
DEFINING NEUROSYMBOLIC AI

Lennert De Smet

Department of Computer Science,
KU Leuven, Belgium
lennert.desmet@kuleuven.be

Luc De Raedt

Department of Computer Science,
KU Leuven, Belgium
and Örebro University, Sweden
luc.deraedt@kuleuven.be

ABSTRACT

Neurosymbolic AI focuses on integrating learning and reasoning, in particular, on unifying logical and neural representations. Despite the existence of an alphabet soup of neurosymbolic AI systems, the field is lacking a generally accepted formal definition of what neurosymbolic models and inference really are. We introduce a formal definition for neurosymbolic AI that makes abstraction of its key ingredients. More specifically, we define neurosymbolic inference as the computation of an integral over a product of a logical and a belief function. We show that our neurosymbolic AI definition makes abstraction of key representative neurosymbolic AI systems.

1 Introduction

Neurosymbolic AI (NeSy) combines more traditional symbolic AI techniques with the latest advances in deep learning. It integrates neural and numerical data processing with symbolic reasoning and background knowledge. The advantages of this combination have already been demonstrated in applications such as providing safety guarantees [Yang et al., 2023], learning from distant signals and supervision by deduction [Augustine et al., 2022], and more [DeLong et al., 2023, Jiao et al., 2024, Sun and Shoukry, 2024]. Neurosymbolic AI is attracting a lot of attention [Hochreiter, 2022, Besold et al., 2017, Garcez and Lamb, 2023, De Raedt et al., 2020, van Harmelen and ten Teije, 2019, d’Avila Garcez et al., 2019, Bader and Hitzler, 2005, van Bekkum et al., 2021, Dash et al., 2021] and has been termed “*the most promising approach to a broad AI*” by Hochreiter [Hochreiter, 2022] and the “*3rd wave in AI*” by Garcez and Lamb [Garcez and Lamb, 2023]. It is mentioned as an innovation trigger on Gartner’s hype cycle¹ and there are now dedicated journals², conferences³, and summer schools⁴ devoted to neurosymbolic AI.

Despite the wide interest, the term neurosymbolic AI is used for many different types of integrations of neural and symbolic AI systems. For instance, Henry Kautz describes six different types of such integrations [Kautz, 2022], some requiring tighter interfaces between the neural and the symbolic component, others looser ones. As Garcez and Lamb [Garcez and Lamb, 2023], we will focus on the original and stricter interpretation of the term neurosymbolic AI, which Garcez and Lamb describe as “*research that integrates in a principled way neural network-based learning with symbolic knowledge representation and logical reasoning*”. This is also the dominant view in the Neurosymbolic AI Journal and Conference. Within this view there exist numerous models, systems and techniques that integrate logic with neural network-based approaches, see for instance [Marra et al., 2024, Garcez et al., 2022, Besold et al., 2017, Hitzler and Sarker, 2021] for overviews. However, the focus in the field is very much on designing bespoke systems that score best on the latest benchmarks, which results in an alphabet soup of systems. This comes at the expense of understanding the underlying principles and commonalities that these systems share, which is hindering progress in the field. What is lacking is a commonly agreed formal definition and framework for specifying, comparing and developing neurosymbolic AI models and problems. It is precisely this gap that this paper wants to bridge.

¹<https://www.gartner.com/en/articles/hype-cycle-for-artificial-intelligence>

²<https://neurosymbolic-ai-journal.com/>

³<https://2025.nesyconf.org/>

⁴<https://neurosymbolic.github.io/nsss2024/>

More specifically, we contribute formal definitions of a neurosymbolic model and neurosymbolic inference. These definitions are based on the observation that the vast majority of neurosymbolic AI models combine *logic* with *beliefs*. The term logic refers to the wide variety of logics that are used in neurosymbolic AI ranging from Boolean logic to first-order and fuzzy logic, as well as their combinations. The term belief refers to a weighting component used by many neurosymbolic AI models [Manhaeve et al., 2021, Winters et al., 2022a, Pryor et al., 2022, Yang et al., 2020] often derived from statistical relational AI models [De Raedt et al., 2016]. Within our framework, neurosymbolic inference can be formally defined as aggregation (or marginalisation) of a product of a logic and a belief function. Our definitions provide a semantic framework for NeSy models that clarifies their components and the way they interact. We will show that many representative classes of NeSy models and tasks, including those based on probabilistic logic [Manhaeve et al., 2021, Yang et al., 2020], fuzzy logic [Badreddine et al., 2022], or soft logic [Pryor et al., 2022], can be cast within our definitions by instantiating the logic, the belief function and the neural networks appropriately. As a consequence, our definitions can be used to relate, compare and develop different NeSy models in a principled manner, as well as to study fundamental properties of NeSy models and tasks.

2 Logic as the symbol level

In Garcez and Lamb’s perspective on neurosymbolic AI, the symbol level is viewed as symbolic knowledge representation and logical reasoning. We will therefore focus on using logical languages, although the proposed definitions in principle apply to other formal languages and automata [De Giacomo and Favorito, 2021, Manginas et al., 2024]. We will also allow for a wide range of semantics for these languages, in order to support Boolean, fuzzy and other logics.

More formally, a language L is a set of sentences over an ordered set $\mathcal{S} = \{s_i\}_{i \in I}$ of symbols s_i with domains $D_i \subseteq \mathbb{D}$ that can be embedded in a shared domain \mathbb{D} and interact with operators \mathcal{O} .⁵ Symbols are assigned values by an interpretation $\omega : \mathcal{S} \rightarrow \mathbb{D}$ and we use $\Omega = \mathbb{D}^{\mathcal{S}}$ to denote the set of all possible interpretations over the symbols \mathcal{S} . Given an interpretation $\omega \in \Omega$ and a sentence $\varphi \in L$, the *semantics* $\mu : L \times \Omega \rightarrow V$ of the language L maps φ to a semantic value $\mu(\varphi, \omega)$ in the interpretation ω , which we will often denote as $\varphi(\omega)$ for brevity. The set V of semantic values is assumed to be embeddable in \mathbb{R}^+ .⁶ If a symbol s_i has a domain D_i that is equal to the set V of semantic values, then it will be called an *atom symbol* and the set of all atom symbols will be denoted as \mathcal{A} . Atom symbols are special as they are directly assigned a meaningful value in an interpretation.

We illustrate the different choices of language and semantics using both fuzzy [Zadeh, 1965, Ruspini, 1991] and Boolean semantics for propositional logic and SMT logic sentences.

Example 2.1 (Boolean propositional logic). Propositional logic is the language that consists of sentences over symbols \mathcal{S} called *propositions* that can be connected with logical operators such as \Rightarrow and \vee . For example, the sentence φ equivalent to

$$\text{happy} \Rightarrow (\text{coffee} \vee \text{publication}), \quad (2.1)$$

states that happiness h is only possible when having either a coffee c or a publication p . Propositional logic can be equipped with a Boolean semantics by using $\mathbb{D} = \{0, 1\}$ as domain for all propositions denoting false and true and considering the interpretations ω as mappings from \mathcal{S} to $V = \{0, 1\}$. The Boolean semantics μ_B of propositional logic formulae then follows inductively. For instance, the above formula φ evaluates to $\varphi_B(\omega) = 1$ for the interpretation

$$\omega(s) = \begin{cases} 1 & \text{if } s \in \{h, c\}, \\ 0 & \text{otherwise.} \end{cases} \quad (2.2)$$

Example 2.2 (Fuzzy propositional logic). Propositional logic can also be equipped with a fuzzy semantics by setting the sets V and \mathbb{D} to the real unit interval $[0, 1]$ and using fuzzy operators such as the continuous T-norms [Gupta and Qi, 1991] to inductively define its semantics μ_F .

For instance, if one uses Łukasiewicz fuzzy logic [Łukasiewicz and Tarski, 1956] with the T-conorm $\min(1, x + y)$ for the disjunction and the negation $1 - x$, the formula φ (Equation 2.1) evaluates to $\varphi_F(\omega) = \min(1, 1 - \omega(h) + \min(1, \omega(c) + \omega(p))) = 1$ for the interpretation

$$\omega(s) = \begin{cases} 0.5 & \text{if } s \in \{c, p\}, \\ 1 & \text{otherwise.} \end{cases} \quad (2.3)$$

That is, having a fuzzy feeling of 0.5 for having a coffee and having a publication is enough to satisfy φ and allow for happiness, while Boolean semantics require absolute certainty on having a coffee or a publication.

⁵The assumption of having a shared domain is made to simplify notation and holds without loss of generality.

⁶This assumption also simplifies notation and is not a hard one, e.g. it is common to represent Boolean truth values \top and falsehood \perp as 1 and 0.

Example 2.3 (Linear SMT logic). The language of linear SMT logic is comprised of sentences over symbols that appear in linear arithmetic operators to form linear arithmetic comparisons that can be connected with logical operators. For example, one can rewrite the propositional sentence of Equation 2.1 as the linear SMT formula

$$h = 1 \Rightarrow (c + p \geq 0), \quad (2.4)$$

where h , c and p are symbols with domain $\{-1, 1\}$. SMT logic can also be given a Boolean semantics by mapping arithmetic comparisons to the set $V = \{0, 1\}$ by choosing 1 for interpretations, i.e. assignments of symbols, that satisfy the comparison according to arithmetic and 0 otherwise.

Numerous other logics exist, such as first-order and temporal logics [De Giacomo et al., 2013], and their semantics can also be defined in terms of assigning values to symbols or sequences of symbols (for temporal logics). Similar analyses can be made for automata [Vardi, 2005, Manginas et al., 2024] and other formal languages [Sun and Shoukry, 2024].

Logical inference can be viewed as inferring whether there are interpretations that satisfy certain constraints w.r.t. their semantic values. For instance, SAT-solvers address the question whether there exists an interpretation ω that satisfies a formula φ , i.e. that satisfies $\varphi_B(\omega) = 1$. In fuzzy logic, one might be interested in considering only those interpretations ω that satisfy $\varphi_F(\omega) > \tau$. For these reasons, it will be convenient to introduce logic functions.

Definition 2.4. (Logic function) A *logic function* l is a function that takes a formula φ and interpretation ω and returns a non-zero semantic value only when the value of $\varphi(\omega)$ is contained in a desired subset V_l of the semantic values V . That is, it is any function $l : L \times \Omega \rightarrow V$ with $l(\varphi, \omega) = 0$ if $\varphi(\omega) \notin V_l$. We call the set V_l the *selection values* of the logic function.

The intuition behind a logic function is that it outputs desired semantic values for selected interpretations of interest based on a logical formula. For most NeSy AI systems, especially those based on Boolean logic, $l(\varphi, \omega) = \varphi(\omega)$. It is only when working with both thresholds and fuzzy logic that more complex logic functions might be necessary, such as $l(\varphi_F, \omega) = [\varphi_F(\omega) > \tau]$, where $[\cdot]$ denotes the Iverson brackets that evaluate to 1 if its argument evaluates to true, and yields the value 0 otherwise.

3 Towards Neurosymbolic Models and Inference

Neurosymbolic AI models extend logic with neural networks and are, as we will show, typically based on two components: a (possibly fuzzy) logic and a (possibly probabilistic) neural belief [Marra et al., 2024]. While the semantic value $\varphi(\omega)$ of a logical formula in an interpretation is captured by the semantic function $\mu(\varphi, \omega)$, the belief component can be captured by a *belief function* $b_\theta(\varphi, \omega)$. The belief $b_\theta(\varphi, \omega)$ can be interpreted as the weight indicating the degree of belief that the interpretation ω satisfies the formula φ . It generally takes the form of a parametrised function $b_\theta : L \times \Omega \rightarrow \mathbb{R}$ with parameters θ that takes an assignment of symbols and a sentence and outputs the belief. This belief is often probabilistic or neurally implemented, which means that θ are the parameters of a neural network or a graphical model

Example 3.1 (Parametrising a Boolean propositional sentence). Consider again the sentence $h \Rightarrow (c \vee p)$ from Example 2.1. Now assume there is a neural network that takes as input an image taken from a camera in your local mathematics department. As output, the network with parameters θ returns three probabilities, one probability $p_{\theta,s}$ for each symbol $s \in \{h, c, p\}$. Together, assuming h , c and p are independent, these probabilities can be combined into a simple probabilistic belief function $b_\theta(\varphi, \omega)$ by taking the product of the probabilities, i.e.

$$b_\theta(\varphi, \omega) = \prod_{s \in \{h, c, p\}} p_{\theta,s}^{\omega(s)} \cdot (1 - p_{\theta,s})^{1-\omega(s)}. \quad (3.1)$$

Definition 3.2 (Neurosymbolic model). A neurosymbolic AI model $(L, \mu, \Omega, b_\theta)$ consists of a logical language L with a semantics μ over interpretations Ω and a belief function b_θ with parameters θ .

Neurosymbolic AI models are used to perform inference. We view inference in a neurosymbolic model $(L, \mu, \Omega, b_\theta)$ as computing the integral graphically illustrated in Figure 1. Neurosymbolic inference can be formally defined through neurosymbolic functionals via measure theory and Lebesgue integration (Appendix A).

Definition 3.3 (Neurosymbolic Inference). Given a neurosymbolic model $(L, \mu, \Omega, b_\theta)$, a logic function l and a measure space $(\Omega, \Sigma_\Omega, m)$, neurosymbolic inference is defined as computing the result of the following *neurosymbolic functional*

$$F_\theta(\varphi) = \int_{\Omega'} l(\varphi, \omega) b_\theta(\varphi, \omega) dm(\omega), \quad (3.2)$$

where $\Omega' \subseteq \Omega$ is a subset of all possible interpretations determined by a subset of the symbols of L .

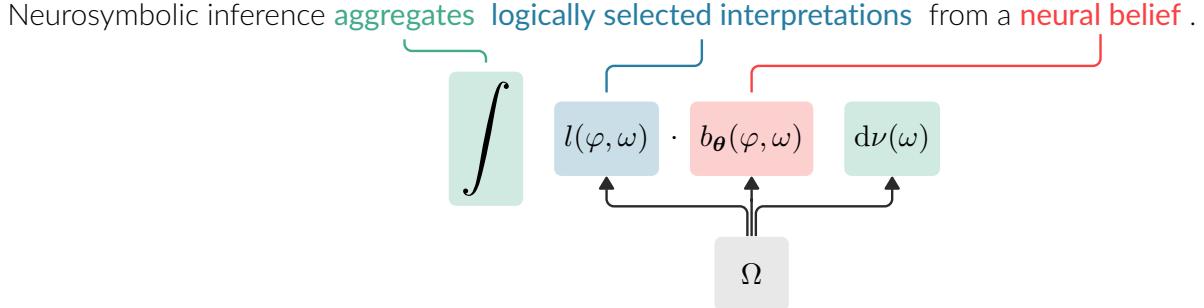


Figure 1: The main intuition behind neurosymbolic inference. Note how the interpretations Ω of the logical language form the interface between neural and symbolic components.

The technical intuition for using Lebesgue integration and measures is given in the next paragraph, which can be skipped for the less technically inclined.

Lebesgue integration generalises the usual Riemannian integration to domains that are not real vector spaces, such as the space of interpretations Ω , by using *measures*. A measure m does nothing else than define a notion of size or volume by taking a set S as input and returning a value $m(S) \in \mathbb{R}$ indicating how large that set is. Such a notion of volume then forms the basis for integration as integration intuitively aggregates function values over infinitesimally small pieces of volume. For instance, the usual Riemann integral $\int_{\mathbb{R}} f(x) dx$ over the real line \mathbb{R} aggregates function values $f(x)$ over infinitesimally small intervals, so the length of the interval can be seen as the measure or notion of volume of Riemann integration. Since the Riemann integral is defined using volumes, it uses the differential notation dx to denote infinitesimal pieces of volume or measurements. In contrast, the Lebesgue integral makes the measure m explicit by using the notation $dm(x)$ instead (Equation 3.2).

This definition of neurosymbolic inference generalises weighted model counting (WMC) [Chavira and Darwiche, 2008] and weighted model integration (WMI) [Belle et al., 2015] beyond purely probabilistic neurosymbolic models [De Smet et al., 2023] through the generality of measure theory and Lebesgue integration. WMC and WMI are recovered from Equation 3.2 by having a finite space of interpretations Ω with a counting measure (WMC) or an infinite space of interpretations with a measure that combines counting with Riemann integration (WMI).

Importantly, Definition 3.3 leads to precise conditions under which neurosymbolic inference is well-defined.

Proposition 3.4 (Well-defined neurosymbolic inference). Let $(L, \mu, \Omega, b_\theta)$ be a neurosymbolic AI model, l a logic function and $(\Omega, \Sigma_\Omega, dm)$ a measure space for which the logic function l and belief function b_θ are measurable, then neurosymbolic inference is well-defined.

The choice of how to parametrise the belief b_θ is completely free and does not necessarily have to involve neural networks. In fact, if one foregoes the use of neural networks and considers probabilistic beliefs, then our definition of a neurosymbolic model reduces to a definition for statistical relational AI (StarAI) [De Raedt et al., 2016] models. Consequently, our view on neurosymbolic inference provides a formal framework for inference in StarAI as well. Such a framework for StarAI was also missing, which shows the utility of a unifying and formal definition.

The question of how to perform learning from the quantities inferred by a neurosymbolic functional can be answered in many different ways. In settings where the belief is parametrised by neural networks, one can define a loss function in terms of the neurosymbolic functional and learn via backpropagation of this loss. In other settings, such as in StarAI, different approaches to learning like expectation maximisation can also be used. In general, our framework does not impose any restrictions on how to perform learning. We only define the semantics of NeSy models and of inference, not of learning, as is usual when defining semantics.

4 Neurosymbolic inference unifies neurosymbolic AI

We will now show that our definitions of neurosymbolic models and inference unifies many prominent neurosymbolic AI frameworks. These frameworks can be characterised based on their language, semantics and parametrisation (Table 1). We will mainly separate systems based on their semantics.

Table 1: How to recover inference in popular neurosymbolic frameworks as neurosymbolic inference following Definition 3.3. (B) indicates the semantics is Boolean in the sense that they are based on true and false values while (F) similarly indicates fuzzy semantics. A semantics indicated with (N) is a semantics that uses neural networks to compute more procedural semantics. Belief parametrisations can take many forms, but we specifically indicate whether they are deep (D), probabilistic (P) or based on probabilistic circuits (PC) [Choi et al., 2020].

System	Language	Semantics	Belief function	Logic function
α ILP [Shindo et al., 2023]	Logic programs	Stable models (B)	D + P	Boolean satisfaction
δ ILP [Law et al., 2018]	Logic programs	Fuzzy	Point prediction (D)	Fuzzy satisfaction
II-NeSy [Baaif and Marquis, 2025]	If-then rules	Preferential entailment [Dubois et al., 1991]	Possiblestic (D)	Preferential satisfaction
DeepProbLog [Manhaeve et al., 2018]	Logic programs	Well-founded (B)	D + P	Boolean satisfaction
DeepSeaProbLog [De Smet et al., 2023]	Distributional logic programs	Measure semantics [Dos Martires et al., 2024] (B)	D + P	Boolean satisfaction
DeepSoftLog [Maene and De Raedt, 2023]	Logic programs + embeddings	Well-founded (B)	D + P	Boolean satisfaction
DeepStochLog [Winters et al., 2022b]	Definite clause grammars	Stochastic (F)	D + P	Fuzzy satisfaction
DL2 [Fischer et al., 2019]	SMT logic	Fuzzy semantics (F)	Point prediction (D)	Fuzzy satisfaction
DLM [Marra et al., 2019]	First-order logic	Boolean (B)	D + P	Boolean satisfaction
DRL [Stoian and Giunchiglia, 2025]	QFLRA	Boolean (B)	D + P	Boolean satisfaction
LRNN [Sorek et al., 2018]	Definite clause logic	Fuzzy (F)	Embedded formula (D)	Fuzzy satisfaction
LTN [Serafini and Garcez, 2016]	First-order logic	Real logic (F)	Embedded formula (D)	Fuzzy satisfaction
NeuPSL [Pryor et al., 2022]	Logic programs	Lukasiewicz (F)	D + P	Fuzzy satisfaction
Neural LP [Yang et al., 2017]	Polytree DKGs	Stochastic (F)	D + P	Fuzzy satisfaction
NeurASP [Yang et al., 2020]	Logic programs	Stable models (B)	D + P	Boolean satisfaction
NLM [Dong et al., 2019]	First-order rules	Neural (N)	Embedded formula (D)	Function evaluation
NLog [Tsamoura et al., 2021]	Logic programs	Boolean (B)	D + P	Boolean satisfaction
NLProlog [Weber et al., 2019]	Horn clauses	Fuzzy (F)	D	Fuzzy satisfaction
NMLN [Marra and Kuželka, 2021]	First-order logic	Boolean (B)	D + P	Boolean satisfaction
NTP [Rocktäschel and Riedel, 2017]	Function-free first-order rules	Neural (N)	Embedded formula (D)	Function evaluation
SBR [Diligenti et al., 2017]	First-order logic	Fuzzy (F)	Point prediction (D)	Fuzzy satisfaction
Scallop [Li et al., 2023]	Logic programs	Provenance (B)	D + P	Algebraic satisfaction
Semantic Loss [Xu et al., 2018]	Propositional logic	Boolean (B)	D + P	Boolean satisfaction
SLASH [Skryagin et al., 2022]	Logic programs	Stable models (B)	D + P + PC	Boolean satisfaction
SPL [Ahmed et al., 2022]	Propositional logic	Boolean (B)	PC	Boolean satisfaction
TensorLog [Cohen et al., 2020]	Polytree DKGs	Stochastic (F)	D + P	Fuzzy satisfaction

4.1 Neurosymbolic AI with Boolean semantics

Boolean logic is fundamental to computer science and is also the foundation of a series of neurosymbolic systems with a *probabilistic interpretation*, such as DeepProbLog [Manhaeve et al., 2018], Neural Markov Logic Networks (NMLN) [Marra and Kuželka, 2021], Semantic Probabilistic Layers (SPL) [Ahmed et al., 2022] and NeurASP [Yang et al., 2020]. All of these systems differ in their choice of language, parametrisation or implementation of Boolean semantics (Table 1), e.g. logic programs with stable model semantics [Gelfond and Lifschitz, 1988] (NeurASP) or propositional logic with Boolean semantics (SPL). However, they are similar in that they are all based on computing probabilities of sentences being true or false. Returning to our running example, each of the systems can compute the probability of the sentence “happiness is only possible when having coffee or a publication” encoded in their respective languages.

Example 4.1 (Probabilistic Boolean neurosymbolic AI). Assuming an independent factorisation b_θ (Equation 3.1), the probability of the sentence $h \Rightarrow (c \vee p)$ being true is by definition

$$\int_{\mathbb{B}^3} \mu_B(h \Rightarrow (c \vee p), \omega) \cdot \prod_{s \in \{h, c, p\}} p_{\theta, s}^{\omega(s)} \cdot (1 - p_{\theta, s})^{1-\omega(s)} dm_C(h, c, p) \quad (4.1)$$

where m_C is a counting measure over the 8 possible binary interpretations. That is, we have neurosymbolic inference with the Boolean semantics μ_B as logic function and a probabilistic belief function. As the space of interpretations Ω is finite, this expression is equivalent to the WMC instance

$$\sum_{\omega \in \mathcal{M}} \prod_{s \in \{h, c, p\}} p_{\theta, s}^{\omega(s)} \cdot (1 - p_{\theta, s})^{1-\omega(s)}, \quad (4.2)$$

where $\mathcal{M} = \{\omega \mid \mu_B(h \Rightarrow (c \vee p), \omega) = 1\}$ is the set of *models* of the sentence $h \Rightarrow (c \vee p)$. Each of the different Boolean neurosymbolic systems would perform this exact computation, but implemented in their own language and semantics. NMLN would directly encode the sentence in first-order logic while SPL encodes sentences in propositional logic. For DeepProbLog and NeurASP, the sentence $h \Rightarrow (c \vee p)$ would be encoded as a logic program with its corresponding well-founded or stable models semantics.

In general, neurosymbolic systems based on Boolean semantics perform inference according to Definition 3.3.

Claim 4.2. Inference in typical neurosymbolic systems based on Boolean semantics corresponds to neurosymbolic inference of the form

$$\int_{\Omega_B} l(\varphi, \omega) b_\theta(\varphi, \omega) dm_B(\omega), \quad (4.3)$$

where $\Omega_B = \mathbb{B}^{\mathcal{A}} \times \mathbb{D}^{\mathcal{S} \setminus \mathcal{A}}$ and m_B is a combination of binary counting measures and Borel measures.

Argument. We show that this statement holds for DeepProbLog, SPL, NeurASP and NMLNs. The foundational inference task in these systems is computing the probability that a sentence φ is true, i.e.

$$\mathbb{P}(\varphi) = \int_{\Omega_B} \varphi_B(\omega) b_{\theta}(\varphi, \omega) dm_B(\omega). \quad (4.4)$$

Hence, the logic function l for DeepProbLog, SPL, NeurASP and NMLNs is equal to the Boolean value $\varphi_B(\omega)$ of the sentence φ in the interpretation ω . The belief function b_{θ} for all systems has to be a probability distribution, yet the form differs per system. We split up three cases for (1) DeepProbLog and NeurASP, (2) SPL, and (3) NMLNs.

(1) DeepProbLog and NeurASP choose an independently factorising probability distribution as belief. That is, their belief function is

$$b_{\theta}(\varphi, \omega) = \prod_{s \in \mathcal{S}} p_{\theta, s}^{\omega(s)} \cdot (1 - p_{\theta, s})^{1 - \omega(s)}, \quad (4.5)$$

where $p_{\theta, s}$ is the probability that the binary symbol s is true.⁷

(2) SPL allows to parametrise the belief as a conditional probabilistic circuit [Shao et al., 2020] that is *compatible* [Choi et al., 2020] with the logical formula φ .

(3) NMLNs see the sentence φ as a first-order theory $\bigwedge_{i=1}^N \varphi_i$ consisting of N sentences. Their belief function is then constructed as the normalised exponentiated sum

$$b_{\theta}(\varphi, \omega) = \frac{1}{Z} e^{\sum_{i=1}^N \lambda_{\theta, i} \cdot \varphi_{i, B}(\omega)} = \frac{1}{Z} \prod_{i=1}^N e^{\lambda_{\theta, i} \cdot \varphi_{i, B}(\omega)}, \quad (4.6)$$

where each $\lambda_{\theta, i}$ is a parametrised weight and Z is the normalising constant over interpretations Ω . \square

In the probabilistic setting where the belief $b_{\theta}(\varphi, \omega)$ is a probability distribution over the set of interpretations Ω , neurosymbolic inference becomes an instance of either weighted model counting (WMC) [Chavira and Darwiche, 2008] or weighted model integration (WMI) [Belle et al., 2015] depending on whether Ω is finite or infinite.

4.2 Neurosymbolic AI with fuzzy semantics

Fuzzy semantics for logical languages [Ruspini, 1991] has enjoyed a lot of interest as a finer-grained alternative to the traditional Boolean semantics. While Boolean semantics is two-valued based on absolute truth and falsehood, fuzzy semantics is infinite-valued and expresses a degree of truth by mapping symbols and sentences to the real unit interval. For neurosymbolic AI, the real-valued nature of fuzzy semantic values can result in a differentiable notion of satisfiability that makes the integration with neural networks easier. However, it does lead to more diverse computations as different systems can be interested in different restrictions of the fuzzy values of a sentence.

Example 4.3 (Fuzzy neurosymbolic AI). Many fuzzy neurosymbolic systems only compute the fuzzy value of a sentence given a single fuzzy interpretation. For example, Logic Tensor Networks (LTN) proposes an intricate way of parametrising the belief b_{θ} for a single fuzzy interpretation ω_{θ} by mapping symbols and operators to tensors and tensor operations followed by computing fuzzy values of sentences in the interpretation ω_{θ} . In case of our running example with the Łukasiewicz T-norm, the belief b_{θ} of LTN would be a Dirac delta distribution δ that gives a single fuzzy value for h , c and p while ignoring all the other possible fuzzy interpretations in the space $\Omega = [0, 1]^3$. This construction leads to neurosymbolic inference of the form

$$\int_{[0,1]^3} \min(1, 1 - \omega(h) + \min(1, \omega(c) + \omega(p))) \delta(\omega - \omega_{\theta}) d\omega d\omega d\omega = \varphi_F(\omega_{\theta}), \quad (4.7)$$

where now $d\omega$, dc and dp are the usual Borel measure on $[0, 1]$. This expression uses fuzzy evaluation as logic function and collapses to the fuzzy value $\varphi_F(\omega_{\theta})$ because of the Dirac delta distribution δ .

In general, our definition of neurosymbolic inference encompasses inference in typical fuzzy neurosymbolic systems. While covering LTN [Badreddine et al., 2022] and SBR [Diligenti et al., 2017] inference requires rewriting their inferred quantities, the rewrite exposes connections to other fuzzy systems like NeuPSL [Pryor et al., 2022].

⁷Other related systems [De Smet et al., 2023] also allow for categorical or continuous distributions resulting in products of probability mass functions or probability densities.

Claim 4.4. Inference in typical neurosymbolic systems based on fuzzy semantics is neurosymbolic inference of the form

$$\int_{\Omega_F} l(\varphi, \omega) b_{\theta}(\varphi, \omega) dm_F(\omega), \quad (4.8)$$

where $\Omega_F = [0, 1]^{\mathcal{A}} \times \mathbb{D}^{\mathcal{S} \setminus \mathcal{A}}$ and m_F is a Borel measure.

Argument. We prove this claim for the cases of (1) logic tensor networks (LTN) and semantic-based regularisation (SBR), and (2) neural probabilistic soft logic (NeuPSL).

(1) LTN and SBR both compute the fuzzy value $\varphi_F(\omega_{\theta})$ of a sentence φ in a single parametrised fuzzy interpretation ω_{θ} . Only considering a single fuzzy interpretation corresponds to choosing a belief function that is a Dirac delta distribution, i.e. $b_{\theta}(\varphi, \omega) = \delta(\omega - \omega_{\theta})$. Indeed, we can use the collapsing property of the Dirac delta distribution δ to write

$$\varphi_F(\omega_{\theta}) = \int_{\Omega_F} \varphi_F(\omega) \delta(\omega - \omega_{\theta}) d\omega, \quad (4.9)$$

where we used notation $d\omega$ because we can use a traditional Riemann integral. Hence, LTN and SBR both have the fuzzy value $\varphi_F(\omega)$ as logic function and a Dirac delta as belief function.

(2) NeuPSL sets the belief function to be the probability distribution

$$b_{\theta}(\varphi, \omega) = \frac{1}{Z} e^{\sum_{i=1}^N \lambda_{\theta,i} \cdot \varphi_{i,F}(\omega)} = \frac{1}{Z} \prod_{i=1}^N e^{\lambda_{\theta,i} \cdot \varphi_{i,F}(\omega)}, \quad (4.10)$$

similarly to NMLNs, but with fuzzy semantics. Its choice of logic function changes from task to task, though NeuPSL generally computes fuzzy expected values. For instance, the usual expected fuzzy value would use fuzzy satisfaction $\varphi_F(\omega)$ as logic function. \square

Note how NeuPSL combines fuzzy logic with a parametrised probabilistic belief over all fuzzy interpretations.

Example 4.5 (Probabilistic fuzzy neurosymbolic AI). Systems like NeuPSL relax the hard, Boolean semantic values of atomic expressions to soft, fuzzy values in Łukasiewicz logic and define a probability distribution $p(\varphi, \omega)$ over the space of fuzzy interpretations for each formula φ . This construction allows computing *fuzzy expectations*, e.g. the expectation of the fuzzy value of the sentence $h \Rightarrow (c \vee p)$

$$\mathbb{E}_{\omega \sim p(\varphi, \omega)} [\varphi_F(\omega)] = \int_{\Omega_F} \min(1, 1 - \omega(h) + \min(1, \omega(c) + \omega(p))) p(\varphi, \omega) d\omega. \quad (4.11)$$

This quantity can then be used to optimise the fuzzy value of a sentence *in expectation* instead of only relying on a point estimate as LTN or SBR does.

Given the example of NeuPSL, it seems fuzzy neurosymbolic systems can become more expressive by parametrising the continuum of fuzzy interpretations. Indeed, instead of parametrising a probability distribution over all fuzzy interpretations, one can use any other expressive belief function that covers more than one fuzzy interpretation. For instance, to maintain a completely fuzzy approach, one could turn the set Ω_F of fuzzy interpretations itself into a fuzzy set by parametrising a membership function $f_m : \Omega_F \rightarrow [0, 1]$ that corresponds to defining a fuzzy belief function.

4.3 Limitations

Our definition of neurosymbolic models (Definition 3.2) and inference (Definition 3.3) make, as we have shown, abstraction of the alphabet soup in neurosymbolic AI. They have been designed to strike a good balance between generality and mathematical complexity. While certain edge cases exist that do not directly fit Definition 3.3, we believe it would not be too hard to extend our definitions to accommodate them. For instance, maximum a posteriori (MAP) inference, which aims to return the most likely interpretation, would require composing Equation 3.2 with a maximisation operation. Such edge cases can be covered again by considering nesting or compositions of Equation 3.2, but would be somewhat more involved.

For similar reasons we have also not given much attention to the choice or construction of the measure in definition 3.3 as the measure is connected to the type of inference. For example, tasks like MAP can be cast as instances of algebraic model counting (AMC) [Kimmig et al., 2017] that aggregate using general algebraic operations, but the measures used in our definitions only consider the usual algebraic structure on the reals \mathbb{R} . However, this problem can be solved by

using generalised measures [Wang and Klir, 2010, Choquet, 1954, Sugeno, 1972], as in fuzzy measure theory [Zadeh, 1978]. For ease of exposition, we leave a more detailed analysis using generalised measures for future work.

Finally, we want to draw attention to the fact that our definitions do not limit the semantics of a neurosymbolic model to be strictly logical. That is, it is allowed to consider neurosymbolic models with unconstrained or more operational notions of semantics, e.g. NLM [Dong et al., 2019] and NTP [Rocktäschel and Riedel, 2017] (Table 1). This freedom allows us to be inclusive, but opens up the debate on what it means for a system to be really symbolic, which usually requires a logical semantics.

5 Related work

This is not the first attempt to arrive at a synthesis and a framework for neurosymbolic AI. For instance, Odense and Garcez [2025] introduce a semantic framework for *encoding* logics into neural networks. However, their emphasis is on the necessary conditions under which a class of neural networks and logical systems can be said to be semantically equivalent. That is, a neural network can be encoded as a logical theory and the other way around. This is in line with Henry Kautz’s category **Neural_{Symbolic}** to produce a neural network from logical rules. Our semantics instead focuses on making the logic and belief functions explicit while being rather implicit about the neural network architecture, which is more in line with Henry Kautz’s category **Neural | Symbolic** where both the logical and neural components remain and one is not reduced to the other.

Another related work is ULLER [van Krieken et al., 2024], which proposes a unified language for learning and reasoning. It aims at the “frictionless sharing of knowledge” across neurosymbolic systems and is intended as an interface language, or even interface system, for contemporary neurosymbolic AI systems. Unlike our approach, ULLER is built upon a fixed first order logic and assigns a concrete semantics for Boolean, fuzzy and probabilistic instances. In contrast, our framework is not looking for a lingua franca for neurosymbolic AI, but rather focuses on defining a wide variety of neurosymbolic models and inference tasks in a mathematically uniform way.

Other noteworthy approaches include van Bekkum et al. [2021], who show how to combine and visualize specific design patterns of learning and reasoning architectures, Dash et al. [2021] who characterise NeSy systems by input formats and loss functions, Slusarz et al. [2023] and Marra et al. [2024] who devise various dimensions of neurosymbolic and statistical relational AI systems on which our definition builds. While these are important developments that can eventually lead to an extensive taxonomy of neurosymbolic AI systems, our definitions identify the essential concepts of neurosymbolic models and show how these concepts define abstract neurosymbolic inference tasks.

6 Conclusion

Motivated by the wide range of existing neurosymbolic AI models and approaches, which appear quite different on the surface level, we proposed a general and unifying definition of inference in neurosymbolic AI systems that integrate neural networks with logics. In our view, neurosymbolic inference consists of computing an integral over a product of a logic function and a belief function. We provided evidence that our framework is general in that it makes abstraction of prominent contemporary systems such as LTNs, NeuPSL, SBR, SPL, DeepProbLog, NeurASP and NMLNs.

We believe that our definition will be useful for developing both the theory of neurosymbolic AI by providing a computational framework for designing, evaluating and comparing different neurosymbolic AI systems and tasks, and for studying their computational and mathematical properties. We also believe it will be useful for developing an operational framework and system in which many existing neurosymbolic AI systems can be emulated.⁸⁹

Acknowledgements

This work project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant agreement No. 101142702), the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme, the Flemish research foundation (FWO) project “Neurosymbolic AI and Constraint Learning” (Project G047124N) and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors would also like to thank the DeepLog team (David Debot, Gabriele Venturato, Giuseppe Marra, Jaron Maene, Lucas Van Praet, Pedro Zuidberg Dos Martires, Rik Adriaensen, Robin Manhaeve, Stefano Colamondo and Vincent Derkinderen) for the many interesting discussions and feedback on earlier drafts of the paper.

⁸<https://research.kuleuven.be/EU/p/he/p1/erc/deeplog>

⁹<https://wms.cs.kuleuven.be/cs/onderzoek/deeplog>

References

- Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Semantic probabilistic layers for neuro-symbolic learning. *Advances in Neural Information Processing Systems*, 35:29944–29959, 2022.
- Eriq Augustine, Connor Pryor, Charles Dickens, Jay Pujara, William Wang, and Lise Getoor. Visual sudoku puzzle classification: A suite of collective neuro-symbolic tasks. In *International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*, 2022.
- Ismaïl Baaj and Pierre Marquis. π -nesy: A possibilistic neuro-symbolic approach. *arXiv preprint arXiv:2504.07055*, 2025.
- Sebastian Bader and Pascal Hitzler. Dimensions of neural-symbolic integration - A structured survey. *CoRR*, abs/0511042, 2005.
- Samy Badreddine, Artur d’Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.
- Vaishak Belle, Andrea Passerini, Guy Van den Broeck, et al. Probabilistic inference in hybrid domains by weighted model integration. In *Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2770–2776. AAAI Press/International Joint Conferences on Artificial Intelligence, 2015.
- Tarek R. Besold, Artur S. d’Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation. *CoRR*, abs/1711.03902, 2017.
- Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6–7):772–799, 2008.
- Y Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA*. URL: <http://starai.cs.ucla.edu/papers/ProbCirc20.pdf>, page 6, 2020.
- Gustave Choquet. Theory of capacities. In *Annales de l’institut Fourier*, volume 5, pages 131–295, 1954.
- William Cohen, Fan Yang, and Kathryn Rivard Mazaitis. Tensorlog: A probabilistic database implemented using deep-learning infrastructure. *Journal of Artificial Intelligence Research*, 67:285–325, 2020.
- Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. How to tell deep neural networks what we know. *arXiv preprint arXiv:2107.10295*, 2021.
- Artur S. d’Avila Garcez, Marco Gori, Luís C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *FLAP*, 6, 2019.
- Giuseppe De Giacomo and Marco Favorito. Compositional approach to translate ltlf/ldlf into deterministic finite automata. In *Proceedings of the international conference on automated planning and scheduling*, volume 31, pages 122–130, 2021.
- Giuseppe De Giacomo, Moshe Y Vardi, et al. Linear temporal logic and linear dynamic logic on finite traces. In *Ijcai*, volume 13, pages 854–860, 2013.
- Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Morgan & Claypool Publishers, 2016. doi: 10.2200/S00692ED1V01Y201601AIM032. URL <https://doi.org/10.2200/S00692ED1V01Y201601AIM032>.
- Luc De Raedt, Sebastijan Dumancić, Robin Manhaeve, and Giuseppe Marra. From statistical relational to neuro-symbolic artificial intelligence. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4943–4950. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/688. URL <https://doi.org/10.24963/ijcai.2020/688>. Survey track.
- Lennert De Smet, Pedro Zuidberg Dos Martires, Robin Manhaeve, Giuseppe Marra, Angelika Kimmig, and Luc De Raedt. Neural probabilistic logic programming in discrete-continuous domains. *UAI*, 2023.
- Lauren Nicole DeLong, Ramon Fernández Mir, Zonglin Ji, Fiona Niamh Coulter Smith, and Jacques D Fleuriot. Neurosymbolic ai for reasoning on biomedical knowledge graphs. *arXiv preprint arXiv:2307.08411*, 2023.
- Michelangelo Diligenti, Marco Gori, and Claudio Saccà. Semantic-based regularization for learning and inference. *Artif. Intell.*, 244, 2017.
- Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. Neural logic machines. In *ICLR*, 2019. URL <https://openreview.net/forum?id=B1xY-hRctX>.

- Pedro Zuidberg Dos Martires, Luc De Raedt, and Angelika Kimmig. Declarative probabilistic logic programming in discrete-continuous domains. *Artificial Intelligence*, 337:104227, 2024.
- Didier Dubois, Jérôme Lang, and Henri Prade. A brief overview of possibilistic logic. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 53–57. Springer, 1991.
- Marc Fischer, Mislav Balunovic, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin Vechev. DL2: training and querying neural networks with logic. In *International Conference on Machine Learning*, pages 1931–1941. PMLR, 2019.
- Artur d’Avila Garcez and Luís C. Lamb. Neurosymbolic AI: The 3rd Wave. *Artificial Intelligence Review*, 2023. doi: 10.1007/s10462-023-10448-w. URL <https://doi.org/10.1007/s10462-023-10448-w>.
- Artur d’Avila Garcez, Sebastian Bader, Howard Bowman, Luis C Lamb, Leo de Penning, BV Illuminoo, Hoifung Poon, and COPPE Gerson Zaverucha. Neural-symbolic learning and reasoning: a survey and interpretation. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 2022.
- Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, volume 88, pages 1070–1080. Cambridge, MA, 1988.
- Madan M Gupta and J11043360726 Qi. Theory of t-norms and fuzzy inference methods. *Fuzzy sets and systems*, 40(3): 431–450, 1991.
- Pascal Hitzler and Md. Kamruzzaman Sarker, editors. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, volume 342 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2021. ISBN 978-1-64368-244-0. doi: 10.3233/FAIA342. URL <https://doi.org/10.3233/FAIA342>.
- Sepp Hochreiter. Toward a broad AI. *Communications of the ACM*, 65(4):56–57, 2022.
- Ying Jiao, Luc De Raedt, and Giuseppe Marra. Valid text-to-sql generation with unification-based deepstochlog. In *International Conference on Neural-Symbolic Learning and Reasoning*, pages 312–330. Springer, 2024.
- Henry A. Kautz. The third AI summer: AAAI Robert S. Engelmore memorial lecture. *AI Mag.*, 43(1):93–104, 2022. doi: 10.1609/aimag.v43i1.19122. URL <https://doi.org/10.1609/aimag.v43i1.19122>.
- Angelika Kimmig, Guy Van den Broeck, and Luc De Raedt. Algebraic model counting. *Journal of Applied Logic*, 22: 46–62, July 2017. ISSN 15708683. doi: 10.1016/j.jal.2016.11.031. URL <https://linkinghub.elsevier.com/retrieve/pii/S157086831630088X>.
- Mark Law, Alessandra Russo, and Krysia Broda. Inductive learning of answer set programs from noisy examples. *CoRR*, abs/1808.08441, 2018. URL <http://arxiv.org/abs/1808.08441>.
- Ziyang Li, Jiani Huang, and Mayur Naik. Scallop: A language for neurosymbolic programming. *Proceedings of the ACM on Programming Languages*, 7(PLDI):1463–1487, 2023.
- Jan Lukasiewicz and Alfred Tarski. Investigations into the sentential calculus. *Logic, semantics, metamathematics*, pages 38–59, 1956.
- Jaron Maene and Luc De Raedt. Soft-unification in deep probabilistic logic. *Advances in Neural Information Processing Systems*, 36:60804–60820, 2023.
- Nikolaos Manginas, George Paliouras, and Luc De Raedt. Nesy: Neurosymbolic automata. *arXiv preprint arXiv:2412.07331*, 2024.
- Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: neural probabilistic logic programming. In *NeurIPS*, 2018.
- Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Neural probabilistic logic programming in DeepProbLog. *Artificial Intelligence*, 298:103504, September 2021. ISSN 00043702. doi: 10.1016/j.artint.2021.103504. URL <https://linkinghub.elsevier.com/retrieve/pii/S0004370221000552>.
- Giuseppe Marra and Ondřej Kuželka. Neural markov logic networks. In *Uncertainty in Artificial Intelligence*, pages 908–917. PMLR, 2021.
- Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Integrating learning and reasoning with deep logic models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 517–532. Springer, 2019.
- Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, page 104062, 2024.
- Simon Odense and Artur d’Avila Garcez. A semantic framework for neurosymbolic computation. *Artificial Intelligence*, 340:104273, 2025.

- Connor Pryor, Charles Dickens, Eriq Augustine, Alon Albalak, William Wang, and Lise Getoor. Neupsl: Neural probabilistic soft logic. *arXiv preprint arXiv:2205.14268*, 2022.
- Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. *Advances in neural information processing systems*, 30, 2017.
- Enrique H Ruspini. On the semantics of fuzzy logic. *International Journal of Approximate Reasoning*, 5(1):45–88, 1991.
- Luciano Serafini and Artur d’Avila Garcez. Logic Tensor Networks: Deep Learning and Logical Reasoning from Data and Knowledge. *arXiv:1606.04422 [cs]*, July 2016. URL <http://arxiv.org/abs/1606.04422>. arXiv: 1606.04422.
- Xiaoting Shao, Alejandro Molina, Antonio Vergari, Karl Stelzner, Robert Peharz, Thomas Liebig, and Kristian Kersting. Conditional sum-product networks: Imposing structure on deep probabilistic architectures. In *International Conference on Probabilistic Graphical Models*, pages 401–412. PMLR, 2020.
- Hikaru Shindo, Viktor Pfanschilling, Devendra Singh Dhami, and Kristian Kersting. α ilp: thinking visual scenes as differentiable logic programs. *Machine Learning*, 112(5):1465–1497, 2023.
- Arseny Skryagin, Wolfgang Stammer, Daniel Ochs, Devendra Singh Dhami, and Kristian Kersting. Neural-Probabilistic Answer Set Programming. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, pages 463–473, 8 2022. doi: 10.24963/kr.2022/48. URL <https://doi.org/10.24963/kr.2022/48>.
- Natalia Slusarz, Ekaterina Komendantskaya, Matthew L Daggitt, Robert Stewart, and Kathrin Stark. Logic of differentiable logics: Towards a uniform semantics of dl. In *Proceedings of 24th International Conference on Logic*, volume 94, pages 473–493, 2023.
- Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezny, Steven Schockaert, and Ondrej Kuzelka. Lifted relational neural networks: Efficient learning of latent relational structures. *Journal of Artificial Intelligence Research*, 62:69–100, 2018.
- Mihaela C Stoian and Eleonora Giunchiglia. Beyond the convexity assumption: Realistic tabular data generation under quantifier-free real linear constraints. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Michio Sugeno. Fuzzy measure and fuzzy integral. *Transactions of the Society of Instrument and Control Engineers*, 8 (2):218–226, 1972.
- Xiaowu Sun and Yasser Shoukry. Neurosymbolic motion and task planning for linear temporal logic tasks. *IEEE Transactions on Robotics*, 2024.
- Efthymia Tsamoura, Timothy Hospedales, and Loizos Michael. Neural-symbolic integration: A compositional perspective. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 5051–5060, 2021.
- Michael van Bekkum, Maaike de Boer, Frank van Harmelen, André Meyer-Vitali, and Annette ten Teije. Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases. *arXiv:2102.11965 [cs]*, March 2021. URL <http://arxiv.org/abs/2102.11965>. arXiv: 2102.11965.
- Frank van Harmelen and Annette ten Teije. A boxology of design patterns for hybrid learning and reasoning systems. *J. Web Eng.*, 18(1-3):97–124, 2019. doi: 10.13052/jwe1540-9589.18133. URL <https://doi.org/10.13052/jwe1540-9589.18133>.
- Emile van Krieken, Samy Badreddine, Robin Manhaeve, and Eleonora Giunchiglia. Uller: A unified language for learning and reasoning. In *International Conference on Neural-Symbolic Learning and Reasoning*, pages 219–239. Springer, 2024.
- Moshe Y Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for concurrency: structure versus automata*, pages 238–266. Springer, 2005.
- Zhenyuan Wang and George J Klir. *Generalized measure theory*, volume 25. Springer Science & Business Media, 2010.
- Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. Nlprolog: Reasoning with weak unification for question answering in natural language. In *ACL*, 2019. doi: 10.18653/v1/p19-1618. URL <https://doi.org/10.18653/v1/p19-1618>.
- Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. DeepStochLog: neural stochastic logic programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10090–10100, 2022a.

- Thomas Winters, Giuseppe Marra, Robin Manhaeve, and Luc De Raedt. Deepstochlog: Neural stochastic logic programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10090–10100, 2022b.
- Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, 2018.
- Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.
- Wen-Chi Yang, Giuseppe Marra, Gavin Rens, and Luc De Raedt. Safe reinforcement learning via probabilistic logic shields. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 5739–5749, 2023.
- Zhun Yang, Adam Ishay, and Joohyung Lee. NeurASP: embracing neural networks into answer set programming. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, pages 1755–1762, 2020.
- Lotfi A Zadeh. Fuzzy sets. *Information and Control*, 1965.
- Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

A Basic of measure theory

Our definition of neurosymbolic inference uses certain basic concepts from measure theory that we outline here.

Definition A.1 (σ -algebra and measurable spaces). Let X be a set, then a σ -algebra Σ on X is a non-empty collection of subsets of X that satisfies the properties

1. $\forall S \in \Sigma : S^c \in \Sigma,$
2. $\forall (S_n)_{n \in \mathbb{N}} : (\forall i \in \mathbb{N} : S_i \in \Sigma) \Rightarrow \bigcup_{n \in \mathbb{N}} S_n \in \Sigma,$
3. $\forall (S_n)_{n \in \mathbb{N}} : (\forall i \in \mathbb{N} : S_i \in \Sigma) \Rightarrow \bigcap_{n \in \mathbb{N}} S_n \in \Sigma.$

That is, a σ -algebra Σ is closed with respect to taking the complement, countable unions and countable intersections. The elements of Σ are called *measurable sets*. If Σ is a σ -algebra on the set X , then the couple (X, Σ) is called a *measurable space*. A function f between two measurable spaces (S, Σ_S) and (T, Σ_T) is called *measurable* if $f^{-1}(T) \in \Sigma_S$ for each $T \in \Sigma_T$.

Example A.2 (A σ -algebra for the Boolean interpretations of propositional logic). Assume we limit the set A of atomic expressions of the language of propositional logic to be finite, e.g. the modern Latin alphabet. In this case, the set of all possible Boolean interpretations is isomorphic to \mathbb{B}^{26} . Any finite set can easily be provided with a σ -algebra by taking the powerset of that set, so a σ -algebra of the set \mathbb{B}^{26} of interpretations could be $\mathcal{P}(\mathbb{B}^{26})$. It is trivial to verify that this collection indeed satisfies the necessary conditions to be a σ -algebra of \mathbb{B}^{26} .

Definition A.3 (Measure and measure spaces). Let (X, Σ) be a measurable space, then a function $m : \Sigma \rightarrow \mathbb{R} \cup \{\infty\}$ is called a *measure* if it satisfies

1. $m(\emptyset) = 0,$
2. **Non-negativity:** $\forall S \in \Sigma : m(S) \geq 0,$
3. **Sigma-additivity:** $\forall (S_n)_{n \in \mathbb{N}} : (\forall i, j, l \in \mathbb{N} : S_i \in \Sigma \wedge S_j \cap S_l = \emptyset) \Rightarrow \sigma(\bigcup_{n \in \mathbb{N}} S_n) = \sum_{n \in \mathbb{N}} m(S_n).$

In other words, a measure is a positive map of subsets of X to the extended real number line that “commutes” with countable unions. If m is a measure on the measurable space (X, Σ) , then the triple (X, Σ, m) is called a *measure space*.

Example A.4 (A measure for the Boolean interpretations of propositional logic). Assume the same setting as in Example A.2 and take the measurable space $(\Omega, \mathcal{P}(\Omega))$ with $\Omega = \mathbb{B}^{26}$. A well-known measure for finite measurable spaces is the *counting measure* m_C that outputs the cardinality of each element of Σ , i.e. $m_C(S) = |S|$.

Given a measure space, one can define a notion of integration that generalises the traditional Riemann integral on real-valued measure spaces to other measure spaces. This notion of integration is based on the *Lebesgue integral* and is constructed by first considering the family of *simple functions*.

Definition A.5 (Simple function). If (X, Σ, m) is a measure space, then a simple function $s(\mathbf{x})$ is a linear combination of indicator functions over disjoint measurable sets with finite measure, i.e.

$$s(\mathbf{x}) = \sum_{i=1}^N a_i \cdot \mathbb{1}_{S_i}(\mathbf{x}), \quad (\text{A.1})$$

where $(S_i)_{i=1}^N$ is a finite sequence of disjoint sets with $S_i \in \Sigma$ and $m(S_i) < \infty$, $a_i \in \mathbb{R}$ and $\mathbb{1}_{S_i}$ is the indicator function on the set S_i . That is,

$$\mathbb{1}_S(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in S, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{A.2})$$

The Lebesgue integral is first defined for the family of simple functions, since they inherently grasp the property of linearity that we are used to from the Riemann integral. Then, the Lebesgue integral for any measurable function $f : X \rightarrow \mathbb{R}$ can be defined as the integral of the simple function “closest from below” to f .

Definition A.6 (Lebesgue integral). The Lebesgue integral of a simple function $s(\mathbf{x}) = \sum_{i=1}^N a_i \cdot \mathbb{1}_{S_i}(\mathbf{x})$ over a measure space (X, Σ, m) is defined as

$$\int s \, dm = \int s(\mathbf{x}) \, dm(\mathbf{x}) = \sum_{i=1}^N a_i \cdot m(S_i). \quad (\text{A.3})$$

Moreover, the Lebesgue integral of a *non-negative* measurable function $f : X \rightarrow \mathbb{R}$ is then defined as

$$\int f \, dm = \int f(\mathbf{x}) \, dm(\mathbf{x}) = \sup \left(\left\{ \int s \, dm \mid s \leq f \text{ and } s \text{ is simple} \right\} \right). \quad (\text{A.4})$$

If f is not non-negative, then we define the two non-negative functions $f^+ = \max(0, f)$ and $f^- = \min(0, f)$ such that

$$\int f \, dm = \int f^+ \, dm - \int f^- \, dm. \quad (\text{A.5})$$