Many-Valued First-Order Logics with Probabilistic Semantics

Thomas Lukasiewicz

Institut für Informatik, Universität Gießen, Arndtstraße 2
D-35392 Gießen, Germany
lukasiewicz@informatik.uni-giessen.de

Abstract. We present n-valued first-order logics with a purely probabilistic semantics. We then introduce a new probabilistic semantics of n-valued first-order logics that lies between the purely probabilistic semantics and the truth-functional semantics of the n-valued Łukasiewicz logics $Ł_n$. Within this semantics, closed formulas of classical first-order logics that are logically equivalent in the classical sense also have the same truth value under all n-valued interpretations. Moreover, this semantics is shown to have interesting computational properties. More precisely, n-valued logical consequence in disjunctive logic programs with n-valued disjunctive facts can be reduced to classical logical consequence in $n-1$ layers of classical disjunctive logic programs. Moreover, we show that n-valued logic programs have a model and a fixpoint semantics that are very similar to those of classical logic programs. Finally, we show that some important deduction problems in n-valued logic programs have the same computational complexity like their classical counterparts.

1 Introduction

One way to give semantics to n-valued first-order logics is to use a truth-functional approach and to define the semantics inductively as it is well-known from classical first-order logics. Another way is to use a probabilistic approach as illustrated by the following example.

Let us assume that we have two experts in a certain field and that these two experts fixed a common first-order language to express their knowledge. Moreover, let us assume that the two experts have a common domain over which their first-order language is interpreted, that they also have a common interpretation of all function symbols, but that each of them has its own interpretation of the predicate symbols (that is, each of the two experts stands for a classical first-order interpretation). Finally, let us assume that it is our job to unify the knowledge of the two experts (that is, to define an interpretation that combines the two classical first-order interpretations). Now, there might be closed formulas on which the two experts agree and those on which they disagree: given a closed formula α, it may be the case that they both think that α is **true**, that they both think that α is **false**, or that one of them thinks that α is **true** and the other one that α is **false**. If we assume that both experts are equally credible, then we could model this situation by assigning α the truth values 1, 0, or $\frac{1}{2}$, respectively. That is, the truth value of a closed formula α is equal to the relative number of experts who think that α is **true**.

Note that we can similarly merge two finite sets of closed formulas over a common classical first-order language. More precisely, if α is contained in both sets, then the truth value of α is 1. If α is contained in one of the sets, then the truth value of α is greater than or equal to $\frac{1}{2}$. Finally, if α is not contained in any of the sets, then the truth value of α is greater than or equal to 0. A more general view on merging knowledge-bases is given, for example, in [33].

Note also that another way of combining classical first-order interpretations of a finite number of experts is given in [13]. In detail, each formula α of a modal logic is assigned as a truth value the set of all experts who think that α is **true**. That is, the work in [13] keeps track of each single expert, while our approach abstracts to relative numbers of experts (since we are interested in the totally ordered set of truth values $\{\frac{0}{k}, \frac{1}{k}, \ldots, \frac{k}{k}\}$ with $k \geq 2$).

Analyzing our *2-experts world* more deeply, we realize that the truth value assignment to formulas respects the laws of probability. That is, the truth value of a closed formula is in $[0, 1]$, the truth value of all tautologies is 1, the truth value of the disjunction $\alpha \vee \beta$ of mutually exclusive closed formulas α and β is the sum of the truth values of α and β, and logically equivalent closed formulas have the same truth value. But, instead of allowing all real numbers in $[0, 1]$ as truth values, we just consider the members of $\{0, \frac{1}{2}, 1\}$ as truth values.

We also realize that the 2-experts world cannot be expressed by any truth-functional 3-valued logic, since the truth value of the conjunction $\alpha \wedge \beta$ of two closed formulas α and β is generally not a function of the truth values of α and β! For example, if the two experts disagree on a closed formula α, then the truth value of both α and its negation $\neg\alpha$ is $\frac{1}{2}$. Thus, the conjunction $\alpha \wedge \alpha$ also gets the truth value $\frac{1}{2}$, whereas the contradiction $\alpha \wedge \neg\alpha$ always gets the truth value 0.

Finally, we observe that our 2-experts world can easily be generalized to a *k-experts world* with $k \geq 2$ and the $k+1$ truth values $\frac{0}{k}, \frac{1}{k}, \ldots, \frac{k}{k}$.

We are arrived at the topic of this paper, which is to explore the area between the purely probabilistic and the truth-functional semantics of n-valued first-order logics. Our aim is to bring together the advantages of two different approaches. While the probabilistic formalism provides a well-defined semantics, truth-functional logics are often more compelling from the computational side (see, for example, [22] and [23] for the subtleties of probabilistic propositional deduction).

In this paper, we now present a new probabilistic semantics of n-valued first-order logics that lies between the purely probabilistic semantics and the truth-functional semantics given by the n-valued Łukasiewicz logics Ł_n. More precisely, the main contributions of this paper can be summarized as follows:

- We present n-valued first-order logics with a purely probabilistic semantics.
- We introduce $\text{Ł}_n^{\star}$-interpretations, which give n-valued first-order formulas a new probabilistic semantics that lies between the purely probabilistic semantics and the truth-functional semantics of the n-valued Łukasiewicz logics Ł_n.
- The introduced $\text{Ł}_n^{\star}$-interpretations have the following nice property: closed formulas of classical first-order logics that are logically equivalent in the classical sense also have the same truth value under all $\text{Ł}_n^{\star}$-interpretations.

- We show that disjunctive logic programming with n-valued disjunctive facts in $Ł_n^\star$ is reducible to classical disjunctive logic programming in $n-1$ layers.
- We present a model and a fixpoint semantics for n-valued logic programming in $Ł_n^\star$, which are very similar to those of classical logic programming.
- We show that some important special cases of n-valued logic programming in $Ł_n^\star$ have the same computational complexity like their classical counterparts.

The rest of this paper is organized as follows. In Section 2, we present n-valued first-order logics with purely probabilistic semantics. In Section 3, we introduce $Ł_n^\star$-interpretations. Sections 4 and 5 discuss disjunctive logic programs with n-valued disjunctive facts and n-valued logic programs, respectively. In Section 6, we discuss important related work. In Section 7, we summarize the main results and give an outlook on future research.

2 Many-Valued First-Order Logics

Let Φ be a first-order vocabulary that contains a set of function symbols and a set of predicate symbols. Let $\mathcal{X}$ be a set of variables. The first part of the following definitions are the usual ones from classical first-order logics. We briefly introduce them here to fix a clear technical background.

We define *terms* by induction as follows. A term is a variable from $\mathcal{X}$ or an expression of the kind $f(t_1, \ldots, t_k)$, where f is a function symbol of arity $k \geq 0$ from Φ and $t_1, \ldots, t_k$ are terms. *Formulas* are defined by induction as follows. If p is a predicate symbol of arity $k \geq 0$ from Φ and $t_1, \ldots, t_k$ are terms, then $p(t_1, \ldots, t_k)$ is a formula (called *atomic formula*). If F and G are formulas, then $\neg F$, $(F \wedge G)$, $(F \vee G)$, and $(F \leftarrow G)$ are formulas. If F is a formula and x is a variable from $\mathcal{X}$, then $\forall x\, F$ and $\exists x\, F$ are formulas.

An *interpretation* $I = (D, \pi)$ consists of a nonempty set D, called *domain*, and a mapping π that assigns to each function symbol from Φ a function of right arity over D and to each predicate symbol from Φ a predicate of right arity over D. A *variable assignment* σ is a mapping that assigns to each variable from $\mathcal{X}$ an element from D. For a variable x from $\mathcal{X}$ and an element d from D, we write $\sigma[x/d]$ to denote the variable assignment that is identical to σ except that it assigns d to x. The variable assignment σ is by induction extended to terms by $\sigma(f(t_1, \ldots, t_k)) = \pi(f)(\sigma(t_1), \ldots, \sigma(t_k))$ for all terms $f(t_1, \ldots, t_k)$. The *truth* of formulas F in I under σ, denoted $I \models_\sigma F$, is inductively defined as follows:

- $I \models_\sigma p(t_1, \ldots, t_k)$ iff $(\sigma(t_1), \ldots, \sigma(t_k)) \in \pi(p)$.
- $I \models_\sigma \neg F$ iff not $I \models_\sigma F$, and $I \models_\sigma (F \wedge G)$ iff $I \models_\sigma F$ and $I \models_\sigma G$.
- $I \models_\sigma \forall x\, F$ iff $I \models_{\sigma[x/d]} F$ for all $d \in D$.
- The truth of the remaining formulas in I under σ is defined by expressing $\vee$, $\leftarrow$, and $\exists$ in terms of $\neg$, $\wedge$, and $\forall$ as usual.

A formula F *is true* in I, or I is a *model* of F, denoted $I \models F$, iff F is true in I under all variable assignments σ. I is a *model* of a set of formulas $\mathcal{F}$, denoted $I \models \mathcal{F}$, iff I is a model of all formulas in $\mathcal{F}$. $\mathcal{F}$ is *satisfiable* iff a model of $\mathcal{F}$ exists. A formula F is a *logical consequence* of a set of formulas $\mathcal{F}$, denoted $\mathcal{F} \models F$, iff each model of $\mathcal{F}$ is also a model of F.

We adopt the classical conventions to eliminate parentheses. Moreover, universally quantified formulas are defined as usual. Finally, the classical notions of ground terms, ground formulas, and ground instances of formulas are adopted.

2.1 Many-Valued First-Order Logics of Probabilities in $\mathbf{Pr}_n$

Next, we introduce a purely probabilistic approach to n-valued first-order logics. For this purpose, let $TV = \{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, 1\}$ with $n \geq 3$ denote the set of *truth values*. Note that the classical truth values **false** and **true** correspond to 0 and 1, respectively. We define the syntax and the probabilistic semantics of n-valued formulas as follows:

An *n-valued formula* is an expression of the kind $tv(F) \geq c$ with a classical formula F and a truth value c from TV (note that $tv(F) \geq c$ will mean that the truth value of F is greater than or equal to c, while $tv(F) \geq c$ together with $tv(\neg F) \geq 1 - c$ will express that the truth value of F is exactly c).

A *Pr_n-interpretation* $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$ consists of $n-1$ interpretations $(D, \pi_1), \ldots, (D, \pi_{n-1})$ over the same domain D such that $\pi_1(f) = \cdots = \pi_{n-1}(f)$ for all function symbols f from Φ. The *truth value* $\mathcal{I}_\sigma(F)$ of a formula F in the Pr_n-interpretation $\mathcal{I}$ under a variable assignment σ is defined by:

$$\mathcal{I}_\sigma(F) \;=\; |\{i \in [1\!:\!n-1] \,|\, (D, \pi_i) \models_\sigma F\}| \,/\, (n-1)\,.$$

An n-valued formula $tv(F) \geq c$ *is true* in $\mathcal{I}$ under σ iff $\mathcal{I}_\sigma(F) \geq c$. An n-valued formula N *is true* in $\mathcal{I}$, or $\mathcal{I}$ is a *model* of N, denoted $\mathcal{I} \models N$, iff N is true in $\mathcal{I}$ under all variable assignments σ. $\mathcal{I}$ is a *model* of a set of n-valued formulas $\mathcal{N}$, denoted $\mathcal{I} \models \mathcal{N}$, iff $\mathcal{I}$ is a model of all n-valued formulas in $\mathcal{N}$. $\mathcal{N}$ is *satisfiable* iff a model of $\mathcal{N}$ exists. N is a *logical consequence* of $\mathcal{N}$, denoted $\mathcal{N} \models N$, iff each model of $\mathcal{N}$ is also a model of N.

Finally, note that we canonically define ground n-valued formulas and ground instances of n-valued formulas.

2.2 Many-Valued First-Order Logics in the Łukasiewicz Logics $\mathbf{Ł}_n$

Among all the truth-functional n-valued logics, the Łukasiewicz logic $\mathrm{Ł}_n$ seems to be the one closest in spirit to our n-valued logic of probabilities. We now briefly describe how n-valued formulas are interpreted in $\mathrm{Ł}_n$. For this purpose, we just have to define $\mathrm{Ł}_n$-interpretations and the truth value of n-valued formulas in $\mathrm{Ł}_n$-interpretations under variable assignments. Note that our $\mathrm{Ł}_n$-interpretations are defined in a nonstandard way by adapting Pr_n-interpretations (of course, this approach is technically quite clumsy, but it will be useful to understand the relationship between Pr_n- and $\mathrm{Ł}_n$-interpretations).

An *$Ł_n$-interpretation* $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$ consists of $n-1$ interpretations $(D, \pi_1), \ldots, (D, \pi_{n-1})$ over the same domain D such that $\pi_1(f) = \cdots = \pi_{n-1}(f)$ for all function symbols f from Φ. The *truth value* $\mathcal{I}_\sigma(F)$ of a formula F in the $\mathrm{Ł}_n$-interpretation $\mathcal{I}$ under a variable assignment σ is inductively defined by:

- $\mathcal{I}_\sigma(p(t_1, \ldots, t_k)) \;=\; |\{i \in [1\!:\!n-1] \,|\, (D, \pi_i) \models_\sigma p(t_1, \ldots, t_k)\}| \,/\, (n-1)\,.$
- $\mathcal{I}_\sigma(\neg F) \;=\; 1 - \mathcal{I}_\sigma(F)$ and $\mathcal{I}_\sigma(F \wedge G) \;=\; \min(\mathcal{I}_\sigma(F), \mathcal{I}_\sigma(G))$.

- $\mathcal{I}_\sigma(F \vee G) = \max(\mathcal{I}_\sigma(F), \mathcal{I}_\sigma(G))$ and $\mathcal{I}_\sigma(F \leftarrow G) = \min(1, \mathcal{I}_\sigma(F) - \mathcal{I}_\sigma(G) + 1)$.
- $\mathcal{I}_\sigma(\forall x\, F) = \min\{\mathcal{I}_{\sigma[x/d]}(F) \mid d \in D\}$.
- $\mathcal{I}_\sigma(\exists x\, F) = \max\{\mathcal{I}_{\sigma[x/d]}(F) \mid d \in D\}$.

3 Many-Valued First-Order Logics of Probabilities in Ł$_n^\star$

In this section, we define a new probabilistic semantics of n-valued formulas that lies between Pr$_n$- and Ł$_n$-interpretations. In this new semantics, we keep the nice property of Pr$_n$-interpretations that closed formulas of the same classical truth value under all classical interpretations are also of the same n-valued truth value under all n-valued interpretations. In particular, contradictory and tautological closed formulas always have the truth values 0 and 1, respectively.

The new semantics is obtained by adapting Pr$_n$-interpretations to Ł$_n$-interpretations. More precisely, we give up some properties of Ł$_n$-interpretations (to keep the above-mentioned nice properties of Pr$_n$-interpretations) by being more cautious in using the truth tables of Ł$_n$. That is, not all logical combinations of formulas are truth-functionally interpreted. We just interpret logical combinations of ground atomic formulas by the truth tables of Ł$_n$. To realize this, we must impose a certain restriction on Pr$_n$-interpretations:

Theorem 1 *Let $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$ be a Pr$_n$-interpretation and let σ be a variable assignment. Let each $d \in D$ have a ground term t_d with $\sigma(t_d) = d$.*

It holds $\mathcal{I}_\sigma(A \wedge B) = \min(\mathcal{I}_\sigma(A), \mathcal{I}_\sigma(B))$ for all ground atomic formulas A and B iff there exists a permutation μ of the elements in $[1:n-1]$ with $\pi_{\mu(1)}(p) \supseteq \pi_{\mu(2)}(p) \supseteq \cdots \supseteq \pi_{\mu(n-1)}(p)$ for all predicate symbols p from Φ.

Proof. '$\Leftarrow$': Let A and B be ground atomic formulas. If μ is a permutation of the elements in $[1:n-1]$ with $\pi_{\mu(1)}(p) \supseteq \pi_{\mu(2)}(p) \supseteq \cdots \supseteq \pi_{\mu(n-1)}(p)$ for all predicate symbols p from Φ, then $(D, \pi_{\mu(i)}) \models_\sigma A$ and $(D, \pi_{\mu(i)}) \models_\sigma B$ for exactly all $1 \leq i \leq \mathcal{I}_\sigma(A) \cdot (n-1)$ and $1 \leq i \leq \mathcal{I}_\sigma(B) \cdot (n-1)$, respectively. Hence, we get $(D, \pi_{\mu(i)}) \models_\sigma A \wedge B$ for exactly all $1 \leq i \leq \min(\mathcal{I}_\sigma(A), \mathcal{I}_\sigma(B)) \cdot (n-1)$.

'$\Rightarrow$': For all ground atomic formulas A let $I_A = \{i \in [1:n-1] \mid (D, \pi_i) \models_\sigma A\}$. For all ground atomic formulas A and B: if $\mathcal{I}_\sigma(A \wedge B) = \min(\mathcal{I}_\sigma(A), \mathcal{I}_\sigma(B))$, then $\mathcal{I}_\sigma(A) \leq \mathcal{I}_\sigma(B)$ iff $I_A \subseteq I_B$. Hence, we can define the desired permutation μ by $\mu(i) = \min((\bigcap\{I_A \mid A \in GAF,\, \mathcal{I}_\sigma(A) \cdot (n-1) \geq i\}) \setminus \{\mu(1), \ldots, \mu(i-1)\})$ for all $i \in [1:n-1]$, where GAF denotes the set of all ground atomic formulas. Note that we canonically assume $\bigcap \emptyset = [1:n-1]$. $\Box$

3.1 Ł$_n^\star$-Interpretations

Inspired by Theorem 1, we can now introduce a natural probabilistic semantics of n-valued formulas that lies between the purely probabilistic semantics of Pr$_n$-interpretations and the truth-functional semantics of Ł$_n$-interpretations. We do this by defining Ł$_n^\star$-interpretations and the truth value of n-valued formulas in Ł$_n^\star$-interpretations under variable assignments.

An *Ł$_n^\star$-interpretation* $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$ consists of $n-1$ interpretations $(D, \pi_1), \ldots, (D, \pi_{n-1})$ over the same domain D such that $\pi_1(f) = \cdots = \pi_{n-1}(f)$

for all function symbols f from Φ and

$$\pi_1(p) \supseteq \pi_2(p) \supseteq \cdots \supseteq \pi_{n-1}(p) \text{ for all predicate symbols } p \text{ from } \Phi. \qquad (1)$$

The *truth value* $\mathcal{I}_\sigma(F)$ of a formula F in the $Ł_n^\star$-interpretation $\mathcal{I}$ under a variable assignment σ is defined by:

$$\mathcal{I}_\sigma(F) \;=\; |\{i \in [1{:}n-1] \,|\, (D, \pi_i) \models_\sigma F\}| \,/\, (n-1)\,.$$

3.2 Properties of $Ł_n^\star$-Interpretations

In this subsection, we explore some basic properties of $Ł_n^\star$-interpretations. We start with showing that (1) is propagated through the structure of all formulas that are built without the logical connectives $\neg$ and $\leftarrow$.

We say that the truth value of a formula F is *compressed* in the $Ł_n^\star$-interpretation $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$ under the variable assignment σ iff

$$\mathcal{I}_\sigma(F) \;=\; \max(\{i \in [1{:}n-1] \,|\, (D, \pi_i) \models_\sigma F\} \cup \{0\}) \,/\, (n-1)\,.$$

Lemma 2 *For all formulas F that are built without the logical connectives $\neg$ and $\leftarrow$, all $Ł_n^\star$-interpretations $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$, and all variable assignments σ, the truth value of F is compressed in $\mathcal{I}$ under σ.*

Proof. The claim is proved by induction on the structure of all formulas that are built without the logical connectives $\neg$ and $\leftarrow$. Let $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$ be an $Ł_n^\star$-interpretations and let σ be a variable assignment.

If F is an atomic formula $p(t_1, \ldots, t_k)$, then the truth value of F is compressed in $\mathcal{I}$ under σ by the definition of $Ł_n^\star$-interpretations. If F is of the form $G \wedge H$ or $G \vee H$, then the truth values of G and H are compressed in $\mathcal{I}$ under σ by the induction hypothesis. Hence, also the truth values of $G \wedge H$ and $G \vee H$ are compressed in $\mathcal{I}$ under σ. Moreover, we get $\mathcal{I}_\sigma(G \wedge H) = \min(\mathcal{I}_\sigma(G), \mathcal{I}_\sigma(H))$ and $\mathcal{I}_\sigma(G \vee H) = \max(\mathcal{I}_\sigma(G), \mathcal{I}_\sigma(H))$. If F is of the form $\forall x\, G$ or $\exists x\, G$, then for all $d \in D$ the truth value of G is compressed in $\mathcal{I}$ under $\sigma[x/d]$ by the induction hypothesis. Hence, also the truth values of $\forall x\, G$ and $\exists x\, G$ are compressed in $\mathcal{I}$ under σ. Furthermore, we get $\mathcal{I}_\sigma(\forall x\, G) = \min\{\mathcal{I}_{\sigma[x/d]}(G) \,|\, d \in D\}$ and $\mathcal{I}_\sigma(\exists x\, G) = \max\{\mathcal{I}_{\sigma[x/d]}(G) \,|\, d \in D\}$. $\Box$

Next, we focus on the relationship between $Ł_n^\star$- and $Ł_n$-interpretations. We show that $Ł_n^\star$- and $Ł_n$-interpretations coincide in the semantics they give to all the formulas that are built without the logical connectives $\neg$ and $\leftarrow$ and in the semantics they give to all logical combinations of these formulas.

Lemma 3 *For all formulas F and G that are built without the logical connectives $\neg$ and $\leftarrow$, all variables x in $\mathcal{X}$, all $Ł_n^\star$-interpretations $\mathcal{I} = (D, \pi_1, \ldots, \pi_{n-1})$, and all variable assignments σ:*

$$\mathcal{I}_\sigma(\neg F) = 1 - \mathcal{I}_\sigma(F) \qquad (2)$$

$$\mathcal{I}_\sigma(F \wedge G) = \min(\mathcal{I}_\sigma(F), \mathcal{I}_\sigma(G)) \qquad (3)$$

$$\mathcal{I}_\sigma(F \vee G) = \max(\mathcal{I}_\sigma(F), \mathcal{I}_\sigma(G)) \qquad (4)$$

$$\mathcal{I}_\sigma(F \leftarrow G) = \min(1, \mathcal{I}_\sigma(F) - \mathcal{I}_\sigma(G) + 1) \qquad (5)$$

$$\mathcal{I}_\sigma(\forall x\, F) = \min\{\mathcal{I}_{\sigma[x/d]}(F) \mid d \in D\} \tag{6}$$
$$\mathcal{I}_\sigma(\exists x\, F) = \max\{\mathcal{I}_{\sigma[x/d]}(F) \mid d \in D\}\,. \tag{7}$$

Proof. The claim (2) is immediate (for all formulas F) by the definition of the truth value of a formula. The claims (3) and (4) hold by the proof of Lemma 2. The claim (5) follows from $\mathcal{I}_\sigma(F \leftarrow G) = \mathcal{I}_\sigma(F \vee \neg G)$ (since the implication connective $\leftarrow$ is semantically expressed in terms of $\neg$ and $\vee$), from $\mathcal{I}_\sigma(F \vee \neg G) = \mathcal{I}_\sigma(F \wedge G) - \mathcal{I}_\sigma(G) + 1$ by the definition of the truth value of a formula, and from (3). Finally, the claims (6) and (7) hold by the proof of Lemma 2. □

Note that Lemma 3 does not hold for all formulas F and G (that is, $Ł_n^\star$- and $Ł_n$-interpretations do not coincide completely in the semantics they give to classical first-order formulas). More precisely, the semantics of $Ł_n^\star$-interpretations cannot be defined inductively by *any* truth table. We give a counter-example to show this. If F and G are atomic formulas with $\mathcal{I}_\sigma(F) = \frac{1}{n-1}$ and $\mathcal{I}_\sigma(G) = \frac{n-2}{n-1}$, then we get $\mathcal{I}_\sigma(F \wedge G) = \min(\mathcal{I}_\sigma(F), \mathcal{I}_\sigma(G)) = \frac{1}{n-1}$. However, also $\mathcal{I}_\sigma(F) = \frac{1}{n-1}$ and thus $\mathcal{I}_\sigma(\neg F) = \frac{n-2}{n-1}$, but it always holds $\mathcal{I}_\sigma(F \wedge \neg F) = 0$.

Especially for handling uncertain beliefs, $Ł_n^\star$-interpretations provide a more intuitive semantics than $Ł_n$-interpretations, as the following example shows.

3.3 Example

Assume an agent has the belief $\frac{1}{2}$ in an event F. $Ł_n^\star$-interpretations then assign the belief 0 to the false event $F \wedge \neg F$ and the belief 1 to the true event $F \vee \neg F$. $Ł_n$-interpretations, however, assign the belief $\frac{1}{2}$ to both $F \wedge \neg F$ and $F \vee \neg F$.

Assume now an agent has the belief $\frac{1}{2}$ in the events F and G. $Ł_n^\star$-interpretations then assign the same belief 1 to the logically equivalent events $F \leftarrow G$ and $F \vee \neg G$. $Ł_n$-interpretations, however, assign the different beliefs 1 and $\frac{1}{2}$ to $F \leftarrow G$ and $F \vee \neg G$, respectively.

4 Disjunctive Logic Programming with Many-Valued Disjunctive Facts in $Ł_n^\star$

In the previous section, we introduced $Ł_n^\star$-interpretations and we showed that their characterizing property (1) is propagated through the structure of all the formulas that are built without $\neg$ and $\leftarrow$. Hence, each n-valued knowledge-base that just contains n-valued formulas of the form $tv(F) \geq 1$ and n-valued formulas that are built without $\neg$ and $\leftarrow$ has a unique splitting into $n-1$ layers of classical knowledge-bases. Moreover, as we show in this section, in disjunctive logic programs with n-valued disjunctive facts, we can reduce logical consequences in $Ł_n^\star$ to classical logical consequences in $n-1$ layers of classical disjunctive logic programs. This result is of particular interest for handling uncertain knowledge in disjunctive deductive databases, since it can directly be used to generalize their extensional part to many-valued disjunctive facts.

4.1 Logical Consequence in $Ł_n^\star$

A *disjunctive fact* is a universally quantified formula $\forall(A_1 \vee \dots \vee A_k)$ with atomic formulas $A_1, \dots, A_k$ and $k \geq 1$. A *disjunctive program clause* is a universally quantified formula $\forall(A_1 \vee \cdots \vee A_k \vee \neg B_1 \vee \cdots \vee \neg B_l)$ with atomic formulas $A_1, \ldots, A_k, B_1, \ldots, B_l$, where $k \geq 1$ and $l \geq 0$. An *n-valued disjunctive fact* is an n-valued formula $tv(F) \geq c$ with a disjunctive fact F. A *disjunctive logic program with n-valued disjunctive facts* is a finite set of n-valued formulas $tv(P) \geq 1$ with disjunctive program clauses P and of n-valued disjunctive facts.

In the sequel, let $\mathcal{P}$ be a disjunctive logic program with n-valued disjunctive facts. Let the Herbrand base $HB_{\mathcal{P}}$ comprise all ground atomic formulas that are constructed with function symbols and predicate symbols from $\mathcal{P}$ (we assume that $\mathcal{P}$ contains at least one 0-ary predicate symbol or at least one predicate symbol and one 0-ary function symbol). We subsequently restrict our considerations to *Herbrand $L_n^\star$-interpretations* $\mathcal{I} = (I_1, \ldots, I_{n-1})$, which consist of $n-1$ classical Herbrand interpretations $I_1, \ldots, I_{n-1}$ with $HB_{\mathcal{P}} \supseteq I_1 \supseteq \cdots \supseteq I_{n-1}$.

We now split $\mathcal{P}$ into $n-1$ layers of disjunctive logic programs. For each truth value $d > 0$ from TV let $\mathcal{P}_d$ contain all disjunctive program clauses P for which there exists a truth value $c \geq d$ from TV with $tv(P) \geq c \in \mathcal{P}$.

The next theorem shows the crucial result that certain logical consequences in $\mathrm{L}_n^\star$ in disjunctive logic programs with n-valued disjunctive facts can be reduced to logical consequences in $n-1$ layers of classical disjunctive logic programs.

Theorem 4 *Let $\mathcal{P}$ be a disjunctive logic program with n-valued disjunctive facts. For all n-valued disjunctive facts $tv(F) \geq d$ with $d > 0$:*

$$\mathcal{P} \models tv(F) \geq d \textit{ in } L_n^\star \textit{ iff } \mathcal{P}_d \models F\,.$$

Proof. '$\Leftarrow$': Let $\mathcal{I} = (I_1, \ldots, I_{n-1})$ be a model of $\mathcal{P}$. Then, we get $I_i \models P$ for all $tv(P) \geq c \in \mathcal{P}$ and all $1 \leq i \leq c \cdot (n-1)$ (here, we use that $\mathcal{P}$ contains only n-valued disjunctive facts and n-valued formulas of the form $tv(P) \geq 1$). Hence, the classical Herbrand interpretation $I_{d \cdot (n-1)}$ is a model of $\mathcal{P}_d$. Since $\mathcal{P}_d$ logically entails F, the interpretation $I_{d \cdot (n-1)}$ is also a model of F. Hence, since $I_1 \supseteq \cdots \supseteq I_{n-1}$, each classical Herbrand interpretation I_i with $1 \leq i \leq d \cdot (n-1)$ is a model of F. Thus, we also get $\mathcal{I} \models tv(F) \geq d$.

'$\Rightarrow$': Let I be a classical Herbrand interpretation that is a model of $\mathcal{P}_d$. We define a model $\mathcal{I} = (I_1, \ldots, I_{n-1})$ of $\mathcal{P}$ as follows. Let $I_i = HB_{\mathcal{P}}$ for all $1 \leq i < d \cdot (n-1)$ (this works, since $\mathcal{P}$ is a disjunctive logic program with n-valued disjunctive facts) and $I_i = I$ for all $d \cdot (n-1) \leq i \leq n-1$ (this would also work for more general n-valued formulas in $\mathcal{P}$). Now, since $\mathcal{P}$ logically entails $tv(F) \geq d$, each classical Herbrand interpretation I_i with $1 \leq i \leq d \cdot (n-1)$ is a model of F. Hence, in particular the interpretation $I_{d \cdot (n-1)}$ is a model of F. That is, the interpretation I is a model of F. $\Box$

Finally, Theorem 4 also shows that logical consequences in $\mathrm{L}_n^\star$ in disjunctive logic programs with n-valued disjunctive facts are independent of the number n of truth values. More precisely, for all disjunctive logic programs with n-valued disjunctive facts $\mathcal{P}$, n-valued disjunctive facts $tv(F) \geq d$, and $m = kn - k + 1$ with $k \geq 1$, we get $\mathcal{P} \models tv(F) \geq d$ in $\mathrm{L}_n^\star$ iff $\mathcal{P} \models tv(F) \geq d$ in $\mathrm{L}_m^\star$.

4.2 Example

Let $\mathcal{P}$ be the following disjunctive logic program with 4-valued disjunctive facts (a, b, and c are 0-ary function symbols, p, q, r, and s are 1-ary predicate symbols, t is a 2-ary predicate symbol, and X and Y are variables):

$$\mathcal{P}' = \{t(X,Y) \leftarrow q(X) \wedge r(Y),\ t(X,Y) \leftarrow q(X) \wedge s(Y),\ r(Y) \vee s(Y) \leftarrow p(Y)\}$$
$$\mathcal{P}\ = \{tv(P) \geq 1 \mid P \in \mathcal{P}'\} \cup \{tv(p(b)) \geq 2/3,\ tv(q(a)) \geq 2/3,\ tv(r(c) \vee s(c)) \geq 1/3\}\,.$$

We get the following three layers of classical disjunctive logic programs:

$$\begin{aligned}\mathcal{P}_{1/3} &= \mathcal{P}' \cup \{p(b),\ q(a),\ r(c) \vee s(c)\}\\ \mathcal{P}_{2/3} &= \mathcal{P}' \cup \{p(b),\ q(a)\}\\ \mathcal{P}_{1} &= \mathcal{P}'\,.\end{aligned}$$

We easily verify that $\mathcal{P}_{1/3} \models t(a,c)$ and $\mathcal{P}_{2/3} \models t(a,b)$. Hence, by Theorem 4, $\mathcal{P} \models tv(t(a,c)) \geq 1/3$ and $\mathcal{P} \models tv(t(a,b)) \geq 2/3$ in $Ł_4^\star$ (and also in $Ł_7^\star$, $Ł_{10}^\star$, ...). Moreover, we easily verify that neither $\mathcal{P}_{2/3} \models t(a,c)$ nor $\mathcal{P}_1 \models t(a,b)$. Hence, by Theorem 4, neither $\mathcal{P} \models tv(t(a,c)) \geq 2/3$ nor $\mathcal{P} \models tv(t(a,b)) \geq 1$ in $Ł_4^\star$ (and these logical consequences also do not hold in $Ł_7^\star$, $Ł_{10}^\star$, ...).

5 Many-Valued Logic Programming in $Ł_n^\star$

In this section, we focus on n-valued logic programming in $Ł_n^\star$. Differently from disjunctive logic programming with n-valued facts in $Ł_n^\star$, it cannot be reduced to classical logic programming in $n-1$ layers. However, the semantic background of $Ł_n^\star$-interpretations easily provides a model and a fixpoint semantics for n-valued logic programming in $Ł_n^\star$, which generalize the model and the fixpoint semantics of classical logic programming (see, for example, [1]).

5.1 Model and Fixpoint Semantics

A *program clause* is a universally quantified formula $\forall(A \vee \neg B_1 \vee \cdots \vee \neg B_l)$ with atomic formulas $A, B_1, \ldots, B_l$ and $l \geq 0$. An *n-valued program clause* is an n-valued formula $tv(P) \geq c$ with a program clause P. We abbreviate n-valued program clauses $tv(\forall(H \vee \neg B_1 \vee \cdots \vee \neg B_k)) \geq c$ by $(H \leftarrow B_1 \wedge \cdots \wedge B_k)[c,1]$. An *$n$-valued logic program* is a finite set of n-valued program clauses.

In the sequel, let $\mathcal{P}$ be an n-valued logic program and let the Herbrand base $HB_\mathcal{P}$ be defined like in Section 4.1. Again, we restrict our considerations to Herbrand $Ł_n^\star$-interpretations, which are now identified with fuzzy sets [35] that are mappings from $HB_\mathcal{P}$ to TV. More precisely, each Herbrand $Ł_n^\star$-interpretation $(I_1, I_2, \ldots, I_{n-1})$ is identified with the fuzzy set $\boldsymbol{I}\colon HB_\mathcal{P} \rightarrow TV$ that is defined by $\boldsymbol{I}(A) = \max(\{c \in TV \setminus \{0\} \mid A \in I_{c\cdot(n-1)}\} \cup \{0\})$ for all $A \in HB_\mathcal{P}$. We subsequently use bold symbols to denote such fuzzy sets.

The fuzzy sets $\boldsymbol{\emptyset}$ and $\boldsymbol{HB}_\mathcal{P}$ are defined by $\boldsymbol{\emptyset}(A) = 0$ and $\boldsymbol{HB}_\mathcal{P}(A) = 1$ for all $A \in HB_\mathcal{P}$. Finally, we define the intersection, the union, and the subset relation for fuzzy sets $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ as usual by $\boldsymbol{S}_1 \cap \boldsymbol{S}_2 = \min(\boldsymbol{S}_1, \boldsymbol{S}_2)$, $\boldsymbol{S}_1 \cup \boldsymbol{S}_2 = \max(\boldsymbol{S}_1, \boldsymbol{S}_2)$, and $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$ iff $\boldsymbol{S}_1 = \boldsymbol{S}_1 \cap \boldsymbol{S}_2$, respectively.

The semantics of n-valued program clauses under Herbrand $\text{Ł}_n^\star$-interpretations is already defined by the semantics of n-valued formulas under Pr_n-interpretations. However, to get a rough idea how to define an immediate consequence operator, it might be useful to briefly focus on the technical details that additionally follow from the properties of $\text{Ł}_n^\star$-interpretations:

Lemma 5 *For all interpretations $\boldsymbol{I} \subseteq \boldsymbol{HB}_{\mathcal{P}}$, all variable assignments σ, and all n-valued program clauses $(H \leftarrow B_1 \wedge \cdots \wedge B_k)[c, 1]$ in $\mathcal{P}$:*

$$tv(H \vee \neg B_1 \vee \cdots \vee \neg B_k) \geq c \textit{ is true in } \boldsymbol{I} \textit{ under } \sigma \textit{ iff}$$
$$\boldsymbol{I}_\sigma(H) \;\geq\; c - 1 + \min(\boldsymbol{I}_\sigma(B_1), \ldots, \boldsymbol{I}_\sigma(B_k))\,.$$

Proof. By (3) and (5) of Lemma 3, we get directly $\boldsymbol{I}_\sigma(H \vee \neg B_1 \vee \cdots \vee \neg B_k) = \boldsymbol{I}_\sigma(H \leftarrow B_1 \wedge \cdots \wedge B_k) = \min(1, \boldsymbol{I}_\sigma(H) - \min(\boldsymbol{I}_\sigma(B_1), \ldots, \boldsymbol{I}_\sigma(B_k)) + 1)$. Hence, the n-valued formula $tv(H \vee \neg B_1 \vee \cdots \vee \neg B_k) \geq c$ is true in $\boldsymbol{I}$ under σ iff $\min(1, \boldsymbol{I}_\sigma(H) - \min(\boldsymbol{I}_\sigma(B_1), \ldots, \boldsymbol{I}_\sigma(B_k)) + 1) \geq c$. This already shows the claim, since $1 \geq c$ for all truth values $c \in TV$. □

We define the monotonic immediate consequence operator $\boldsymbol{T}_{\mathcal{P}}$ on the set of all subsets of $\boldsymbol{HB}_{\mathcal{P}}$ as follows. For all $\boldsymbol{I} \subseteq \boldsymbol{HB}_{\mathcal{P}}$ and $H \in HB_{\mathcal{P}}$:

$$\begin{aligned}\boldsymbol{T}_{\mathcal{P}}(\boldsymbol{I})(H) = \max(\{c - 1 + \min(\boldsymbol{I}(B_1), \ldots, \boldsymbol{I}(B_k)) \mid (H \leftarrow B_1 \wedge \cdots \wedge B_k)[c, 1] \\ \text{is a ground instance of an } n\text{-valued program clause in } \mathcal{P}\} \cup \{0\})\,.\end{aligned}$$

Note that we canonically define $\min(\boldsymbol{I}(B_1), \ldots, \boldsymbol{I}(B_k)) = 1$ for $k = 0$.

For all $\boldsymbol{I} \subseteq \boldsymbol{HB}_{\mathcal{P}}$, we define $\boldsymbol{T}_{\mathcal{P}} \uparrow \omega(\boldsymbol{I})$ as the union of all $\boldsymbol{T}_{\mathcal{P}} \uparrow n(\boldsymbol{I})$ with $n < \omega$, where $\boldsymbol{T}_{\mathcal{P}} \uparrow 0(\boldsymbol{I}) = \boldsymbol{I}$ and $\boldsymbol{T}_{\mathcal{P}} \uparrow (n+1)(\boldsymbol{I}) = \boldsymbol{T}_{\mathcal{P}}(\boldsymbol{T}_{\mathcal{P}} \uparrow n(\boldsymbol{I}))$ for all $n < \omega$.

Next, we show that the immediate consequence operator $\boldsymbol{T}_{\mathcal{P}}$ makes sense. That is, that each model of $\mathcal{P}$ corresponds to a pre-fixpoint of $\boldsymbol{T}_{\mathcal{P}}$:

Lemma 6 *$\boldsymbol{I} \subseteq \boldsymbol{HB}_{\mathcal{P}}$ is a model of $\mathcal{P}$ iff $\boldsymbol{T}_{\mathcal{P}}(\boldsymbol{I}) \subseteq \boldsymbol{I}$.*

Proof. The claim follows immediately from Lemma 5. □

To arrive at the main theorem, it just remains to show the continuity of $\boldsymbol{T}_{\mathcal{P}}$:

Lemma 7 *$\boldsymbol{T}_{\mathcal{P}}$ is continuous.*

Proof sketch. We must show that $\bigcup\{\boldsymbol{T}_{\mathcal{P}}(\boldsymbol{I}_i) \mid i \geq 0\} = \boldsymbol{T}_{\mathcal{P}}(\bigcup\{\boldsymbol{I}_i \mid i \geq 0\})$ for all sequences $\boldsymbol{I}_0 \subseteq \boldsymbol{I}_1 \subseteq \cdots \subseteq \boldsymbol{HB}_{\mathcal{P}}$. This can be done similarly to van Emden's proof of continuity of the operator T_P in his quantitative deduction [34]. □

Finally, the desired model and fixpoint semantics of n-valued logic programs is expressed by the following important theorem:

Theorem 8 $\bigcap\{\boldsymbol{I} \mid \boldsymbol{I} \subseteq \boldsymbol{HB}_{\mathcal{P}},\, \boldsymbol{I} \models \mathcal{P}\} = \mathit{lfp}(\boldsymbol{T}_{\mathcal{P}}) = \boldsymbol{T}_{\mathcal{P}} \uparrow \omega(\emptyset)\,.$

Proof. The first equality holds, since the monotonic operator $\boldsymbol{T}_{\mathcal{P}}$ has a least fixpoint $\mathit{lfp}(\boldsymbol{T}_{\mathcal{P}})$ that is equal to its least pre-fixpoint. Now, by Lemma 6, this least pre-fixpoint is equal to $\bigcap\{\boldsymbol{I} \mid \boldsymbol{I} \subseteq \boldsymbol{HB}_{\mathcal{P}},\, \boldsymbol{I} \models \mathcal{P}\}$. The second equality follows from the continuity of $\boldsymbol{T}_{\mathcal{P}}$ by Lemma 7. □

Theorem 8 provides a characterization of logical consequence in $Ł_n^\star$ as follows. The n-valued logic program $\mathcal{P}$ logically entails in $Ł_n^\star$ the ground n-valued atomic formula $tv(A) \geq d$ iff $\boldsymbol{T}_\mathcal{P}\!\uparrow\!\omega(\emptyset)(A) \geq d$.

Hence, since the fixpoint semantics is independent of the number n of truth values, the notion of logical consequence in $Ł_n^\star$ in n-valued logic programs is also independent of the number n of truth values. In detail, for all n-valued logic programs $\mathcal{P}$, ground n-valued atomic formulas $tv(A) \geq d$, and $m = kn - k + 1$ with $k \geq 1$, we get $\mathcal{P} \models tv(A) \geq d$ in $Ł_n^\star$ iff $\mathcal{P} \models tv(A) \geq d$ in $Ł_m^\star$.

5.2 Example

Let $\mathcal{P}$ be the following 4-valued logic program (a, b, c, and d are 0-ary function symbols, e and p are 2-ary predicate symbols, and X, Y, and Z are variables):

$$\begin{aligned}\mathcal{P} = \{&(e(a,b) \leftarrow)[1,1],\ (e(b,c) \leftarrow)[1,1],\ (e(c,d) \leftarrow)[2/3,1],\\ &(p(X,Y) \leftarrow e(X,Y))[1,1],\ (p(X,Z) \leftarrow p(X,Y) \wedge p(Y,Z))[2/3,1]\}\,.\end{aligned}$$

The line of computation of the fixpoint iteration is reported by the following members of $\boldsymbol{T}_\mathcal{P}\!\uparrow\!1(\emptyset)$, $\boldsymbol{T}_\mathcal{P}\!\uparrow\!2(\emptyset)$, $\boldsymbol{T}_\mathcal{P}\!\uparrow\!3(\emptyset)$, and $\boldsymbol{T}_\mathcal{P}\!\uparrow\!4(\emptyset)$:

$$\begin{aligned}(e(a,b),1),(e(b,c),1),(e(c,d),2/3) &\in \boldsymbol{T}_\mathcal{P}\!\uparrow\!1(\emptyset)\\ (p(a,b),1),(p(b,c),1),(p(c,d),2/3) &\in \boldsymbol{T}_\mathcal{P}\!\uparrow\!2(\emptyset)\\ (p(a,c),2/3),(p(b,d),1/3) &\in \boldsymbol{T}_\mathcal{P}\!\uparrow\!3(\emptyset)\\ (p(a,d),1/3) &\in \boldsymbol{T}_\mathcal{P}\!\uparrow\!4(\emptyset)\,.\end{aligned}$$

Hence, $\mathcal{P}$ logically entails in $Ł_4^\star$ (and also in $Ł_7^\star$, $Ł_{10}^\star$, ...) the ground 4-valued atomic formula $tv(p(a,d)) \geq 1/3$, since $(p(a,d),1/3) \in \boldsymbol{T}_\mathcal{P}\!\uparrow\!\omega(\emptyset) = \boldsymbol{T}_\mathcal{P}\!\uparrow\!4(\emptyset)$.

5.3 Computational Complexity

In this section, we analyze the computational complexity of three special cases of deciding whether a ground n-valued atomic formula $tv(A) \geq c$ is a logical consequence in $Ł_n^\star$ of an n-valued logic program. They generalize the decision problems that define the computational complexity of propositional logic programming, the data complexity of datalog, and the program complexity of datalog (see, for example, [5] for a survey). Crucially, we now show that the computational complexities of the three deduction problems in $Ł_n^\star$ are no worse than the computational complexities of their more specialized classical counterparts.

Theorem 9 *Let $\mathcal{P}$ be a ground n-valued logic program, let A be a ground atomic formula, and let c be a truth value from TV. The problem of deciding whether $tv(A) \geq c$ is a logical consequence in $Ł_n^\star$ of $\mathcal{P}$ is P-complete.*

Proof. The problem is P-hard, since it generalizes the P-complete problem of deciding whether a ground atomic formula is a classical logical consequence of a ground classical logic program.

It is in P, since the number of applications of the operator $\boldsymbol{T}_\mathcal{P}$ to reach $\boldsymbol{T}_\mathcal{P} \uparrow \omega(\emptyset)$ is bounded by $n - 1$ times the number of ground atomic formulas in $\mathcal{P}$ and since each application of $\boldsymbol{T}_{\mathcal{P}}$ runs in polynomial time in the size of $\mathcal{P}$. Note that, in fact, the number of applications of $\boldsymbol{T}_{\mathcal{P}}$ to reach $\boldsymbol{T}_{\mathcal{P}} \uparrow \omega(\emptyset)$ is even bounded by the number of ground atomic formulas in $\mathcal{P}$. To show this, let us simply assume that $\boldsymbol{T}_{\mathcal{P}} \uparrow \omega(\emptyset)$ contains a pair (A, tv) that is not contained in $\boldsymbol{T}_{\mathcal{P}} \uparrow m(\emptyset)$, where m is the number of ground atomic formulas in $\mathcal{P}$. Hence, the derivation tree of (A, tv) must contain at least one path from a leaf to the root (A, tv) with first a node (B, tv_1) and thereafter another node (B, tv_2), where B is a ground atomic formula and $tv_1, tv_2 \in TV$ with $tv_1 < tv_2$. However, the immediate consequence operator $\boldsymbol{T}_{\mathcal{P}}$ shows that the truth values are decreasing along every path from a leaf to the root. That is, we also get $tv_1 \geq tv_2$. □

Theorem 10 *Let all function symbols in Φ be of arity 0.*

a) Let $\mathcal{P}$ be a fixed n-valued logic program, let $\mathcal{F}$ be a varying set of ground n-valued atomic formulas, let A be a ground atomic formula, and let c be a truth value from TV. The problem of deciding whether $tv(A) \geq c$ is a logical consequence in $L_n^\star$ of $\mathcal{P} \cup \mathcal{F}$ is P-complete.

b) Let $\mathcal{P}$ be a varying n-valued logic program, let $\mathcal{F}$ be a fixed set of ground n-valued atomic formulas, let A be a ground atomic formula, and let c be a truth value from TV. The problem of deciding whether $tv(A) \geq c$ is a logical consequence in $L_n^\star$ of $\mathcal{P} \cup \mathcal{F}$ is DEXPTIME-complete.

Proof. a) The problem is P-hard, since it generalizes the P-complete problem that defines the data complexity of datalog.

It is in P by Theorem 9, since the number of all ground instances of n-valued program clauses in $\mathcal{P} \cup \mathcal{F}$ (which are constructed with all 0-ary function symbols from $\mathcal{P} \cup \mathcal{F}$) is polynomial in the input size of $\mathcal{F}$.

b) The problem is DEXPTIME-hard, since it generalizes the DEXPTIME-complete problem that defines the program complexity of datalog.

It is in DEXPTIME by Theorem 9, since the number of all ground instances of n-valued program clauses in $\mathcal{P} \cup \mathcal{F}$ (which are built with all 0-ary function symbols from $\mathcal{P} \cup \mathcal{F}$) is exponential in a polynom of the input size of $\mathcal{P}$. □

6 Related Work

The literature already contains quite extensive work on probabilistic logics and on truth-functional many-valued logics separately. However, to the best of our knowledge, an integration of both has never been studied so far.

Probabilistic propositional logics and their various dialects are thoroughly studied in the literature (see, for example, the pioneering work in [29] and [10]). Their extensions to probabilistic first-order logics can be classified into first-order logics in which probabilities are defined over a set of possible worlds and those in which probabilities are given over the elements of the domain (see, for example, [2] and [15]). The first ones are suitable for representing degrees of belief, while the latter are appropriate for describing statistical knowledge. The same classification holds for approaches to probabilistic logic programming (see, for example, [27], [26], [28], [24], and [25]).

Many approaches to truth-functional finite-valued logic programming are restricted to three or four truth values (see, for example, [18], [11], [12], [3], [30], and [8]). Among these approaches, the one closest in spirit to n-valued logic programming in $Ł_n^\star$ is the 3-valued approach in [18]. Note that in contrast to [18], however, n-valued logic programming in $Ł_n^\star$ is especially much more general and also well-grounded on a probabilistic semantics.

Our n-valued logic programming in $Ł_n^\star$ is closely related to van Emden's infinite-valued quantitative deduction in [34]. More precisely, both approaches have a common semantic background in probabilistic logic programming as introduced in [24]. That is, interpreted by probability distributions over possible worlds that satisfy an extension of (1), the probabilistic logic programs in [24] coincide in their fixpoint semantics with n-valued logic programs in $Ł_n^\star$ (if the implication connective is interpreted as material implication) and with van Emden's quantitative logic programs (if the implication connective is interpreted as conditional probability). Note that probability distributions over possible worlds represent experts worlds in which the number of experts is variable and in which the (not necessarily equal) credibilities of the experts sum up to 1.

Crucially, in contrast to van Emden's quantitative deduction, our n-valued logic programming in $Ł_n^\star$ does not have to struggle with an infinite number of truth values in its proof theory (which will be presented in future work).

Both n-valued logic programming in $Ł_n^\star$ and also van Emden's quantitative deduction can be regarded as special cases of generalized annotated logic programming (where annotations may contain variables and function symbols: see, for example, [16]). Moreover, n-valued logic programming in $Ł_n^\star$ is generalized by signed formula logic programming as discussed in [21]. However, we think that it is important in its own right, since generalized annotated logic programming and also signed formula logic programming (with their increased expressiveness) generally do not seem to have the computational properties of classical logic programming (see [19] and [21]). Finally, we also consider n-valued logic programming in $Ł_n^\star$ as a new important starting point for giving a natural semantics to disjunctions and negations in many-valued logic programming (which is a very interesting topic of future research). In particular, the implied natural semantics of negations will differ from the classical semantics of negations in current truth-functional many-valued logic programming.

Note that annotated logic programs with only constants as annotations (like the closely related signed formula approach in [14]) are not expressive enough to generalize n-valued logic programs in $Ł_n^\star$, since they cannot represent truth value assignments on the level of n-valued program clauses (they can just represent logical relationships between n-valued facts).

Finally, the way in which truth values are propagated through the implication connective of many-valued program clauses distinguishes our n-valued logic programming in $Ł_n^\star$ from many other approaches like, for example, the propositional approach in [9] and the work in [17] (we interpret the implication connective as material implication, while the latter approaches define the truth value of the head of a many-valued program clause as conjunction of the truth value of the many-valued program clause itself and the truth value of its body).

7 Summary and Outlook

We presented n-valued first-order logics with a purely probabilistic semantics. We then introduced $\L_n^\star$-interpretations as a new probabilistic semantics of n-valued first-order logics that lies between the purely probabilistic semantics and the truth-functional semantics of the n-valued Łukasiewicz logics $\L_n$.

We showed that $\L_n^\star$-interpretations have very interesting properties from both the semantic and the computational point of view. In particular, closed formulas of classical first-order logics that are logically equivalent in the classical sense also have the same truth value under all $\L_n^\star$-interpretations. Moreover, disjunctive logic programming with n-valued disjunctive facts in $\L_n^\star$ can be reduced to classical disjunctive logic programming in $n-1$ layers. Moreover, n-valued logic programs in $\L_n^\star$ were shown to have a model and a fixpoint semantics that are very similar to those of classical logic programs. Finally, some important deduction problems in n-valued logic programs in $\L_n^\star$ were shown to have the same computational complexity like their classical counterparts.

Future research may concern the extension of $\L_n^\star$-interpretations to probability distributions over possible worlds (by which we obtain experts worlds in which the number of experts is variable and in which their credibilities is not necessarily equal). Another topic of future research is n-valued logic programming in $\L_n^\star$ with negations and disjunctions. Finally, it would be interesting to extend the language of n-valued formulas to also allow reasoning on the level of the full n-valued first-order language (similar to, for example, [15]).

8 Acknowledgements

I am very grateful to the referees for their useful comments. I am also very grateful to Thomas Eiter for having presented this paper at the conference.

References

1. K. R. Apt. Logic programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 10, pages 493–574. The MIT Press, 1990.
2. F. Bacchus, A. Grove, J. Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of beliefs. *Artif. Intell.*, 87:75–143, 1996.
3. H. A. Blair and V. S. Subrahmanian. Paraconsistent logic programming. *Theor. Comput. Sci.*, 68:135–154, 1989.
4. R. Carnap. *Logical Foundations of Probability.* University of Chicago Press, Chicago, 1950.
5. E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. In *Proc. of the 12th Annual IEEE Conference on Computational Complexity*, pages 82–101, 1997.
6. B. de Finetti. *Theory of Probability.* J. Wiley, New York, 1974.
7. A. Dekhtyar and V. S. Subrahmanian. Hybrid probabilistic programs. In *Proc. of the 14th International Conference on Logic Programming*, pages 391–405, 1997.
8. J. P. Delahaye and V. Thibau. Programming in three-valued logic. *Theor. Comput. Sci.*, 78:189–216, 1991.
9. G. Escalada-Imaz and F. Manyà. Efficient interpretation of propositional multi-valued logic programs. In *Proc. 5th International Conference on Information Pro-*

cessing and Management of Uncertainty in Knowledge-Based Systems (IPMU '94), volume 945 of *LNCS*, pages 428–239. Springer, 1995.
10. R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87:78–128, 1990.
11. M. Fitting. Partial models and logic programming. *Theor. Comput. Sci.*, 48:229–255, 1986.
12. M. Fitting. Bilattices and the semantics of logic programming. *J. Logic Program.*, 11(1–2):91–116, 1991.
13. M. Fitting. Many-valued modal logics II. *Fundam. Inf.*, 17:55–73, 1992.
14. R. Hähnle. Exploiting data dependencies in many-valued logics. *J. Appl. Non-Class. Log.*, 6(1):49–69, 1996.
15. J. Y. Halpern. An analysis of first-order logics of probability. *Artif. Intell.*, 46:311–350, 1990.
16. M. Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *J. Logic Program.*, 12(3–4):335–367, 1992.
17. L. V. S. Lakshmanan and F. Sadri. Probabilistic deductive databases. In *Proc. of the International Logic Programming Symposium*, pages 254–268, 1994.
18. J.-L. Lassez and M. J. Maher. Optimal fixedpoints of logic programs. *Theor. Comput. Sci.*, 39:15–25, 1985.
19. S. M. Leach and J. J. Lu. Query processing in annotated logic programming: Theory and implementation. *J. Intell. Inf. Syst.*, 6(1):33–58, 1996.
20. J. W. Lloyd. *Foundations of Logic Programming.* Springer, Berlin, 2nd ed., 1987.
21. J. J. Lu. Logic programming with signs and annotations. *J. Log. Comput.*, 6(6):755–778, 1996.
22. T. Lukasiewicz. Magic inference rules for probabilistic deduction under taxonomic knowledge. In *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 354–361. Morgan Kaufmann Publishers, 1998.
23. T. Lukasiewicz. Probabilistic deduction with conditional constraints over basic events. In *Principles of Knowledge Representation and Reasoning: Proc. of the 6th International Conference*, pages 380–391. Morgan Kaufmann Publishers, 1998.
24. T. Lukasiewicz. Probabilistic logic programming. In *Proc. of the 13th European Conference on Artificial Intelligence*, pages 388–392. J. Wiley & Sons, 1998.
25. R. T. Ng. Semantics, consistency, and query processing of empirical deductive databases. *IEEE Trans. Knowl. Data Eng.*, 9(1):32–49, 1997.
26. R. T. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Inf. Comput.*, 101:150–201, 1992.
27. R. T. Ng and V. S. Subrahmanian. A semantical framework for supporting subjective and conditional probabilities in deductive databases. *J. Autom. Reasoning*, 10(2):191–235, 1993.
28. R. T. Ng and V. S. Subrahmanian. Stable semantics for probabilistic deductive databases. *Inf. Comput.*, 110:42–83, 1994.
29. N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28:71–88, 1986.
30. T. C. Przymusinski. Three-valued nonmonotonic formalisms and semantics of logic programs. *Artif. Intell.*, 49:309–343, 1991.
31. N. Rescher. *Many-valued Logic.* McGraw-Hill, New York, 1969.
32. J. B. Rosser and A. R. Turquette. *Many-valued Logics.* North-Holland, 1952.
33. V. S. Subrahmanian. Amalgamating knowledge bases. *ACM Trans. Database Syst.*, 19(2):291–331, 1994.
34. M. H. van Emden. Quantitative deduction and its fixpoint theory. *J. Logic Program.*, 3(1):37–53, 1986.
35. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.