Artificial Intelligence 346 (2025) 104361

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

A semantics for probabilistic hybrid knowledge bases with function symbols

Marco Alberti [a], Evelina Lamma [b], Fabrizio Riguzzi [a], Riccardo Zese [c,*]

[a] *Dipartimento di Matematica e Informatica, University of Ferrara, Via Saragat, 1, 44122 Ferrara, Italy*
[b] *Dipartimento di Ingegneria, University of Ferrara, Via Saragat, 1, 44122 Ferrara, Italy*
[c] *Dipartimento di Scienze Chimiche, Farmaceutiche e Agrarie, University of Ferrara, Via Luigi Borsari, 46, 44121 Ferrara, Italy*

ARTICLE INFO

Keywords:
Hybrid knowledge bases
Minimal knowledge with negation as failure
Probability
Distribution semantics

ABSTRACT

Hybrid Knowledge Bases (HKBs) successfully integrate Logic Programming (LP) and Description Logics (DL) under the Minimal Knowledge with Negation as Failure semantics. Both world closure assumptions (open and closed) can be used in the same HKB, a feature required in many domains, such as the legal and health-care ones. In previous work, we proposed (function-free) Probabilistic HKBs, whose semantics applied Sato's distribution semantics approach to the well-founded HKB semantics proposed by Knorr et al. and Lyu and You. This semantics relied on the fact that the grounding of a function-free Probabilistic HKB (PHKB) is finite. In this article, we extend the PHKB language to allow function symbols, obtaining PHKB^{FS}. Because the grounding of a PHKB^{FS} can be infinite, we propose a novel semantics which does not require the PHKB^{FS}'s grounding to be finite. We show that the proposed semantics extends the previously proposed semantics and that, for a large class of PHKB^{FS}, every query can be assigned a probability.

1. Introduction

Knowledge representation and reasoning in complex domains such as law [1] or health-care [27] require coping with open domains while adopting the closed-world assumption in order to infer negative information. To this purpose, several authors proposed languages combining Description Logics (DLs), that accommodate the former requirement, and Logic Programming (LP), that provide the latter. Among these, we can cite Description Logic Programs [25], or Hybrid Knowledge Bases (HKBs) by [36]. In particular, the latter combines logic program and a DL Knowledge Base following a semantics based on the logic of Minimal Knowledge with Negation as Failure (MKNF) [29]. This formalization exhibits desirable properties: faithfulness, the preservation of the semantics of both formalisms when the other is absent; tightness, the absence of layering of LP and DL; and flexibility, the possibility of viewing each predicate under both open- and closed-world assumptions.

However, these proposals lack an important feature when reasoning with complex, real-world, domains: the capability of dealing with uncertain information. Considering LP and DL separately, there are many proposals introducing probability in these logics. Regarding LP, in the Probabilistic Logic Programming (PLP) field [44,47] there is a plethora of approaches (e.g., PRISM [51], Logic Programs with Annotated Disjunctions [55], and ProbLog [20]) mostly based on the distribution semantics [51], where a program defines a probability distribution over normal Logic Programs, called worlds, from which the probability of a query is obtained.

* Corresponding author.
E-mail addresses: marco.alberti@unife.it (M. Alberti), evelina.lamma@unife.it (E. Lamma), fabrizio.riguzzi@unife.it (F. Riguzzi), riccardo.zese@unife.it (R. Zese).

https://doi.org/10.1016/j.artint.2025.104361
Received 28 February 2024; Received in revised form 22 January 2025; Accepted 13 May 2025

Available online 20 May 2025

Considering DL languages, their combination with probability theory was also amply studied, with proposals exploiting graphical models, such as Bayesian networks [17,15], or Markov networks [24]; or reasoning with intervals of probability values, such as the approaches such based on Nilsson's probabilistic logic [37] (e.g., [23,31,33,12]).

A step in the direction of creating an integrating framework featuring probabilistic LP and DL can be seen in the definition of semantics applying the distribution semantics of PLP to DLs, such as DISPONTE (for "DIstribution Semantics for Probabilistic ONTologiEs") [9].

Despite the number of proposals to combine probability and logics, the problem of combining complex languages exhibiting both open- and closed-world assumptions with probability theory received some attention only in the last few years, with works such as Bayesian Description Logic Programs [42], Probabilistic DL-Programs [32] and Probabilistic Hybrid Knowledge Bases [2]. However, as argued by [36], the first two proposals present drawbacks when compared to MKNF-based HKBs.

In Probabilistic Hybrid Knowledge Bases (PHKBs) [2], facts of the logic programs and DL axioms may be annotated with a probability value. PHKBs have a distribution semantics in the style of [51] based on the well-founded semantics for HKBs [27]. The main limitation of [2] is that the LP part cannot contain function symbols.

In this paper we extend the PHKB semantics to cope with function symbols. The resulting Probabilistic HKB^{FS}s (PHKB^{FS}s) contain both (probabilistic) LP rules and (probabilistic) DL axioms. On the line of [2], we extend the semantics based on the well-founded MKNF semantics with the treatment of function symbols and probability. Since LP with function symbols is Turing-complete [18], PHKB^{FS}s are also Turing-complete, making them a full probabilistic programming language [8], thus greatly enhancing the expressive power of PHKBs.

The main motivation behind the introduction of function symbols is to increase the expressivity of the language allowing the representation of infinite domains and recursive data structures such as lists, trees, time, etc., similarly to what [13] did for Answer Set Programming.

We show that, for a large class of PHKB^{FS}, the semantics assigns a probability to every query. We do so by proving that each query is associated to a measurable set using two operators, and their iterated fixpoint, leveraging the definition of the semantics for non-probabilistic HKB proposed by Alberti et al. [3] for function-free HKBs and by [4] for HKBs with function symbols.

The proof exploits the fact that the probability measure for PHKB^{FS}s is the product of two measures and follows the same approach of the proof that the semantics assigns a probability to every query for probabilistic LP with functions symbols and continuous random variables [6]: in that case as well the probability measure of the program is the product of two measures, one for the discrete and one for the continuous part.

The paper is organized as follows. Related work is discussed in Section 2. In Section 3, we provide some background on LP and DLs, and their probabilistic extensions, and on MKNF-based HKBs. In Section 4, we present the iterated fixpoint definition of the well-founded semantics for non-probabilistic HKBs. In Section 5, we introduce Probabilistic Hybrid Knowledge Bases and we prove that their semantics is well-defined. We conclude and outline future work in Section 6.

2. Related work

Probabilistic extensions have been proposed for several of the languages that integrate DL and LP; in the following, we review some of them. In general, as argued by [36], all these languages present drawbacks when compared to MKNF-based HKBs.

In FOProbLog [11], the knowledge base is composed of disjunctive clauses, where each disjunct is a first order formula annotated with a probability. Probabilities act as constraints, and a model is any distribution that satisfies the constraints; in this way, the semantics defines a probability range for a query, while our approach returns the exact probability of the query. Another important difference regards negation. In FOProbLog, inference is performed by translating the knowledge base into a ProbLog program, following Stickel's PTTP approach [53] to build a FOL theorem prover using an LP proof procedure. However, being a FOL language, FOProbLog does not support default negation.

Description Logic Programs [25] is an intersection of DL and LP: in other words, they can be seen as the fragment of DL that can be expressed in LP or as the fragment of LP that can be expressed in DL. While this approach achieves interoperability between the allowed fragments of LP and DL, important expressive features are not supported: namely, default negation in LP rules, reasoning about unknown individuals and existential quantification in consequents. On the opposite, PHKB^{FS}s allow the use of more expressive DL fragments.

In Bayesian Description Logic Programs (BDLPs) [42], each rule is annotated with two values, representing the probability that the head is true and false when the body is true; a BDLP encodes a Bayesian network where each ground atom is a node and rules represent conditional probabilities. Compared to PHKB^{FS}s, BDLPs inherit the reduced expressiveness of the underlying language.

A translation from subsets of OWL Lite to variants of Datalog is proposed by [38], and of their probabilistic extensions to probabilistic Datalog by Fuhr [22]. In particular, the OWL Lite^{-} language is translated to Datalog, while the more expressive OWL Lite^{EQ} language is translated to Datalog^{EQ} (i.e., Datalog with equality in heads); probabilistic extensions of both DL languages are translated to probabilistic Datalog. However, as argued by Lukasiewicz et al. [35], the supported DL fragments are the same as in Description Logic Programs [25], which limits the expressiveness of the resulting probabilistic language.

Poole's Independent Choice Logic (ICL) [40] combines probability and logic using the notion of choice space. A choice space is a set of pairwise disjoint subsets of a program's Herbrand base, called alternatives; each element of each alternative is an atomic choice. Intuitively, in each world only one atomic choice is true for each alternative; a probability distribution is defined over the elements of each alternative, inducing a probability distribution over possible worlds. Atomic choices can occur in rule bodies; at the semantic level, the effect is that in each world only some rules are selected. Compared to ICL, PHKB^{FS} semantics defines a probability distribution over worlds where the choices consider single facts or axioms instead of set of facts.

Lukasiewicz's Probabilistic DL-Programs [32] integrate DL-Programs [21] with ICL: besides the choice space with its probability distribution, they are composed of a DL ontology and a set of non-disjunctive LP rules; probabilistic extensions of both the answer sets and the well-founded semantics of DL-Programs are given. In DL-Programs, the atoms that occur in DL axioms cannot be the head of rules; therefore, rules cannot be used to define DL predicates, and the integration is not tight. This limitation is inherited by Probabilistic DL-Programs, while it does not affect PHKB^{FS}s.

Lukasiewicz et al. [35] proposed a probabilistic extension of a tight integration of DL and LP based on Disjunctive DL-Programs [34], which are composed of a disjunctive logic program and a set of DL-Lite$_{\mathcal{A}}$ axioms. Answer set and well-founded semantics are provided. Lukasiewicz et al. [35] extend Disjunctive DL-Programs by means of ICL where the LP part is restricted to normal logic programs. In particular, a probabilistic DL-program is composed of a DL-Lite$_{\mathcal{A}}$ ontology, a normal logic program, an ICL choice space and a probability distribution over the choice space. The probabilistic semantics defines tight lower and upper bounds for the probability of a conditional query of the form $b|a$ where a and b are ground atoms, in terms of the answer sets or well-founded model determined by the selected atomic choices. The authors also provide an anytime algorithm to approximate the lower and upper bounds. As observed by Alferes et al. [5], DL-programs require the ontology to be decomposable into a positive and a negative part, which is satisfied by DL-Lite$_{\mathcal{A}}$ but restricts the applicability to general DLs; the same holds for probabilistic DL-programs compared to PHKB^{FS}s. Moreover, in probabilistic DL-programs the ontology is deterministic, while in PHKB^{FS}s DL axioms can be probabilistic.

3. Background

3.1. Logic programs

We assume familiarity with standard First Order Logic terminology (see Appendix C). In this work, we follow the common LP practice of denoting predicate and function symbols with alphanumeric strings starting with a lowercase letter, and variables with alphanumeric strings starting with an uppercase letter. A *literal* l is either an atom a (*positive literal*) or its *default negation* $\sim a$ (*negative literal*). A *normal logic program* P is a finite set of formulas, called *clauses* or *rules*, of the form

$$h \leftarrow b_1, \dots, b_n$$

where h is an atom and all the b_is are literals. h is called the *head* of the clause and the conjunction $b_1, \dots, b_n$ is called the *body*. If the body is empty the clause is called a *fact*.

A term, atom, literal or clause is *ground* if it does not contain variables. A *substitution* θ is an assignment of terms to variables: $\theta = \{V_1/t_1, \dots, V_n/t_n\}$. The *application of a substitution* $\theta = \{V_1/t_1, \dots, V_n/t_n\}$ *to a term atom, literal or clause* r, indicated with $r\theta$, is the replacement of each variable V_i occurring in r and in θ with t_i. $r\theta$ is called an *instance* of r. θ is a *grounding* for r if $r\theta$ is ground.

The *Herbrand universe* $\mathcal{U}_P$ of a logic program P is the set of all the ground terms that can be built from the constant and function symbols in the program, respecting the function symbols' arities.

The grounding of a program P, indicated as *ground*(P), is obtained by substituting terms from the Herbrand universe $\mathcal{U}_P$ for the variables in the clauses of P in all possible ways.

The Herbrand universe of a program P is finite if P does not contain function symbols, otherwise it is denumerable (if P contains at least one constant). Therefore, if P does not contain function symbols, its grounding *ground*(P) is finite, while if P contains function symbols and at least one variable and one constant, *ground*(P) is denumerable.

3.2. ProbLog

Among the several equivalent languages for PLP under the distribution semantics, we consider ProbLog [20], which will make the treatment simpler.

A *ProbLog program* $\mathcal{P} = (\mathcal{R}, \mathcal{F})$ consists of a finite set $\mathcal{R}$ of (*certain*) LP rules and a finite set $\mathcal{F}$ of *probabilistic facts* of the form

$$p_i :: a_i,$$

where $p_i \in (0, 1)$ and a_i is an atom, meaning that we have evidence of the truth of each ground instantiation $a_i\theta$ of a_i with probability p_i and of its falsity with probability $1 - p_i$ (see definitions in Appendix A).

For simplicity, we assume that the atoms in probabilistic facts do not unify with the head of any rule. Note that to ensure this property we can rewrite without loss of generality the following ProbLog program

$$\begin{array}{l} p :: a. \\ a \leftarrow body, \end{array}$$

as

$$\begin{array}{l} p :: a'. \\ a \leftarrow a'. \\ a \leftarrow body. \end{array}$$

These two programs are equivalent when we consider their models excluding a'.

A *grounded ProbLog program* $\mathcal{P} = (\mathcal{R}, \mathcal{F})$ differs from a ProbLog program because $\mathcal{R}$ and $\mathcal{F}$ are ground and may be infinite. Given a ProbLog program $\mathcal{P} = (\mathcal{R}, \mathcal{F})$, its *grounding* $ground(\mathcal{P})$ is defined as $(ground(\mathcal{R}), ground(\mathcal{F}))$. Then $ground(\mathcal{P})$ is a grounded ProbLog program. In fact, if $\mathcal{P}$ contains function symbols, the grounding of $\mathcal{R}$ and $\mathcal{F}$ may be denumerable.

With a slight abuse of notation, in the following sometimes we will use $\mathcal{F}$ to indicate the set of atoms a_i that occur in probabilistic facts. The meaning of $\mathcal{F}$ will be clear from the context.

In the following we briefly report notions about the semantics of ProbLog programs without and with function symbols. We refer to Appendix E.1 for detailed description of the semantics of ProbLog programs without function symbols and to Appendix E.2 for that of the semantics of ProbLog with function symbols.

3.2.1. The semantics of ProbLog programs without function symbols

For a ProbLog program $\mathcal{P} = (\mathcal{R}, \mathcal{F})$ without functions symbols, $ground(\mathcal{R})$ and $ground(\mathcal{F})$ are finite. From the grounding $ground(\mathcal{P})$, we generate normal programs called *worlds* by including in a program the set of certain rules and a subset of the probabilistic facts, in all possible ways. In other words, a world w is obtained by selecting or not each (ground) probabilistic fact. Call $W_\mathcal{P}$ the set of all possible worlds. Since $\mathcal{F}$ is finite, so is $W_\mathcal{P}$.

Given a ground atom q, define function $Q : W_\mathcal{P} \to \{0, 1\}$ as

$$Q(w) = \begin{cases} 1 & \text{if } w \vDash q \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $v \vDash q$ means that q is true in the well-founded model of w (see Appendix D). The distribution of Q is defined by $P(Q = 1)$ ($P(Q = 0)$ is given by $1 - P(Q = 1)$) and we indicate $P(Q = 1)$ with $P(q)$.

We can now compute $P(q)$ as

$$P(q) = \sum_{w \in W_\mathcal{P} : w \vDash q} \prod_{p::a \in \mathcal{F} : a \in w} p \prod_{p::a \in \mathcal{F} : a \notin w} (1 - p)$$

Example 1. The program

$$\mathcal{F} = \{0.3 :: connectionIsTransitive. \tag{2}$$

$$0.2 :: edge(bill, stephanie).\} \tag{3}$$

$$\mathcal{R} = \{edge(bill, john). \tag{4}$$

$$edge(john, stephanie). \tag{5}$$

$$connected(X, Y) \leftarrow edge(X, Y). \tag{6}$$

$$connected(X, Y) \leftarrow connected(X, Z), edge(Z, Y), \tag{7}$$

$$connectionIsTransitive.\} \tag{8}$$

models the connections between users in a social network. *bill* and *stephanie* are directly connected ($edge/2$ predicate) with probability 0.2 because an interaction occurred between them. Two facts model that *bill* and *john* are friends, and so are *john* and *stephanie*. Two users are connected if they are directly connected. Moreover, the probabilistic fact *connectionIsTransitive* models that $connected/2$ is also the transitive closure of direct connection with probability 0.3.

This program has two probabilistic facts, so there are four worlds: one that contains both facts, one that contains none, and two containing one each. The query $connected(bill, stephanie)$ is true in three of them, i.e., those containing at least one probabilistic fact, and false in the world that does not contain any probabilistic fact. The query's probability is $0.2 \times 0.3 + 0.8 \times 0.3 + 0.2 \times 0.7 = 0.44$.

3.2.2. The semantics of ProbLog programs with function symbols

When the program contains functions symbols, $ground(\mathcal{F})$ may be infinite.

Example 2 *(Spillover - ProbLog).* Let us consider the following ProbLog program $\mathcal{P}$ where

$$\mathcal{F} = \{0.8 :: mutated(t).$$

$$0.6 :: spillover(Y).\}$$

$$\mathcal{R} = \{spillover\_count(X, s(Y)) \leftarrow virus(X), mutated(X), \qquad \text{(a)}$$

$$spillover\_count(X, Y), spillover(Y).$$

$$spillover\_count(X, 0) \leftarrow virus(X).$$

$$virus(t).\}$$

This program counts the number of spillover events of a mutated virus: if a virus is mutated, the spillover count is Y and a spillover event happens, then the count is also $Y+1$, represented as the function symbol $s(Y)$ in the head of rule *(a)*. Moreover, the program defines a virus called t, asserts that spillover events happen with probability 0.6 and that t is mutated with probability 0.8. The grounding of $\mathcal{F}$ is

$$
\begin{aligned}
ground(\mathcal{F}) = \{&0.6 :: spillover(0)\\
&0.6 :: spillover(s(0))\\
&\dots && (9)\\
&0.6 :: spillover(t)\\
&0.6 :: spillover(s(t))\\
&\dots && (10)\\
&0.8 :: mutated(t).\}
\end{aligned}
$$

Let us introduce some terminology. An *atomic choice* indicates whether a ground probabilistic fact $p :: f$ is selected or not and is represented with the pair (f,k) where $k \in \{0,1\}$. $k=1$ means that the fact is selected, $k=0$ that it is not. A set of atomic choices is *consistent* if only one alternative is selected for the same probabilistic fact, i.e., it does not contain atomic choices $(f,0)$ and $(f,1)$ for any f. A *composite choice* κ is a consistent set of atomic choices. A *selection* σ (also called total composite choice) contains one atomic choice for every probabilistic fact. A selection σ identifies a *world* w_σ, i.e., a logic program containing the rules $\mathcal{R}$ and fact f for each atomic choice $(f,1)$ of σ. Let $W_\mathcal{P}$ be the set of worlds, which may be uncountable [46].

The *set of worlds* ω_κ *compatible with a composite choice* κ is $\omega_\kappa = \{w_\sigma \in W_\mathcal{P} \mid \kappa \subseteq \sigma\}$. Therefore, a composite choice identifies a set of worlds. For programs with function symbols, ω_κ may be uncountable.

To compute the probability of a ground atom we need to resort to a different concept. Given a probabilistic logic program $\mathcal{P}$, a ground atom q and a composite choice κ, we say that κ is an *explanation* of q if $\forall w \in \omega_\kappa : w \vDash q$. We say that a set of composite choices K is *covering* for q if $\{w \mid w \in W_\mathcal{P} \wedge w \vDash q\} \subseteq \omega_K$.

If q has a countable set K of countable explanations that is covering with respect to q, Q represents a random variable, since $\{w \mid w \in W_\mathcal{P} \wedge w \vDash q\} = \omega_K \in \Omega_\mathcal{P}$. For brevity, we indicate $P(Q=1)$ with $P(q)$ and we say that $P(q)$ is *well-defined* according to the distribution semantics. If the probability of all ground atoms in the grounding of a probabilistic logic program $\mathcal{P}$ is well-defined, then $\mathcal{P}$ is *well-defined*.

Riguzzi [46,47] proved that any query to a sound ProbLog program has a countable set of countable explanations that is covering, so it can be assigned a probability so that the program is well-defined.

3.3. Description logics

DLs are decidable fragments of First Order Logic used to model ontologies [10]. Usually their syntax is based on concepts and roles, corresponding to unary and binary predicates, respectively. In the following, for the sake or simplicity, we briefly recall one of the simplest DLs, $\mathcal{ALC}$. However, the semantics proposed in this paper can exploit any DL; see [7] for a complete introduction to DLs.

$\mathcal{ALC}$'s alphabet is composed of a set $\mathbf{C}$ of *atomic concepts*, a set $\mathbf{R}$ of *atomic roles* and a set $\mathbf{I}$ of individuals. A *concept* C is defined by:

$$C ::= C_1 \mid \bot \mid \top \mid (C \sqcap C) \mid (C \sqcup C) \mid \neg C \mid \exists R.C \mid \forall R.C$$

where $C_1 \in \mathbf{C}$ and $R \in \mathbf{R}$.

A *TBox* T is a finite set of *concept inclusion axioms* $C \sqsubseteq D$, where C and D are concepts. An *ABox* A is a finite set of *concept membership axioms* $a : C$ and *role membership axioms* $(a,b) : R$, where C is a concept, $R \in \mathbf{R}$ and $a,b \in \mathbf{I}$. An $\mathcal{ALC}$ knowledge base $O = T \cup A$ is the union of a TBox and an ABox.

DL axioms can be mapped to FOL formulas by the transformation π shown in Table 1 for the $\mathcal{ALC}$ DL [52]. π is applied to concepts as follows:

$$
\begin{aligned}
\pi_x(A) &= A(x)\\
\pi_x(\neg C) &= \neg \pi_x(C)\\
\pi_x(C \sqcap D) &= \pi_x(C) \wedge \pi_x(D)\\
\pi_x(C \sqcup D) &= \pi_x(C) \vee \pi_x(D)\\
\pi_x(\exists R.C) &= \exists y.R(x,y) \wedge \pi_y(C)\\
\pi_x(\forall R.C) &= \forall y.R(x,y) \rightarrow \pi_y(C)
\end{aligned}
$$

Table 1
Translation of $\mathcal{ALC}$ axioms into FOL.

Axiom	Translation
$C \sqsubseteq D$	$\forall x.\pi_x(C) \rightarrow \pi_x(D)$
$a : C$	$\pi_a(C)$
$(a, b) : R$	$R(a, b)$

3.4. Probabilistic description logics

DISPONTE [9,57,48] applies the distribution semantics to probabilistic ontologies [51]. A DISPONTE *knowledge base* (KB) is a pair $\mathcal{O} = (\mathcal{A}, \mathcal{E})$, where $\mathcal{A}$ is a finite set of DL axioms, that we call *certain*, and $\mathcal{E}$ is a finite set of *probabilistic axioms* of the form

$$p_i :: e_i$$

where p_i is a real number in $[0, 1]$ and e_i is a DL axiom.

As for ProbLog, from a DISPONTE KB we obtain non-probabilistic KBs by taking the certain axioms and adding a subset of the probabilistic axioms in all possible ways. We call *worlds* the resulting KBs while $W_{\mathcal{O}}$ is the set of worlds.

Given an axiom q, define the function $Q : W_{\mathcal{O}} \rightarrow \{0, 1\}$ as in Eq. (1). The distribution of Q is defined by $P(Q = 1)$ ($P(Q = 0)$ is given by $1 - P(Q = 1)$) and we indicate $P(Q = 1)$ with $P(q)$.

We can now compute $P(q)$ as

$$P(q) = \sum_{w \in W_{\mathcal{O}} : w \models q} \prod_{p :: a \in \mathcal{E} : a \in w} p \prod_{p :: a \in \mathcal{E} : a \notin w} (1 - p)$$

For a detailed description of DISPONTE, we refer to Appendix E.3.

Example 3. Consider the following KB, based on the social network domain of Example 1:

$$\mathcal{A} = \{\mathit{influencer} \sqsubseteq \mathit{social}\} \tag{11}$$

$$\mathcal{E} = \{0.7 :: \mathit{famousPerson} \sqsubseteq \mathit{influencer} \tag{12}$$

$$0.1 :: \mathit{jack} : \mathit{famousPerson}\} \tag{13}$$

This probabilistic DL KB models that we believe *jack* is a famous person (modelled by concept *famousPerson*) with probability 0.1, and that famous people are influencers with probability 0.7. Finally, the KB models that an influencer is a social person. As in Example 1, there are 4 worlds: one containing both axioms from $\mathcal{E}$, one containing none of them, and two containing one each. All the 4 worlds also contain the axiom in $\mathcal{A}$. The query $\mathit{jack} : \mathit{social}$ is true only in the one containing both axioms from $\mathcal{E}$, so $P(\mathit{jack} : \mathit{social}) = 0.7 \times 0.1 = 0.07$.

3.5. MKNF hybrid knowledge bases

The Minimal Knowledge with Negation as Failure (MKNF) logic [29], inspired by several works [28,45] on epistemic query answering on non-monotonic databases, supports epistemic queries on logic programs.

The MKNF formula is a First Order Logic formula (see Appendix C) augmented with the modal operators **K** and **not** , i.e., the same of formula (C.1) with the additional alternatives $\mathbf{K}\,\psi$ and $\mathbf{not}\,\psi$.

Hybrid Knowledge Bases [36], which integrate DL and LP in one formalism, adopt MKNF as its semantical foundation. [36] point out that MKNF-based Hybrid Knowledge Bases possess desirable properties that competing languages lack, at least in part: faithfulness, i.e., the semantics of each formalism is preserved when the other is absent; tightness, i.e., the LP and DL portions of a knowledge base do not need to be in separate layers; flexibility, i.e., both the open and closed world assumption can be employed in the definition of the same predicate; and decidability.

Definition 1. A Hybrid Knowledge Base (HKB) is a pair $H = (P, O)$, where P is a normal logic program (Section 3.1), possibly with function symbols, and O is a DL KB (Section 3.3).

An HKB (P, O) is *positive* if no negative literals occur in P. An HKB (P, O) is *ground* if P is ground. The *grounding* of an HKB (P, O) is given by $(O, \mathit{ground}(P))$, where the constants used in the grounding are those appearing in (P, O).

Note that, differently from the definition by Motik and Rosati [36], disjunctions are not allowed in LP rule heads.

In the rest of the paper, when we say that a HKB $H_1 = (O_1, P_1)$ is a subset of a HKB $H_2 = (O_2, P_2)$ ($H_1 \subseteq H_2$), we mean that $O_1 \subseteq O_2$ and $P_1 \subseteq P_2$. For a given HKB $H = (P, O)$, an atom in P is a *DL-atom* if its predicate occurs in O, a non-DL-atom otherwise.

Definition 2 *(DL-safety).* An LP rule is *DL-safe* if each of its variables occurs in at least one positive non-DL-atom in the body; a HKB is *DL-safe* if all its LP-rules are DL-safe.

A transformation can be defined from a HKB $H = (P, O)$ to an MKNF formula by extending the standard transformation π for DL axioms (Table 1) to support LP rules:

- if r is a rule of the form $h \leftarrow a_1, \ldots, a_n, \sim b_1, \ldots, \sim b_m$ where all a_i and b_j are atoms and $\mathbf{X}$ is the tuple of all variables in r, then $\pi(r) = \forall \mathbf{X}(\mathbf{K}\, a_1 \wedge \ldots \wedge \mathbf{K}\, a_n \wedge \mathbf{not}\, b_1 \wedge \ldots \wedge \ldots \mathbf{not}\, b_m \supset \mathbf{K}\, h)$
- $\pi(P) = \bigwedge_{r \in P} \pi(r)$
- $\pi((P, O)) = \mathbf{K}\, \pi(O) \wedge \pi(P)$

HKB allow to reason with both closure assumptions, as the following example shows.

Example 4 *(Viral Marketing).* In a marketing campaign a company wants to grant discounts to customers in a social network, maximizing the return on their investment by avoiding to assign discounts to loner customers, who are less likely to talk to other people about the company products, making the discount ineffective. However, due to fair competition laws, such discounts cannot be granted to people considered influencers, in order to avoid surreptitious advertising. People are considered loner if they are not known to be social and in our case being social means being connected to at least an influencer. This requires the closed world assumption using, for example, default negation of LP. If direct connections (say, the social network's friendship relation) are represented by a directed graph whose nodes are people, connections can be modelled as the transitive closure of direct connections, which is also easily computable in LP. However, the user may want to model that a person is connected to an unknown influencer; this is not possible in logic programming, but it is supported in description logics. This scenario cannot be modelled by LP or DL alone; it needs both.

A domain involving two users, *bill* and *stephanie*, can be modelled with the following HKB, which we will use as a running example.

$$
\begin{array}{ll}
\mathit{loner} \sqsubseteq \mathit{ineffective} & \mathit{person}(\mathit{bill}). \\
\exists \mathit{connected}.\mathit{influencer} \sqsubseteq \mathit{social} & \mathit{influencer}(\mathit{stephanie}). \\
\mathit{loner}(X) \leftarrow \mathit{person}(X), \sim \mathit{social}(X). & \mathit{connected}(X, Y) \leftarrow \mathit{edge}(X, Y). \\
\mathit{discount}(X) \leftarrow \mathit{person}(X), \sim \mathit{ineffective}(X). & \mathit{connected}(X, Y) \leftarrow \mathit{connected}(X, Z), \\
\mathit{edge}(\mathit{bill}, \mathit{stephanie}). & \qquad \mathit{edge}(Z, Y).
\end{array}
$$

This HKB models that a loner person is ineffective for the marketing campaign, that a loner person is someone not social, and that someone is social if they are connected with an influencer.

Suppose that we do not know any influencer of the network, but we know that *bill* is connected with at least one influencer; this cannot be represented in LP alone, but in DL we can specify

$$\mathit{bill} : \exists \mathit{connected}.\mathit{influencer}$$

If we add this axiom to the above HKB and we remove the facts about *stephanie*, *bill* still remains eligible for discount, even if we do not know the identity of the influencer.

The MKNF transformation of this HKB is:

$$
\begin{array}{l}
\mathbf{K}\, \pi(O) \wedge \pi(P) = \\
\mathbf{K}\, (\forall X : (\mathit{ineffective}(X) \subseteq \mathit{loner}(X)) \wedge \\
\quad \forall X : (\mathit{social}(X) \subseteq \exists Y : (\mathit{connected}(X, Y) \wedge \mathit{influencer}(Y)))) \wedge \\
\forall X : (\mathbf{K}\, \mathit{person}(X) \wedge \mathbf{not}\, \mathit{social}(X) \supset \mathbf{K}\, \mathit{loner}(X)) \wedge \\
\forall X : (\mathbf{K}\, \mathit{person}(X) \wedge \mathbf{not}\, \mathit{ineffective}(X) \supset \mathbf{K}\, \mathit{discount}(X)) \wedge \\
\mathbf{K}\, \mathit{edge}(\mathit{bill}, \mathit{stephanie}) \wedge \mathbf{K}\, \mathit{person}(\mathit{stephanie}) \wedge \\
\mathbf{K}\, \mathit{person}(\mathit{bill}) \wedge \mathbf{K}\, \mathit{influencer}(\mathit{stephanie}) \wedge \\
\forall X, Y : (\mathbf{K}\, \mathit{edge}(X, Y) \supset \mathbf{K}\, \mathit{connected}(X, Y)) \wedge \\
\forall X, Y, Z : (\mathbf{K}\, \mathit{connected}(X, Z) \wedge \mathbf{K}\, \mathit{edge}(Z, Y) \supset \mathbf{K}\, \mathit{connected}(X, Y)) \wedge \\
\forall X : (\mathbf{K}\, \mathit{person}(X) \wedge \mathbf{K}\, \mathit{influencer}(X) \supset \mathbf{K}\, \mathit{social}(X))
\end{array}
$$

The MKNF transformation defines a semantics for HKBs: MKNF formulas can have two-valued [29] and three-valued [27] semantics, so the semantics of an HKB can be defined as the (two or three-valued) semantics of MKNF formula resulting from the transformation. The three-valued MKNF semantics, which is more relevant to our work, is recalled in Appendix G.

3.6. Well founded HKB semantics

Knorr et al. [27] defined the well-founded model of an MKNF formula as the MKNF model Appendix G that, intuitively, leaves as much as possible undefined. In particular, the authors define a "more knowledge derivable" relation between MKNF interpretation pairs: $(M_1, N_1) \succcurlyeq_k (M_2, N_2)$ iff $M_1 \subseteq M_2$ and $N_2 \subseteq N_1$. An HKB's three-valued MKNF model (M, N) that is minimal w.r.t. $\succcurlyeq_k$ (i.e., if (M_1, N_1) is also a three-valued model, then $(M_1, N_1) \succcurlyeq_k (M, N)$) is defined to be a *well-founded model*. Not all HKBs have a unique

$$
\begin{array}{lll}
\mathbf{P}_0 & = & \emptyset \\
\mathbf{P}_1 & = & \{edge(bill, stephanie), \\
 & & person(bill), \\
 & & influencer(stephanie), \\
 & & connected(bill, stephanie), \\
 & & social(bill), person(stephanie), \\
 & & social(stephanie)\} \\
\mathbf{P}_2 & = & \mathbf{P}_1 \\
\mathbf{P}_3 & = & \mathbf{N}_2 = \{edge(bill, stephanie), \\
 & & person(bill), \\
 & & influencer(stephanie), \\
 & & discount(bill), \\
 & & connected(bill, stephanie), \\
 & & social(bill), person(stephanie), \\
 & & social(stephanie)\} \\
\mathbf{P}_4 & = & \mathbf{P}_3 = \mathbf{P}_\omega
\end{array}
\qquad
\begin{array}{lll}
\mathbf{N}_0 & = & \mathsf{KA}(H) \\
\mathbf{N}_1 & = & \{edge(bill, stephanie), \\
 & & person(bill), \\
 & & influencer(stephanie), \\
 & & loner(bill), discount(bill), \\
 & & connected(bill, stephanie), \\
 & & social(bill), ineffective(bill) \\
 & & person(stephanie), \\
 & & social(stephanie)\} \\
\mathbf{N}_2 & = & \{edge(bill, stephanie), \\
 & & person(bill), \\
 & & influencer(stephanie), \\
 & & discount(bill), \\
 & & connected(bill, stephanie), \\
 & & social(bill), person(stephanie), \\
 & & social(stephanie)\} \\
\mathbf{N}_3 & = & \mathbf{N}_2 \\
\mathbf{N}_4 & = & \mathbf{N}_3 = \mathbf{N}_\omega
\end{array}
$$

Fig. 1. Building of $\mathbf{P}_\omega$ and $\mathbf{N}_\omega$ for Example 4, step by step.

well-founded model; *MKNF-coherent* HKBs [30] have a unique well-founded model that is characterized by a partition of the atoms that occur in rules. We present this class below.

Knorr et al. [27] consider only DL-safe $H = (P, O)$ because they want to disallow infinite sets of individuals. The grounding of a DL-safe HKB without function symbols is finite. Note that, if an HKB is DL-safe it has the same two-valued MKNF models of its grounding [36]. In the following, we assume that the HKB H is obtained by grounding.

The set of *known atoms* of H, $\mathsf{KA}(H)$, is the set of all the atoms appearing in P.

Definition 3. A *partition* of $\mathsf{KA}(H)$ is a pair (P, N) such that $P \subseteq N \subseteq \mathsf{KA}(H)$; (P, N) is *exact* if $P = N$.

Intuitively, P is a set of true atoms and N a set of true or undefined atoms. Given $S \subseteq \mathsf{KA}(H)$, the *objective knowledge of O with respect to S* is the set

$$\mathsf{OB}_{O,S} = \{\pi(O)\} \cup S \tag{14}$$

The operators R_H, D_H and T_H derive atoms that are consequences of a positive HKB H and a set S of atoms. $R_H(S)$ is the set of immediate consequences due to rules, i.e., the heads of rules in P whose bodies are composed of atoms that belong to S; $D_H(S)$ is the set of immediate consequences due to axioms, i.e., the atoms from $\mathsf{KA}(H)$ entailed by $\mathsf{OB}_{O,S}$; and $T_H(S) = R_H(S) \cup D_H(S)$. Given an HKB H and a set of atoms $S \subseteq \mathsf{KA}(H)$, the following transformations, which yield positive knowledge bases, are defined: the *MKNF transformation* H/S is $(O, P/S)$, where P/S is the set of rules $h \leftarrow a_1, \dots, a_m$ such that there exists in P a rule $h \leftarrow a_1, \dots, a_m, \sim b_1, \dots, \sim b_n$ with $\{b_1, \dots, b_n\} \cap S = \emptyset$, and the *MKNF-coherent transformation* $H//S$ is $(O, P//S)$, where $P//S$ is the set of rules $h \leftarrow a_1, \dots, a_m$ such that there exists a rule $h \leftarrow a_1, \dots, a_m, \sim b_1, \dots, \sim b_n$ in P with $\{b_1, \dots, b_m\} \cap S = \emptyset$ and $\mathsf{OB}_{O,S} \not\models \neg h$.

Since, as shown by [27], T_H is monotonic if H is a ground positive HKB, the following transformations of sets of atoms are well defined: $\Gamma_H(S) = \mathsf{lfp}(T_{H/S})$ and $\Gamma'_H(S) = \mathsf{lfp}(T_{H//S})$. Using these transformations, the sequences of sets of atoms $\mathbf{P}$ and $\mathbf{N}$ are defined as follows: $\mathbf{P}_0 = \emptyset$, $\mathbf{N}_0 = \mathsf{KA}(H)$, $\mathbf{P}_{n+1} = \Gamma_H(\mathbf{N}_n)$ and $\mathbf{N}_{n+1} = \Gamma'_H(\mathbf{P}_n)$, $\mathbf{P}_\omega = \bigcup \mathbf{P}_i$, $\mathbf{N}_\omega = \bigcap \mathbf{N}_i$.

The pair $(\mathbf{P}_\omega, \mathbf{N}_\omega)$ is called H's *alternating fix-point partition.*

Example 5. Fig. 1 shows the computation of the alternating fixpoint partition for the HKB of Example 4.

The HKBs such that the alternating fix-point partition defines a three-valued MKNF model are called *MKNF-coherent* [30].

Definition 4 *(MKNF-coherent HKB, Definition 10 from [30]).* An HKB H is *MKNF-coherent* if (I_P, I_N), where $I_P = \{I \mid I \models \mathsf{OB}_{O,\mathbf{P}_\omega}\}$ and $I_N = \{I \mid I \models \mathsf{OB}_{O,\mathbf{N}_\omega}\}$, is a three-valued MKNF model of H.

Note that we use here a slightly different definition with respect to that given by Liu and You [30] since they adopt a different syntax in the rules' definition, postponing the **K** operator to every positive literal. In this article, for simplicity, we assume the presence of the operator.

For MKNF-coherent HKBs, the model determined by the alternating fix-point partition as in Definition 4 is the unique well-founded model.

Theorem 1 *(Unique well-founded model of an MKNF-coherent HKB, Proposition 2 from [30]). If H is an MKNF-coherent HKB, then it has the unique well-founded model* $(\{I \mid I \models \mathsf{OB}_{O,\mathbf{P}_\omega}\}, \{I \mid I \models \mathsf{OB}_{O,\mathbf{N}_\omega}\})$

4. Iterated fixpoint semantics for HKBs with function symbols

In the original HKB language, function symbols are not allowed. However, this is a feature that is useful in many domains, as shown in Example 2, that describes the behavior of a virus, which can mutate and spillover may happen due to each mutation.

Alberti et al. [4] extended the HKB syntax with function symbols and presented an iterated fixpoint semantics for this new language (HKB^{FS}). They proved that the semantics coincides with that of Knorr et al. [27] and Liu and You [30] in the case of HKBs not including function symbols, and therefore can be considered an extension of that semantics to the case with function symbols.

Definition 5. An HKB^{FS} H is a tuple (P, O) where P is a logic program that may contain function symbols and O is a DL KB. A grounded HKB^{FS} differs from an HKB^{FS} because P is ground and may be denumerable. The grounding $ground(H)$ of an HKB^{FS} H is $(ground(P), O)$ where the grounding uses all the symbols from (P, O). $ground(H)$ is a grounded HKB^{FS}.

Definition 6. A 2-valued interpretation I for an HKB^{FS} H is a subset of $\mathsf{KA}(H)$.

Two-valued interpretations form a complete lattice where the partial order is defined as $I \leq J$ if $I \subseteq J$. For a set T of two-valued interpretations, the least upper bound and greatest lower bound always exist and are respectively

$$\mathrm{lub}(T) = \bigcup_{I \in T} I$$

and

$$\mathrm{glb}(T) = \bigcap_{I \in T} I.$$

The top element is $\mathsf{KA}(H)$ and the bottom element is $\emptyset$.

Definition 7. A 3-valued interpretation $\mathcal{I}$ for an HKB^{FS} H is a pair (I_T, I_F) where I_T and I_F are subsets of $\mathsf{KA}(H)$. $\mathcal{I}$ is consistent if I_T and I_F are disjoint, i.e., $I_T \cap I_F = \emptyset$

Given a 3-valued interpretation $\mathcal{I} = (I_T, I_F)$, an atom a is true in it if $a \in I_T$, false in it if $a \in I_F$, undefined in it otherwise. Moreover, let $\mathcal{I}_T = I_T$ and $\mathcal{I}_F = I_F$.

Three-valued interpretations form a complete lattice where the partial order is defined as $(I_F, I_T) \leq (I'_F, I'_T)$ if $I_T \subseteq I'_T$ and $I_F \subseteq I'_F$. For a set T of three-valued interpretations, the least upper bound and greatest lower bound always exist and are respectively

$$\mathrm{lub}(T) = \left(\bigcup_{(I_T, I_F) \in T} I_T, \bigcup_{(I_T, I_F) \in T} I_F \right)$$

and

$$\mathrm{glb}(T) = \left(\bigcap_{(I_T, I_F) \in T} I_T, \bigcap_{(I_T, I_F) \in T} I_F \right)$$

The top element is $(\mathsf{KA}(H), \mathsf{KA}(H))$ and the bottom element is $(\emptyset, \emptyset)$.

We denote by ${Int_3}^H$ the set of 3-valued interpretations for an HKB^{FS} H.

Definition 8. Given a grounded HKB^{FS} $H = (P, O)$, and a 3-valued interpretation $\mathcal{I} = (I_T, I_F)$ for H, we define the operators $OpTrue^H_{\mathcal{I}} : 2^{\mathsf{KA}(H)} \rightarrow 2^{\mathsf{KA}(H)}$ and $OpFalse^H_{\mathcal{I}} : 2^{\mathsf{KA}(H)} \rightarrow 2^{\mathsf{KA}(H)}$ as

- $OpTrue^H_{\mathcal{I}}(Tr) = \{a \in \mathsf{KA}(H) \mid$ there is a clause $a \leftarrow b_1, ..., b_m, \sim c_1, \dots, \sim c_n$ in P such that for every i $(1 \leq i \leq m)$ b_i is true in $\mathcal{I}$ or $b_i \in Tr$, and for every j $(1 \leq j \leq n)$ c_j is false in $\mathcal{I}\} \cup \{a \in \mathsf{KA}(H) | \mathsf{OB}_{O, I_T \cup Tr} \models a\}$;
- $OpFalse^H_{\mathcal{I}}(Fa) = \{a \in \mathsf{KA}(H) \mid \mathsf{OB}_{O, I_T} \models \neg a$, or, for every clause $a \leftarrow b_1, ..., b_m, \sim c_1, \dots, \sim c_n$ in P, there is some i $(1 \leq i \leq m)$ such that b_i is false in $\mathcal{I}$ or $b_i \in Fa$, or there is some j $(1 \leq j \leq n)$ such that c_j is true in $\mathcal{I}\} \cap \{a \in \mathsf{KA}(H) | \mathsf{OB}_{O, \mathsf{KA}(H) \setminus (I_F \cup Fa)} \not\models a\}$

In words, $OpTrue^H_{\mathcal{I}}(Tr)$ represents the true atoms that can be derived from H knowing $\mathcal{I}$ and true atoms Tr, while $OpFalse^H_{\mathcal{I}}(Fa)$ represents the false atoms that can be derived from H by knowing $\mathcal{I}$ and false atoms Fa.

$$
\begin{array}{llll}
I_{T0} = & \emptyset & I_{F0} = & \emptyset \\
I_{T1} = & \{virus(t), & I_{F1} = & \mathsf{KA}(H) \setminus I_{T1} \setminus \{safe(t)\} \\
& mutated(t), & & \\
& spillover\_count(t,0), & & \\
& spillover\_count(t,s(0)), & & \\
& spillover\_count(t,s(s(0))), & & \\
& at\_least\_two\_spillovers(t) & & \\
& \dots\} & & \\
I_{T2} = & I_{T1} & I_{F2} = & \mathsf{KA}(H) \setminus I_{T1} \\
I_{T3} = & I_{T2} & I_{F3} = & I_{F2}
\end{array}
$$

Fig. 2. Iterations of the IFP^H operator for Example 6.

Given an HKB^{FS} H and a 3-valued interpretation $\mathcal{I}$, since $OpTrue^H_{\mathcal{I}}$ and $OpFalse^H_{\mathcal{I}}$ are monotonic in their argument (see Propositions 3 and 4 in Appendix H), they both have least and greatest fixpoints. So, it is possible to define the following iterative operator on a 3-valued interpretation $\mathcal{I}$.

Definition 9 *(Iterated Fixed Point for an HKB^{FS}).* For an HKB^{FS} H, we define $IFP^H : {Int_3}^H \to {Int_3}^H$ as

$$IFP^H(\mathcal{I}) = (\mathsf{lfp}(OpTrue^H_{\mathcal{I}}), \mathsf{gfp}(OpFalse^H_{\mathcal{I}}))$$

By virtue of being monotonic (see Proposition 5 in Appendix H), IFP^H admits a least fixpoint for each HKB^{FS} H, which we define as the semantics of the HKB^{FS}.

Definition 10 *(Iterated fixpoint semantics for an HKB^{FS}).* Given an HKB^{FS} H, its iterated fixpoint semantics is $\mathsf{lfp}(IFP^H)$.

Example 6 *(Spillover [4]).* Let $H = (P, O)$, where

$$
\begin{aligned}
P = \ & \{spillover\_count(X, s(Y)) \leftarrow virus(X), mutated(X), \\
& \quad spillover\_count(X, Y). \\
& spillover\_count(X, 0) \leftarrow virus(X). \\
& at\_least\_two\_spillovers(X) \leftarrow virus(X), spillover\_count(X, s(s(Y)))\} \\
& safe(X) \leftarrow virus(X), {\sim} at\_least\_two\_spillovers(X)\} \\
& virus(t).\} \\
O = \ & \{\exists mutation.\top \sqsubseteq mutated \\
& t : \exists mutation.\top\}
\end{aligned}
$$

This HKB^{FS} states that t is a virus and there is at least a mutation of t. If there exists at least one mutation for an individual, it is mutated. We model the series of spillover events by means of predicate *spillover_count*. A virus is safe if it had at most one spillover.

Fig. 2 shows the computation of the iterated fixpoint semantics for the HKB^{FS} H. Given the presence of the function symbol $s(\cdot)$, the model is infinite.

Each $\mathcal{I}_m = (I_{Tm}, I_{Fm})$, for $m = 1, 2, 3$ is determined by the fixpoints of $OpTrue^H_{\mathcal{I}_{m-1}}$ and $OpFalse^H_{\mathcal{I}_{m-1}}$ as follows.

- $OpTrue^H_{\mathcal{I}_0} \uparrow 0 = \emptyset$,
- $OpTrue^H_{\mathcal{I}_0} \uparrow 1 = OpTrue^H_{\mathcal{I}_0} \uparrow 0 \cup \{virus(t), mutated(t)\}$,
- $OpTrue^H_{\mathcal{I}_0} \uparrow 2 = OpTrue^H_{\mathcal{I}_0} \uparrow 1 \cup \{spillover\_count(t, 0)\}$,
- $OpTrue^H_{\mathcal{I}_0} \uparrow 3 = OpTrue^H_{\mathcal{I}_0} \uparrow 2 \cup \{spillover\_count(t, s(0))\}$,
- $OpTrue^H_{\mathcal{I}_0} \uparrow 4 = OpTrue^H_{\mathcal{I}_0} \uparrow 3 \cup \{spillover\_count(t, s(s(0)))\}$,
- $OpTrue^H_{\mathcal{I}_0} \uparrow 5 = OpTrue^H_{\mathcal{I}_0} \uparrow 4 \cup \{at\_least\_two\_spillovers(t)\}$,

and so on to the least fixpoint I_{T1}.

- $OpFalse^H_{\mathcal{I}_0} \downarrow 0 = \mathsf{KA}(H)$
- $OpFalse^H_{\mathcal{I}_0} \downarrow 1 = OpFalse^H_{\mathcal{I}_0} \downarrow 0 \setminus \{virus(t), mutated(t), safe(t)\}$
- $OpFalse^H_{\mathcal{I}_0} \downarrow 2 = OpFalse^H_{\mathcal{I}_0} \downarrow 1 \setminus \{spillover\_count(t, 0)\}$

- $OpFalse^H_{\mathcal{I}_0} \downarrow 3 = OpFalse^H_{\mathcal{I}_0} \downarrow 2 \setminus \{spillover\_count(t, s(0))\}$
- $OpFalse^H_{\mathcal{I}_0} \downarrow 4 = OpFalse^H_{\mathcal{I}_0} \downarrow 3 \setminus \{spillover\_count(t, s(s(0)))\}$
- $OpFalse^H_{\mathcal{I}_0} \downarrow 5 = OpFalse^H_{\mathcal{I}_0} \downarrow 4 \setminus \{at\_least\_two\_spillovers(t)\}$

and so on to the greatest fixpoint I_{F1}.

- $OpTrue^H_{\mathcal{I}_1} \uparrow 0 = \emptyset$
- $OpTrue^H_{\mathcal{I}_1} \uparrow 1 = I_{T1}$

which is the least fixpoint.

- $OpFalse^H_{\mathcal{I}_1} \downarrow 0 = \mathsf{KA}(H)$
- $OpFalse^H_{\mathcal{I}_1} \downarrow 1 = OpFalse^H_{\mathcal{I}_1} \downarrow 0 \setminus \{virus(t), mutated(t)\}$. In this case, $safe(t)$ is kept because $at\_least\_two\_spillovers((t))$ is true in $\mathcal{I}_1$.
- $OpFalse^H_{\mathcal{I}_1} \downarrow 2 = OpFalse^H_{\mathcal{I}_1} \downarrow 1 \setminus \{spillover\_count(t, 0)\}$
- $OpFalse^H_{\mathcal{I}_1} \downarrow 3 = OpFalse^H_{\mathcal{I}_1} \downarrow 2 \setminus \{spillover\_count(t, s(0))\}$
- $OpFalse^H_{\mathcal{I}_1} \downarrow 4 = OpFalse^H_{\mathcal{I}_1} \downarrow 3 \setminus \{spillover\_count(t, s(s(0)))\}$

to the greatest fixpoint $I_{F2} = \mathsf{KA}(H) \setminus I_{T1}$.

For all m, it holds that

$$OpTrue^H_{\mathcal{I}_2} \uparrow m = OpTrue^H_{\mathcal{I}_1} \uparrow m \tag{15}$$
$$OpFalse^H_{\mathcal{I}_2} \downarrow m = OpFalse^H_{\mathcal{I}_1} \downarrow m \tag{16}$$

so $\mathcal{I}_2 = \mathcal{I}_3 = \mathsf{lfp}(IFP^H)$.

For function-free HKB^{FS}s, which are also HKBs, Knorr et al.'s alternating fixpoint partition and the iterated fixpoint (Definition 10) coincide, modulo a set complement operation.

Theorem 2 *(From [4]). Given a function-free HKB^{FS} $H = (P, O)$, let $\mathsf{lfp}(IFP^H) = (I_T, I_F)$. Then $(I_T, \mathsf{KA}(H) \setminus I_F)$ is H's alternating fixpoint partition.*

We call $\mathsf{lfp}(IFP^H)$ the *well-founded model* of H and we indicate it with $\mathsf{WFM}(H)$. If a is an atom and $\mathsf{WFM}(H) = (I_T, I_F)$, we write $H \vDash a$ if $a \in I_T$ and $H \vDash \sim a$ if $a \in I_F$. We call the well-founded model *total* if $I_T \cup I_F = \mathsf{KA}(H)$.

5. Probabilistic hybrid knowledge bases with function symbols

In this section we define a language of Probabilstic Hybrid Knowledge Bases that extends the one by [2] by allowing function symbols. The proofs of all the theorems of this section can be found in Appendix J.

A PHKB^{FS} is composed of a ProbLog program (Section 3.2), possibly containing function symbols, and a DISPONTE knowledge base (Section 3.4).

Definition 11. A PHKB^{FS} $\mathcal{H} = (\mathcal{P}, \mathcal{O})$ is composed of

- a ProbLog program $\mathcal{P} = (\mathcal{R}, \mathcal{F})$
- a DISPONTE knowledge base $\mathcal{O} = (\mathcal{A}, \mathcal{E})$

Sometimes it will be convenient to represent $\mathcal{H}$ with the 4-tuple $(\mathcal{R}, \mathcal{F}, \mathcal{A}, \mathcal{E})$.

Example 7 *(Spillover, Probabilistic).* Let us consider the following PHKB^{FS} $\mathcal{H} = (\mathcal{P}, \mathcal{O})$ as the probabilistic version of the HKB of Example 6, where

$$\begin{aligned}\mathcal{R} = \; & spillover\_count(X, s(Y)) \leftarrow virus(X), mutated(X), \\ & \qquad spillover\_count(X, Y), spillover(X, Y). \\ & spillover\_count(X, 0) \leftarrow virus(X). \\ & at\_least\_two\_spillovers(X) \leftarrow virus(X), spillover\_count(X, s(s(Y))) \\ & safe(X) \leftarrow virus(X), \sim at\_least\_two\_spillovers(X)\end{aligned}$$

$virus(t)$.

$\mathcal{F} = 0.6 :: spillover(X, Y)$

$\mathcal{A} = \exists mutation.\top \sqsubseteq mutated$

$\mathcal{E} = 0.8 :: t : \exists mutation.\top$

This PHKB^{FS} models a series of spillover events. In particular, there is a probability of 80% that at least a mutation arises from t. Moreover, the probability of a spillover to happen is 60%.

Definition 12. A PHKB^{FS} $\mathcal{H} = (\mathcal{P}, \mathcal{O})$ is *grounded* if $\mathcal{P}$ is a grounded ProbLog program. The grounding $ground(\mathcal{H})$ of a $\mathcal{H} = (\mathcal{P}, \mathcal{O})$ is $\mathcal{H} = (ground(\mathcal{P}), \mathcal{O})$. $ground(\mathcal{H})$ is a grounded PHKB^{FS}.

Given a grounded PHKB^{FS} $\mathcal{H} = (\mathcal{P}, \mathcal{O})$ with $\mathcal{P} = (\mathcal{R}, \mathcal{F})$, the set $\mathsf{KA}(\mathcal{H})$ of *known atoms* in $\mathcal{H}$ is the set of all the atoms that occur in $\mathcal{R}$ or $\mathcal{F}$.

In order to define a probability measure for a PHKB^{FS}, we can define a σ-algebra for the PHKB^{FS} as the product σ-algebra (see Definition 25 in Appendix A) of its LP and DL portions, as follows.

Definition 13. Given a grounded PHKB^{FS} $\mathcal{H} = (\mathcal{P}, \mathcal{O})$, let $(W_{\mathcal{P}}, \Omega_{\mathcal{P}}, \mu_{\mathcal{P}})$ and $(W_{\mathcal{O}}, \Omega_{\mathcal{O}}, \mu_{\mathcal{O}})$ be the probability measures for $\mathcal{P}$ and $\mathcal{O}$ respectively. The probability measure for $\mathcal{H}$ is the product measure

$$(W_{\mathcal{H}}, \Omega_{\mathcal{H}}, \mu_{\mathcal{H}}) = (W_{\mathcal{P}} \times W_{\mathcal{O}}, \Omega_{\mathcal{P}} \otimes \Omega_{\mathcal{O}}, \mu_{\mathcal{P}} \cdot \mu_{\mathcal{O}})$$

Definition 14. A world w of a PHKB^{FS} $\mathcal{H}$ is an element of $W_{\mathcal{H}}$ and is of the form $w = (w_{\mathcal{P}}, w_{\mathcal{O}})$, where $w_{\mathcal{P}} = (\mathcal{R}_w, \mathcal{F}_w)$ and $w_{\mathcal{O}} = (\mathcal{A}_w, \mathcal{E}_w)$.

In the definition above, $\mathcal{R}_w = \mathcal{R}$ and $\mathcal{A}_w = \mathcal{A}$ are respectively the set of rules and certain axioms contained in world w, while $\mathcal{F}_w$ and $\mathcal{E}_w$ are respectively the set of probabilistic facts from $\mathcal{F}$ and of probabilistic axioms from $\mathcal{E}$ selected to be included in w without their probability.

Definition 15 *(Sound PHKB^{FS}).* A PHKB^{FS} $\mathcal{H}$ is *sound* if and only if, for each world w of $\mathcal{H}$, $\mathsf{WFM}(w)$ is total.

A query q is an atom from $\mathsf{KA}(\mathcal{H})$. The probability of a query can be defined as $P(q) = \mu_{\mathcal{H}}(\{w \mid w \in W_{\mathcal{H}}, w \vDash q\})$. In order for the probability to be well defined, we have to prove that the set $\{w \mid w \in W_{\mathcal{H}}, w \vDash q\}$ is measurable, i.e., that it belongs to $\Omega_{\mathcal{H}}$. We do so in the remainder of this section.

The semantics of PHKB^{FS}s is based on the Iterated Fixpoint semantics defined in Section 4, where two-/three-valued interpretations are defined. In this section we need to define *parameterized* two-/three-valued interpretations. Basically, a two-valued parameterized interpretation associates to each atom the set of composite choices that identify the sets of worlds where the atom is true, or false; a three-valued parameterized interpretation associates to each atom two sets of composite choices, characterizing the sets of worlds where the atom is true and false, respectively.

Definition 16 *(Parameterized two-valued interpretations).* Given a grounded PHKB^{FS} $\mathcal{H}$, a *parameterized positive two-valued interpretation* $\mathscr{T}r$ is a set of pairs (a, ϕ_a) with $a \in \mathsf{KA}(\mathcal{H})$ and $\phi_a \in \Omega_{\mathcal{H}}$. Similarly, a *parameterized negative two-valued interpretation* $\mathscr{F}a$ is a set of pairs $(a, \phi_{\sim a})$ with $a \in \mathsf{KA}(\mathcal{H})$ and $\phi_{\sim a} \in \Omega_{\mathcal{H}}$.

Following this definition, the intuition is that (a, ϕ_a) means that a is true in the worlds of ϕ_a. On the other side $(a, \phi_{\sim a})$ means that a is false in the worlds of $\phi_{\sim a}$.

Parameterized two-valued interpretations form a complete lattice where the partial order is defined as $\mathscr{I} \leq \mathscr{J}$ if $\forall (a, \phi_a) \in \mathscr{I}, (a, \theta_a) \in \mathscr{J}$: $\phi_a \subseteq \theta_a$. For a set T of parameterized two-valued interpretations, the least upper bound and greatest lower bound always exist and are respectively

$$\text{lub}(T) = \{(a, \bigcup_{\mathscr{I} \in T, (a, \phi_a) \in \mathscr{I}} \phi_a) \mid a \in \mathsf{KA}(\mathcal{H})\}$$

and

$$\text{glb}(T) = \{(a, \bigcap_{\mathscr{I} \in T, (a, \phi_a) \in \mathscr{I}} \phi_a) \mid a \in \mathsf{KA}(\mathcal{H})\}.$$

The top element is $\{(a, W_{\mathcal{P}} \times W_{\mathcal{O}}) \mid a \in \mathsf{KA}(\mathcal{H})\}$ and the bottom element is $\{(a, \emptyset) \mid a \in \mathsf{KA}(\mathcal{H})\}$.

Definition 17 *(Parameterized three-valued interpretations).* Given a grounded PHKB^{FS} $\mathcal{H}$, a *parameterized three-valued interpretation* $\mathscr{I}$ is a set of triples $(a, \phi_a, \phi_{\sim a})$ with $a \in \mathsf{KA}(\mathcal{H})$, $\phi_a \in \Omega_{\mathcal{H}}$ and $\phi_{\sim a} \in \Omega_{\mathcal{H}}$. A parameterized three-valued interpretation $\mathscr{I}$ is *consistent* if $\forall (a, \phi_a, \phi_{\sim a}) \in \mathscr{I} : \phi_a \cap \phi_{\sim a} = \emptyset$.

Parameterized three-valued interpretations form a complete lattice where the partial order is defined as $\mathscr{I} \leq \mathscr{J}$ if $\forall (a, \phi_a, \phi_{\sim a}) \in \mathscr{I}, (a, \theta_a, \theta_{\sim a}) \in \mathscr{J}$: $\phi_a \subseteq \theta_a$ and $\phi_{\sim a} \subseteq \theta_{\sim a}$. For a set T of parameterized three-valued interpretations, the least upper bound and greatest lower bound always exist and are respectively

$$\mathrm{lub}(T) = \{(a, \bigcup_{\mathscr{I} \in T, (a, \phi_a, \phi_{\sim a}) \in \mathscr{I}} \phi_a, \bigcup_{\mathscr{I} \in T, (a, \phi_a, \phi_{\sim a}) \in \mathscr{I},} \phi_{\sim a}) \mid a \in \mathsf{KA}(\mathcal{H})\}$$

and

$$\mathrm{glb}(T) = \{(a, \bigcap_{\mathscr{I} \in T, (a, \phi_a, \phi_{\sim a}) \in \mathscr{I}} \phi_a, \bigcap_{\mathscr{I} \in T, (a, \phi_a, \phi_{\sim a}) \in \mathscr{I}} \phi_{\sim a}) \mid a \in \mathsf{KA}(\mathcal{H})\}.$$

The top element is

$$\{(a, W_{\mathcal{P}} \times W_{\mathcal{O}}, W_{\mathcal{P}} \times W_{\mathcal{O}}) \mid a \in \mathsf{KA}(\mathcal{H})\}$$

and the bottom element is

$$\{(a, \emptyset, \emptyset) \mid a \in \mathsf{KA}(\mathcal{H})\}.$$

As in the case of the HKB semantics (Sect. 4), we will be interested in pairs of sets of axioms and atoms such that the objective knowledge of the set of axioms w.r.t. the set of atoms (Formula (14)) entails an atom.

Definition 18 *($POpTrue^{\mathcal{H}}_{\mathscr{I}}(\mathscr{T}r)$ and $POpFalse^{\mathcal{H}}_{\mathscr{I}}(\mathscr{F}a)$).* For a grounded PHKB^{FS} $\mathcal{H} = (\mathcal{R}, \mathcal{F}, \mathcal{A}, \mathcal{E})$, a parameterized two-valued positive interpretation

$$\mathscr{T}r = \{(a, \theta_a) | a \in \mathsf{KA}(\mathcal{H})\},$$

a parameterized two-valued negative interpretation

$$\mathscr{F}a = \{(a, \theta_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\},$$

and a parameterized three-valued interpretation

$$\mathscr{I} = \{(a, \phi_a, \phi_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\},$$

we define $POpTrue^{\mathcal{H}}_{\mathscr{I}}(\mathscr{T}r) = \{(a, \gamma_a) \mid a \in \mathsf{KA}(\mathcal{H})\}$ where

$$\gamma_a = \begin{cases} \omega_{\{\{(a,1)\}\}} \times W_{\mathcal{O}} & \text{if } a \in \mathcal{F} \\ \Big(\bigcup_{a \leftarrow b_1, \dots, b_m, \sim c_1, \dots \sim c_n \in \mathcal{R}} \bigcap_{i=1,\dots,m} (\phi_{b_i} \cup \theta_{b_i}) \bigcap_{i=1,\dots,n} \phi_{\sim c_i} \Big) & \\ \bigcup \Big(\bigcup_{\substack{G \subseteq \mathsf{KA}(\mathcal{H}) \\ E \subseteq \mathcal{E}, \\ \mathsf{OB}_{\mathcal{A} \cup E, G} \models a}} \bigcap_{g \in G} (\phi_g \cup \theta_g) \bigcap (W_{\mathcal{P}} \times \omega_{\{(e,1) \mid e \in E\}}) \Big) & \text{otherwise} \end{cases}$$

and $POpFalse^{\mathcal{H}}_{\mathscr{I}}(\mathscr{F}a) = \{(a, \gamma_{\sim a}) \mid a \in \mathsf{KA}(\mathcal{H})\}$ where

$$\gamma_{\sim a} = \begin{cases} \omega_{\{\{(a,0)\}\}} \times W_{\mathcal{O}} & \text{if } a \in \mathcal{F} \\ \Big(\bigcap_{a \leftarrow b_1, \dots, b_m, \sim c_1, \dots \sim c_n \in \mathcal{R}} \bigcup_{i=1,\dots,m} (\phi_{\sim b_i} \cup \theta_{\sim b_i}) \bigcup_{i=1,\dots,n} \phi_{c_i} & \\ \bigcup_{\substack{G \subseteq \mathsf{KA}(\mathcal{H}) \\ E \subseteq \mathcal{E}, \\ \mathsf{OB}_{\mathcal{A} \cup E, G} \models \neg a}} \bigcap_{g \in G} \phi_g \bigcap (W_{\mathcal{P}} \times \omega_{\{\{(e,1) \mid e \in E\}\}}) \Big) & \\ \bigcap_{\substack{G \subseteq \mathsf{KA}(\mathcal{H}) \\ E \subseteq \mathcal{E}, \\ \mathsf{OB}_{\mathcal{A} \cup E, G} \models a}} \Big(\bigcup_{g \in G} (\phi_{\sim g} \cup \theta_{\sim g}) \bigcup_{e \in E} (W_{\mathcal{P}} \times \omega_{\{\{(e,0)\}\}}) \Big) & \text{otherwise} \end{cases}$$

Proposition 1 *(Monotonicity of $POpTrue^{\mathcal{H}}_{\mathscr{I}}$ and $POpFalse^{\mathcal{H}}_{\mathscr{I}}$). $POpTrue^{\mathcal{H}}_{\mathscr{I}}$ and $POpFalse^{\mathcal{H}}_{\mathscr{I}}$ are monotonic in their argument.*

$POpTrue^{\mathcal{H}}_{\mathscr{I}}$ and $POpFalse^{\mathcal{H}}_{\mathscr{I}}$ are monotonic so they both have a least fixpoint and a greatest fixpoint.

Definition 19 *(Iterated fixed point for a PHKB^{FS}).* For a grounded PHKB^{FS} $\mathcal{H}$, and a parameterized three-valued interpretation $\mathscr{I}$, let $PIFP^{\mathcal{H}}(\mathscr{I})$ be defined as

$$PIFP^{\mathcal{H}}(\mathscr{I}) = \{(a, \phi_a, \phi_{\sim a}) \mid (a, \phi_a) \in \mathsf{lfp}(POpTrue^{\mathcal{H}}_{\mathscr{I}}),$$
$$(a, \phi_{\sim a}) \in \mathsf{gfp}(POpFalse^{\mathcal{H}}_{\mathscr{I}})\}.$$

Proposition 2 *(Monotonicity of $PIFP^{\mathcal{H}}$). $PIFP^{\mathcal{H}}$ is monotonic.*

The monotonicity property ensures that $PIFP^{\mathcal{H}}$ has a least fixpoint. Let us denote $\mathsf{lfp}(PIFP^{\mathcal{H}})$ with $\mathsf{WFMP}(\mathcal{H})$. We call *depth* of P the smallest ordinal δ such that $PIFP^{\mathcal{H}} \uparrow \delta = \mathsf{WFMP}(P)$.

Given a parameterized positive two-valued interpretation $\mathcal{T}r = \{(a, \phi_a | a \in \mathsf{KA}(\mathcal{H})\}$ and a world $w = (w_{\mathcal{P}}, w_{\mathcal{O}})$ of $\mathcal{H}$, the *projection of $\mathcal{T}r$ with respect to w* is the two-valued interpretation $\mathcal{T}r^w = \{a | w \in \phi_a\}$. Given a parameterized negative two-valued interpretation $\mathcal{F}a = \{(a, \phi_{\sim a} | a \in \mathsf{KA}(\mathcal{H})\}$ and a world $w = (w_{\mathcal{P}}, w_{\mathcal{O}})$ of $\mathcal{H}$, the *projection of $\mathcal{F}a$ with respect to w* is the two-valued interpretation $\mathcal{F}a^w = \{a | w \in \phi_{\sim a}\}$. Given a parameterized three-valued interpretation $\mathscr{I} = \{(a, \phi_a, \phi_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\}$ and a world $w = (w_{\mathcal{P}}, w_{\mathcal{O}})$ of $\mathcal{H}$, the *projection of $\mathscr{I}$ with respect to w* is the three-valued interpretation $\mathscr{I}^w = (\mathcal{I}_T^w, \mathcal{I}_F^w)$ where $\mathcal{I}_T = \{(a, \phi_a) | (a, \phi_a, \phi_{\sim a}) \in \mathscr{I}\}$ and $\mathcal{I}_F = \{(a, \phi_{\sim a}) | (a, \phi_a, \phi_{\sim a}) \in \mathscr{I}\}$.

Lemma 1 *(Model Equivalence). For a grounded PHKB^{FS} $\mathcal{H} = (\mathcal{R}, \mathcal{F}, \mathcal{A}, \mathcal{E})$, for every world w and iteration α, we have:*

$$IFP^w \uparrow \alpha = (PIFP^{\mathcal{H}} \uparrow \alpha)^w$$

Now we can prove that $PIFP^{\mathcal{H}}$ is sound and complete.

Lemma 2 *(Soundness and completeness of $PIFP^{\mathcal{H}}$). For a sound grounded PHKB^{FS} $\mathcal{H}$, let $PIFP^{\mathcal{H}} \uparrow \alpha = \{(a, \phi_a^\alpha, \phi_{\sim a}^\alpha) | a \in \mathsf{KA}(\mathcal{H})\}$ for all α. For every atom $a \in \mathsf{KA}(\mathcal{H})$ and world w there is an iteration α_0 such that for all $\alpha > \alpha_0$ we have:*

$$w \in \phi_a^\alpha \leftrightarrow \mathsf{WFM}(w) \vDash a \tag{17}$$

$$w \in \phi_{\sim a}^\alpha \leftrightarrow \mathsf{WFM}(w) \vDash \sim a \tag{18}$$

Now we can prove that every query for every sound program is well-defined.

Theorem 3 *(Well-definedness of the distribution semantics). For a sound grounded PHKB^{FS} $\mathcal{H}$, for every atom $a \in \mathsf{KA}(\mathcal{H})$, $\mu_{\mathcal{H}}(\{w \mid w \in W_{\mathcal{H}}, w \vDash a\})$ is well-defined.*

Example 8 *(Spillover, Probabilistic Query).* Let us consider the PHKB^{FS} of Example 7 and the queries $q_1 = \mathit{safe}(t)$ and $q_2 = \mathit{spillover\_count}(t, s(0))$.

Tables 2 and 3 show the computation of the first iteration of the $PIFP^{\mathcal{H}}$ operator for, respectively:

1. $\mathit{mutation}(t, Y)$ and $\mathit{mutated}(t)$ to show how the operators deal with DL axioms.
2. $\mathit{safe}(t)$ to show how the operator work with a LP rule.

For the complete computation of the first iteration of the $PIFP^{\mathcal{H}}$ operator we refer to Tables from I.5 to I.10 in Appendix I.

In the tables, fact f is $spillover(X, Y)$ while axiom e is

$$t : \exists mutation \top.$$

Each column shows the sets of composite choices associated to an atom of $\mathsf{KA}(\mathcal{H})$ at each step of the inner and outer operators. In particular, for each atom a, the line labelled $\mathscr{I}^\alpha$ shows the sets of composite choices S_a and $S_{\sim a}$ such that $S_a = \omega_{\phi_a}$ and $S_{\sim a} = \omega_{\phi_{\sim a}}$ where $\mathsf{PIFP}^{\mathcal{H}} \uparrow \alpha = \{(a, \phi_a, \phi_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\}$; the lines labelled $\mathcal{T}r^\delta$ (resp. $\mathcal{F}a^\delta$) show the set of composite choices $S_a = \omega_{\theta_a}$ and $S_{\sim a} = \omega_{\theta_{\sim a}}$ such that $(a, \theta_a) \in POpTrue^{\mathcal{H}}_{\mathsf{PIFP}^{\mathcal{H}} \uparrow \alpha} \uparrow \delta$ (resp. $(a, \theta_a) \in POpFalse^{\mathcal{H}}_{\mathsf{PIFP}^{\mathcal{H}} \uparrow \alpha} \uparrow \delta$).

A covering set of explanations for $q_1 = \mathit{safe}(t)$ is

$$\{\{(e, 0)\}, \{(f\{X/t, Y/0\}, 0)\}, \{(e, 1), (f\{X/t, Y/0\}, 1), (f\{X/t, Y/s(0)\}, 0)\}\}$$

The probability of $\mathit{safe}(t)$ is $(0.2 + 0.4 - (0.2 \times 0.4)) + 0.8 \times 0.6 \times 0.4 = 0.52 + 0.192 = 0.712$, that is the noisy-or of the two first choices (which share the world where both are not present) plus the probability of the latter choice. A covering set of explanations for $q_2 = \mathit{spillover\_count}(t, s(0))$ is

$$\{\{(e, 1), (f\{X/t, Y/0\}, 1)\}\}.$$

The probability of $\mathit{spillover\_count}(t, s(0))$ is $0.8 \times 0.6 = 0.48$

Table 2
Iterations of the operators for $mutation(t, Y)$, and $mutated(t)$.

	mutation(t,Y)	mutated(t)
$\mathcal{I}^0$	$\emptyset, \emptyset$	$\emptyset, \emptyset$
$\mathcal{T}r^1$	$\{\{(e,1)\}\}$	$\emptyset$
$\mathcal{T}r^2$	$\{\{(e,1)\}\}$	$\{\{(e,1)\}\}$
$\mathcal{T}r^3$	$\{\{(e,1)\}\}$	$\{\{(e,1)\}\}$
$\mathcal{T}r^4$	$\{\{(e,1)\}\}$	$\{\{(e,1)\}\}$
$\mathcal{F}a^1$	$\{\{(e,0)\}\}$	$\{\emptyset\}$
$\mathcal{F}a^2$	$\{\{(e,0)\}\}$	$\{\{(e,0)\}\}$
$\mathcal{F}a^3$	$\{\{(e,0)\}\}$	$\{\{(e,0)\}\}$
$\mathcal{F}a^4$	$\{\{(e,0)\}\}$	$\{\{(e,0)\}\}$
$\mathcal{I}^1$	$\{\{(e,1)\}\}, \{\{(e,0)\}\}$	$\{\{(e,1)\}\}, \{\{(e,0)\}\}$

Table 3
Iterations of the operators for $safe(t)$.

	safe(t)
$\mathcal{I}^0$	$\emptyset, \emptyset$
$\mathcal{T}r^1$	$\emptyset$
$\mathcal{T}r^2$	$\emptyset$
$\mathcal{T}r^3$	$\emptyset$
$\mathcal{T}r^4$	$\emptyset$
$\mathcal{T}r^5$	$\emptyset$
$\mathcal{T}r^6$	$\{\{(e,0)\}, \{(f\{X/t, Y/0\}, 0)\}\}$
$\mathcal{T}r^7$	$\{\{(e,0)\}, \{(f\{X/t, Y/0\}, 0)\}, \{(e,1), (f\{X/t, Y/0\}, 1), (f\{X/t, Y/s(0)\}, 0)\}\}$
$\mathcal{T}r^8$	$\{\{(e,0)\}, \{(f\{X/t, Y/0\}, 0)\}, \{(e,1), (f\{X/t, Y/0\}, 1), (f\{X/t, Y/s(0)\}, 0)\}\}$
$\mathcal{F}a^1$	$\{\emptyset\}$
$\mathcal{F}a^2$	$\{\emptyset\}$
$\mathcal{F}a^3$	$\{\emptyset\}$
$\mathcal{F}a^4$	$\{\emptyset\}$
$\mathcal{F}a^5$	$\{\emptyset\}$
$\mathcal{F}a^6$	$\{\{(e,1), (f\{X/t, Y/0\}, 1), (f\{X/t, Y/s(0)\}, 1)\}\}$
$\mathcal{F}a^7$	$\{\{(e,1), (f\{X/t, Y/0\}, 1), (f\{X/t, Y/s(0)\}, 1)\}\}$
$\mathcal{I}^1$	$\{\{(e,0)\}, \{(f\{X/t, Y/0\}, 0)\}, \{(e,1), (f\{X/t, Y/0\}, 1), (f\{X/t, Y/s(0)\}, 0)\}\}$, $\{\{(e,1), (f\{X/t, Y/0\}, 1), (f\{X/t, Y/s(0)\}, 1)\}\}$

6. Conclusions and future work

In this article we have presented a probabilistic extension of Hybrid Knowledge Bases with Function Symbols (HKB^{FS}s) introduced by [4], that in turn extend Hybrid Knowledge Bases by [36] with function symbols. The semantics of HKB^{FS}s combines LP and DLs while exhibiting desirable properties such as faithfulness and tightness. The resulting Probabilistic HKB^{FS}s (PHKB^{FS}s) contain both (probabilistic) LP rules and DL axioms. PHKB^{FS}s are equipped with a semantics based on the well-founded MKNF semantics extended with the treatment of probability. Using this semantics, we are able to assign a probability to every query to sound programs.

In the future we plan to study restrictions to make query answering in PHKB^{FS}s decidable and identify necessary and sufficient conditions that ensures the soundness of PHKB^{FS}. We will consider programs to be finitely ground ([13,14]) or having strongly bounded term size ([50,46]) which are identified as conditions ensuring decidability of logic programs.

We also plan to equip this semantics with a reasoner for computing the probability of a query given a PHKB^{FS}, extending the **SLG**($\mathcal{O}$) proof procedure [5] for MKNF HKBs, which is sound and complete for the well-founded semantics. The reasoner, as intended at the moment, will apply the **SLG**($\mathcal{O}$) proof procedure [5], integrating the TRILL reasoner [58,56] as the DL oracle and PITA [49] to cope with the PLP part of the probabilistic HKB.

CRediT authorship contribution statement

Marco Alberti: Writing – review & editing, Writing – original draft, Investigation, Formal analysis, Conceptualization. **Evelina Lamma:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Fabrizio Riguzzi:** Writing – review & editing, Writing – original draft, Investigation, Formal analysis, Conceptualization. **Riccardo Zese:** Writing – review & editing, Writing – original draft, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been partially supported by the Spoke 1 "FutureHPC & BigData" of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 - Next Generation EU (NGEU) - CUP F77G22000120006 and by "FAIR – Future Artificial Intelligence Research" cascade project of Partenariato Esteso PE00000013 – Spoke 8 "Pervasive AI" funded by MUR PNRR - M4C2 - Investimento 1.3 (Decreto Direttoriale MUR n. 341 of 15th March 2022) - Next Generation EU (NGEU) - CUP J33C22002830006. Marco Alberti, Fabrizio Riguzzi and Riccardo Zese are members of the "Gruppo Nazionale Calcolo Scientifico - Istituto Nazionale di Alta Matematica" (GNCS-INdAM).

Appendix A. Probability theory

In this section, we review some background on probability theory, in particular Kolmogorov probability theory, that will be needed in the following. Most of the definitions are taken from [16] and [47].

We define the *sample space* W as the set composed by the elements that are outcomes of the random process we want to model. For instance, if we consider the toss of a coin whose outcome could be heads h or tails t, the sample space is defined as $W^{coin} = \{h, t\}$. If we throw 2 coins, then $W^{2coins} = \{(h,h),(h,t),(t,h),(t,t)\}$. If the number of coins is infinite then $W^{coins} = \{(o_1, o_2, \ldots) \mid o_i \in \{h, t\}\}$.

Definition 20 *(σ-Algebra).* A non-empty set Ω of subsets of W is a σ-algebra on the set W iff:

- $W \in \Omega$
- Ω is closed under complementation: $\omega \in \Omega \Rightarrow \omega^c = \Omega \setminus \omega \in \Omega$
- Ω is closed under countable union: if $\omega_i \in \Omega \Rightarrow \bigcup_i w_i \in \Omega$

The elements of a σ-algebra Ω are called *measurable sets* or *events*, Ω is called event space and (W, Ω) is called *measurable space*. When W is finite, Ω is usually the powerset of W, but, in general, it is not necessary that every subset of W must be present in Ω. For example, to model a coin toss, we can consider the set of events $\Omega^{coin} = \mathbb{P}(W^{coin})$ and $\{h\}$ an event corresponding to the outcome heads.

Definition 21 *(Minimal σ-algebra).* Let C be an arbitrary non-empty collection of subsets of W. The intersection of all σ-algebras containing all the elements of C is called the σ-algebra generated by C or the minimal $sigma$-algebra containing C. It is denoted by $\sigma(C)$. Moreover, $\sigma(C)$ always exists and is unique [16].

Now we introduce the definition of probability measure:

Definition 22 *(Probability measure).* Given a measurable space (W, Ω), a probability measure is a finite set function $\mu : \Omega \to \mathbb{R}$ that satisfies the following three axioms (called Kolmogorov axioms):

- a_1: $\mu(\omega) \geq 0 \; \forall \, \omega \in \Omega$
- a_2: $\mu(W) = 1$
- a_3: μ is *countably additive* (or σ-additive): if $O = \{\omega_1, \omega_2, \ldots\} \subseteq \Omega$ is a countable collection of pairwise disjoint sets, then $\mu(\bigcup_{\omega \in O}) = \sum_i \mu(\omega_i)$

Axioms a_1 and a_2 state that we measure the probability of an event with a number between 0 and 1. Axiom a_3 states that the probability of the union of disjoint events is equal to the sum of the probability of every single event. (W, Ω, μ) is called a *probability space*.

For example, if we consider the toss of a coin, $(W^{coin}, \Omega^{coin}, \mu^{coin})$ with $\mu^{coin}(\emptyset) = 0$, $\mu^{coin}(\{h\}) = 0.5$, $\mu^{coin}(\{t\}) = 0.5$ and $\mu^{coin}(\{h, t\}) = 1$ is a probability space.

Definition 23 *(Measurable function).* Given a probability space (W, Ω, μ) and a measurable space (S, Σ), a function X: W → S is measurable if $X^{-1}(\sigma) = \{w \in W \mid X(w) \in \sigma\} \in \Omega$, $\forall \sigma \in \Sigma$.

Definition 24 *(Random variable).* Let (W, Ω, μ) be a probability space and (S, Σ) be a measurable space. A measurable function $X : W \to S$ is a random variable. The elements of S are called values of X. We indicate with $P(X \in \sigma)$ for all $\sigma \in \Sigma$ the probability that a random variable X has value in σ, that is, $\mu(X^{-1}(\sigma))$. If S is countable, X is a discrete random variable. If S is uncountable, X is a continuous random variable.

The *probability distribution* of a discrete random variable is defined as $P(X \in \{x\})\, \forall x \in S$ and it is often abbreviated with $P(X = x)$ or $P(x)$.

In the following, we will need to consider the product of measurable spaces.

Definition 25 *(Product σ-algebra).* Given two measurable spaces (W_1, Ω_1) and (W_2, Ω_2), the product σ-algebra $\Omega_1 \otimes \Omega_2$ is defined as $\Omega_1 \otimes \Omega_2 = \sigma(\{\omega_1 \times \omega_2 \mid \omega_1 \in \Omega_1, \omega_2 \in \Omega_2\})$. The result of $\Omega_1 \otimes \Omega_2$ is different from the Cartesian product $\Omega_1 \times \Omega_2$ because it is the minimal σ-algebra generated by all the possible couples of elements from Ω_1 and Ω_2. $\Omega_1 \otimes \Omega_2$ is also called a tensor product.

Theorem 4 *(Theorem 6.3.1 from [16]). Given two probability spaces (W_1, Ω_1, μ_1) and (W_2, Ω_2, μ_2), there exists a unique probability space (W, Ω, μ) such that $W = W_1 \times W_2$, $\Omega = \Omega_1 \otimes \Omega_2$ and*

$$\mu(\omega_1 \times \omega_2) = \mu_1(\omega_1) \cdot \mu_2(\omega_2)$$

for $\omega_1 \in \Omega_1$ and $\omega_2 \in \Omega_2$. Measure μ is called the product measure *of μ_1 and μ_2 and is denoted also by $\mu_1 \times \mu_2$. Moreover, for any $\omega \in \Omega$, let's define its* sections *as*

$$\omega^{(1)}(w_1) = \{w_2 \mid (w_1, w_2) \in \omega\} \quad \omega^{(2)}(w_2) = \{w_1 \mid (w_1, w_2) \in \omega\}.$$

Then, both $\omega^{(1)}(w_1)$ and $\omega^{(2)}(w_2)$ are measurable according to (W_2, Ω_2, μ_2) and (W_1, Ω_1, μ_1) respectively, i.e., $\omega^{(1)}(w_1) \in \Omega_2$ and $\omega^{(2)}(w_2) \in \Omega_1$. $\mu_2(\omega^{(1)}(w_1))$ and $\mu_1(\omega^{(2)}(w_2))$ are well-defined real functions, the first on W_1 and the second on W_2.

Measure $\mu = \mu_1 \times \mu_2$ for every $\omega \in \Omega$ also satisfies

$$\mu(\omega) = \int_{W_2} \mu_1(\omega^{(2)}(w_2)) d\mu_2 = \int_{W_1} \mu_2(\omega^{(1)}(w_1)) d\mu_1.$$

When sample spaces are countable, integrals are replaced by summations. So if both W_1 and W_2 are countable, we obtain

$$\mu(\omega) = \sum_{w_2 \in W_2} \mu_1(\omega^{(2)}(w_2)) = \sum_{w_1 \in W_1} \mu_2(\omega^{(1)}(w_1)) d\mu_1.$$

Appendix B. Set theory

A *one-to-one* function $f : A \to B$ is such that if $f(a) = f(b)$, then $a = b$, i.e., no element of B is the image of more than one element of A. A set A is *equipotent* with a set B if there exists a one-to-one function from A to B. A set A is *denumerable* if it is equipotent to the set of natural numbers $\mathbb{N}$. A set A is *countable* if there exists a one-to-one correspondence between the elements of A and the elements of some subset B of the set of natural numbers. Otherwise, A is termed *uncountable*. If A is countable and $B = \{1, 2, \ldots, n\}$, then A is called *finite* with n elements. $\emptyset$ (empty set) is considered a finite set with 0 elements. We define *powerset* of any set A, indicated with $\mathbb{P}(A)$, the set of all subsets including the empty set. For any reference space S and subset A of S, we denote with A^c the *complement* of A, i.e., $S \setminus A$, the set of all elements of S that do not belong to A.

An *order* on a set A is a binary relation $\leq$ that is reflexive, antisymmetric and transitive. If a set A has an order relation $\leq$, it is termed a *partially ordered set*, sometimes abbreviated with *ordered set*. A partial order $\leq$ on a set A is called a *total order* if $\forall a, b \in A,\ a \geq b$ or $b \geq a$. In this case, A is called *totally ordered*. The *upper bound* of a subset A of some ordered set B is an element $b \in B$ such that $\forall a \in A,\ a \leq b$. If $b \leq b'$ for all upper bounds b', then b is the *least upper bound* (lub). The definitions for *lower bound* and *greatest lower bound* (glb) are similar. If glb and lub exist, they are unique. A partially ordered set $(A, \leq)$ is a *complete lattice* if glb and lub exist for every subset S of A. A complete lattice A always has a *top element* $\top$ such that $\forall a \in A,\ a \leq \top$ and a *bottom element* $\bot$ such that $\forall a \in A,\ \bot \leq a$. A function $f : A \to B$ between two partially order set A and B is called *monotonic* if, $\forall a, b \in A,\ a \leq b$ implies that $f(a) \leq f(b)$. For an in-depth treatment of this topic see [19].

Appendix C. First order logic

A *signature* is a triple $(\Sigma_c, \Sigma_f, \Sigma_p)$ where Σ_c is a set of *constants*, Σ_f is a set of *function symbols*, each with an associated natural number called *arity*, and Σ_p is a set of *predicate symbols* with arity, containing the equality binary predicate $\approx$. A *term* is a constant, variable, or a function symbol applied to as many terms as the symbol's arity. A *first order formula* is

$$\psi ::= \mathtt{true} | P(t_1, \ldots, t_n) | \neg\psi | \psi \wedge \psi | \exists x : \psi \tag{C.1}$$

where $P(t_1, \ldots, t_n)$ is called an *atom*, P is a predicate symbol of arity n and the t_is are terms. In the formula $\exists x : \psi$, the occurrences of variable x in ψ are in the *scope* of the $\exists x$ *quantifier*; a variable not in the scope of any quantifier is *free*; a formula with no free variables is *closed*. Common syntactic shortcuts are shown in Table C.4.

Let Σ be a signature and Δ a non-empty set called universe. Then a *first order interpretation* I over Σ and Δ maps each $c \in \Sigma_c$ to an object $c^I \in \Delta$, each $f \in \Sigma_f$ to a function $f^I : \Delta^n \to \Delta$ (where n is f's arity) and each $p \in \Sigma_p$ to a relation $p^I \subseteq \Delta^n$ (where n is p's arity).

Table C.4
Syntactic shortcuts in FOL.

formula	is a shortcut for
$\psi_1 \vee \psi_2$	$\neg(\neg\psi_1 \wedge \neg\psi_2)$
$\forall x : \psi$	$\neg(\exists x : \neg\psi)$
$\psi_1 \supset \psi_2$	$\neg\psi_1 \vee \psi_2$
$\psi_1 \equiv \psi_2$	$(\psi_1 \supset \psi_2) \wedge (\psi_2 \supset \psi_1)$
`false`	$\neg$`true`
$t_1 \approx t_2$	$\approx(t_1, t_2)$
$t_1 \not\approx t_2$	$\neg(t_1 \approx t_2)$

Appendix D. Well-founded semantics for normal logic programs

The well-founded semantics [54] assigns a three-valued model to a normal logic program. We give here the alternative definition of the WFS of Przymusinski [43] that is based on an iterated fixpoint.

A *three-valued interpretation* $\mathcal{I}$ is a pair (I_T, I_F) where I_T and I_F are subsets of $\mathcal{B}_P$ and represent, respectively, the set of true and false atoms. So a is true in $\mathcal{I}$ if $a \in I_T$ and is false in $\mathcal{I}$ if $a \in I_F$, and $\sim a$ is true in $\mathcal{I}$ if $a \in I_F$ and is false in $\mathcal{I}$ if $a \in I_T$. If $a \notin I_T$ and $a \notin I_F$, then a assumes the third truth value, *undefined*.

The set $Int3$ of three-valued interpretations for a program P forms a complete lattice where the partial order $\leq$ is defined as $(I_T, I_F) \leq (J_T, J_F)$ if $I_T \subseteq J_T$ and $I_F \subseteq J_F$. The least upper bound and greatest lower bound are defined as $\mathsf{lub}(X) = \bigcup_{\mathcal{I} \in X} \mathcal{I}$ and $\mathsf{glb}(X) = \bigcap_{\mathcal{I} \in X} \mathcal{I}$. The bottom and top element are, respectively, $(\emptyset, \emptyset)$ and $(\mathcal{B}_P, \mathcal{B}_P)$. Let *Int2* be the set of two-valued interpretations.

Definition 26 *($OpTrue_{\mathcal{I}}^P$ and $OpFalse_{\mathcal{I}}^P$ operators).* For a normal program P and a three-valued interpretation $\mathcal{I}$, we define the operators $OpTrue_{\mathcal{I}}^P : Int2 \rightarrow Int2$ and $OpFalse_{\mathcal{I}}^P : Int2 \rightarrow Int2$ as

$OpTrue_{\mathcal{I}}^P(Tr) =$ $\{a | a$ is not true in $\mathcal{I}$; and there is a clause $b \leftarrow l_1, ..., l_n$ in P, a grounding substitution θ such that $a = b\theta$ and for every $1 \leq i \leq n$ either $l_i\theta$ is true in $\mathcal{I}$, or $l_i\theta \in Tr\}$;

$OpFalse_{\mathcal{I}}^P(Fa) =$ $\{a | a$ is not false in $\mathcal{I}$; and for every clause $b \leftarrow l_1, ..., l_n$ in P and grounding substitution θ such that $a = b\theta$ there is some i $(1 \leq i \leq n)$ such that $l_i\theta$ is false in $\mathcal{I}$ or $l_i\theta \in Fa\}$.

$OpTrue_{\mathcal{I}}^P$ and $OpFalse_{\mathcal{I}}^P$ are both monotonic [43], so they both have least and greatest fixpoints. Let us now define an iterated fixpoint operator.

Definition 27 *(Iterated fixpoint).* For a normal program P, let IFP^P: $Int3 \rightarrow Int3$ be defined as

$$\mathsf{IFP}^P(\mathcal{I}) = \mathcal{I} \cup \langle \mathsf{lfp}(OpTrue_{\mathcal{I}}^P), \mathsf{gfp}(OpFalse_{\mathcal{I}}^P)\rangle.$$

IFP^P is monotonic [43] and thus has a least fixpoint $\mathsf{lfp}(\mathsf{IFP}^P)$. The well-founded model $\mathsf{WFM}(P)$ of P is $\mathsf{lfp}(\mathsf{IFP}^P)$.

If $\mathsf{WFM}(P) = (I_T, I_F)$ and $I_T \cup I_F = \mathcal{B}_P$, then the well-founded model is called *total* or *two-valued* and the program *dynamically stratified*.

Appendix E. Probabilistic semantics

In this section we describe in detail the semantics of ProbLog without function symbols (Appendix E.1) and with function symbols (Appendix E.2), and the DISPONTE semantics (Appendix E.3).

E.1. ProbLog programs' semantics without function symbols

Let $\mathcal{P} = (\mathcal{R}, \mathcal{F})$ a ProbLog program without functions symbols, $ground(\mathcal{R})$ and $ground(\mathcal{F})$ are finite. From the grounding $ground(\mathcal{P})$, we generate normal programs called *worlds* by including in a program the set of certain rules and a subset of the probabilistic facts, in all possible ways. Call $W_{\mathcal{P}}$ the set of all possible worlds. Since $\mathcal{F}$ is finite, so is $W_{\mathcal{P}}$ and $(W_{\mathcal{P}}, \mathbb{P}(W_{\mathcal{P}}))$ is a measurable space, where $\mathbb{P}(\cdot)$ is the powerset function. Thus the measurable sets or events are the sets of worlds.

Define function $\rho_{\mathcal{P}} : W_{\mathcal{P}} \rightarrow \mathbb{R}$ as

$$\rho_{\mathcal{P}}(w) = \prod_{p::a \in \mathcal{F} : a \in w} p \prod_{p::a \in \mathcal{F} : a \notin w} (1 - p)$$

and function $\mu_{\mathcal{P}} : \mathbb{P}(W_{\mathcal{P}}) \rightarrow \mathbb{R}$ as

$$\mu_{\mathcal{P}}(\omega) = \sum_{w \in \omega} \rho_{\mathcal{P}}(w)$$

Then $(W_\mathcal{P}, \mathbb{P}(W_\mathcal{P}), \mu_\mathcal{P})$ is a probability space and $\mu_\mathcal{P}$ is a probability measure.

Let $Q : W_\mathcal{P} \to \{0,1\}$ the function

$$Q(w) = \begin{cases} 1 & \text{if } w \vDash q \\ 0 & \text{otherwise} \end{cases} \tag{E.1}$$

where $v \vDash q$ means that q is true in the well-founded model of w. We assume that each world has a total well-founded model, i.e., each ground atom is either true or false in the world and cannot be undefined. We call programs satisfying this property *sound*. This means that if $w \nvDash q$, q is false in the well-founded model, so $Q(w) = 0$ only for false atoms.

Since the set of events is the powerset, then $Q^{-1}(\gamma) \in \mathbb{P}(W_\mathcal{P})$ for all $\gamma \subseteq \{0,1\}$ so Q is a random variable. The distribution of Q is defined by $P(Q=1)$ ($P(Q=0)$ is given by $1 - P(Q=1)$) and we indicate $P(Q=1)$ with $P(q)$.

So, $P(q)$ can be computed as

$$P(q) = \mu_\mathcal{P}(Q^{-1}(\{1\})) = \mu_\mathcal{P}(\{w | w \in W_\mathcal{P}, w \vDash q\}) = \sum_{w \in W_\mathcal{P} : w \vDash q} \rho_\mathcal{P}(w)$$

E.2. ProbLog programs' semantics with function symbols

When the program contains functions symbols, $ground(\mathcal{F})$ may be infinite. If $\mathcal{F}$ is infinite, as in Example 2, $\rho_\mathcal{P}(w)$ is a denumerable product of numbers in $(0,1)$ bounded away from 1, so $\rho_\mathcal{P}(w) = 0$ for any w and a different probability space must be defined.

We briefly recall that:

- an *atomic choice* (f,k) indicates whether a ground probabilistic fact $p :: f$ is selected ($k=1$) or not ($k=0$);
- a set of atomic choices is *consistent* if only one alternative is selected for the same probabilistic fact;
- a *composite choice* κ is a consistent set of atomic choices;
- a *selection* σ contains one atomic choice for every probabilistic fact and identifies a *world* w_σ;
- $W_\mathcal{P}$ is the set of worlds, which may be uncountable [46];
- the *set of worlds* ω_κ *compatible with a composite choice* κ is $\omega_\kappa = \{w_\sigma \in W_\mathcal{P} \mid \kappa \subseteq \sigma\}$ and may be uncountable.

Given a composite choice κ we define function $\rho_\mathcal{P}$ as

$$\rho_\mathcal{P}(\kappa) = \prod_{(f_i,1)\in\kappa} p_i \prod_{(f_i,0)\in\kappa} 1 - p_i.$$

Given a set of composite choices K, the *set of worlds* ω_K *compatible with* K is defined as $\omega_K = \bigcup_{\kappa\in K} \omega_\kappa$. Two sets K_1 and K_2 of composite choices are *equivalent* if $\omega_{K_1} = \omega_{K_2}$, that is, they identify the same set of worlds. If the union of two composite choices κ_1 and κ_2 is not consistent, then κ_1 and κ_2 are *incompatible*. We define *pairwise incompatible* a set K of composite choices if $\forall\kappa_1 \in K, \forall\kappa_2 \in K$, $\kappa_1 \neq \kappa_2$ implies that κ_1 and κ_2 are incompatible. If K is a pairwise incompatible set of composite choices, define $\mu_c(K) = \sum_{\kappa\in K} \rho_\mathcal{P}(\kappa)$.

Given a general set K of composite choices, we can construct a pairwise incompatible equivalent set through the technique of *splitting*. In detail, if f is a fact and κ is a composite choice that does not contain an atomic choice (f,k) for any k, the *split* of κ on f can be defined as the set of composite choices $S_{\kappa,f} = \{\kappa \cup \{(f,0)\}, \kappa \cup \{(f,1)\}\}$. In this way, κ and $S_{\kappa,f}$ identify the same set of possible worlds, i.e., $\omega_\kappa = \omega_{S_{\kappa,f}}$, and $S_{\kappa,f}$ is pairwise incompatible. It turns out that, given a set of composite choices, by repeatedly applying splitting it is possible to obtain an equivalent mutually incompatible set of composite choices [41].

Theorem 5 *(Existence of a pairwise incompatible set of composite choices [41]). Given a finite set K of composite choices, there exists a finite set K' of pairwise incompatible composite choices equivalent to K.*

Theorem 6 *(Equivalence of the measure of two equivalent pairwise incompatible finite set of finite composite choices [39]). If K_1 and K_2 are both pairwise incompatible finite sets of finite composite choices such that they are equivalent, then $\mu_c(K_1) = \mu_c(K_2)$.*

For a probabilistic logic program $\mathcal{P}$ and a ground atom q, we define function $Q : W_\mathcal{P} \to \{0,1\}$ as for the case of no function symbols, Eq. (1). As for programs without function symbols, we consider only sound programs, i.e., programs where each world has a total well-founded model.

Given a probabilistic logic program $\mathcal{P}$, we call $\Omega_\mathcal{P}$ the set of sets of worlds identified by countable sets of countable composite choices, i.e., $\Omega_\mathcal{P} = \{\omega_K \mid K \text{ is a countable set of countable composite choices}\}$.

Lemma 3 *(σ-algebra of a program, Lemma 2 of [47]). $\Omega_\mathcal{P}$ is a σ-algebra over $W_\mathcal{P}$.*

We can now define a function $\mu_\mathcal{P} : \Omega_\mathcal{P} \to [0,1]$. Given $K = \{\kappa_1, \kappa_2, \dots\}$, consider the sequence $\{K_n \mid n \geq 1\}$ where $K_n = \{\kappa_1, \dots, \kappa_n\}$. K_n is an increasing sequence and so $\lim_{n\to\infty} K_n$ exists and is equal to $\bigcup_{n=1}^{\infty} K_n = K$ [16]. Consider the sequence $\{K'_n \mid n \geq 1\}$ constructed as follows: $K'_1 = \{\kappa_1\}$, and K'_n is obtained by the union of K'_{n-1} with the splitting of each element of K'_{n-1} with κ_n. It is possible to prove by induction that K'_n is pairwise incompatible and equivalent to K_n.

Since $\rho_{\mathcal{P}}(\kappa) = 0$ for infinite composite choices, we can compute $\mu_c(K'_n)$ for each K'_n. Considering $\lim_{n\to\infty} \mu_c(K'_n)$, we have the following lemma.

Lemma 4 *(Existence of the limit of the measure of countable union of countable composite choices, Lemma 3 from [47]).* $\lim_{n\to\infty} \mu_c(K'_n)$ *exists.*

We can now introduce the definition of the probability space of a program.

Theorem 7 *(Probability space of a program, Theorem 8 from [47]). Given a set of composite choices $K = \{\kappa_1, \kappa_2, \dots\}$ and a pairwise incompatible set of composite choices K'_n equivalent to $\{\kappa_1, \dots, \kappa_n\}$, the triple $\langle W_{\mathcal{P}}, \Omega_{\mathcal{P}}, \mu_{\mathcal{P}}\rangle$ with*

$$\mu_{\mathcal{P}}(\omega_K) = \lim_{n\to\infty} \mu_c(K'_n)$$

is a probability space.

As already reported in Section 3.2.2, given a probabilistic logic program $\mathcal{P}$, a ground atom q and a composite choice κ, we say that κ is an *explanation* of q if $\forall w \in \omega_\kappa : w \vDash q$. We say that a set of composite choices K is *covering* for q if $\{w \mid w \in W_{\mathcal{P}} \wedge w \vDash q\} \subseteq \omega_K$.

If q has a countable set K of countable explanations that is covering with respect to q, Q represents a random variable, since $\{w \mid w \in W_{\mathcal{P}} \wedge w \vDash q\} = \omega_K \in \Omega_{\mathcal{P}}$. For brevity, we indicate $P(Q = 1)$ with $P(q)$ and we say that $P(q)$ is *well-defined* according to the distribution semantics. If the probability of all ground atoms in the grounding of a probabilistic logic program $\mathcal{P}$ is well-defined, then $\mathcal{P}$ is *well-defined*.

E.3. DISPONTE semantics

Given a DISPONTE *knowledge base* (KB) $\mathcal{O} = (\mathcal{A}, \mathcal{E})$, as for ProbLog, we obtain the set of worlds $W_{\mathcal{O}}$, where each world is built by taking the certain axioms and adding a subset of the probabilistic axioms in all possible ways. We can define the query random variable as for ProbLog without function symbols (Eq. (1) in Section 3.2.1), so the sample space is $W_{\mathcal{O}}$, the event space $\Omega_{\mathcal{O}}$ is the powerset of $W_{\mathcal{O}}$.

Define function $\rho_{\mathcal{O}} : W_{\mathcal{O}} \to \mathbb{R}$ as

$$\rho_{\mathcal{O}}(w) = \prod_{p::a\in\mathcal{E}:a\in w} p \prod_{p::a\in\mathcal{E}:a\notin w} (1-p)$$

and function $\mu_{\mathcal{O}} : \mathbb{P}(W_{\mathcal{O}}) \to \mathbb{R}$ as

$$\mu_{\mathcal{O}}(\omega) = \sum_{w\in\omega} \rho_{\mathcal{O}}(w)$$

Then $(W_{\mathcal{O}}, \mathbb{P}(W_{\mathcal{O}}), \mu_{\mathcal{O}})$ is a probability space and $\mu_{\mathcal{O}}$ is a probability measure.

Given an axiom q, define the function $Q : W_{\mathcal{O}} \to \{0, 1\}$ as in Eq. (1). Since the set of events is the powerset, then $Q^{-1}(\gamma) \in \mathbb{P}(W_{\mathcal{O}})$ for all $\gamma \subseteq \{0, 1\}$ so Q is a random variable. The distribution of Q is defined by $P(Q = 1)$ ($P(Q = 0)$ is given by $1 - P(Q = 1)$) and we indicate $P(Q = 1)$ with $P(q)$.

We can now compute $P(q)$ as

$$P(q) = \mu_{\mathcal{O}}(Q^{-1}(\{1\})) = \mu_{\mathcal{O}}(\{w | w \in W_{\mathcal{O}}, w \vDash q\}) = \sum_{w\in W_{\mathcal{O}}:w\vDash q} \rho_{\mathcal{O}}(w)$$

Appendix F. Ordinal numbers, mappings and fixpoints

We denote the set of *ordinal numbers* with Ω. Ordinal numbers extend the definition of natural numbers. The elements of Ω are called *ordinals* and are represented with lower case Greek letters. Ω is *well-ordered*, i.e., is a totally ordered set and every subset of it has a smallest element. The smallest element of Ω is 0. Given two ordinals α and β, we say that α is a *predecessor* of β, or equivalently β is a *successor* of α, if $\alpha < \beta$. If α is the largest ordinal smaller than β, α is termed *immediate predecessor*. The *immediate successor* of α is the smallest ordinal larger than α, denoted as $\alpha + 1$. Every ordinal has an immediate successor called *successor ordinal*. Ordinals that have predecessors but no immediate predecessor are called *limit ordinals*. So, ordinal numbers can be limit ordinals or successor ordinals.

The first elements of Ω are the naturals $0, 1, 2, \dots$ After all the natural numbers comes ω, the first *infinite ordinal*. Successors of ω are $\omega + 1$, $\omega + 2$ and so on. The generalization of the concept of sequence for ordinal number is the so-called *transfinite sequence*. The technique of induction for ordinal numbers is called *transfinite induction*: this states that, if a property $P(\alpha)$ is defined for all ordinals α, to prove that it is true for all ordinals we need to assume that $P(\beta)$ is true $\forall \beta < \alpha$ and then prove that $P(\alpha)$ is true. Transfinite induction proofs are usually structured in two steps: prove $P(\alpha)$ for α both successor and limit ordinal.

Consider a lattice A. A *mapping* is a function $f : A \to A$. It is monotonic if $f(x) \le f(y)$, $\forall x, y \in A,\ x \le y$. If $a \in A$ and $f(a) = a$, then a is a *fixpoint*. The *least fixpoint* is the smallest fixpoint. The *greatest fixpoint* can be defined analogously.

We define *increasing ordinal powers* of a monotonic mapping f as $f \uparrow 0 = \bot$, $f \uparrow (\alpha+1) = f(f(\alpha))$ if α is a successor ordinal and $f \uparrow \alpha = \text{lub}(\{f \uparrow \beta \mid \beta < \alpha\})$ if α is a limit ordinal. Similarly, *decreasing ordinal powers* are defined as $f \downarrow 0 = \top$, $f \downarrow \alpha = f(f(\alpha-1))$ if α is a successor ordinal and $f \downarrow \alpha = \text{glb}(\{f \downarrow \beta \mid \beta < \alpha\})$ if α is a limit ordinal. If A is a complete lattice and f a monotonic mapping, then the set of fixpoints of f in A is also a lattice (Knaster-Tarski theorem [26]). Moreover, f has a least fixpoint (lfp(A)) and a greatest fixpoint (gfp(A)). See [26] for a complete analysis of the topic.

Appendix G. Three-valued MKNF semantics [27]

The truth of an MKNF formula ψ is defined relatively to a *three-valued MKNF structure* $(I, \mathcal{M}, \mathcal{N})$, which consists of a first-order[1] interpretation I over a universe Δ and two pairs $\mathcal{M} = (M, M_1)$ and $\mathcal{N} = (N, N_1)$ of sets of first-order interpretations over Δ where $M_1 \subseteq M$ and $N_1 \subseteq N$. M and M_1 can be seen as the sets possible worlds where ψ is true or not false, respectively, for the purpose of evaluating the truth value of $\mathbf{K}\psi$. N and N_1 serve the same purpose for defining the truth value of $\mathbf{not}\,\psi$.

Satisfaction of a closed formula by a three-valued MKNF structure is defined as follows (where p is a predicate, ψ is a formula, the values `true`, `undefined` and `false` follow the order `false` < `undefined` < `true`, and ϵ^I represents the individual or relation in the domain of discourse assigned to ϵ by the interpretation I:

$(I,\mathcal{M},\mathcal{N})(p(t_1,\dots,t_n))$	`true` iff $(t_1^I,\dots,t_n^I) \in p^I$ `false` iff $(t_1^I,\dots,t_n^I) \notin p^I$
$(I,\mathcal{M},\mathcal{N})(\neg\psi)$	`true` iff $(I,\mathcal{M},\mathcal{N})(\psi) =$ `false`, `undefined` iff $(I,\mathcal{M},\mathcal{N})(\psi) =$ `undefined`, `false` iff $(I,\mathcal{M},\mathcal{N})(\psi) =$ `true`
$(I,\mathcal{M},\mathcal{N})(\psi_1 \wedge \psi_2)$	$min\{(I,\mathcal{M},\mathcal{N})(\psi_1),(I,\mathcal{M},\mathcal{N})(\psi_2)\}$
$(I,\mathcal{M},\mathcal{N})(\psi_1 \supset \psi_2)$	`true` iff $(I,\mathcal{M},\mathcal{N})(\psi_1) < (I,\mathcal{M},\mathcal{N})(\psi_2)$, `false` otherwise
$(I,\mathcal{M},\mathcal{N})(\exists x : \psi)$	$max\{(I,\mathcal{M},\mathcal{N})(\psi)[\alpha/x] \| \alpha \in \Delta\}$
$(I,\mathcal{M},\mathcal{N})(\mathbf{K}\psi)$	`true` iff $(J,(M,M_1),\mathcal{N})(\psi) =$ `true` for all $J \in M$, `false` iff $(J,(M,M_1),\mathcal{N})(\psi) =$ `false` for some $J \in M_1$, `undefined` otherwise
$(I,\mathcal{M},\mathcal{N})(\mathbf{not}\,\psi)$	`true` iff $(J,\mathcal{M},(N,N_1))(\psi) =$ `false` for some $J \in N_1$, `false` iff $(J,\mathcal{M},(N,N_1))(\psi) =$ `true` for all $J \in N$, `undefined` otherwise

An *MKNF interpretation* over a universe Δ is a non-empty set of first order interpretations over Δ. An *MKNF interpretation pair* (M, N) over a universe Δ consists of two MKNF interpretations M, N over Δ, with $\emptyset \subset N \subseteq M$. An MKNF interpretation pair (M, N) *satisfies* a closed MKNF formula ψ iff, for each $I \in M$, $(I,(M,N),(M,N))(\psi) =$ `true`. If there exists an MKNF interpretation pair satisfying ψ, then ψ is *consistent*. An MKNF interpretation pair (M, N) over a universe Δ is a *three-valued MKNF model* for a given closed MKNF formula ψ if

- (M, N) satisfies ψ and
- for each MKNF interpretation pair (M', N') over Δ with $M \subseteq M'$ and $N \subseteq N'$, where at least one of the inclusions is proper and $M' = N'$ if $M = N$, there is $I' \in M'$ such that $(I',(M',N'),(M,N))(\psi) =$ `false`. In other words, M and N cannot be extended while satisfying ψ; intuitively, the semantics implements minimal knowledge by requiring as many possible worlds as possible.

Appendix H. Details on iterated fixpoint semantics for HKB

Proposition 3 *(From [4]). Given an HKB^{FS} H and a 3-valued interpretation $\mathcal{I}$ for H, $OpTrue^H_{\mathcal{I}}$ and $OpFalse^H_{\mathcal{I}}$ are both monotonic in their argument.*

Proposition 4 *(From [4]). Given an HKB^{FS} H, $OpTrue^H_{\mathcal{I}}$ and $OpFalse^H_{\mathcal{I}}$ are monotonic in $\mathcal{I}$, i.e., if $\mathcal{I}$ and $\mathcal{I}'$ are three-valued interpretations for H such that $\mathcal{I} \le \mathcal{I}'$, then*

1. *for each $Tr \subseteq \mathsf{KA}(H)$, $OpTrue^H_{\mathcal{I}}(Tr) \subseteq OpTrue^H_{\mathcal{I}'}(Tr)$*
2. *for each $Fa \subseteq \mathsf{KA}(H)$, $OpFalse^H_{\mathcal{I}}(Fa) \subseteq OpFalse^H_{\mathcal{I}'}(Fa)$.*

Proposition 5 *(From [4]). For each HKB^{FS} H, IFP^H is monotonic w.r.t. the order relation among 3-valued interpretations defined in Definition 7.*

[1] A summary of First Order Logic is in Appendix C.

Table I.5
Iterations of the operators for *virus*(t), *mutation*(t,Y), and *spillover*($t,0$).

	virus(t)	mutation(t,Y)	spillover(t,0)
$\mathcal{F}^0$	$\emptyset,\emptyset$	$\emptyset,\emptyset$	$\emptyset,\emptyset$
$\mathcal{T}r^1$	$\{\emptyset\}$	$\{\{(e,1)\}\}$	$\{\{(f\{X/t,Y/0\},1)\}\}$
$\mathcal{T}r^2$	$\{\emptyset\}$	$\{\{(e,1)\}\}$	$\{\{(f\{X/t,Y/0\},1)\}\}$
$\mathcal{T}r^3$	$\{\emptyset\}$	$\{\{(e,1)\}\}$	$\{\{(f\{X/t,Y/0\},1)\}\}$
$\mathcal{T}r^4$	$\{\emptyset\}$	$\{\{(e,1)\}\}$	$\{\{(f\{X/t,Y/0\},1)\}\}$
$\mathcal{F}a^1$	$\{\emptyset\}$	$\{\{(e,0)\}\}$	$\{\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}a^2$	$\{\emptyset\}$	$\{\{(e,0)\}\}$	$\{\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}a^3$	$\{\emptyset\}$	$\{\{(e,0)\}\}$	$\{\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}a^4$	$\{\emptyset\}$	$\{\{(e,0)\}\}$	$\{\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}^1$	$\{\emptyset\},\emptyset$	$\{\{(e,1)\}\},\{\{(e,0)\}\}$	$\{\{(f\{X/t,Y/0\},1)\}\},\{\{(f\{X/t,Y/0\},0)\}\}$

Table I.6
Iterations of the operators for *spillover*($t,s(0)$), and *mutated*(t).

	spillover(t,s(0))	mutated(t)
$\mathcal{F}^0$	$\emptyset,\emptyset$	$\emptyset,\emptyset$
$\mathcal{T}r^1$	$\{\{(f\{X/t,Y/s(0)\},1)\}\}$	$\emptyset$
$\mathcal{T}r^2$	$\{\{(f\{X/t,Y/s(0)\},1)\}\}$	$\{\{(e,1)\}\}$
$\mathcal{T}r^3$	$\{\{(f\{X/t,Y/s(0)\},1)\}\}$	$\{\{(e,1)\}\}$
$\mathcal{T}r^4$	$\{\{(f\{X/t,Y/s(0)\},1)\}\}$	$\{\{(e,1)\}\}$
$\mathcal{F}a^1$	$\{\{(f\{X/t,Y/s(0)\},0)\}\}$	$\{\emptyset\}$
$\mathcal{F}a^2$	$\{\{(f\{X/t,Y/s(0)\},0)\}\}$	$\{\{(e,0)\}\}$
$\mathcal{F}a^3$	$\{\{(f\{X/t,Y/s(0)\},0)\}\}$	$\{\{(e,0)\}\}$
$\mathcal{F}a^4$	$\{\{(f\{X/t,Y/s(0)\},0)\}\}$	$\{\{(e,0)\}\}$
$\mathcal{F}^1$	$\{\{(f\{X/t,Y/s(0)\},1)\}\},\{\{(f\{X/t,Y/s(0)\},0)\}\}$	$\{\{(e,1)\}\},\{\{(e,0)\}\}$

Table I.7
Iterations of the operators for *spillover_count*($t,0$), and *spillover_count*($t,s(0)$).

	spillover_count(t,0)	spillover_count(t,s(0))
$\mathcal{F}^0$	$\emptyset,\emptyset$	$\emptyset,\emptyset$
$\mathcal{T}r^1$	$\emptyset$	$\emptyset$
$\mathcal{T}r^2$	$\{\emptyset\}$	$\emptyset$
$\mathcal{T}r^3$	$\{\emptyset\}$	$\{\{(e,1),(f\{X/t,Y/0\},1)\}\}$
$\mathcal{T}r^4$	$\{\emptyset\}$	$\{\{(e,1),(f\{X/t,Y/0\},1)\}\}$
$\mathcal{F}a^1$	$\{\emptyset\}$	$\{\emptyset\}$
$\mathcal{F}a^2$	$\{\emptyset\}$	$\{\emptyset\}$
$\mathcal{F}a^3$	$\{\emptyset\}$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}a^4$	$\{\emptyset\}$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}^1$	$\{\emptyset\},\emptyset$	$\{\{(e,1),(f\{X/t,Y/0\},1)\}\},\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\}\}$

Appendix I. $PIFP^{\mathcal{H}}$ operator iteration for Example 8

Let us consider the PHKB^{FS} of Example 7 and the queries $q_1 = safe(t)$ and $q_2 = spillover\_count(t, s(0))$.

Tables from I.5 to I.10 show the computation of the first iteration of the $PIFP^{\mathcal{H}}$ operator. In the tables, fact f is $spillover(X,Y)$ while axiom e is

$$t : \exists mutation\top.$$

Each column shows the sets of composite choices associated to an atom of $\mathsf{KA}(\mathcal{H})$ at each step of the inner and outer operators. In particular, for each atom a, the line labelled $\mathcal{F}^\alpha$ shows the sets of composite choices S_a and $S_{\sim a}$ such that $S_a = \omega_{\phi_a}$ and $S_{\sim a} = \omega_{\phi_{\sim a}}$ where $\mathsf{PIFP}^{\mathcal{H}} \uparrow \alpha = \{(a, \phi_a, \phi_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\}$; the lines labelled $\mathcal{T}r^\delta$ (resp. $\mathcal{F}a^\delta$) show the set of composite choices $S_a = \omega_{\theta_a}$ and $S_{\sim a} = \omega_{\theta_{\sim a}}$ such that $(a, \theta_a) \in POpTrue^{\mathcal{H}}_{\mathsf{PIFP}^{\mathcal{H}}\uparrow\alpha} \uparrow \delta$ (resp. $(a, \theta_a) \in POpFalse^{\mathcal{H}}_{\mathsf{PIFP}^{\mathcal{H}}\uparrow\alpha} \uparrow \delta$).

Table I.8
Iterations of the operators for *spillover_count*$(t, s(s(0)))$.

	spillover_count(t,s(s(0)))
$\boldsymbol{\mathscr{I}}^0$	$\emptyset, \emptyset$
$\mathcal{T}r^1$	$\emptyset$
$\mathcal{T}r^2$	$\emptyset$
$\mathcal{T}r^3$	$\emptyset$
$\mathcal{T}r^4$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\}$
$\mathcal{T}r^5$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\}$
$\mathcal{F}a^1$	$\{\emptyset\}$
$\mathcal{F}a^2$	$\{\emptyset\}$
$\mathcal{F}a^3$	$\{\emptyset\}$
$\mathcal{F}a^4$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}a^5$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$
$\mathcal{F}a^6$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$
$\boldsymbol{\mathscr{I}}^1$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\},$ $\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$

Table I.9
Iterations of the operators for *at_least_two_spillovers*(t).

	at_least_two_spillovers(t)
$\boldsymbol{\mathscr{I}}^0$	$\emptyset, \emptyset$
$\mathcal{T}r^1$	$\emptyset$
$\mathcal{T}r^2$	$\emptyset$
$\mathcal{T}r^3$	$\emptyset$
$\mathcal{T}r^4$	$\emptyset$
$\mathcal{T}r^5$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\}$
$\mathcal{T}r^6$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\}$
$\mathcal{F}a^1$	$\{\emptyset\}$
$\mathcal{F}a^2$	$\{\emptyset\}$
$\mathcal{F}a^3$	$\{\emptyset\}$
$\mathcal{F}a^4$	$\{\emptyset\}$
$\mathcal{F}a^5$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{F}a^6$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$
$\mathcal{F}a^7$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$
$\boldsymbol{\mathscr{I}}^1$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\},$ $\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$

Table I.10
Iterations of the operators for *safe*(t).

	safe(t)
$\boldsymbol{\mathscr{I}}^0$	$\emptyset, \emptyset$
$\mathcal{T}r^1$	$\emptyset$
$\mathcal{T}r^2$	$\emptyset$
$\mathcal{T}r^3$	$\emptyset$
$\mathcal{T}r^4$	$\emptyset$
$\mathcal{T}r^5$	$\emptyset$
$\mathcal{T}r^6$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\}\}$
$\mathcal{T}r^7$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$
$\mathcal{T}r^8$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\}$
$\mathcal{F}a^1$	$\{\emptyset\}$
$\mathcal{F}a^2$	$\{\emptyset\}$
$\mathcal{F}a^3$	$\{\emptyset\}$
$\mathcal{F}a^4$	$\{\emptyset\}$
$\mathcal{F}a^5$	$\{\emptyset\}$
$\mathcal{F}a^6$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\}$
$\mathcal{F}a^7$	$\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\}$
$\boldsymbol{\mathscr{I}}^1$	$\{\{(e,0)\},\{(f\{X/t,Y/0\},0)\},\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},0)\}\},$ $\{\{(e,1),(f\{X/t,Y/0\},1),(f\{X/t,Y/s(0)\},1)\}\}$

Appendix J. Proof for Section 5

Proposition 6 *(Monotonicity of $POpTrue^{\mathcal{H}}_{\mathcal{I}}$ and $POpFalse^{\mathcal{H}}_{\mathcal{I}}$). $POpTrue^{\mathcal{H}}_{\mathcal{I}}$ and $POpFalse^{\mathcal{H}}_{\mathcal{I}}$ are monotonic in their argument.*

Proof. Here we only consider $POpTrue^{\mathcal{H}}_{\mathcal{I}}$, since the proof for $POpFalse^{\mathcal{H}}_{\mathcal{I}}$ can be constructed in a similar way. We have to prove that if $\mathcal{T}r_1 \leq \mathcal{T}r_2$ then $POpTrue^{\mathcal{H}}_{\mathcal{I}}(\mathcal{T}r_1) \leq POpTrue^{\mathcal{H}}_{\mathcal{I}}(\mathcal{T}r_2)$. By definition, $\mathcal{T}r_1 \leq \mathcal{T}r_2$ means that

$$\forall (a, \phi_a) \in \mathcal{T}r_1, (a, \theta_a) \in \mathcal{T}r_2 : \phi_a \subseteq \theta_a.$$

Let (a, ϕ'_a) be the elements of $POpTrue^{\mathcal{H}}_{\mathcal{I}}(\mathcal{T}r_1)$ and (a, θ'_a) the elements of $POpTrue^{\mathcal{H}}_{\mathcal{I}}(\mathcal{T}r_2)$. To prove the monotonicity, we have to prove that $\phi'_a \subseteq \theta'_a$

If $a \in \mathcal{F}$ then $\phi'_a = \theta'_a = \omega_{\{\{(a,1)\}\}} \times W_{\mathcal{O}}$. If $a \in \mathsf{KA}(\mathcal{H}) \setminus \mathcal{F}$, then ϕ'_a and θ'_a are given by expressions that have the same structure and are monotonic in ϕ_b and θ_b, respectively. Since $\forall b \in \mathsf{KA}(\mathcal{H}) : \phi_b \subseteq \theta_b$, then $\phi'_a \subseteq \theta'_a$. $\square$

Proposition 7 *(Monotonicity of $PIFP^{\mathcal{H}}$). $PIFP^{\mathcal{H}}$ is monotonic.*

Proof. As above, we have to prove that, in the case that $\mathcal{I}_1 \leq \mathcal{I}_2$, then

$$PIFP^{\mathcal{H}}(\mathcal{I}_1) \leq PIFP^{\mathcal{H}}(\mathcal{I}_2).$$

By definition, $\mathcal{I}_1 \leq \mathcal{I}_2$ means that

$$\forall (a, \phi_a, \phi_{\sim a}) \in \mathcal{I}_1, (a, \theta_a, \theta_{\sim a}) \in \mathcal{I}_2 : \phi_a \subseteq \theta_a, \phi_{\sim a} \subseteq \theta_{\sim a}.$$

Let $(a, \phi'_a, \phi'_{\sim a})$ be the elements of $PIFP^{\mathcal{H}}(\mathcal{I}_1)$ and $(a, \theta'_a, \theta'_{\sim a})$ the elements of $PIFP^{\mathcal{H}}(\mathcal{I}_2)$. We have to prove that $\phi'_a \subseteq \theta'_a$ and $\phi'_{\sim a} \subseteq \theta'_{\sim a}$. This is a direct consequence of the monotonicity of $POpTrue^{\mathcal{H}}_{\mathcal{I}}$ and $POpFalse^{\mathcal{H}}_{\mathcal{I}}$ in $\mathcal{I}$, which can be proved as in Proposition 1. $\square$

Lemma 5 *(Model Equivalence). For a grounded PHKB^{FS} $\mathcal{H} = (\mathcal{R}, \mathcal{F}, \mathcal{A}, \mathcal{E})$, for every world w and iteration α, we have:*

$$IFP^w \uparrow \alpha = (PIFP^{\mathcal{H}} \uparrow \alpha)^w$$

Proof. We prove it by double transfinite induction. If α is a successor ordinal, assume that

$$IFP^w \uparrow \alpha - 1 = (PIFP^{\mathcal{H}} \uparrow \alpha - 1)^w$$

For any α, δ, let $\mathcal{I}^\alpha$ be $PIFP^{\mathcal{H}} \uparrow \alpha = \{(a, \phi^\alpha_a, \phi^\alpha_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\}$, $POpTrue^{\mathcal{H}}_{\mathcal{I}^{\alpha-1}} \uparrow \delta = \{(a, \theta^\delta_a) | a \in \mathsf{KA}(\mathcal{H})\}$, $POpFalse^{\mathcal{H}}_{\mathcal{I}^{\alpha-1}} \downarrow \delta = \{(a, \theta^\delta_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\}$ and $\mathcal{I}^\alpha_w = (I^\alpha_T, I^\alpha_F)$ be $IFP^w \uparrow \alpha$

Let us perform transfinite induction on the iterations of $OpTrue^w_{\mathcal{I}^{\alpha-1}_w}$ and $OpFalse^w_{\mathcal{I}^{\alpha-1}_w}$. Consider a successor ordinal δ and assume that

$$\begin{aligned} OpTrue^w_{\mathcal{I}^{\alpha-1}_w} \uparrow (\delta - 1) =& (POpTrue^{\mathcal{H}}_{\mathcal{I}^{\alpha-1}} \uparrow (\delta - 1))^w \\ OpFalse^w_{\mathcal{I}^{\alpha-1}_w} \downarrow (\delta - 1) =& (POpFalse^{\mathcal{H}}_{\mathcal{I}^{\alpha-1}} \downarrow (\delta - 1))^w \end{aligned}$$

We now prove that

$$\begin{aligned} OpTrue^w_{\mathcal{I}^{\alpha-1}_w} \uparrow \delta =& (POpTrue^{\mathcal{H}}_{\mathcal{I}^{\alpha-1}} \uparrow \delta)^w \\ OpFalse^w_{\mathcal{I}^{\alpha-1}_w} \downarrow \delta =& (POpFalse^{\mathcal{H}}_{\mathcal{I}^{\alpha-1}} \uparrow \delta)^w \end{aligned}$$

Pick an atom $a \in \mathsf{KA}(\mathcal{H})$ such that $w \in \theta^\delta_a$. If $a \in \mathcal{F}$, $w \in \theta^\delta_a$ means that a is a fact in w and $a \in OpTrue^w_{\mathcal{I}^{\alpha-1}_w} \uparrow \delta$.

If $a \notin \mathcal{F}$ and $w \in \theta^\delta_a$ where

$$\begin{aligned} \theta^\delta_a =& \Big(\bigcup_{a \leftarrow b_1, \ldots, b_m, \sim c_1, \ldots \sim c_n \in \mathcal{R}} \bigcap_{i=1,\ldots,m} (\phi^{\alpha-1}_{b_i} \cup \theta^{\delta-1}_{b_i}) \bigcap_{i=1,\ldots,n} \phi^{\alpha-1}_{\sim c_i} \Big) \\ & \bigcup \Big(\bigcup_{\substack{G \subseteq \mathsf{KA}(\mathcal{H}) \\ E \subseteq \mathcal{E}, \\ \mathsf{OB}_{\mathcal{A} \cup E, G} \models a}} \bigcap_{g \in G} (\phi^{\alpha-1}_g \cup \theta^{\delta-1}_g) \bigcap (W_{\mathcal{P}} \times \omega_{\{\{(e,1) \mid e \in E\}\}}) \Big) \end{aligned}$$

This means that either

1. there is a rule $a \leftarrow b_1, \ldots, b_m, \sim c_1, \ldots, \sim c_n \in \mathcal{R}$ such that $w \in \theta^{\delta-1}_{b_i} \cup \phi^{\alpha-1}_{b_i}$ for $i = 1 \ldots m$ and $w \in \phi^{\alpha-1}_{\sim c_j}$ for $j = 1 \ldots n$; or

2. there is a subset G of $\mathsf{KA}(\mathcal{H})$ and a subset E of $\mathcal{E}$ such that $\mathsf{OB}_{\mathcal{A}\cup E,G} \models a$, $\forall g \in G : w \in \phi_g^{\alpha-1} \vee w \in \theta_g^{\delta-1}$ and $\forall e \in E : e \in \mathcal{E}_w$

In case 1, by the inductive assumption, then either each $b_i \in OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow (\delta-1)$ or $b_i \in I_T^{\alpha-1}$ and each $c_j \in I_F^{\alpha-1}$ so $a \in OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow \delta$.

In case 2, there exist sets G and E such that it holds that for $\forall g \in G : g \in I_T^{\alpha-1} \cup OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow (\delta-1)$ for the inductive hypothesis, and $E \subset \mathcal{A}_w$ because $w \in \theta_a^\delta$, so $\mathsf{OB}_{w,I_T^{\alpha-1}\cup OpTrue^w_{\mathcal{I}_w^{\alpha-1}}\uparrow(\delta-1)} \models a$ and $a \in OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow \delta$.

In the other direction, consider $a \in OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow \delta$. If $a \in \mathcal{F}$, then a must be a fact in w because no rule of $\mathcal{R}$ has a in the head so $w \in \omega_{\{\{(a,1)\}\}} \times W_\mathcal{O}$ and $w \in \theta_a^\delta$.

If $a \notin \mathcal{F}$ then either

1. there is a rule $a \leftarrow b_1, \dots, b_m, \sim c_1, \dots, \sim c_n \in \mathcal{R}$ such that, for $i = 1 \dots m$, $b_i \in OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow (\delta-1)$ or $\mathcal{I}_w^{\alpha-1} \models a$ and, for $j = 1 \dots n$, $\mathcal{I}_w^{\alpha-1} \models \sim c_j$ or
2. $\mathsf{OB}_{\mathcal{A},I_T^{\alpha-1}\cup OpTrue^w_{\mathcal{I}_w^{\alpha-1}}\uparrow(\delta-1)} \models a$

In case 1, by the inductive assumption, $w \in \theta_{b_i}^{\delta-1} \cup \phi_{b_i}^{\alpha-1}$ for $i = 1 \dots m$ and $w \in \phi_{\sim c_j}^{\alpha-1}$ for $j = 1 \dots n$, so $w \in \theta_a^\delta$

In case 2, pick $G = I_T^{\alpha-1} \cup OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow (\delta-1)$ and $E = \mathcal{E}_w$. Then it holds that $\mathsf{OB}_{\mathcal{A}\cup E,G} \models a$, so $w \in \theta_a^\delta$.

Now suppose $w \in \theta_{\sim a}^\delta$. If $a \in \mathcal{F}$, $w \in \theta_{\sim a}^\delta$ means that a is not a fact in w and no rule has a in the head so $a \in OpFalse^w_{\mathcal{I}_w^{\alpha-1}} \downarrow \delta$.

If $a \notin \mathcal{F}$ then

$$\theta_{\sim a}^\delta = \Big(\bigcap_{a\leftarrow b_1,\dots,b_m,\sim c_1,\dots\sim c_n\in\mathcal{R}} \bigcup_{i=1,\dots,m} (\phi_{\sim b_i}^{\alpha-1} \cup \theta_{\sim b_i}^{\delta-1}) \bigcup_{i=1,\dots,n} \phi_{c_i}^{\alpha-1}$$
$$\bigcup_{\substack{G\subseteq \mathsf{KA}(\mathcal{H})\\ E\subseteq\mathcal{E},\\ \mathsf{OB}_{\mathcal{A}\cup E,G}\models\neg a}} \bigcap_{g\in G} \phi_g^{\alpha-1} \bigcap (W_\mathcal{P} \times \omega_{\{\{(e,1)\ \mid\ e\in E\}\}}) \Big)$$
$$\bigcap \bigcap_{\substack{G\subseteq \mathsf{KA}(\mathcal{H})\\ E\subseteq\mathcal{E},\\ \mathsf{OB}_{\mathcal{A}\cup E,G}\models a}} \Big(\bigcup_{g\in G} (\phi_{\sim g}^{\alpha-1} \cup \theta_{\sim g}^{\delta-1}) \bigcup_{e\in E} (W_\mathcal{P} \times \omega_{\{\{(e,0)\}\}}) \Big)$$

This means that

1. either
   (a) for all rules $a \leftarrow b_1, \dots, b_m, \sim c_1, \dots, \sim c_n \in \mathcal{R}$ either there exists an i such that $w \in \phi_{\sim b_i}^{\alpha-1} \cup \theta_{\sim b_i}^{\delta-1}$ or a j such that $w \in \phi_{c_j}^{\alpha-1}$,
   (b) or there is a subset G of $\mathsf{KA}(\mathcal{H})$ and a subset E of $\mathcal{E}$ such that $\mathsf{OB}_{\mathcal{A}\cup E,G} \models \neg a$, $\forall g \in G : w \in \phi_g^{\alpha-1}$ and $\forall e \in E : e \in \mathcal{E}_w$,
2. and for all subsets G of $\mathsf{KA}(\mathcal{H})$ and subsets E of $\mathcal{E}$ such that $\mathsf{OB}_{\mathcal{A}\cup E,G} \models a$, $\exists g \in G : w \in \phi_{\sim g}^{\alpha-1} \vee w \in \theta_{\sim g}^{\delta-1}$ or $\exists e \in E : e \notin \mathcal{E}_w$

In case 1a, by the inductive assumption, for all rules there exists a b_i such that $b_i \in OpFalse^w_{\mathcal{I}_w^{\alpha-1}} \downarrow (\delta-1)$ or $b_i \in I_T^{\alpha-1}$ or there exists a c_j such that $c_j \in I_F^{\alpha-1}$.

In case 1b, for the sets G and E that satisfy the condition, it holds that for $\forall g \in G : g \in OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow (\delta-1) \cup I_T^{\alpha-1}$ for the inductive hypothesis, $E \subset \mathcal{E}_w$ and $\mathsf{OB}_{\mathcal{A}\cup E,I_T^{\alpha-1}} \models \neg a$.

In case 2, for all sets G and E such that $\mathsf{OB}_{\mathcal{A}\cup E,G} \models a$, it holds that for $\exists g \in G : g \in OpTrue^w_{\mathcal{I}_w^{\alpha-1}} \uparrow (\delta-1) \cup I_T^{\alpha-1}$ for the inductive hypothesis or $\exists e \in E : e \notin \mathcal{E}_w$ so $\mathsf{OB}_{\mathcal{A}\cup\mathcal{E}_w,\mathsf{KA}(\mathcal{H})\setminus(OpFalse^w_{\mathcal{I}_w^{\alpha-1}}\downarrow(\delta-1)\cup I_F^{\alpha-1})} \not\models a$.

Conjoining conditions 1a and 2 it holds that $a \in OpFalse^w_{\mathcal{I}_w^{\alpha-1}} \downarrow \delta$. Similarly, conjoining conditions 1b and 2 it holds that $a \in OpFalse^w_{\mathcal{I}_w^{\alpha-1}} \downarrow \delta$.

In the other direction, consider $a \in OpFalse^w_{\mathcal{I}_w^{\alpha-1}} \downarrow \delta$. If $a \in \mathcal{F}$, then a must not be a fact in w and $w \in \omega_{\{\{(a,0)\}\}} \times W_\mathcal{O}$ so $w \in \theta_a^\delta$.

If $a \notin \mathcal{F}$ then

1. either
   (a) for all rules $a \leftarrow b_1, \dots, b_m, \sim c_1, \dots, \sim c_n \in \mathcal{R}$ there exists an i such that $b_i \in OpFalse^w_{\mathcal{I}_w^{\alpha-1}} \downarrow (\delta-1)$ or $b_i \in I_T^{\alpha-1}$ or there exists a j such tat $c_j \in I_F^{\alpha-1}$.
   (b) or $\mathsf{OB}_{\mathcal{A}\cup\mathcal{E}_w,I_T^{\alpha-1}} \models \neg a$,
2. and $\mathsf{OB}_{\mathcal{A}\cup\mathcal{E}_w,\mathsf{KA}(\mathcal{H})\setminus(OpFalse^w_{\mathcal{I}_w^{\alpha-1}}\downarrow(\delta-1)\cup I_F^{\alpha-1})} \not\models a$

In case 1a, by the inductive assumption, for all rules there exists an i such that $w \in \phi^{\alpha-1}_{\sim b_i} \cup \theta^{\delta-1}_{\sim b_i}$ or there exists a j such that $w \in \phi^{\alpha-1}_{c_j}$.

In case 1b, pick $G = I^{\alpha-1}_T$ and $E = \mathcal{E}_w$. Then it holds that $\mathsf{OB}_{\mathcal{A}\cup E,G} \models \neg a$

In case 2, consider $G = \mathsf{KA}(\mathcal{H}) \setminus (OpFalse^w_{\mathcal{I}^{\alpha-1}_w} \downarrow (\delta-1) \cup I^{\alpha-1}_F)$ and $E = \mathcal{E}_w$. Then it holds that $\mathsf{OB}_{\mathcal{A}\cup E,G} \not\models a$. Since description logics are monotonic, this means that $\mathsf{OB}_{\mathcal{A}\cup E',G'} \not\models a$ holds also for all subsets G' of G and E' of E. Let's consider the supersets G'' of G and E'' of E. For each pair (G'', E'') such that $\mathsf{OB}_{\mathcal{A}\cup E'',G''} \models a$, either $\exists g \in G'' : w \in \phi^{\alpha-1}_{\sim g} \vee w \in \theta^{\delta-1}_{\sim g}$ or $\exists e \in E'' : e \notin \mathcal{E}_w$.

Joining conditions 1a and 2 it holds that $w \in \theta^{\delta}_{\sim a}$. Similarly, joining conditions 1b and 2 it holds that $w \in \theta^{\delta}_{\sim a}$.

Consider now δ a limit ordinal, so $\theta^{\delta}_a = \bigcup_{\mu<\delta} \theta^{\mu}_a$ and $\theta^{\delta}_{\sim a} = \bigcap_{\mu<\delta} \theta^{\mu}_{\sim a}$.

$a \in OpTrue^w_{\mathcal{I}^{\alpha-1}_w} \uparrow \delta$, iff there exists a $\mu < \delta$ such that

$$a \in OpTrue^w_{\mathcal{I}^{\alpha-1}_w} \uparrow \mu.$$

For the inductive hypothesis, $w \in \theta^{\delta}_a$.

$\sim a \in OpFalse^w_{\mathcal{I}^{\alpha-1}_w} \downarrow \delta$, iff, for all $\mu < \delta$,

$$\sim a \in OpFalse^w_{\mathcal{I}^{\alpha-1}_w} \downarrow \mu.$$

For the inductive hypothesis, $w \in \theta^{\delta}_{\sim a}$.

Since

$$\begin{aligned} OpTrue^w_{\mathcal{I}^{\alpha-1}_w} &= (POpTrue^{\mathcal{H}}_{\mathcal{J}^{\alpha-1}} \uparrow \delta)^w \\ OpFalse^w_{\mathcal{I}^{\alpha-1}_w} &= (POpFalse^{\mathcal{H}}_{\mathcal{J}^{\alpha-1}} \downarrow \delta)^w \end{aligned}$$

holds for any δ, then

$$\begin{aligned} \mathsf{lfp}(OpTrue^w_{\mathcal{I}^{\alpha-1}_w}) &= (\mathsf{lfp}(POpTrue^{\mathcal{H}}_{\mathcal{J}^{\alpha-1}}))^w \\ \mathsf{gfp}(OpFalse^w_{\mathcal{I}^{\alpha-1}_w}) &= (\mathsf{gfp}(POpFalse^{\mathcal{H}}_{\mathcal{J}^{\alpha-1}}))^w \end{aligned}$$

and

$$IFP^w \uparrow \alpha = (PIFP^{\mathcal{H}} \uparrow \alpha)^w$$

Consider now α a limit ordinal. Then $\phi^{\alpha}_a = \bigcup_{\beta<\alpha} \phi^{\beta}_a$ and $\phi^{\alpha}_{\sim a} = \bigcup_{\beta<\alpha} \phi^{\beta}_{\sim a}$.

$a \in IFP^w \uparrow \alpha$, iff there exists a $\beta < \alpha$ such that

$$a \in IFP^w \uparrow \beta = (PIFP^{\mathcal{H}} \uparrow \beta)$$

For the inductive hypothesis, $w \in \phi^{\delta}_a$.

$\sim a \in IFP^w \uparrow \alpha$, iff, for all $\beta < \alpha$,

$$\sim a \in IFP^w \uparrow \beta.$$

For the inductive hypothesis, $w \in \phi^{\delta}_{\sim a}$. □

Lemma 6 *(Soundness and completeness of $PIFP^{\mathcal{H}}$). For a sound grounded PHKB^{FS} $\mathcal{H}$, let $PIFP^{\mathcal{H}} \uparrow \alpha = \{(a, \phi^{\alpha}_a, \phi^{\alpha}_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\}$ for all α. For every atom $a \in \mathsf{KA}(\mathcal{H})$ and world w there is an iteration α_0 such that for all $\alpha > \alpha_0$ we have:*

$$w \in \phi^{\alpha}_a \leftrightarrow \mathsf{WFM}(w) \models a \tag{J.1}$$

$$w \in \phi^{\alpha}_{\sim a} \leftrightarrow \mathsf{WFM}(w) \models \sim a \tag{J.2}$$

Proof. $\mathsf{WFM}(w) \models a$ means that there exists a α_0 such that $\forall \alpha : \alpha \geq \alpha_0 \rightarrow IFP^w \uparrow \alpha \models a$. For Lemma 1, this happens if and only if $w \in \phi^{\alpha}_a$. Similarly, $\mathsf{WFM}(w) \models \sim a$ implies that there exists a α_0 such that $\forall \alpha : \alpha \geq \alpha_0 \rightarrow IFP^w \uparrow \alpha \models \sim a$. As before, for Lemma 1, $w \in \phi^{\alpha}_{\sim a}$. □

Theorem 8 *(Well-definedness of the distribution semantics). For a sound grounded PHKB^{FS} $\mathcal{H}$, for every atom $a \in \mathsf{KA}(\mathcal{H})$, $\mu_{\mathcal{H}}(\{w \mid w \in W_{\mathcal{H}}, w \models a\})$ is well-defined.*

Proof. Let $PIFP^{\mathcal{H}} \uparrow \delta = \{(a, \phi^{\delta}_a, \phi^{\delta}_{\sim a}) | a \in \mathsf{KA}(\mathcal{H})\}$, where δ denotes the depth of the program. For Lemma 2, $\{w \mid w \in W_P, w \models a\} = \phi^{\delta}_a$.

Each iteration of $POpTrue^{\mathcal{H}}_{\mathcal{J}^{\alpha}}$ and $POpFalse^{\mathcal{H}}_{\mathcal{J}^{\alpha}}$ generates sets using a countable number of unions and intersection, since the set of rules is countable. So $\phi^{\delta}_a \in \Omega_{\mathcal{H}}$, $\{w \mid w \in W_{\mathcal{H}}, w \models a\}$ is measurable and $\mu_{\mathcal{H}}(\{w \mid w \in W_{\mathcal{H}}, w \models a\})$ is well-defined. □

Data availability

No data was used for the research described in the article.

References

[1] M. Alberti, A.S. Gomes, R. Gonçalves, J. Leite, M. Slota, Normative systems represented as hybrid knowledge bases, in: J. Leite, P. Torroni, T. Ågotnes, G. Boella, L. van der Torre (Eds.), Computational Logic in Multi-Agent Systems - 12th International Workshop, CLIMA XII, Barcelona, Spain, Proceedings, Springer, Berlin, 2011, pp. 330–346.

[2] M. Alberti, E. Lamma, F. Riguzzi, R. Zese, Probabilistic hybrid knowledge bases under the distribution semantics, in: G. Adorni, S. Cagnoni, M. Gori, M. Maratea (Eds.), AI*IA 2016: Advances in Artificial Intelligence, 21st Congress of the Italian Association for Artificial Intelligence, Pisa, Springer, Berlin, 2016, pp. 364–376.

[3] M. Alberti, E. Lamma, F. Riguzzi, R. Zese, Iterated fixpoint well-founded semantics for hybrid knowledge bases, in: Monica Dario Della, A. Murano, S. Rubin, L. Sauro (Eds.), Joint Proceedings of the 18th Italian Conference on Theoretical Computer Science and the 32nd Italian Conference on Computational Logic, Sun SITE Central Europe, Aachen, Germany, 2017, pp. 248–261.

[4] M. Alberti, R. Zese, F. Riguzzi, E. Lamma, An iterative fixpoint semantics for MKNF hybrid knowledge bases with function symbols, in: Y. Lierler, J.F. Morales, C. Dodaro, V. Dahl, M. Gebser, T. Tekle (Eds.), Proceedings 38th International Conference on Logic Programming, ICLP 2022 Technical Communications / Doctoral Consortium, Haifa, Israel, 31st July 2022 - 6th, August 2022, 2022, pp. 65–78.

[5] J.J. Alferes, M. Knorr, T. Swift, Query-driven procedures for hybrid MKNF knowledge bases, ACM Trans. Comput. Log. 14 (2013) 16:1–16:43, https://doi.org/10.1145/2480759.2480768.

[6] D. Azzolini, F. Riguzzi, E. Lamma, A semantics for hybrid probabilistic logic programs with function symbols, Artif. Intell. 294 (2021) 103452, https://doi.org/10.1016/j.artint.2021.103452.

[7] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, New York, NY, USA, 2003.

[8] G. Barthe, J.P. Katoen, A. Silva (Eds.), Foundations of Probabilistic Programming, Cambridge University Press, 2020.

[9] E. Bellodi, E. Lamma, F. Riguzzi, S. Albani, A distribution semantics for probabilistic ontologies, in: F. Bobillo, R. Carvalho, P.C.G. da Costa, C. d'Amato, N. Fanizzi, K.B. Laskey, K.J. Laskey, T. Lukasiewicz, T. Martin, M. Nickles, M. Pool (Eds.), 7th International Workshop on Uncertainty Reasoning for the Semantic Web, Sun SITE Central Europe, Aachen, Germany, 2011, pp. 75–86, http://ceur-ws.org/Vol-778/paper7.pdf.

[10] A. Borgida, On the relative expressiveness of description logics and predicate logics, Artif. Intell. 82 (1996) 353–367, https://doi.org/10.1016/0004-3702(96)00004-5.

[11] M. Bruynooghe, T. Mantadelis, A. Kimmig, B. Gutmann, J. Vennekens, G. Janssens, L. De Raedt, Problog technology for inference in a probabilistic first order logic, in: ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16–20, 2010, Proceedings, IOS Press, 2010, pp. 719–724.

[12] A. Calì, T. Lukasiewicz, L. Predoiu, H. Stuckenschmidt, Tightly coupled probabilistic description logic programs for the semantic web, J. Data Semant. XII (2009) 95–130.

[13] F. Calimeri, S. Cozza, G. Ianni, N. Leone, Computable functions in ASP: theory and implementation, in: M.G. de la Banda, E. Pontelli (Eds.), Logic Programming, 24th International Conference, ICLP 2008, Udine, Italy, December 9-13 2008, Proceedings, Springer, 2008, pp. 407–424.

[14] F. Calimeri, S. Cozza, G. Ianni, N. Leone, Finitely recursive programs: decidability and bottom-up computation, AI Commun. 24 (2011) 311–334, https://doi.org/10.3233/AIC-2011-0509.

[15] İ.İ. Ceylan, R. Peñaloza, Bayesian description logics, in: M. Bienvenu, M. Ortiz, R. Rosati, M. Simkus (Eds.), Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17–20, 2014, Sun SITE Central Europe, Aachen, 2014, pp. 447–458.

[16] Y.S. Chow, H. Teicher, Probability Theory: Independence, Interchangeability, Martingales. Springer Texts in Statistics, Springer, 2012.

[17] C. d'Amato, N. Fanizzi, T. Lukasiewicz, Tractable reasoning with Bayesian description logics, in: S. Greco, T. Lukasiewicz (Eds.), Scalable Uncertainty Management, Second International Conference, SUM 2008, Naples, Italy, October 1-3, 2008, Proceedings, Springer, Berlin, 2008, pp. 146–159.

[18] E. Dantsin, T. Eiter, G. Gottlob, A. Voronkov, Complexity and expressive power of logic programming, ACM Comput. Surv. 33 (2001) 374–425, https://doi.org/10.1145/502807.502810.

[19] B.A. Davey, H.A. Priestley, Introduction to Lattices and Order, 2 ed., Cambridge University Press, 2002.

[20] L. De Raedt, A. Kimmig, H. Toivonen, ProbLog: a probabilistic Prolog and its application in link discovery, in: M.M. Veloso (Ed.), 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), AAAI Press, 2007, pp. 2462–2467.

[21] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, H. Tompits, Combining answer set programming with description logics for the semantic web, Artif. Intell. 172 (2008) 1495–1539, https://doi.org/10.1016/j.artint.2008.04.002.

[22] N. Fuhr, Probabilistic datalog: implementing logical information retrieval for advanced applications, J. Am. Soc. Inf. Sci. 51 (2000) 95–110.

[23] R. Giugno, T. Lukasiewicz, P-SHOQ(D): a probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web, in: S. Flesca, S. Greco, N. Leone, G. Ianni (Eds.), Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, Proceedings, Springer, 2002, pp. 86–97.

[24] G. Gottlob, T. Lukasiewicz, G.I. Simari, Conjunctive query answering in probabilistic datalog+/- ontologies, in: S. Rudolph, C. Gutierrez (Eds.), 5th International Conference on Web Reasoning and Rule Systems (RR 2011), Galway, Ireland, Springer, Berlin, 2011, pp. 77–92.

[25] B.N. Grosof, I. Horrocks, R. Volz, S. Decker, Description logic programs: combining logic programs with description logic, in: G. Hencsey, B. White, Y.R. Chen, L. Kovács, S. Lawrence (Eds.), Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, 2003, ACM, New York, NY, USA, May 20-24, 2003, pp. 48–57.

[26] P. Hitzler, A. Seda, Mathematical Aspects of Logic Programming Semantics, Chapman & Hall/CRC Studies in Informatics Series, CRC Press, 2016.

[27] M. Knorr, J.J. Alferes, P. Hitzler, Local closed world reasoning with description logics under the well-founded semantics, Artif. Intell. 175 (2011) 1528–1554, https://doi.org/10.1016/j.artint.2011.01.007.

[28] H.J. Levesque, Foundations of a functional approach to knowledge representation, Artif. Intell. 23 (1984) 155–212, https://doi.org/10.1016/0004-3702(84)90009-2.

[29] V. Lifschitz, Nonmonotonic databases and epistemic queries, in: J. Mylopoulos, R. Reiter (Eds.), 12th International Joint Conference on Artificial Intelligence (IJCAI 1991), Morgan Kaufmann, San Francisco, CA, USA, 1991, pp. 381–386.

[30] F. Liu, J.H. You, Three-valued semantics for hybrid mknf knowledge bases revisited, Artif. Intell. 252 (2017) 123–138, https://doi.org/10.1016/j.artint.2017.08.003.

[31] T. Lukasiewicz, Probabilistic default reasoning with conditional constraints, Ann. Math. Artif. Intell. 34 (2002) 35–88.

[32] T. Lukasiewicz, Probabilistic description logic programs, Int. J. Approx. Reason. 45 (2007) 288–307, https://doi.org/10.1016/j.ijar.2006.06.012.

[33] T. Lukasiewicz, Expressive probabilistic description logics, Artif. Intell. 172 (2008) 852–883.

[34] T. Lukasiewicz, A novel combination of answer set programming with description logics for the semantic web, IEEE Trans. Knowl. Data Eng. 22 (2010) 1577–1592, https://doi.org/10.1109/TKDE.2010.111.

[35] T. Lukasiewicz, L. Predoiu, H. Stuckenschmidt, Tightly integrated probabilistic description logic programs for representing ontology mappings, Ann. Math. Artif. Intell. 63 (2011) 385–425, https://doi.org/10.1007/s10472-012-9280-3.

[36] B. Motik, R. Rosati, Reconciling description logics and rules, J. ACM 57 (2010) 30:1–30:62, https://doi.org/10.1145/1754399.1754403.

[37] N.J. Nilsson, Probabilistic logic, Artif. Intell. 28 (1986) 71–87.

[38] H. Nottelmann, N. Fuhr, Adding probabilities and rules to owl lite subsets based on probabilistic datalog, Int. J. Uncertain. Fuzziness 14 (2006) 17–42, https://doi.org/10.1142/S0218488506003819.

[39] D. Poole, Probabilistic Horn abduction and Bayesian networks, Artif. Intell. 64 (1993) 81–129.

[40] D. Poole, The independent choice logic for modelling multiple agents under uncertainty, Artif. Intell. 94 (1997) 7–56.

[41] D. Poole, Abducing through negation as failure: stable models within the independent choice logic, J. Log. Program. 44 (2000) 5–35, https://doi.org/10.1016/S0743-1066(99)00071-0.

[42] L. Predoiu, H. Stuckenschmidt, A probabilistic framework for information integration and retrieval on the semantic web, 2007.

[43] T.C. Przymusinski, Every logic program has a natural stratification and an iterated least fixed point model, in: Proceedings of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-1989), ACM Press, 1989, pp. 11–21.

[44] L.D. Raedt, K. Kersting, S. Natarajan, D. Poole, Statistical Relational Artificial Intelligence: Logic, Probability, and Computation, Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 10, 2016, pp. 1–189.

[45] R. Reiter, On asking what a database knows, in: J.W. Lloyd (Ed.), Computational Logic: Symposium Proceedings, Brussels, November 13/14, 1990, Springer, Berlin Heidelberg, Berlin, Heidelberg, 1990, pp. 96–113.

[46] F. Riguzzi, The distribution semantics for normal programs with function symbols, Int. J. Approx. Reason. 77 (2016) 1–19, https://doi.org/10.1016/j.ijar.2016.05.005.

[47] F. Riguzzi, Foundations of Probabilistic Logic Programming Languages, Semantics, Inference and Learning, second edition, River Publishers, 2022.

[48] F. Riguzzi, E. Bellodi, E. Lamma, R. Zese, Probabilistic description logics under the distribution semantics, Semant. Web 6 (2015) 447–501, https://doi.org/10.3233/SW-140154.

[49] F. Riguzzi, T. Swift, The PITA system: tabling and answer subsumption for reasoning under uncertainty, Theory Pract. Log. Program. 11 (2011) 433–449, https://doi.org/10.1017/S147106841100010X.

[50] F. Riguzzi, T. Swift, Terminating evaluation of logic programs with finite three-valued models, ACM Trans. Comput. Log. 15 (2014) 32:1–32:38, https://doi.org/10.1145/2629337.

[51] T. Sato, A statistical learning method for logic programs with distribution semantics, in: L. Sterling (Ed.), Logic Programming, Proceedings of the Twelfth International Conference on Logic Programming, Tokyo, Japan, June 13–16, 1995, MIT Press, 1995, pp. 715–729.

[52] U. Sattler, D. Calvanese, R. Molitor, Relationships with other formalisms, in: Description Logic Handbook, 2003, pp. 137–177.

[53] M.E. Stickel, A prolog technology theorem prover: implementation by an extended prolog compiler, J. Autom. Reason. 4 (1988) 353–380, https://doi.org/10.1007/BF00297245.

[54] A. Van Gelder, K.A. Ross, J.S. Schlipf, The well-founded semantics for general logic programs, J. ACM 38 (1991) 620–650.

[55] J. Vennekens, S. Verbaeten, M. Bruynooghe, Logic programs with annotated disjunctions, in: B. Demoen, V. Lifschitz (Eds.), 20th International Conference on Logic Programming (ICLP 2004), Springer, Berlin, Heidelberg, 2004, pp. 431–445.

[56] R. Zese, Probabilistic Semantic Web: Reasoning and Learning, Studies on the Semantic Web, vol. 28, IOS Press, Amsterdam, 2017.

[57] R. Zese, E. Bellodi, E. Lamma, F. Riguzzi, F. Aguiari, Semantics and inference for probabilistic description logics, in: F. Bobillo, R.N. Carvalho, P.C. Costa, C. d'Amato, N. Fanizzi, K.B. Laskey, K.J. Laskey, T. Lukasiewicz, M. Nickles, M. Pool (Eds.), Uncertainty Reasoning for the Semantic Web III, in: Lecture Notes in Computer Science, vol. 8816, Springer International Publishing, 2014, pp. 79–99.

[58] R. Zese, E. Bellodi, F. Riguzzi, G. Cota, E. Lamma, Tableau reasoning for description logics and its extension to probabilities, Ann. Math. Artif. Intell. 82 (2018) 101–130, https://doi.org/10.1007/s10472-016-9529-3.