Calculi and Models for Non-Horn Knowledge Bases Containing Neural and Evaluable Predicates

Alexander Sakharov

Synstretch
Framingham, Massachusetts, USA

Abstract

This paper investigates logical foundations of the derivation of literals from non-Horn knowledge bases with fuzzy predicates. Some of the predicates are defined by neural networks, and some are defined by recursive functions. This inference excludes reasoning by contradiction, and it is characterized by means of substructural single-succedent sequent calculi with non-logical axioms expressing knowledge base rules and facts. The semantics of this inference is specified by constrained real-valued models. Lower bounds of fuzzy truth values of ground literals are calculated by traversing sequent calculus derivations of the literals.

Keywords: non-Horn rule, sequent calculus, fuzzy knowledge base, real-valued logic, neural-symbolic computing

1 Introduction

The languages of logic programs and knowledge bases (KB) are usually based on first-order logic (FOL) [28]. Most commonly, KB facts are atoms or literals. Atoms are expressions $P(t_1, ..., t_k)$ where P is a predicate and $t_1, ..., t_k$ are terms. Literals are atoms or their negations. Non-Horn rules are expressions $A \Leftarrow A_1 \wedge ... \wedge A_k$, where $A, A_1, ..., A_k$ are literals. In Horn rules, A and all A_i are atoms. In normal logic programs, A is an atom and A_i are literals.

Horn KBs have a limited inference power. The advantages of non-Horn KB over normal logic programs are discussed in [30]. The semantics of non-Horn KBs is given by classical 2-valued FOL models. FOL calculi are used as the proof theories of non-Horn KBs. Nonetheless, inference for KBs and logic programs differs significantly from inference in FOL. Most importantly, the outcome of this inference and its intermediate steps is literal sets as opposed to arbitrary FOL formulas.

KBs and logic programs may include computable (aka evaluable) functions and predicates [20]. The values of terms composed of constants and evaluable functions are calculated during inference. Also, the truth values of atoms of evaluable predicates with constant arguments are calculated, not derived.

Evaluable functions and predicates may be partial. Evaluable predicates do not have to be boolean, they may yield multiple truth values.

Recent advances in AI made it possible to implement some predicates as neural networks [9,32,15,31,13]. Representing predicates by neural networks is also known as relational embedding [7]. The fuzzy truth values of atoms of these neural predicates with constant arguments are calculated. These values are real numbers. For some predicates, the calculation of fuzzy truth values of atoms with constant arguments can be implemented by other means than neural networks.

The principle of Reductio Ad Absurdum (RAA) states that if A is deduced from a hypothesis that is A's complement, then A is derivable. Reasoning by contradiction, i.e. with using RAA, is not quite adequate for KBs with evaluable predicates [29]. It will be explained later that reasoning by contradiction is not appropriate for KBs with neural predicates either.

The aim of this paper is to specify model and proof theories for inference from KBs containing neural and evaluable predicates along with other predicates that are derivable from KB rules and facts. In section 3, KB inference without RAA is characterized by sequent calculi with a limited set of logical and structural rules and with non-logical axioms that are images of KB facts and rules. In section 4, the semantics of inference from KBs containing neural and evaluable predicates is specified by constrained real-valued models. It is also shown how to calculate lower bounds of the truth values of derived ground literals.

2 Non-Horn Knowledge Bases With Fuzzy Predicates

Let us recall some definitions which will be used later. A KB is called consistent if no atom is a fact or is derivable from this KB, along with its negation being derivable or a fact. A literal is called ground if it does not contain variables. A substitution is a finite set of mappings of variables to terms. The result of applying a substitution to a formula or set of formulas is called its instance.

We consider inference of ground literals, which are called goals, from non-Horn KBs. These KBs may contain predicates specified by neural networks, which are used to approximate the truth values of atoms of these predicates with constant arguments. Fuzzy truth values are usually represented by real numbers from interval $[0, 1]$. For non-Horn KBs, it is more convenient to use interval $[-1, 1]$ for the representation of truth values. One represents true, minus one represents false. Other real numbers from interval $[-1, 1]$ represent fuzzy truth values.

These KBs may also contain evaluable functions and predicates [20]. We assume that evaluable functions and predicates are defined as recursive functions in a functional programming language or as algorithms in a procedural programming language. The truth values yielded by the algorithms implementing evaluable predicates could also be fuzzy, i.e. they could be from interval $[-1, 1]$.

Terms of evaluable functions with constant arguments are evaluated as soon as they appear in KB derivations. The same applies to atoms of neural and evaluable predicates with constant arguments. The evaluation may not terminate, in which case it is assumed that the truth value is zero. Any complete search strategy for inference from KBs with evaluable and neural predicates should continue and-or search [28] simultaneously with the evaluations including neural computations. If the evaluation of ground atom $A(...)$ yields a positive value above a certain threshold $h > 0$, then $A(...)$ is considered a KB fact. If the evaluation of this atom yields a negative value below $-h$, then $\neg A(...)$ is considered a fact.

All other predicates will be called derivable. As explained in [30], derivable predicates should be considered partial by default. In the presence of neural predicates, the truth values of ground atoms of derivable predicates should also be real numbers from interval $[-1, 1]$, that is, derivable predicates like neural ones are fuzzy. It is expected that fuzzy truth values higher than h are assigned to some facts. One is the default truth value for KB facts. Let $|A|$ denote the truth value of ground literal A.

We rely on the traditional definition of truth functions in fuzzy KBs [4]. The following equation defines the truth values for negation: $|\neg A| = -|A|$. The use of this truth function for negation is limited to the calculation of the truth values of negatibve literals of neural and evaluable predicates. The truth values for conjunction are defined by the following equation: $|A_1 \wedge ... \wedge A_k| = \min\{|A_1|, ..., |A_k|\}$. The use of this truth function for conjuction is limited to the calculation of the truth values of the bodies of KB rules.

Truth functions for disjunctions will not be used here, and the use of implication truth functions will be indirect in the KBs under consideration. The meaning of KB rules is that the truth value of the rule body is a lower bound of the truth value of the head. Given that KB rules are implications and assuming that KB rules are not fuzzy, this semantics of KB rules is consistent with several implication truth functions for t-norms. For the Lukasiewicz, Godel, and product t-norms, $|A \Rightarrow B| = 1$ if $|A| \leq |B|$ [12].

It is explained in [30] why reasoning by contradiction is questionable for KBs with evaluable predicates. The same argument applies to KBs containing neural predicates. Consider two KB rules $P \Leftarrow Q$ and $P \Leftarrow \neg Q$. Here is reasoning by contradiction using these rules. Suppose P is false. The first rule implies that Q is false, and hence P is true by the second rule. Now suppose $|P| = 0$. If $|Q| = 0$ as well, then both rules are satisfied, but they do not provide any evidence that P is true or $|P| > 0$ at least.

3 Sequent Calculi

Let $-A$ denote the complement of A, i.e. it is the negation of atom A, and the atom of negative literal A. A sequent is $\Gamma \vdash \Pi$ where Γ is an antecedent and Π is a succedent. Antecedents and succedents are multisets of formulas. KB inference and logic programming are concerned about the derivation of literals, i.e. sequents of the form $\vdash A$ where A is a literal. Consider single-succedent calculi in which formulas are literals. The only structural rule is *cut*.

$$\frac{\Gamma \vdash A \quad A, \Pi \vdash B}{\Gamma, \Pi \vdash B}\ cut$$

These sequent calculi do not have logical axioms. The following rule is the only logical rule. It replaces the standard negation rules.

$$\frac{A, \Gamma \vdash B}{-B, \Gamma \vdash -A}\ swap$$

KB facts and rules can be treated as non-logical axioms [22]. Sequents of the form $\vdash A$ represent facts, and rules are represented by sequents of the form $A_1, ..., A_n \vdash A$ where $A, A_1, ..., A_n$ are literals. Variables can be replaced by any terms in instances of these axioms.

Definition 3.1 L_{cs} is the set of sequent calculus instances in which formulas are literals, succedents contain one literal, the structural rule is *cut*, the logical rule is *swap*, and non-logical axioms represent KB rules and facts.

Arguably, L_{cs} are some of the simplest calculi formalizing KB inference without RAA. Alternatively, this inference could be formalized by calculi whose sequents contain atoms only. Yet another option is to define calculi based on clauses, i.e. disjunctions, as opposed to sequents. We chose L_{cs} because their single-succedent sequents comprised of literals copy KB rules. The other two options require KB rule transformations. L_{cs} rules embody two fundamental logical principles: *cut* corresponds to Modus Ponens and *swap* corresponds to Modus Tollens.

Usually, if a formal theory is inconsistent, then any formula is derivable in this theory. This is why inconsistent theories are discarded. In reality, KBs may have bugs and may be inconsistent. Proliferation of inconsistencies is limited in L_{cs}. Unlike sequent calculi for FOL, nothing else could be derived in L_{cs} from sequents $\vdash A$ and $\vdash \neg A$ alone. Nonproliferation of inconsistencies is important in KB development because bugs do not lead to a mass of gibberish results in this case.

Theorem 3.2 *(normal form) Any L_{cs} derivation of literal G can be transformed into such L_{cs} derivation of G that the premise of every swap is a KB rule and the transformed derivation tree contains the same KB fact instances as the original derivation tree.*

Proof. Consider a L_{cs} derivation. Let us replace *swap* with the two following rules and adjust the derivation by replacing *swap* with the $L\neg$ rule followed by the $R\neg$ rule.

$$\frac{\Gamma \vdash A}{-A, \Gamma \vdash}\ L- \qquad\qquad \frac{B, \Gamma \vdash}{\Gamma \vdash -B}\ R-$$

The $L-$ and $R-$ rules can be moved upward.

$$\cfrac{\cfrac{\Gamma \vdash A \quad A, \Pi \vdash B}{\Gamma, \Pi \vdash B}}{-B, \Gamma, \Pi \vdash} \quad \rightarrow \quad \cfrac{\Gamma \vdash A \quad \cfrac{A, \Pi \vdash B}{A, -B, \Pi \vdash}}{-B, \Gamma, \Pi \vdash}$$

$$\dfrac{\dfrac{\Gamma \vdash A \quad A, \Pi, B \vdash}{\Gamma, \Pi, B \vdash}}{\Gamma, \Pi \vdash -B} \quad \rightarrow \quad \dfrac{\Gamma \vdash A \quad \dfrac{A, \Pi, B \vdash}{A, \Pi \vdash -B}}{\Gamma, \Pi \vdash -B}$$

$$\dfrac{\dfrac{B, \Gamma \vdash A \quad A, \Pi \vdash}{B, \Gamma, \Pi \vdash}}{\Gamma, \Pi \vdash -B} \quad \rightarrow \quad \dfrac{\dfrac{A, \Pi \vdash}{\Pi \vdash -A} \quad \dfrac{\dfrac{B, \Gamma \vdash A}{-A, B, \Gamma \vdash}}{-A, \Gamma \vdash -B}}{\Gamma, \Pi \vdash -B}$$

By repeatedly applying these permutations, all applications of the $L-/R-$ rules can be moved above all applications of *cut*. Since $R\neg$ always follows $L\neg$, the succedents of the premises of all *cut* rules are single literals. Any sequence of $L-/R-$ rules applied to a KB rule or fact can be either discarded or replaced by one *swap* rule. The above permutations do not change the set of KB fact instances. Hence, the transformed derivation satisfies the statement of this theorem. □

Theorem 3.3 *L_{cs} is sound and complete with respect to the derivation of ground literals in FOL without RAA.*

Proof. It is proved in [29] that ground literal L is derivable from KB facts and rules in FOL without RAA if and only if $-L$ is refutable by resolution in which the factoring rule is not used and at least one premise of every resolution step is not $-L$ or its descendant. Consider such resolution refutation. The resolution steps that are not ascendants of the endclause are discarded. Let us ground this refutation and then exclude the step that resolves $-L$. There is only one such step because at least one premise of every resolution step is not $-L$ or its descendant. As a result, L is added to every descendant clause of this step including the endclause which becomes L.

Let us traverse this resolution tree bottom-up and map every resolution step to an application of *cut* in L_{cs}. Sequent $\vdash L$ is the conclusion of the last *cut* in the respective L_{cs} derivation tree. The premises of every *cut* in this tree are uniquely determined by the resolution step. The succedent of the *cut* conclusion is also the succedent of the second premise, and the succedent of the first premise is the principal formula of this *cut*. Every leaf node in the L_{cs} derivation tree is an instance of a KB fact, KB rule, or the conclusion of *swap* applied to an instance of a KB rule.

Now consider a ground normal-form L_{cs} derivation of sequent $\vdash L$. Every application of *cut* in this derivation corresponds to a resolution step but ground instances of KB rules and facts are used in this resolution derivation instead of the rules and facts. The endclause of this resolution derivation is L.

The lifting lemma [6] states that if clause A is an instance of A', B is an instance of B', and C is the resolvent of A and B, then there is such clause C' that C is its instance, and C' is the resolvent of A' and B'. It is well-known that the lifting lemma can be generalized onto arbitrary resolution derivations: If C is the endclause of a resolution derivation with input clauses $A_1, ..., A_n$ which are instances of $A'_1, ..., A'_n$, respectively, then there is such resolution derivation with input clauses $A'_1, ..., A'_n$ and endclause C' that C is an instance of C'. The proof is a straightforward induction on the depth of resolution derivations.

As a consequence of this generalization of the lifting lemma, there is a resolution tree with the input comprised of KB rules and facts treated as clauses and with such endclause L' that L is its instance. A step resolving L' and $–L$ is added to this derivation. The resolvent of this step is the empty clause, and $–L$ occurs in one premise of the last step only. □

4 Constrained Real-Valued Models

Models are usually defined by truth tables (or functions) for logical connectives so that the truth values of ground formulas can be calculated. No other formulas than literals are produced during KB derivations. Because of this, legitimate models for KB inference can be defined by a negation truth function and by constraints on truth values in ground instances of facts and rules as opposed to truth tables for other logical connectives.

Definition 4.1 An assignment of real numbers from interval $[-1, 1]$ to ground literals is a $\mathcal{M}_r$ model if $|\neg A| = -|A|$ for any ground atom A and the following constraints are satisfied:
1. A is a ground KB fact instance: $|A| > h$
2. $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$ is a ground KB rule instance:
   a. If $|A_i| \geq h$ for $i = 1...k$, then $|A_0| \geq \min\{|A_1|, ..., |A_k|\}$.
   b. For $j = 1, ..., k$, if $|A_0| \leq -h$ and $|A_i| \geq h$ for $i = 1...j-1$ and $i = j+1...k$, then $-|A_j| \geq -|A_0|$.

Constraint 2a expresses the semantics of KB rules: the truth value of the body is less or equal to the truth value of the head, min is employed as the truth function for conjunctions of literals in the bodies. Constraint 2b is a consequence of this semantics of KB rules with fuzzy literals. Consider the case that $|A_i|$ are positive for $i = 1...j-1$ and $i = j+1...k$, and $|A_0|$ is negative. In this case, inequality $|A_0| \geq \min\{|A_1|, ..., |A_k|\}$ implies that $-|A_j| \geq -|A_0|$.

Literal A is valid regarding $\mathcal{M}_r$ models if $|A'| > h$ for all groundings A' of literal A in all $\mathcal{M}_r$ models. The constraints of $\mathcal{M}_r$ models can also be considered in the context of sequents as opposed to KB facts and rules. These constraints hold for non-logical axioms of L_{cs}.

Definition 4.2 The set of obscure occurrences of literals in derivation τ is defined recursively as the minimal set of literal occurrences satisfying the following two conditions.
- If sequent $–A_0, A_1, ..., A_{j-1}, A_{j+1}, ..., A_k \vdash –A_j$ from τ is the conclusion of *swap* applied to KB rule instance $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$, then $A_1, ..., A_{j-1}, A_{j+1}, ..., A_k$ are obscure in τ.
- If sequent $A_1, ..., A_k \vdash A_0$ occurs in τ and A_0 is obscure in τ, then $A_1, .., A_k$ are obscure in it.

Let $m(\tau) = \min_{A \in \mathcal{F}} |A|$ where $\mathcal{F}$ is the set of non-obscure occurrences of ground KB fact instances in derivation τ. If $\mathcal{F} = \emptyset$, then $m(\tau) = 1$.

Theorem 4.3 *(soundness) If τ is a ground L_{cs} derivation of literal G, then $|G| \geq m(\tau) \geq h$ for all $\mathcal{M}_r$ models.*

Proof. Let us transform τ to the normal form defined in Theorem 3.2. The set of literals in τ is the same as the set of literals in its normal form. We will prove by induction on the depth of normal-form derivations that $|D| \geq \min\{|A_i|, ..., |A_j|, m(\mu)\}$ for the endsequent $A_1, ..., A_k \vdash D$ of any derivation μ, where $A_i, ..., A_j$ are non-obscure literal occurrences in μ among $A_1, ..., A_k$. As a corollary, $|G| \geq m(\tau)$. Inequality $m(\tau) \geq h$ holds because $|A| \geq h$ for all ground fact instances A.

Base: The depth of derivation μ is zero. In this case, G is an instance of a KB fact, and the above inequality holds.

Induction step. Suppose the inequality under consideration is satisfied for all derivations whose depth is less or equal n. Suppose the depth of μ is $n+1$. If the endsequent $A_1, ..., A_k \vdash D$ is a KB rule instance, then this sequent does not contain KB fact instances, and inequality $|D| \geq \min\{|A_1|, ..., |A_k|, m(\mu)\}$ holds due to constraint 2a. None of $A_1, ..., A_k$ is obscure in μ. If the last rule in μ is *swap*, then its premise is a KB rule, μ does not contain KB fact instances, and inequality $|D| \geq \min\{|A_1|, m(\mu)\}$ holds due to constraint 2b. Literals $A_2, ..., A_k$ are obscure in μ.

Now let the last rule in μ be *cut*, the first premise of this *cut* be $B_1, ..., B_k \vdash C_1$, and the second premise be $C_1, ..., C_m \vdash D$. The conclusion of this *cut* is $B_1, ..., B_k, C_2, ..., C_m \vdash D$. If δ is the derivation ending in $B_1, ..., B_k \vdash C_1$, $B_b, ..., B_{b'}$ are the non-obscure literals of this antecedent in δ, ν is the derivation ending in $C_1, ..., C_m \vdash D$, $C_c, ..., C_{c'}$ are the non-obscure literals of the antecedent of the latter sequent in ν, then $|C_1| \geq \min\{|B_b|, ..., |B_{b'}|, m(\mu)\}$ and $|D| \geq \min\{|C_c|, ..., |C_{c'}|, m(\nu)\}$ by the induction assumption.

If C_1 is obscure in ν, then it is also obscure in μ. In this case, all literal occurrences from δ including $B_1, ..., B_k$ are obscure in μ. This is proved by a straightforward induction on the depth of normal-form derivations. In the case of obscure C_1, the set of non-obscure literal occurrences of μ is the same as the set of non-obscure literal occurrences of ν. Therefore, $|D| \geq \min\{|C_c|, ..., |C_{c'}|, m(\mu)\}$.

Now, consider the case that C_1 is not obscure in ν, i.e. C_c is C_1. By a straightforward induction on the depth of normal-form derivations, the succedent of the endsequent of any derivation γ is not obscure in γ. As a corollary, C_1 is not obscure in μ. The remaining literals among $C_c, ..., C_{c'}$ are not obscure in μ either. Let $C_{c''}$ follow C_c in this list of literals. By combining the inequalities for both premises, we get $|D| \geq \min\{|B_b|, ..., |B_{b'}|, m(\delta), |C_{c''}|, ..., |C_{c'}|, m(\nu)\}$. The set of non-obscure occurrences of KB fact instances of μ is the union of the respective sets of δ and ν. Hence, $|D| \geq \min\{|B_1|, ..., |B_k|, |C_{c''}|, ..., |C_{c'}|, m(\mu)\}$. □

This theorem establishes that $m(\tau)$ is a conservative approximation of the truth values of G in $\mathcal{M}_r$ models. The proof of Theorem 3.3 shows that resolution refutations can be transformed to normal-form L_{cs} derivations in a linear time of the size of the refutations. It is clear from the proof of Theorem 4.3 that the calculation of a lower bound of $|G|$ can be done in a single postorder traversal of the derivation tree. Detecting obscure literal occurrences is done simultaneously with the calculation of m values for sequent succedents during this traversal. Hence, this calculation takes a linear time of the size of G's derivation in L_{cs}. Efficient resolution methods can implement inference from non-Horn KBs containing neural and evaluable predicates, and then lower bounds of fuzzy truth values of the goals are calculated.

Theorem 4.4 *(completeness) If $|G| \geq h$ in all $\mathcal{M}_r$ models for ground literal G, then there exists a derivation of G in L_{cs}.*

Proof. Suppose G is not derivable in L_{cs} from KB facts and rules. Let us look at model M in which $|B| = 1$ for every ground literal B that is derivable from KB facts and rules, $|C| = -1$ for every such ground literal C that $-C$ is derivable, and $|D| = 0$ for every other ground literal D. Such model M exists for any consistent KB, and $|G| = 0$ in M.

Constraint 1 holds for M because ground instances of facts are derivable. Suppose constraint 2a is violated for ground KB rule instance $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$. In this case, $|A_i| = 1$ for $i = 1...k$, and all sequents $\vdash A_i$ are derivable in L_{cs}. Hence, A_0 is derivable from the latter by k applications of *cut* to $A_1, ..., A_k \vdash A_0$ and to every $\vdash A_i$ for $i = 1...k$. Hence, constraint 2a could not be violated for this rule instance.

If we suppose that constraint 2b is violated for ground KB rule instance $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$, then all A_i for $i = 1...j-1$ and $i = j+1...k$ are derivable in L_{cs}, and $-A_0$ is also derivable. Sequent $-A_0, A_1, ..., A_{j-1}, A_{j+1}, ..., A_k \vdash -A_j$ is derived by applying *swap* to this KB rule instance. $-A_j$ is derivable by application of *cut* to this sequent and to $\vdash -A_0$ followed by $k-1$ applications of *cut* using $\vdash A_i$ for $i = 1...j-1$ and $i = j+1...k$ as the first premise. Consequently, constraint 2b could not be violated for this rule instance. Therefore M is a $\mathcal{M}_r$ model and the assumption about G not being derivable in L_{cs} cannot be true. □

5 Related Work

An overview of KB inference methods including resolution-based methods can be found in [28]. Ordered resolution is recognized as one of the most efficient inference methods [3]. It is used in modern theorem provers [17]. Ordered resolution has been adapted to inference from non-Horn KBs without RAA [29].

Description logics [2] and other logics with more limited capabilities than FOL are relevant to KB inference. Inference without RAA is used in argumentation logics [16]. The proof theory suggested in that paper is natural deduction without the RAA rule. Other arumentation logics with limited inference capabilities have been proposed in [5].

Inference from fuzzy KBs is focused on numerical calculations approximating truth values. Forward chaining normally serves as the inference mechanism for fuzzy KBs [4]. KB inference without RAA is more powerful than the forward application of Modus Ponens in chaining and it can be efficiently implemented by leveraging on advanced resolution methods. For non-Horn KBs with neural and evaluable predicates, symbolic inference is done first and then approximate truth values are computed by traversing the derivation trees. In contrast to fuzzy truth functions for logical connectives [12], we utilize constraints imposed by KB rules on fuzzy truth values.

Paper [7] is a comprehensive survey of recent work in the area of neural-symbolic computing. Neural-symbolic systems integrate neural networks and inference methods. In particular, neural networks are used for guiding symbolic inference [33,14,26]. Integration of neural and fuzzy systems is analyzed in [1].

Paper [27] introduces a neural-symbolic method employing weighted real-valued functions for calculating lower and upper bounds of the truth values of FOL formulas. Inference is implemented as alternating upward and downward passes over the structure of the formulas. Truth value bounds are adjusted during these passes. Modus Ponens and Modus Tollens are used to get truth value bounds. In our work, KB rules play the role of premises of Modus Ponens, and swapped KB rules can be viewed as premises of Modus Tollens.

Non-Horn KBs containing neural and evaluable predicates are similar to possibilistic logic [10] in the sense that in both of them, real numbers are associated with derived ground literals. A survey of fuzzy proof theories in which numbers indicating truthness are attached to FOL formulas is presented in [11]. The major difference of our approach is that literals are the only FOL formulas involved in the KB formalism considered here.

ProbLog [25] extends Prolog by associating probabilities with facts. It is assumed that all ground instances of a non-ground fact are mutually independent and share the same probability. ProbLog engines calculate approximate probabilities for inference goals. Since Prolog has positive goals only, negation as failure is adopted in ProbLog to derive negative goals. Non-Horn KBs with neural and evaluable predicates are not probabilistic, they use constraints on the truth values of literals for getting lower bounds of the truth values of derived goals. Inference of negative goals from non-Horn KBs is direct, which helps avoid controversies related to negation as failure [8].

DeepProbLog [19] extends ProbLog by allowing neural networks to be associated with facts instead of probabilities. The probabilities of ground instances of a fact are calculated by the neural network associated with the respective predicate. This is similar to our assumption except for the interpretation of the values yielded by neural networks. We follow their traditional interpretation as fuzzy truth values of ground facts.

Sequent calculus derivations for Horn formulas are researched in [21]. Substructural sequent calculi have been investigated for decades [24,23]. L_{cs} instances are substructural calculi. The set of L_{cs} calculi is particularly tailored to inference from non-Horn KBs with neural and evaluable predicates. The replacement of the two negation rules with the swap rule makes L_{cs} single-succedent, which is essential for the approximation of truth values.

L_{cs} instances contain non-logical axioms which represent KB rules and facts. The cut rule is a core of these calculi. Properties of sequent calculi with non-logical axioms in the form of so-called mathematical basic sequents are investigated in [22]. Axioms corresponding to KB rules/facts can be transformed to mathematical basic sequents.

Like L_{cs}, LK_{-c} calculi from [30] contain non-logical axioms representing KB rules and facts. LK_{-c} calculi characterize inference of literals from non-Horn KBs without using RAA. Those calculi have the same inference power as L_{cs} but they employ standard negation rules as opposed to the swap rule, they allow multiple literals in succedents. LK_{-c} derivations cannot be directly used for the approximation of fuzzy truth values.

6 Conclusion

The language of non-Horn KBs is much simpler than the language of FOL. Negation is a connective in this language. Conjunction with a variable number of arguments and implication are embedded in KB rules but they are not standalone connectives in the language. Non-Horn KBs with evaluable and neural predicates integrate reasoning, computation, and neural networks. They are neural-symbolic systems [18]. These KBs bear similarities with fuzzy KBs [4], but the inference is symbolic. There exist efficient inference methods [29] which can be directly applied to non-Horn KBs containing neural and evaluable predicates.

The calculi and models presented here are comprehensible. Both non-logical axioms of L_{cs} and the constraints in $\mathcal{M}_r$ models are projections of KB facts and rules. The most important feature of our characterization of non-Horn KBs with evaluable and neural predicates is that L_{cs} derivations provide sufficient information for the calculation of lower bounds of the truth values of the derivation goals. Hilbert-type systems are less adequate for characterizing these KBs because they would explicitly include other logical connectives, possibly non-standard ones.

It is feasible to get multiple L_{cs} derivations of the same goal. These derivations of one literal may give various approximations of the truth value of this literal. It may be beneficial to skip some facts with truth values close to h during the derivation process. The design of efficient inference methods capturing higher truth values is beyond the scope of this paper. Investigation of the applicability of other fuzzy truth functions [12] to non-Horn KBs containing neural and evaluable predicates is a topic for future research.

References

[1] Abraham, A., *Adaptation of fuzzy inference system using neural learning*, in: *Fuzzy systems engineering*, Springer, 2005 pp. 53–83.

[2] Baader, F., I. Horrocks and U. Sattler, *Description logics*, Foundations of Artificial Intelligence **3** (2008), pp. 135–179.

[3] Bachmair, L. and H. Ganzinger, *Resolution theorem proving*, in: *Handbook of automated reasoning*, Elsevier, 2001 pp. 19–99.

[4] Barros, L. C. d., R. C. Bassanezi and W. A. Lodwick, "A first course in fuzzy logic, fuzzy dynamical systems, and biomathematics: theory and applications," Springer, 2017.

[5] Besnard, P. and A. Hunter, *A review of argumentation based on deductive arguments*, Handbook of Formal Argumentation (2018), pp. 437–484.

[6] Chang, C.-L. and R. C.-T. Lee, "Symbolic logic and mechanical theorem proving," Academic press, 1973.

[7] d'Avila Garcez, A. S., M. Gori, L. C. Lamb, L. Serafini, M. Spranger and S. N. Tran, *Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning*, Journal of Applied Logics **6** (2019), pp. 611–632.

[8] Denecker, M., M. Truszczynski and J. Vennekens, *About negation-as-failure and the informal semantics of logic programming*, Association for Logic Programming (2017).

[9] Dong, H., J. Mao, T. Lin, C. Wang, L. Li and D. Zhou, *Neural logic machines*, in: *International Conference on Learning Representations*, 2019.

[10] Dubois, D. and H. Prade, *Possibilistic logic-an overview.*, in: *Computational logic*, 2014, pp. 197–255.

[11] Gottwald, S., "A treatise on many-valued logics," Research Studies Press, 2001.

[12] Hájek, P., "Metamathematics of fuzzy logic," Springer Science & Business Media, 2013.

[13] Hong, J. and T. P. Pavlic, *An insect-inspired randomly, weighted neural network with random fourier features for neuro-symbolic relational learning*, in: *Neural-Symbolic Learning and Reasoning* (2021).

[14] Huang, D., P. Dhariwal, D. Song and I. Sutskever, *Gamepad: A learning environment for theorem proving*, in: *International Conference on Learning Representations*, 2019.

[15] Ishihara, T., K. Hayashi, H. Manabe, M. Shimbo and M. Nagata, *Neural tensor networks with diagonal slice matrices*, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 506–515.

[16] Kakas, A. C., P. Mancarella and F. Toni, *On argumentation logic and propositional logic*, Studia Logica **106** (2018), pp. 237–279.

[17] Kovács, L. and A. Voronkov, *First-order theorem proving and vampire*, in: *International Conference on Computer Aided Verification*, Springer, 2013, pp. 1–35.

[18] Lamb, L. C., A. S. d'Avila Garcez, M. Gori, M. O. R. Prates, P. H. C. Avelar and M. Y. Vardi, *Graph neural networks meet neural-symbolic computing: A survey and perspective*, in: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020, pp. 4877–4884.

[19] Manhaeve, R., S. Dumančić, A. Kimmig, T. Demeester and L. De Raedt, *Neural probabilistic logic programming in deepproblog*, Artificial Intelligence **298** (2021), p. 103504.

[20] McCune, W., *Otter 3.3 reference manual and guide*, Technical report, Argonne National Lab. (2003).

[21] Miller, D., G. Nadathur, F. Pfenning and A. Scedrov, *Uniform proofs as a foundation for logic programming*, Annals of Pure and Applied logic **51** (1991), pp. 125–157.

[22] Negri, S. and J. Von Plato, "Structural proof theory," Cambridge University Press, 2001.

[23] Ono, H., *Logics without the contraction rule and residuated lattices*, Australasian Journal of Logic (2010).

[24] Paoli, F., "Substructural logics: a primer," Springer Science & Business Media, 2013.

[25] Raedt, L. D. and A. Kimmig, *Probabilistic (logic) programming concepts*, Machine Learning **100** (2015), pp. 5–47.

[26] Rawson, M. and G. Reger, *A neurally-guided, parallel theorem prover*, in: *International Symposium on Frontiers of Combining Systems*, Springer, 2019, pp. 40–56.

[27] Riegel, R., A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma et al., *Logical neural networks*, arXiv preprint arXiv:2006.13155 (2020).

[28] Russell, S. and P. Norvig, "Artificial Intelligence: A Modern Approach," Prentice Hall Press, 2009, 3rd edition.

[29] Sakharov, A., *Inference methods for evaluable knowledge bases*, in: *Software Engineering Application in Informatics*, Lecture Notes in Networks and Systems (2021), pp. 499–510.

[30] Sakharov, A., *A logical characterization of evaluable knowledge bases*, in: *14th International Conference on Agents and Artificial Intelligence*, 2022, pp. 681–688.

[31] Santoro, A., D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia and T. Lillicrap, *A simple neural network module for relational reasoning*, Advances in neural information processing systems **30** (2017).

[32] Serafini, L. and A. S. d'Avila Garcez, *Logic tensor networks: Deep learning and logical reasoning from data and knowledge*, in: *Neural-Symbolic Learning and Reasoning* (2016).

[33] Wang, M., Y. Tang, J. Wang and J. Deng, *Premise selection for theorem proving by deep graph embedding*, Advances in neural information processing systems **30** (2017).