# MV-Datalog+/-: Effective Rule-based Reasoning with Uncertain Observations (Extended Abstract)*

**Matthias Lanzinger**[1] , **Stefano Sferrazza**[1,2] and **Georg Gottlob**[1]
[1] Department of Computer Science, University of Oxford
[2] TU Wien
matthias.lanzinger@cs.ox.ac.uk, stefano.sferrazza@cs.ox.ac.uk, georg.gottlob@cs.ox.ac.uk

## Abstract

Modern data processing applications often combine information from a variety of complex sources. Oftentimes, some of these sources, like Machine-Learning systems or crowd-sourced data, are not strictly binary but associated with some degree of confidence in the observation. Ideally, reasoning over such data should take this additional information into account as much as possible. To this end, we propose extensions of Datalog and Datalog$^{\pm}$ to the semantics of Łukasiewicz logic **Ł**, one of the most common fuzzy logics. We show that such an extension preserves important properties from the classical case and how these properties can lead to efficient reasoning procedures for these new languages.

## 1 Introduction

Datalog and its extensions are important languages for databases access and lie at the foundation of many rule-based reasoning formalisms. Continuous theoretical and technical improvements have led to the development of highly efficient Datalog systems for widespread practical use in a variety of applications (see e.g., [Maier *et al.*, 2018]).

However, in many such real-world applications the observations and facts that serve as the input are not actually certain but rather associated with some (possibly unknown) level of uncertainty. Particularly important in this context are settings where observations are made by Machine Learning (ML) systems. Consider a database that contains a relation which labels images of animals with the class of animal observed in the image. In modern settings, such labels are commonly derived from the output of ML systems that attempt to classify the image contents. Generally, such systems output possible labels for an image, together with a score that can be interpreted as the level of confidence in the correctness of the label (cf., [Mishkin *et al.*, 2017; Lee *et al.*, 2017]). When such labelling is used in further logical inference, the level of confidence in the individual labels would ideally also be reflected in any knowledge inferred from these uncertain observations. However, classical logical reasoning provides no way to consider such information and requires the projection of uncertainty levels to be either true or false, usually via some simple numerical threshold. Beyond the overall loss of information, this can also lead to problematic outcomes where logical conclusions are true despite their derivation being based on observations made with only moderate confidence. With observations from ML systems becoming a commonplace feature of many modern reasoning settings, the projection to binary truth values severely limits the immense potential of integrating ML observations with logic programming.

There are two natural ways to interpret uncertain observations as described in the example above. One can consider them as the likelihood of the fact being true, i.e., the level of confidence in a fact is interpreted as a probability of the fact holding true. Such an interpretation has been widely studied in the context of Problog [Raedt *et al.*, 2007], Markov Logic Networks [Richardson and Domingos, 2006], and probabilistic databases [Suciu *et al.*, 2011; Cavallo and Pittarelli, 1987]. However, generally, such formalisms make strong assumptions on the pairwise probabilistic independence of all tuples which can be difficult to satisfy. An approach to probabilistic reasoning in Datalog$^{\pm}$ based on the chase procedure was proposed recently by [Bellomarini *et al.*, 2020].

Alternatively, one can express levels of confidence in a fact in terms of *degrees of truth* as in *fuzzy logics* (cf. [Hájek, 1998]). That is, a fact that is considered to be true to a certain degree, in accordance with the level of confidence in the observation. A variety of approaches to combining logic programming with fuzzy logic have been studied in the past. Pioneering work in this area by [Achs and Kiss, 1995], [Ebrahim, 2001], and [Vojtás, 2001] introduced an early foundational idea for fuzzy logic programming. Similar ideas have also been proposed in fuzzy description logics, see e.g., [Bobillo and Straccia, 2016; Borgwardt and Peñaloza, 2017; Stoilos *et al.*, 2007], or [Lukasiewicz and Straccia, 2008] for an overview of fuzzy DLs.

One limiting factor of the existing logic programming approaches in our intended setting is the reliance on the *Gödel t-norm* (see, [Preining, 2010]) as the basis for many-valued semantics. The Gödel t-norm of two formulas $\phi$, $\psi$ is the minimum of the respective truth degrees of $\phi$ and $\psi$. While this

*This is an extended abstract of paper [Lanzinger *et al.*, 2022a] that received the Best Paper Award at the International Conference for Logic Programming (ICLP) 2022 simplifies the resulting logics in certain technical respects, the resulting semantics are not ideal for our envisioned applications in which we want to combine uncertain observations from AI systems with large certain knowledge bases.

Finally, Probabilistic Soft Logic (PSL) was introduced by [Bach *et al.*, 2017] as a framework for fuzzy reasoning with semantics based on Łukasiewicz logic. PSL is a significantly different language with different use cases than our proposed languages. It allows for negation in rule bodies as well as disjunction in the head, in addition to certain forms of aggregation. Semantically it treats all rules as soft constraints (i.e., they might not hold true) and aims to find interpretations that satisfy rules as much as possible. In the context of learning such behaviour is desirable but in terms of reasoning the resulting formalism deviates from many typical characteristics of the Datalog formalisms that we aim to generalise here.

In the following, we provide a high-level overview of our proposed formalisms. Full details and further discussion are available in the full version of this paper [Lanzinger *et al.*, 2022a].

## 2 Łukasiewicz Logic and $K$-fuzzy Models

In this paper, we will study the generalisations of Datalog and Datalog$^{\pm}$ to infinite-valued Łukasiewicz logic **Ł** (see, [Hájek, 1998]). Some of our techniques are particular to Łukasiewicz logic and do not readily apply to other many-valued semantics. For some relational signature $\sigma$, we consider the following logical language, where $R$ is a relational atom (in $\sigma$) and a *formula* $\varphi$ is defined via the grammar

$$\varphi ::= R \mid \varphi \odot \varphi \mid \varphi \oplus \varphi \mid \varphi \to \varphi \mid \neg\varphi$$

For a signature $\sigma$ and countably infinite domain $Dom$, let $GAtoms$ be the set of all ground atoms with respect to $\sigma$ and $Dom$. A *truth assignment* is a function $\nu\colon GAtoms \to [0,1]$, intuitively assigning a degree of truth in the real interval $[0,1]$ to every ground atom. By slight abuse of notation, we also apply $\nu$ to formulas to express the truth of ground formulas $\gamma$, $\gamma'$ according to the following inductive definitions.

$$\begin{aligned}
\nu(\neg\gamma) &= 1 - \nu(\gamma)\\
\nu(\gamma \odot \gamma') &= \max\{0, \nu(\gamma) + \nu(\gamma') - 1\}\\
\nu(\gamma \oplus \gamma') &= \min\{1, \nu(\gamma) + \nu(\gamma')\}\\
\nu(\gamma \to \gamma') &= \min\{1, 1 - \nu(\gamma) + \nu(\gamma')\}
\end{aligned}$$

That is, $\odot$ is the usual Łukasiewicz *t-norm* and $\oplus$ is the corresponding *t-conorm*, which take the place of conjunction and disjunction, respectively. Note that implication $\gamma \to \gamma'$ is equivalent to $\neg\gamma \oplus \gamma'$.

For rational $K \in (0,1]$ we say that a formula $\varphi$ is $K$-*satisfied* by $\nu$ if for every grounding $\gamma$ of $\varphi$ over $Dom$ it holds that $\nu(\gamma) \geq K$. Whenever we make use of $K$ in this context we assume it to be rational. In the context of rule-based reasoning, it may be of particular interest to observe that an implication is 1-satisfied exactly when the head is at least as true as the body. For a set of formulas $\Pi$, we say that a truth assignment $\nu$ is a $K$*-fuzzy model* if all formulas in $\Pi$ are $K$-satisfied by $\nu$.

In place of the database in the classical setting, we instead consider *(finite) partial truth assignments*, that is, partial functions $\tau : GAtoms \to (0,1]$ that are defined for a finite number of ground atoms. Let $(\Pi, \tau)$ be a pair where $\Pi$ is a set of formulas and $\tau$ is a partial truth assignment, a $K$*-fuzzy model* of $(\Pi, \tau)$ is a $K$-fuzzy model $\nu$ of $\Pi$ where $\nu(G) = \tau(G)$ for every ground atom $G$ for which $\tau$ is defined. Whenever we talk about formulas $\Pi$ and partial truth assignments $\tau$ we use $ADom$ to refer to their *active domain*, i.e., the subset of the domain that is mentioned in either $\Pi$ or $\{G \in GAtoms \mid \tau(G) \text{ is defined.}\}$. We write $GAtoms[Adom]$ to indicate $GAtoms$ restricted to groundings over $Adom$. $K$*-fuzzy models* and their theory were introduced and studied in the context of logic programming by [Ebrahim, 2001] but under different many-valued semantics based on the Gödel t-norm. For further discussion of the difference in semantics see the full paper [Lanzinger *et al.*, 2022a].

## 3 MV-Datalog

**Definition 1** (MV-Datalog Program)**.** An *MV-Datalog program* $\Pi$ is a set of **Ł** formulas of the form

$$B_1 \odot B_2 \odot \ldots \odot B_n \to H$$

where all $B_i$, for $1 \leq i \leq n$, and $H$ are relational atoms.

As partial truth assignments $\tau\colon GAtoms \to (0,1]$ take the place of the database in a classical Datalog setting. we also refer to such $\tau$ as databases in the context of MV-Datalog. We call a pair $\Pi, \tau$ of an MV-Datalog program and database an *MV-instance*. We can map a MV-Datalog program $\Pi$ naturally to a Datalog program by substituting $\odot$ with $\land$. We refer to the resulting Datalog program as $\Pi^{crisp}$. For the respective crisp version of $\tau$, we write $D_\tau$ for the (classical) database containing all facts $G$ for which $\tau(G)$ is defined.

Analogous to fact entailment in classical Datalog, we consider the following decision problem K-TRUTH, deciding whether a fact is true to at least a degree $c$ in all models.

K-TRUTH

| | |
|---|---|
| *Input* | MV-instance $(\Pi, \tau)$, ground atom $G$, $c \in [0,1]$ |
| *Output* | $\nu(G) \geq c$ for all $K$-fuzzy models $\nu$ of $(\Pi, \tau)$? |

MV-Datalog is a proper extension of Datalog in the sense that for $K = 1$ and when all ground atoms in the database are assigned truth 1, its models coincide exactly with those of Datalog programs.

Fact entailment in Datalog is typically decided by deriving a minimal model of program and database and checking whether the fact holds true in the (unique) minimal model. The ability to use minimal models as a representative to decide entailment is the key to efficient reasoning in Datalog. In the rest of this Section, we will discuss how the same is also possible for MV-Datalog. For two truth assignment $\nu$, $\mu$ we write $\nu \leq \mu$ when for all $G \in GAtoms$, it holds that $\nu(G) \leq \mu(G)$. We similarly write $\nu < \mu$ if $\nu \leq \mu$ and $\nu(G) < \mu(G)$ for at least one $G \in GAtoms$.

**Definition 2.** Let $\Pi, \tau$ be an MV-instance. We say that a $K$-fuzzy model $\mu$ of $(\Pi, \tau)$ is *minimal* if for every $K$-fuzzy model $\nu$ of $(\Pi, \tau)$ it holds that $\mu \leq \nu$.

In contrast to Datalog, MV-instances do not always have models. Consider a program consisting only of rule $R(x) \to S(x)$. For any $K > 0$, there is a database $\tau$ that assigns truth 1 to $R(a)$ and some truth less than $K$ to $S(a)$ such that the rule is not $K$-satisfied. This is a consequence of the definition of a $K$-fuzzy model $\nu$ of $(\Pi, \tau)$, which requires the truth values in $\nu$ to agree exactly with $\tau$, for every fact for which $\tau$ is defined. In some settings, it may be desirable to relax this slightly and consider $K$-fuzzy models $\nu$ where $\nu(G) \geq \tau(G)$ where $\tau(G)$ is defined. Our semantics cover such a relaxation since it can be simulated by straightforward rewriting of the program: for every relation symbol $R$ that occurs in $\tau$ add rule $R(\mathbf{x}) \rightarrow R'(\mathbf{x})$, and replace all other occurrences of $R$ in the program by $R'$. Nonetheless, for satisfiable instances, a suitable minimal model indeed always exists.

**Proposition 1.** *Let* $\Pi$, $\tau$ *be an MV-instance. For every rational* $K \in (0, 1]$*, if* $(\Pi, \tau)$ *is* $K$*-satisfiable, then there exists a unique minimal* $K$*-fuzzy model for* $(\Pi, \tau)$*.*

It is not difficult to see that deciding K-TRUTH is straightforward once we have such a unique minimal $K$-fuzzy model $\mu$. From the definition of the minimality of such models, it directly follows that for every ground fact $G$ we have $\mu(G) \geq c$ if and only if $\nu(G) \geq c$ for all $K$-fuzzy models of $(\Pi, \tau)$. The remaining question thus is how to find such models. To this end, we provide a characterisation of the minimal $K$-fuzzy model for some $(\Pi, \tau)$ in terms of a linear program $Opt^K_{\Pi,\tau}$. The constraints in the program are induced by the oblivious chase procedure (see, [Calì *et al.*, 2013])[1] for $\Pi^{crisp}, D_\tau$.

Let $(\Pi, \tau)$ be an MV-instance. Let $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ be the ground rules that occur[2] in an execution of the oblivious chase on $\Pi^{crisp}, D_\tau$. Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ be all of the ground atoms occurring in rules in $\Gamma$. For every $G_i \in \mathcal{G}$ we associate $G_i$ with a variable $x_i$ in our linear program that intuitively will represent the truth degree of $G_i$. For $\gamma_j$ of the form $G_{j_1} \odot \cdots \odot G_{j_\ell} \rightarrow G_{j_{head}}$ define

$$\text{Eval}(\gamma_j) := \sum_{k=1}^{\ell} (1 - x_{j_k}) + x_{j_{head}}$$

which directly expresses the satisfaction of rule $\gamma_j$, with variable $x_{j_k}$ representing the truth of $G_{j_k}$. The linear program $Opt^K_{\Pi,\tau}$ is then defined as follows

$$\begin{array}{lll} \text{minimise} & \sum_{i=1}^{n} x_i & \\ \text{subject to} & \text{Eval}(\gamma_j) \geq K & \text{for } 1 \leq j \leq m \\ & x_i = \tau(G_i) & \text{for } i \text{ where } \tau(G_i) \text{ is defined} \\ & 1 \geq x_i \geq 0 & \text{for } 1 \leq i \leq n \end{array} \tag{1}$$

By construction, any feasible solutions $\mathbf{x}$ of $Opt^K_{\Pi,\tau}$ induces a $K$-fuzzy model $\nu_\mathbf{x}$ that assigns $\nu_\mathbf{x}(G_i) = x_i$ and 0 to all other ground atoms not in $\mathcal{G}$ and vice versa. In the full paper, we additionally prove that optimality in the program corresponds exactly to the minimality of the induced $K$-fuzzy model. This then gives us a concrete and practical procedure for finding $K$-fuzzy minimal models and thus deciding K-TRUTH.

[1] While the use of the oblivious chase without existential quantification is untypical we explicitly require all rules induced by oblivious applications for our characterisation. Furthermore, this allows for a simpler transition to the semantics for existential rules in the following section.

[2] For a formal definition of this step see [Lanzinger *et al.*, 2022a].

**Theorem 2.** *Let* $\Pi$, $\tau$ *be an MV-instance. Then* $\mathbf{x}$ *is an optimal solution of* $Opt^K_{\Pi,\tau}$ *if and only if* $\nu_\mathbf{x}$ *is a minimal* $K$*-fuzzy model of* $(\Pi, \tau)$

**Corollary 3.** *Fix a rational* $K \in (0, 1]$. K-TRUTH *is in* **P** *with respect to data complexity. Moreover,* 1-TRUTH *is* **P***-complete in data complexity.*

# 4 MV-Datalog$^\pm$

Existential quantification in the head of rules provides a natural way of dealing with the common problem of incomplete or missing data by declaratively stating that certain facts must exist. Such usage is especially powerful in applications where rule-based reasoning is used for complex data analysis (see e.g., [Bellomarini *et al.*, 2017]). In the following, we introduce MV-Datalog$^\pm$ as an extension of MV-Datalog with a focus on providing a fuzzy reasoning language that is useful for such applications.

To obtain semantics matching our intuition for such a system, we propose an alternative to the commonly studied semantics for existential quantification for Łukasiewicz logic. We then identify certain *preferred models* that exhibit desired conceptual and computational behaviour as the basis for reasoning in MV-Datalog$^\pm$.

## 4.1 Strong Existential Quantification in Łukasiewicz Semantics

The semantics of existential quantification in Łukasiewicz logic is traditionally defined as $\nu(\exists \mathbf{x} \varphi(\mathbf{x})) = \sup\{\nu(\varphi[\mathbf{x}/\mathbf{c}]) \mid \mathbf{c} \in Dom^{|x|}\}$ where $\varphi[\mathbf{x}/\mathbf{c}]$ is the substitution of variables $\mathbf{x}$ in $\varphi$ by constants $\mathbf{c}$ (cf. [Hájek, 1998]). However, with our focus set on practical applications, we propose alternative semantics for existential quantification that match the semantics of $\oplus$. To contrast between the aforementioned semantics of $\exists$ we refer to our semantics as *strong existential quantification*.

$$\nu(\exists \mathbf{x}\, \varphi(\mathbf{x})) = \min \left\{ 1, \sum_{\mathbf{c} \in Dom^{|x|}} \nu(\varphi[\mathbf{x}/\mathbf{c}]) \right\}$$

We refer to **Ł** extended with strong existential quantification as **Ł**$_\exists$ (following standard syntax of first-order existential quantification). Such semantics for quantification in Łukasiewicz logic have also recently been studied more generally in a purely logical context by [Fjellstad and Olsen, 2021].

**Definition 3.** A MV-Datalog$^\pm$ program is a set of **Ł**$_\exists$ formulas that are either MV-Datalog rules or of the form

$$B_1 \odot B_2 \odot \cdots \odot B_n \rightarrow \exists \mathbf{x}\, \varphi(\mathbf{x}, \mathbf{y})$$

where $B_i$ are relational atoms, $\mathbf{y}$ are the variables in the body, and $\varphi$ is a formula that contains only a relational atom that uses all variables of $\mathbf{x}$. We refer to a pair of a MV-Datalog$^\pm$ program and a database as an *MV*$^\pm$*-instance*.

As in Datalog$^\pm$, our truth assignments are not constrained to the active domain of $\Pi$ and $\tau$ but allow for the introduction of "new" constants (i.e., from $Dom \setminus Adom$). As noted above, the introduction of new constants provides a natural mechanism to deductively handle missing and incomplete data. We propose the use of strong existential quantification to obtain specific desirable behaviour in case of incomplete information, as illustrated by the following example.

**Example 1.** We consider the following example rule, expressing that every company has a key person, to illustrate the difference between the two semantics for $\exists$.

$$Company(y) \quad \rightarrow \exists x \; KeyPerson(x, y)$$

Suppose we have the following database, we know for certain that $\mathrm{Acme}$ is a company and we are $0.8$ degrees confident that $\mathrm{Amy}$ is the key person of $\mathrm{Acme}$.

$$\tau(KeyPerson(\mathrm{Amy}, \mathrm{Acme})) = 0.8, \; \tau(Company(\mathrm{Acme})) = 1$$

We discuss the case in a 1-fuzzy model in the following. With existential semantics via the supremum of matching ground atoms, the first rule implies the existence of some new $KeyPerson(\mathrm{N}_1, \mathrm{Acme})$ with truth degree 1. With strong existential quantification, the first rule implies only truth degree $0.2$ for some $KeyPerson(\mathrm{N}_1, \mathrm{Acme})$ since the known observation $KeyPerson(\mathrm{Amy}, \mathrm{Acme})$ already contributes $0.8$ degrees of truth to the head.

Intuitively, the new constant $\mathrm{N}_1$ assumes the role of the unknown other object that possibly is the key person. In traditional semantics for $\exists$ we have to infer (in a 1-fuzzy model) that $\mathrm{N}_1$ is certain to control $\mathrm{Acme}$. In contrast, under strong existential semantics, the confidence in $\mathrm{N}_1$ being the key person is determined by the known observation that $\mathrm{Amy}$ might be the key person.

### 4.2 Preferred Models for MV-Datalog$^\pm$

When considered in full generality, our proposed strong existential quantification semantics create a number of complex theoretical challenges. In particular, it can be necessary for a model to introduce multiple new constants per ground rule to satisfy an existential quantification in certain situations as illustrated by the following example.

**Example 2.** Consider database $\tau$ with $\tau(S(a)) = 0.8, \tau(T(a)) = 0.2$ and program $\Pi$

$$\begin{array}{ll} S(x) & \rightarrow \exists y \; P(x, y) \\ P(x, y) & \rightarrow T(x) \end{array}$$

Then there is no 1-fuzzy model with only one new constant. However, there is a solution where the facts $P(a, n_1), P(a, n_2), P(a, n_3), P(a, n_4)$, with new constants $n_1, \ldots, n_4$, are all assigned truth degree $0.2$.

We focus our development on desired behaviour for practical applications that avoids some of the theoretical corner cases. We formalise this motivation via the notion of *preferred models*, which are intended to capture the desired output of an MV-Datalog$^\pm$ system in a way that is practically meaningful while also admitting good computational properties. Intuitively, a preferred model of $\Pi, \tau$ has two vital properties. First, the model must relate to some execution of the oblivious chase procedure on $\Pi^{crisp}, D_\tau$ (we say the model has an *oblivious base*). This restricts the scope of the unusual behaviour illustrated in Example 2. Second, a preferred model is minimal with respect to some partial ordering of all models with an oblivious base. Looking back at Example 2 again, it is easy to see that plain minimality of models is no longer a useful concept as we can arbitrarily "shift" truth between any of the four facts $P(a, n_i)$, e.g., have three of them be true to degree $0.1$ and the fourth with truth $0.5$. The partial order used in the definition instead orders the models by truth assignments in some "core" part of the model.

Naturally then, an MV$^\pm$-instance does not necessarily have a unique preferred model. However, analogous to minimal models in MV-Datalog we are still able to characterise them via linear programs. That is, for MV$^\pm$-instance $\Pi, \tau$ we show how to construct a linear program $\exists\text{-}Opt^K_{\Pi,\tau}$, extending the ideas from the previous construction of $Opt^K_{\Pi,\tau}$. Note that any of the preferred models is a sensible representative solution that can be used to fact entailment for instances that are satisfiable by a model with an oblivious base.

**Theorem 4.** *Let $\Pi$, $\tau$ be a MV-Datalog program and database where $OLim(\Pi^{crisp}, D_\tau)$ is finite. The following two statements hold.*

1. *If $(\Pi, \tau)$ has a preferred $K$-fuzzy model, then $\exists\text{-}Opt^K_{\Pi,\tau}$ is feasible.*
2. *For any optimal solution $\mathbf{x}$ of $\exists\text{-}Opt^K_{\Pi,\tau}$, we have that $\nu_\mathbf{x}$ is a preferred $K$-fuzzy model for $(\Pi, \tau)$.*

## 5 Conclusion & Future Work

We introduced MV-Datalog and MV-Datalog$^\pm$ as generalisations of classical Datalog and Datalog$^\pm$ to the semantics of infinite-valued Łukasiewicz logic. We propose semantics for logic programming for both of these formalisms that are motivated by applications for interfacing classical data with data from ML systems and crowd-sourcing. Reconsidering many-valued generalisations of Datalog$^\pm$ with traditional semantics for existential quantification is an interesting next step for future work. Such semantics would align with the most prominent fuzzy DLs and it is of natural interest whether a corresponding decidable fuzzy Datalog$^\pm$ language can capture such fuzzy DLs, analogous to how DL-Lite and $\mathcal{EL}$ are captured by guarded Datalog$^\pm$. Initial work in this direction has been outlined in recent follow-up work [Lanzinger *et al.*, 2022b].

## Acknowledgements

Stefano Sferrazza was supported by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18013]. Georg Gottlob is a Royal Society Research Professor and acknowledges support by the Royal Society in this role through the "RAISON DATA" project (Reference No. RP\R1\201074). Matthias Lanzinger and Stefano Sferrazza acknowledge support by the Royal Society "RAISON DATA" project (Reference No. RP\R1\201074). For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

## References

[Achs and Kiss, 1995] Ágnes Achs and Attila Kiss. Fuzzy extension of datalog. *Acta Cybern.*, 12(2):153–166, 1995.

[Bach *et al.*, 2017] Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-Loss Markov Random Fields and Probabilistic Soft Logic. *J. Mach. Learn. Res.*, 18:109:1–109:67, 2017.

[Bellomarini *et al.*, 2017] Luigi Bellomarini, Georg Gottlob, Andreas Pieris, and Emanuel Sallinger. Swift logic for big data and knowledge graphs. In *Proc. IJCAI*, pages 2–10. ijcai.org, 2017.

[Bellomarini *et al.*, 2020] Luigi Bellomarini, Eleonora Laurenza, Emanuel Sallinger, and Evgeny Sherkhonov. Reasoning Under Uncertainty in Knowledge Graphs. In *RuleML+RR*, pages 131–139. Springer, 2020.

[Bobillo and Straccia, 2016] Fernando Bobillo and Umberto Straccia. The fuzzy ontology reasoner fuzzydl. *Knowl. Based Syst.*, 95:12–34, 2016.

[Borgwardt and Peñaloza, 2017] Stefan Borgwardt and Rafael Peñaloza. Fuzzy description logics - A survey. In *Proc. SUM*, volume 10564 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2017.

[Calì *et al.*, 2013] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *JAIR*, 48:115–174, 2013.

[Cavallo and Pittarelli, 1987] Roger Cavallo and Michael Pittarelli. The theory of probabilistic databases. In *VLDB'87*, pages 71–81. Morgan Kaufmann, 1987.

[Ebrahim, 2001] Rafee Ebrahim. Fuzzy logic programming. *Fuzzy Sets Syst.*, 117(2):215–230, 2001.

[Fjellstad and Olsen, 2021] Andreas Fjellstad and Jan-Fredrik Olsen. $\mathrm{IKT}^{\omega}$ and Łukasiewicz-Models. *Notre Dame Journal of Formal Logic*, 62(2):247 – 256, 2021.

[Hájek, 1998] Petr Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer, 1998.

[Lanzinger *et al.*, 2022a] Matthias Lanzinger, Stefano Sferrazza, and Georg Gottlob. Mv-datalog+-: Effective rule-based reasoning with uncertain observations. *Theory Pract. Log. Program.*, 22(5):678–692, 2022.

[Lanzinger *et al.*, 2022b] Matthias Lanzinger, Stefano Sferrazza, and Georg Gottlob. New perspectives for fuzzy datalog (extended abstract). In *Proce. Datalog-2.0*, volume 3203 of *CEUR Workshop Proceedings*, pages 42–47. CEUR-WS.org, 2022.

[Lee *et al.*, 2017] Han S. Lee, Heechul Jung, Alex A. Agarwal, and Junmo Kim. Can deep neural networks match the related objects?: A survey on imagenet-trained classification models. *CoRR*, abs/1709.03806, 2017.

[Lukasiewicz and Straccia, 2008] Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *J. Web Semant.*, 6(4):291–308, 2008.

[Maier *et al.*, 2018] David Maier, K. Tuncay Tekle, Michael Kifer, and David Scott Warren. Datalog: concepts, history, and outlook. In *Declarative Logic Programming: Theory, Systems, and Applications*. ACM / Morgan & Claypool, 2018.

[Mishkin *et al.*, 2017] Dmytro Mishkin, Nikolay Sergievskiy, and Jiri Matas. Systematic evaluation of convolution neural network advances on the imagenet. *Comput. Vis. Image Underst.*, 161:11–19, 2017.

[Preining, 2010] Norbert Preining. Gödel logics - A survey. In *Proc. LPAR*, volume 6397 of *Lecture Notes in Computer Science*, pages 30–51. Springer, 2010.

[Raedt *et al.*, 2007] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proc. IJCAI*, pages 2462–2467, 2007.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro M Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

[Stoilos *et al.*, 2007] Giorgos Stoilos, Giorgos B. Stamou, Jeff Z. Pan, Vassilis Tzouvaras, and Ian Horrocks. Reasoning with very expressive fuzzy description logics. *J. Artif. Intell. Res.*, 30:273–320, 2007.

[Suciu *et al.*, 2011] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[Vojtás, 2001] Peter Vojtás. Fuzzy logic programming. *Fuzzy Sets Syst.*, 124(3):361–370, 2001.