that immediately stop and output NO. And if the assignment satisfies $\{Q, \mathbf{E}\}$, then branch into $2a_1b_2$ paths that immediately stop and output YES, and $2(a_2 - a_1)b_2$ paths that immediately stop and output NO. We claim that more than half of the computation paths of this Turing machine yield YES iff $\mathbb{P}(Q|\mathbf{E}) > \gamma$ (with respect to the fixed probabilities selected nondeterministically by the base machine). Given this claim, the initial nondeterministic selection of probabilities by the base machine can be checked using the auxiliary machine as an oracle, and membership to $\mathsf{NP}^{\mathsf{PP}}$ is established. (Note that if $\overline{\mathbb{P}}(\mathbf{E}) = 0$, then every selection of lower/upper probabilities by the base machine leads to a call to the auxiliary machine that yields NO, as there will be a_2b_2 paths that yield YES and a_2b_2 paths that yield NO in the auxiliary machine. So, if $\overline{\mathbb{P}}(\mathbf{E}) = 0$ the base machine stops with NO, as desired.)

Thus the remaining claim to prove is whether more than half of the computation paths of the auxiliary machine yield YES iff $\mathbb{P}(Q|\mathbf{E}) > \gamma$. Denote by N the number of truth assignments that can be selected by the auxiliary machine, and note that $N = N_1 + N_2 + N_3$, where N_1 is the number of assignments that do not satisfy $\mathbf{E}$, N_2 is the number of assignments that satisfy $\{\neg Q, \mathbf{E}\}$, and N_3 is the number of assignments that satisfy $\{Q, \mathbf{E}\}$. Note that by construction we have $N_1/N = 1 - \mathbb{P}(\mathbf{E})$, $N_2/N = \mathbb{P}(\neg Q, \mathbf{E})$, and $N_3/N = \mathbb{P}(Q, \mathbf{E})$. The number of computation paths that yield YES is $a_2b_2N_1 + 2a_2b_1N_2 + 2a_1b_2N_3$, and the total number of computation paths is $a_2b_2N_1 + a_2b_2N_1 + 2b_1a_2N_2 + 2(b_2 - b_1)a_2N_2 + 2a_1b_2N_3 + 2(a_2 - a_1)b_2N_3 = 2a_2b_2N$. Hence the number of paths that yield YES divided by the total number of paths is

$$(N_1(1/2) + (b_1/b_2)N_2 + (a_1/a_2)N_3)/N = (1 - \mathbb{P}(\mathbf{E}))/2 + b\mathbb{P}(\neg Q, \mathbf{E}) + a\mathbb{P}(Q, \mathbf{E}),$$

7 In our previous work we used the term *data complexity*, but *query complexity* seems more appropriate.

8 Query and domain complexity are respectively related to the existing notions of lqe-liftability and liftability [32,58]; lqe-liftability means that query complexity is polynomial, and liftability means that domain complexity is polynomial.

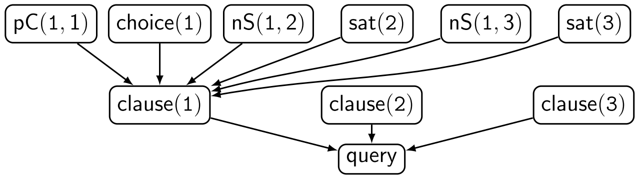


Fig. 10. The grounding of the relational credal network that solves E-MAJSAT with $k = 1$ and ϕ in Expression (19). Only ancestors of clause(1) are shown; groundings of positiveSat, negativeSat, positiveChoice and negativeChoice that are set to false are not shown (and the other ones have their names abbreviated); only groundings of choice and sat that do affect clause(1) are shown.

and this quantity is strictly larger than 1/2 iff

$$a\mathbb{P}(Q, \mathbf{E}) + b\mathbb{P}(\neg Q, \mathbf{E}) > \mathbb{P}(\mathbf{E})/2 \qquad \text{iff} \qquad a\mathbb{P}(Q|\mathbf{E}) + b\mathbb{P}(\neg Q|\mathbf{E}) > 1/2$$

as we can assume that $\mathbb{P}(\mathbf{E}) > 0$ (otherwise the output is NO as already described). If $\gamma < 1/2$, the latter inequality is equivalent to

$$\mathbb{P}(Q|\mathbf{E}) + (1 - 2\gamma)/(2 - 2\gamma)(1 - \mathbb{P}(Q|\mathbf{E})) > 1/2$$

and this is equivalent to $\mathbb{P}(Q|\mathbf{E}) > \gamma$; if instead $\gamma \geq 1/2$, we likewise obtain the inequality $(1/(2\gamma))\mathbb{P}(Q|\mathbf{E}) > 1/2$, equivalent to $\mathbb{P}(Q|\mathbf{E}) > \gamma$. Hence the number of computation paths (of the auxiliary Turing machine) that yield YES is larger than the total number of computation paths iff $\mathbb{P}(Q|\mathbf{E}) > \gamma$, as desired. This completes the proof of membership.

To prove hardness, we adapt the proof of a similar result for Bayesian networks [15]. Take an E-MAJSAT problem with pair (ϕ, k), where ϕ is in CNF with m clauses and propositions $A_1, \ldots, A_n$; each clause has up to n literals (or is trivially true and can be replaced say by $A_1 \vee \neg A_1$). If the number of clauses m is smaller than n, then add trivial clauses such as $A_1 \vee \neg A_1$ until $m = n$. These clauses do not change the output of MAJSAT. If instead $n < m$, then add fresh propositions $A_{n+1}, \ldots, A_m$. These propositions do not change the output of E-MAJSAT. Introduce unary relations sat($\mathcal{x}$) and choice($\mathcal{x}$); impose $\mathbb{P}(\text{sat}(\mathcal{x})) = 1/2$, $\mathbb{P}(\text{choice}(\mathcal{x})) \in [0, 1]$.

To understand the idea, take a domain $\{\mathtt{1}, \ldots, \mathtt{n}\}$ containing n elements (i.e., $\mathtt{n}$ represents n); given our previous discussion we have $n = m$. The elements of the domain serve a dual purpose, indexing both propositions and clauses. Then sat($\mathcal{x}$) refers to proposition $A_{\mathcal{x}}$ for $\mathcal{x}$ in $\{\mathtt{k+1}, \ldots, \mathtt{n}\}$, while choice($\mathcal{x}$) refers to proposition $A_{\mathcal{x}}$ for $\mathcal{x}$ in $\{\mathtt{1}, \ldots, \mathtt{k}\}$.

Now introduce four binary relations positiveSat($\mathcal{x}, \mathcal{y}$), negativeSat($\mathcal{x}, \mathcal{y}$), positiveChoice($\mathcal{x}, \mathcal{y}$), negativeChoice($\mathcal{x}, \mathcal{y}$). Adopt $\mathbb{P}\left(\text{positiveSat}(\mathcal{x}, \mathcal{y})\right) = 1/2$ and likewise for the other three relations.

Finally, introduce:

$$\begin{aligned}\text{clause}(\mathcal{x}) \Leftrightarrow \exists \mathcal{y} : &(\text{positiveSat}(\mathcal{x}, \mathcal{y}) \wedge \text{sat}(\mathcal{y}))\\ &\vee\ (\text{negativeSat}(\mathcal{x}, \mathcal{y}) \wedge \neg\text{sat}(\mathcal{y}))\\ &\vee\ (\text{positiveChoice}(\mathcal{x}, \mathcal{y}) \wedge \text{choice}(\mathcal{y}))\\ &\vee\ (\text{negativeChoice}(\mathcal{x}, \mathcal{y}) \wedge \neg\text{choice}(\mathcal{y})),\\ \text{query} \Leftrightarrow \forall \mathcal{x} : &\ \text{clause}(\mathcal{x}).\end{aligned}$$

Take evidence $\mathbf{E}$ as follows. For each clause, run over the literals. Consider the ith clause, and its non-negated literal A_j. Then if $j > k$, set positiveSat(i, j) to true; otherwise, set positiveChoice(i, j) to true. And consider negated literal $\neg A_j$. Then if $j > k$, set negativeSat(i, j) to true; otherwise, set negativeChoice(i, j) to true. Set all other groundings of positiveSat, negativeSat, positiveChoice and negativeChoice to false. Clearly, $\overline{\mathbb{P}}(\mathbf{E}) = \underline{\mathbb{P}}(\mathbf{E}) = 2^{-4n^2} > 0$.

Now decide whether $\overline{\mathbb{P}}(\text{query} = 1|\mathbf{E}) > 1/2$. If YES, the E-MAJSAT problem is accepted, if NO, it is not accepted. Hence we have the desired reduction. The reduction is polynomial, as the domain size is directly related to the input, and the evidence is quadratic in the domain size; the number of relations is fixed (i.e., it does not depend on the input).

To better understand the construction just described, consider an E-MAJSAT problem with $k = 1$ and

$$\phi = (A_1 \vee \neg A_2 \vee \neg A_3) \wedge (\neg A_1 \vee A_2 \vee A_2). \tag{19}$$

Note that when A_1 is false, we have more than half of assignments to A_2 and A_3 such that ϕ is true; hence the desired response here is YES. Fig. 10 shows a portion of the network produced by our construction. Note that a meaningless clause $A_1 \vee \neg A_1$ is assumed appended to ϕ. We note that for this decoupled credal network we have $\overline{\mathbb{P}}(\text{query} = 1|\mathbf{E}) = 3/4 > 1/2$, as expected. □

To recap, we have seen that decoupled relational networks expressed in FFFO have PSPACE-complete inferential complexity and NP$^{\text{PP}}$-complete query complexity. It is interesting to investigate whether tractability can be achieved by restricting the specification language, as we have done before for propositional languages. We address these questions by considering fragments of FFFO inspired by popular description logics.

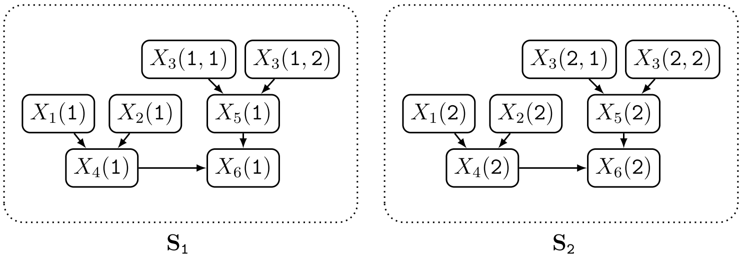


Fig. 11. Grounding assessments (10), (11), (13), (15), (20) and (21) with respect to domain $\mathcal{D} = \{1, 2\}$.

So consider the restricted first-order syntax allowing only:

- unary relations $X(\mathscr{x})$ and binary relations $X(\mathscr{x}, \mathscr{y})$,
- definition axioms of the type

$$X(\mathscr{x}) \Leftrightarrow Y_1(\mathscr{x}) \wedge \ldots \wedge Y_m(\mathscr{x}),$$

  where $Y_1, \ldots, Y_m$ are the parents of X, and of the type

$$X(\mathscr{x}) \Leftrightarrow \exists \mathscr{y} : Y(\mathscr{x}, \mathscr{y}),$$

- and (interval-valued) probabilistic assessments

$$\mathbb{P}(X(\mathscr{x}) = 1) \in [\alpha, \beta] \quad \text{and} \quad \mathbb{P}\left(X(\mathscr{x}, \mathscr{y}) = 1\right) \in [\alpha, \beta].$$

This language was called DLDiet in Ref. [15], as it is inspired by the popular description logic DL-Lite [6,7].

Example 15. Consider again Example 10, and substitute Expression (14) by axioms

$$X_5(\mathscr{x}) \Leftrightarrow \exists \mathscr{y} : X_3(\mathscr{x}, \mathscr{y}), \tag{20}$$

$$X_6(\mathscr{x}) \Leftrightarrow X_4(\mathscr{x}) \wedge X_5(\mathscr{x}). \tag{21}$$

These axioms are sentences in DLDiet. A grounded credal network obtained with the network specified through (10), (11), (13), (15), (20) and (21) is shown in Fig. 11. □

As in the example above, the grounding of a relational credal network specified using DLDiet contains disconnected components. That is, each element a of the domain produces a "slice" $\mathbf{S}_a$ containing groundings of unary predicates such as $X(a)$, and groundings of binary predicates of the form $Y(a, b)$, where b ranges over the domain. The dotted rectangles in Fig. 11 represent the slices $\mathbf{S}_1$ and $\mathbf{S}_2$, relative to elements 1 and 2 of the domain, respectively. Thus inference can focus on a single "slice" and this leads to tractable inference, as the following result shows.

Theorem 16. INF$^+$(DLDiet) *can be solved in polynomial time for decoupled credal networks.*

Proof. We prove the result by deriving a polynomial-time computable expression for $\overline{\mathbb{P}}(Q\,|\mathbf{E})$. We abuse the notation and use the same symbol to refer to a set of (positive) assignments and to its corresponding set of variables. Consider the grounded graph and an extreme probability measure $\mathbb{P}$ in the strong extension. Let $\mathbf{U}$ contain all groundings of unary relations in the network, $\mathbf{U}^r$ be the subset of $\mathbf{U}$ containing only ground atoms with no parents, $\mathbf{B}$ be the set of all groundings of binary relations, and $\mathbf{U}^\exists$ be the subset of $\mathbf{U}$ containing only ground atoms whose parents are in $\mathbf{B}$. For instance, in the grounded network of Fig. 11, $\mathbf{U} = \{X_1(1), X_1(2), X_2(1), X_2(2), X_4(1), X_4(2), X_5(1), X_5(2), X_6(1), X_6(2)\}$, $\mathbf{U}^r = \{X_1(1), X_1(2), X_2(1), X_2(2)\}$, $\mathbf{B} = \{X_3(1,1), X_3(1,2), X_3(2,1), X_3(2,2)\}$, and $\mathbf{U}^\exists = \{X_5(1), X_5(2)\}$. Assume at first that $\{Q\} \cap \mathbf{B} = \emptyset$. Suppose there is a ground atom $X \in \mathbf{U} \setminus (\mathbf{U}^r \cup \mathbf{U}^\exists)$ that appears and has no descendants in $\{Q\} \cup \mathbf{E}$. Let $\mathbf{Y}$ be the assignment $\{Y = 1 | Y \in \mathrm{pa}(X)\}$, and set $\mathbf{Z} = (\{Q\} \cup \mathbf{E}) \setminus \{X = 1\}$. Since $X \Leftrightarrow \bigwedge_{Y \in \mathrm{pa}(X)} Y$, it follows that $\mathbb{P}(\{Q\} \cup \mathbf{E}) = \mathbb{P}(X = 1|\mathbf{Y})\,\mathbb{P}(\mathbf{Y} \cup \mathbf{Z}) = \mathbb{P}(\mathbf{Y} \cup \mathbf{Z})$. Add $\mathbf{Y}$ to $\mathbf{Z}$, and repeat this operation while possible. We conclude that

$$\begin{aligned}\mathbb{P}(\{Q\} \cup \mathbf{E}) &= \mathbb{P}\left(\left(\mathrm{An}(\{Q\} \cup \mathbf{E}) \cap \mathbf{U}^\exists\right) \cup (\mathbf{E} \cap \mathbf{B}) \cup \left(\mathrm{An}(\{Q\} \cup \mathbf{E}) \cap \mathbf{U}^r\right)\right) \\ &= \prod_{\substack{X \in \mathrm{An}(\{Q\} \cup \mathbf{E}) \cap \mathbf{U}^\exists, \\ \mathrm{pa}(X) \cap \mathbf{E} = \emptyset}} \mathbb{P}(X) \prod_{X \in \mathbf{E} \cap \mathbf{B}} \mathbb{P}(X) \prod_{X \in \mathrm{An}(\{Q\} \cup \mathbf{E}) \cap \mathbf{U}^r} \mathbb{P}(X),\end{aligned}$$

where An(**X**) denotes the ancestors of the ground atoms appearing in a set of assignments **X**. By a similar reasoning we see that

$$\mathbb{P}(\mathbf{E}) = \prod_{\substack{X \in \mathrm{An}(\mathbf{E}) \cap \mathbf{U}^{\exists},\\ \mathrm{pa}(X) \cap \mathbf{E} = \emptyset}} \mathbb{P}(X) \prod_{X \in \mathbf{E} \cap \mathbf{B}} \mathbb{P}(X) \prod_{X \in \mathrm{An}(\mathbf{E}) \cap \mathbf{U}^{r}} \mathbb{P}(X),$$

whence $\overline{\mathbb{P}}(\mathbf{E}) = \prod_{\substack{X \in \mathrm{An}(\mathbf{E}) \cap \mathbf{U}^{\exists},\\ \mathrm{pa}(X) \cap \mathbf{E} = \emptyset}} \overline{\mathbb{P}}(X) \prod_{X \in \mathbf{E} \cap \mathbf{B}} \overline{\mathbb{P}}(X) \prod_{X \in \mathrm{An}(\mathbf{E}) \cap \mathbf{U}^{r}} \overline{\mathbb{P}}(X)$. Note that we can verify if this latter expression is non-null efficiently by inspecting whether any X in $\mathbf{E} \cap \mathbf{B}$ or $\mathrm{An}(\mathbf{E}) \cap \mathbf{U}^{r}$ has upper probability zero, or if all parents of $X \in \mathrm{An}(\mathbf{E}) \cap \mathbf{U}^{\exists}$ have upper probability zero. Let $\mathbf{A} = \mathrm{An}(Q) \setminus \mathrm{An}(\mathbf{E})$ be the ancestors of Q that are not ancestors of variables in **E**. We have that

$$\mathbb{P}(Q \,|\mathbf{E}) = \mathbb{P}(\{Q\} \cup \mathbf{E}) / \mathbb{P}(\mathbf{E}) = \prod_{\substack{X \in \mathbf{A} \cap \mathbf{U}^{\exists},\\ \mathrm{pa}(X) \cap \mathbf{E} = \emptyset}} \mathbb{P}(X) \prod_{X \in \mathbf{A} \cap \mathbf{U}^{r}} \mathbb{P}(X),$$

hence $\overline{\mathbb{P}}(Q \,|\mathbf{E}) = \prod_{\substack{X \in \mathbf{A} \cap \mathbf{U}^{\exists},\\ \mathrm{pa}(X) \cap \mathbf{E} = \emptyset}} \overline{\mathbb{P}}(X) \prod_{X \in \mathbf{A} \cap \mathbf{U}^{r}} \overline{\mathbb{P}}(X)$. Note that all ground atoms in $\mathbf{A} \cap \mathbf{U}$ refer to the same element of the domain, say a. Thus, the sets $\mathbf{A} \cap \mathbf{U}^{r}$ and $\mathbf{A} \cap \mathbf{U}^{\exists}$ can be constructed by inspecting the relational graph without resorting to grounding. Consider $X(a) \in \mathbf{U}^{\exists}$. By construction, its upper probability is

$$\begin{aligned}\overline{\mathbb{P}}(X(a) = 1) &= \overline{\mathbb{P}}\left(\exists y : Y(a, y) = 1\right) = 1 - \underline{\mathbb{P}}\left(\forall y : Y(a, y) = 0\right) \\ &= 1 - \left(1 - \overline{\mathbb{P}}\left(Y(\mathbf{x}, y) = 1\right)\right)^{N},\end{aligned}$$

where we have used the independence of root nodes in the grounded network in the last equality.

It remains to address the case when $Q = \{X(a, b) = 1\}$ for $X(a, b) \in \mathbf{B}$. First, we check whether $\overline{\mathbb{P}}(\mathbf{E}) > 0$, as before. If that is the case, we proceed. If $X(a, b)$ is not an ancestor of **E**, then $\overline{\mathbb{P}}(Q \,|\mathbf{E}) = \overline{\mathbb{P}}(Q) = \overline{\mathbb{P}}\left(X(\mathbf{x}, y)\right)$, which is part of the input. Otherwise, we can apply the same reasoning as before and assume that the evidence contains the assignment $\{X'(a) = 1\}$, where $X'(\mathbf{x}) \Leftrightarrow \exists y : X(\mathbf{x}, y)$. Let $\mathbf{E}'$ be the intersection of the groundings of $X(a, y)$ and **E**. Since $X'(a)$ d-separates $X(a, b)$ and $\mathbf{E} \setminus \mathbf{E}'$, we have that $\overline{\mathbb{P}}(Q \,|\mathbf{E}) = \overline{\mathbb{P}}\left(Q \,|\{X'(a) = 1\}, \mathbf{E}'\right)$. Now, if $\mathbf{E}'$ is non-empty, then $\mathbb{P}\left(X'(a) = 1 | \mathbf{E}', Q\right) = 1$, and (by Bayes' rule) $\overline{\mathbb{P}}(Q \,|\mathbf{E}) = \overline{\mathbb{P}}\left(X(\mathbf{x}, y = 1)\right)$. If $\mathbf{E}' = \emptyset$, then (also by Bayes' rule):

$$\overline{\mathbb{P}}\left(Q \,| X'(a) = 1\right) = \left(1 + \underline{\mathbb{P}}\left(X'(a) = 1 | X(a, b) = 0\right) \frac{1 - \overline{\mathbb{P}}\left(X(\mathbf{x}, y) = 1\right)}{\overline{\mathbb{P}}\left(X(\mathbf{x}, y) = 1\right)}\right)^{-1}.$$

The value of $\underline{\mathbb{P}}\left(X'(a) = 1 | X(a, b) = 0\right)$ can be deduced from a simple counting argument: $\underline{\mathbb{P}}\left(X'(a) = 1 | X(a, b) = 0\right) = 1 - \left(1 - \underline{\mathbb{P}}\left(X(\mathbf{x}, y) = 1\right)\right)^{N-1}$. Hence, $\overline{\mathbb{P}}(Q \,|\mathbf{E})$ can be computed in polynomial time. $\square$

The restriction to positive assignments is important, as allowing "negative" evidence produces a PP-hard problem (as noted right after Theorem 5). However, query complexity remains tractable if we allow negation in the language. So consider a generalization of DLDiet that allows also negation in the definition axioms. Call this language DLDiet$^{\neg}$. As this language is a superset of Prop$(\wedge, \neg)$, INF(DLDiet$^{\neg}$) is NP$^{\mathsf{PP}}$-hard for decoupled networks (by Theorem 4). Query complexity, on the other hand, remains tractable.

Theorem 17. QINF(DLDiet$^{\neg}$) *is computable in polynomial time for decoupled relational networks.*

Proof. We construct a polynomial algorithm for $\overline{\mathbb{P}}\left(X_Q(a) = 1 | \mathbf{E}\right)$. The case for query $\{X_Q(a, b) = 1\}$ is analogous. The proof repeatedly uses the fact that the number of relations M in the network is assumed constant, hence, any computation that takes time $O(f(M))$, for any function f is considered constant.

Recall that the grounding of a relational credal network specified using DLDiet consists of "slices" (i.e., disconnected components), each slice containing unary atoms of the form $X(a)$ for some element a of the domain, and binary atoms $X(a, b)$ for all elements b. We will show that inference can be carried out on each slice independently, and only on slices containing evidence and/or query. And since each slice can be computed efficiently (once the number of relations is fixed), the result follows.

To see that this is indeed true, assume, without loss of generality, that the domain is a list of integers $1, 2, \ldots, \mathtt{N}$, and that the relative slices of the grounded network are $\mathbf{S}_1, \ldots, \mathbf{S}_{\mathtt{N}}$. It follows that every extreme distribution of the strong extension factorizes as $\prod_{a=1}^{\mathtt{N}} \mathbb{P}(\mathbf{S}_a)$. Now consider a single slice $\mathbf{S}_a$ relative to some element a. Call **U** the nodes of the form $X(a)$ (i.e., groundings of unary relations) in slice $\mathbf{S}_a$, and let $\mathbf{U}^{\exists}$ be the subset of **U** containing all nodes whose parents are groundings of the form $X(a,b)$ for some element b. Consider the credal network induced by a node $X(a)$ in $\mathbf{U}^{\exists}$ and its parents. The graph of this network is an inverted tree, with leaf $X(a)$ and roots $X(a,b)$. For example, in the network in Fig. 11, the set $\mathbf{U}^{\exists}$ relative to slice $\mathbf{S}_1$ contains only $X_5(1)$; its induced network is composed of node $X_5(1)$ and its parents $X_3(1,1)$ and $X_3(1,2)$. The upper and lower probabilities for such $X(a)$ can be computed in polynomial time in the size of the input (which includes a list of atoms in the evidence and the a list of the elements of the domain), as in the proof of Theorem 16 (alternatively, we could run the 2U algorithm [25]). And since the number of relations M is assumed constant, we can perform such a computation for every node in $\mathbf{U}^{\exists}$ still in polynomial time (there at most $O(M)$ nodes in $\mathbf{U}^{\exists}$). Now collect all nodes in $\mathbf{U}$. The collected nodes induce a credal network with $O(M)$ nodes. Hence, we can enumerate in polynomial time all joint distributions $\mathbb{P}(\mathbf{U}) = \mathbb{P}\left(\mathbf{U} \setminus \mathbf{U}^{\exists} | \mathbf{U}^{\exists}\right) \prod_{X \in \mathbf{U}^{\exists}} \mathbb{P}(X)$ induced by a choice of upper and lower probabilities of root nodes of the network induced by $\mathbf{U}$ (note that $\mathbf{U}^{\exists}$ are root nodes in this network).

To check whether $\overline{\mathbb{P}}(\mathbf{E})$ is positive go through every slice that contains some atom in the evidence. For each slice, enumerate all respective joint distributions and compute the maximum probability of $\mathbf{E} \cap \mathbf{S}_a$. If any slice has upper probability of $\mathbf{E} \cap \mathbf{S}_a$ zero, halt and return NO. Otherwise, we have established that $\overline{\mathbb{P}}(\mathbf{E}) > 0$. So consider the slice $\mathbf{S}_a$ where a is the element in the query $X_q(a)$. Again, obtain all joint distributions over variables in that slice and for each distribution compute $\mathbb{P}(Q|\mathbf{E} \cap \mathbf{S}_a)$. The solution is the maximum over $\mathbb{P}(Q|\mathbf{E} \cap \mathbf{S}_a)$. □

The proof above also shows that inference is computable in linear time in the domain size for credal networks specified in DLDiet$^{\neg}$. Hence, the domain complexity of such networks is polynomial.

We have so far focused on the complexity of decoupled credal networks. We now turn attention to coupled credal networks. As noted before, in coupled credal networks upper probabilities may be attained at probability distributions that are *not* products of extreme points of local credal sets (Example 12). Hence we cannot repeat all proofs used for decoupled credal networks. To proceed, we adopt a simplified framework. The true analysis of decoupled networks is a challenging endeavor that we leave as future work.

The framework we adopt is as follows. Once more, we assume that a directed acyclic graph is given, where each node is a relation of bounded arity and such that

- if X is a root node then X is associated with a *finite* set of probability values $\mathbb{P}(X(\varkappa_1,\dots,\varkappa_k)=1)$; and
- if X is not a root node, then X is associated with a definition axiom, as before.

The semantics of this credal network is similar to the previous semantics of coupled networks, except that we consider only the Bayesian networks obtained as the product of probability measures associated with the root nodes (so that we do not assume convexity nor closure). Since these measures are finite, there is a finite (but exponential) number of such Bayesian networks. An inference is simply a conditional obtained from this set (over the set of measures that assign positive probability to the evidence).

Then we obtain:

Theorem 18. INF(FFFO) *is* PSPACE-*complete for coupled credal networks specified by a finite set of probability values.*

Proof. The proof of membership uses the same arguments as the proof of Theorem 13; the only difference is that we must select probability values instead of extreme points. The proof of hardness is simply a copy of the proof of hardness in Theorem 13. □

Theorem 19. QINF(FFFO) *is* PP-*complete for coupled credal networks specified by a finite set of probability values.*

Proof. PP-hardness is obtained by encoding an E-MAJSAT problem without "choice" propositions in the proof of Theorem 14 (thus obtaining a MAJSAT problem). Note that MAJSAT is a PP-complete problem [57]. To prove pertinence to PP, we will use the fact that PP is closed under union, a celebrated result in complexity theory [4]. Take a fixed relational credal network and note that there is a fixed (and possibly large) number of relational Bayesian networks that can be generated by selecting, for each root node, one of the finitely many probability values associated (the number of Bayesian networks, say M, is exponential in the number of root nodes n, but n is a fixed quantity in this analysis). Each one of these M relational Bayesian networks specifies a language that consists of a set of strings: these are the strings consisting of domain $\mathcal{D}$, query Q, and evidence $\mathbf{E}$, and that satisfy $\overline{\mathbb{P}}(\mathbf{E}) > 0$ and $\overline{\mathbb{P}}(Q|\mathbf{E}) > 0$. So our overall problem is: given an input string with $\mathcal{D}$, Q, and $\mathbf{E}$, accept it if any one of those M sets of strings contains it. But note that each set of strings defines a PP-complete decision problem (the problem of accepting the strings), as each relational Bayesian network can be grounded into a polynomially larger Bayesian network, and inference can be conducted in the latter network. So our main problem is to consider a set of strings that is the union of the M sets of strings; because PP is closed under union, the main problem is in PP as well. □

We finish by returning to the DLDiet language. First, we note that polynomial inference is still obtained, even for coupled networks:

Theorem 20. $\mathsf{INF}^+(\mathsf{DLDiet})$ *can be solved in polynomial time for coupled credal networks.*

Proof. Repeat the proof of Theorem 16 in its entirety. At the end, note that, because all the ground atoms for a relation are assigned the same probability, the result is the same irrespective of whether the coupled or decoupled semantics is adopted (note that as the decoupled semantics is a "relaxed" version of the coupled semantics, inference with the former produces an outer approximation for the latter). □

And we present the analogue to Theorem 17:

Theorem 21. $\mathsf{QINF}^+(\mathsf{DLDiet}^\neg)$ *is computable in polynomial time for coupled credal networks specified by a finite set of probability values.*

Proof. The proof is similar to the proof of Theorem 19. For each relation, enumerate all the possible values for probabilities associated to root nodes. Since the model is fixed, this takes polynomial time. For each such configuration, we obtain a Bayesian network where inference is polynomial [15]. □

6. Conclusion

In this paper we have explored the balance of expressivity and complexity in credal networks. In a recent paper, we have proposed a framework for the analogue analysis of Bayesian networks [15], so the present paper can be understood as a first step in extending that framework to credal networks.

We have here discussed both propositional and relational languages, and for relational languages we have studied inferential and query complexities. Theorems 5 and 16 reveal classes of credal networks that admit polynomial inference, a property shared by few other classes [20]; the results are surprising in that they reproduce the polynomial character of Bayesian networks under the same languages. We believe that applications may benefit from the probabilistic version of DL-Lite that is discussed in Theorem 16, as many practical ontologies use relatively simple constructs that are captured by DL-Lite [6]. And in the opposite direction, Theorems 14 and 19 show intriguing distinctions between Bayesian and credal networks, as in the latter case there is more than one reasonable semantics to choose from, and the choice does have an impact on complexity (under some conditions).

Perhaps the most compelling aspect of our framework is the number of questions it raises. Consider a simple fact. It is usually assumed that one can arbitrarily choose between computing an upper or a lower probability, as they are directly related by $\overline{\mathbb{P}}(A|B) = 1 - \underline{\mathbb{P}}\left(A^c|B\right)$ [59], where A and B are events and A^c is the complement of A. But if a language does not have negation, it may not be possible to formulate $\underline{\mathbb{P}}\left(A^c|B\right)$ as a query, and it may then be harder to produce a lower probability than an upper probability. This sort of phenomena can only be explored when we pay attention to the specification language. In fact, one might argue that the key difference between Bayesian and credal networks lies in the language that is used to express assessments.

There are many additional languages to explore concerning the complexity of credal networks. For instance, there are several fragments of function-free first-order logic that are widely used, such as monadic logic [5]; there are guarded fragments and description logics such as $\mathcal{EL}$ and $\mathcal{ALC}$ [3]; there are also languages based on second-order logic and various modal logics. For all these logical languages, one can investigate inferential and query complexity, not only for inference, but also for other problems of common interest such as maximum a posteriori configurations [23]. All such questions await detailed investigation.

Acknowledgements

The first author is partially supported by CNPq (grant 305395/2010-6). The second author is partially supported by the São Paulo Research Foundation (FAPESP, grant #2016/01055-1).

We would like to thank Jasper De Bock for an extremely careful review of this paper, and for suggesting many improvements that we have gladly adopted. We also thank the reviewers of this paper, who made very useful comments and led us to improve the paper significantly.

References

[1] K.A. Andersen, J.N. Hooker, Bayesian logic, Decis. Support Syst. 11 (1994) 191–210.
[2] T. Augustin, F.P.A. Coolen, G. de Cooman, M.C.M. Troffaes, Introduction to Imprecise Probabilities, Wiley, 2014.
[3] F. Baader, W. Nutt, Basic description logics, in: Description Logic Handbook, Cambridge University Press, 2002, pp. 47–100.
[4] R. Beigel, N. Reingold, D. Spielman, PP is closed under intersection, J. Comput. Syst. Sci. 50 (2) (1995) 191–202.
[5] E. Börger, E. Grädel, Y. Gurevich, The Classical Decision Problem, Springer, 1997.
[6] D. Calvanese, G.D. Giacomo, D. Lembo, M. Lenzerini, R. Rosati, DL-Lite: tractable description logics for ontologies, in: AAAI, 2005, pp. 602–607.
[7] D. Calvanese, G.D. Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: the DL-Lite family, J. Autom. Reason. 39 (2007) 385–429.
[8] G. Coletti, R. Scozzafava, Probabilistic Logic in a Coherent Setting, Trends Log., vol. 15, Kluwer, Dordrecht, 2002.

[9] F.G. Cozman, Credal networks, Artif. Intell. 120 (2000) 199–233.
[10] F.G. Cozman, Graphical models for imprecise probabilities, Int. J. Approx. Reason. 39 (2–3) (2005) 167–184.
[11] F.G. Cozman, Sets of probability distributions, independence, and convexity, Synthese 186 (2) (2012) 577–600.
[12] F.G. Cozman, Independence for full conditional probabilities: structure, factorization, non-uniqueness, and Bayesian networks, Int. J. Approx. Reason. 54 (2013) 1261–1278.
[13] F.G. Cozman, C.P. de Campos, J.C.F. da Rocha, Probabilistic logic with independence, Int. J. Approx. Reason. 49 (2008) 3–17.
[14] F.G. Cozman, C.P. de Campos, J.S. Ide, J.C.F. da Rocha, Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments, in: Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2004, pp. 104–111.
[15] F.G. Cozman, D.D. Mauá, Bayesian networks specified using propositional and relational constructs: combined, data, and domain complexity, in: AAAI Conference on Artificial Intelligence, 2015.
[16] F.G. Cozman, D.D. Mauá, On the complexity of propositional and relational credal networks, in: International Symposium on Imprecise Probability: Theories and Applications, 2015, pp. 97–105.
[17] F.G. Cozman, R.B. Polastro, F.I. Takiyama, K.C. Revoredo, Computing inferences for relational Bayesian networks based on $\mathcal{ALC}$ constructs, in: Uncertainty Reasoning for the Semantic Web III, in: Lect. Notes Comput. Sci., vol. 8816, 2014, pp. 21–40.
[18] C. d'Amato, N. Fanizzi, T. Lukasiewicz, Tractable reasoning with Bayesian description logics, in: S. Greco, T. Lukasiewicz (Eds.), International Conference on Scalable Uncertainty Management, in: Lect. Notes Comput. Sci., vol. 5291, Springer, 2008, pp. 146–159.
[19] A. Darwiche, Modeling and Reasoning with Bayesian Networks, Cambridge University Press, 2009.
[20] C.P. de Campos, F.G. Cozman, The inferential complexity of Bayesian and credal networks, in: International Joint Conference on Artificial Intelligence, IJCAI, Edinburgh, United Kingdom, 2005, pp. 1313–1318.
[21] G. de Cooman, E. Miranda, M. Zaffalon, Independent natural extension, Artif. Intell. 175 (2011) 1911–1950.
[22] G. de Cooman, M. Zaffalon, Updating beliefs with incomplete observations, Artif. Intell. 159 (2004) 75–125.
[23] D.D. Maua, C. Polpo de Campos, F.G. Cozman, The complexity of MAP inference in Bayesian networks specified through logical languages, in: International Joint Conference on Artificial Intelligence, 2015, pp. 889–895.
[24] H.B. Enderton, A Mathematical Introduction to Logic, Academic Press, 1972.
[25] E. Fagiuoli, M. Zaffalon, 2U: an exact interval propagation algorithm for polytrees with binary variables, Artif. Intell. 106 (1) (1998) 77–107.
[26] D. Fierens, G. van den Broeck, J. Renkens, D. Shrerionov, B. Gutmann, G. Janssens, L. de Raedt, Inference and learning in probabilistic logic programs using weighted Boolean formulas, Theory Pract. Log. Program. 15 (3) (2014) 358–401.
[27] M. Gardner, Problems involving questions of probability and ambiguity, Sci. Am. 201 (4) (1959) 174–182.
[28] M. Gardner, The Colossal Book of Mathematics: Classic Puzzles, Paradoxes, and Problems, W. W. Norton and Company, New York, 2001.
[29] L. Getoor, B. Taskar, Introduction to Statistical Relational Learning, MIT Press, 2007.
[30] R. Isaac, The Pleasures of Probability, Springer, 1995.
[31] M. Jaeger, Relational Bayesian networks, in: D. Geiger, P.P. Shenoy (Eds.), Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, California, 1997, pp. 266–273.
[32] M. Jaeger, G.V.D. Broeck, Liftability of probabilistic inference: upper and lower bounds, in: 2nd Statistical Relational AI (StaRAI-12) Workshop, 2012.
[33] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[34] D. Koller, A. Pfeffer, Object-oriented Bayesian networks, in: Conference on Uncertainty in Artificial Intelligence, 1997, pp. 302–313.
[35] L.V.S. Lakshmanan, F. Sadri, Probabilistic deductive databases, in: Symposium on Logic Programming, 1994, pp. 254–268.
[36] I. Levi, The Enterprise of Knowledge, MIT Press, Cambridge, Massachusetts, 1980.
[37] L. Libkin, Elements of Finite Model Theory, Springer, 2004.
[38] M.L. Littman, J. Goldsmith, M. Mundhenk, The computational complexity of probabilistic planning, J. Artif. Intell. Res. 9 (1998) 1–36.
[39] T. Lukasiewicz, Probabilistic logic programming with conditional constraints, ACM Trans. Comput. Log. 2 (3) (2001) 289–339.
[40] T. Lukasiewicz, L. Predoiu, H. Stuckenschmidt, Tightly integrated probabilistic description logic programs for representing ontology mappings, Ann. Math. Artif. Intell. 63 (3/4) (2011) 385–425.
[41] T. Lukasiewicz, U. Straccia, Managing uncertainty and vagueness in description logics for the semantic web, J. Web Semant. 6 (2008) 291–308.
[42] F. Mosteller, Fifty Challenging Problems in Probability, Addison-Wesley, 1965.
[43] R. Ng, V.S. Subrahmanian, Probabilistic logic programming, Inf. Comput. 101 (2) (1992) 150–201.
[44] L. Ngo, P. Haddawy, Answering queries from context-sensitive probabilistic knowledge bases, Theor. Comput. Sci. 171 (1–2) (1997) 147–177.
[45] C.H. Papadimitriou, Computational Complexity, Addison-Wesley Publishing, 1994.
[46] J.D. Park, MAP complexity results and approximation methods, in: A. Darwiche, N. Friedman (Eds.), Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, California, 2002, pp. 388–396.
[47] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, California, 1988.
[48] J. Pearl, Causality: Models, Reasoning, and Inference, Cambridge University Press, Cambridge, United Kingdom, 2000.
[49] D. Poole, Probabilistic Horn abduction and Bayesian networks, Artif. Intell. 64 (1993) 81–129.
[50] D. Poole, The independent choice logic for modelling multiple agents under uncertainty, Artif. Intell. 94 (1/2) (1997) 7–56.
[51] D. Poole, Abducing through negation as failure: stable models within the independent choice logic, J. Log. Program. 44 (2000) 5–35.
[52] D. Poole, First-order probabilistic inference, in: International Joint Conference on Artificial Intelligence, IJCAI, 2003, pp. 985–991.
[53] L.D. Raedt, Logical and Relational Learning, Springer, 2008.
[54] F. Riguzzi, E. Bellodi, E. Lamma, R. Zese, Epistemic and statistical probabilistic ontologies, in: International Workshop on Uncertainty Reasoning for the Semantic Web, 2012, pp. 1–12.
[55] T. Sato, Y. Kameya, Parameter learning of logic programs for symbolic-statistical modeling, J. Artif. Intell. Res. 15 (2001) 391–454.
[56] U. Sattler, D. Calvanese, R. Molitor, Relationship with other formalisms, in: Description Logic Handbook, Cambridge University Press, 2003, pp. 142–183.
[57] L.G. Valiant, The complexity of enumeration and reliability problems, SIAM J. Comput. 8 (3) (1979) 410–421.
[58] G. van den Broeck, On the completeness of first-order knowledge compilation for lifted probabilistic inference, in: Neural Processing Information Systems, 2011.
[59] P. Walley, Statistical Reasoning with Imprecise Probabilities, Chapman and Hall, London, 1991.
[60] K. Weichselberger, The theory of interval-probability as a unifying concept for uncertainty, Int. J. Approx. Reason. 24 (2–3) (2000) 149–170.