A Logical Characterization of Evaluable Knowledge Bases

Alexander Sakharov
Synstretch, Framingham, MA, U.S.A.

Keywords: Evaluable Function/predicate, Reductio Ad Absurdum, 3-Valued Model, Substructural Sequent Calculus.

Abstract: Evaluable knowledge bases are comprised of non-Horn rules with partial predicates and functions, some of them are defined as recursive functions. This paper investigates logical foundations of the derivation of literals from evaluable knowledge bases without reasoning by contradiction. The semantics of this inference is specified by constrained 3-valued models. The derivation of literals without reasoning by contradiction is characterized by means of sequent calculi with non-logical axioms expressing knowledge base rules and facts. The logical rules of these calculi include only the negation rules, and cut is the only essential structural rule.

1 INTRODUCTION

A great deal of research has been devoted to logical characterizations of emerging AI methods. Discovering logical characterizations is important because it brings clarity to the methods. Semantics establish the legitimacy of method results. Calculi make the results of these methods explainable. Besides, the apparatus of mathematical logic can be used to analyze AI methods and compare them. Both first-order logic (FOL) and nonstandard logics are used in these characterizations. Recent advances in AI have given rise to interest in nonstandard logics including description logics, argumentation logics, default logic, etc. One advantage of nonstandard characterizations is that they are usually simpler than FOL.

The languages of logic programs and knowledge bases (KB) are usually based on FOL (Russell and Norvig, 2009). And FOL is normally used for the characterization of relevant inference methods. Nonetheless, inference for KBs and logic programs differs significantly from inference in FOL. The outcome of this inference and its intermediate steps is literals as opposed to arbitrary FOL formulas. The number of rules and especially the number of facts involved in KB inference may be huge whereas FOL axiom sets tend to be compact. The interpretation of negation may be different from FOL, which entails nonstandard semantics and calculi. KBs and logic programs may include computable (aka evaluable) functions and predicates (McCune, 2003).

Generally, KB facts are atoms or literals. Atoms are expressions $P(t_1,\ldots,t_k)$ where P is a predicate and $t_1,\ldots,t_k$ are terms. Literals are atoms or their negations. Non-Horn rules are expressions $A \Leftarrow A_1 \wedge \ldots \wedge A_k$, where $A, A_1, \ldots, A_k$ are literals. FOL formulas can be expressed by equisatisfiable conjunctions of non-Horn clauses. In Horn rules, A and all A_i are atoms. In normal logic programs, A is an atom and A_i are literals.

The semantics of KB with Horn or non-Horn rules/facts is given by classical 2-valued FOL models. FOL calculi are used as the proof theories of Horn and non-Horn KBs. FOL models and proof theories are not adequate for normal logic programs, the respective logical characterizations are nonmonotonic. Availability of rules with both positive and negative heads in non-Horn KBs helps avoid controversies related to logical characterizations of normal logic programs (Denecker et al., 2017). Half-finished non-Horn KBs are not prone to erroneous derivations of negative literals unlike nonmonotonic systems.

Evaluable functions and predicates may be partial, and thus, their calculation may not yield a result. In FOL, it is implicitly assumed that all functions and predicates are total, they are not associated with algorithms, and terms with constant arguments are not mapped to constants unless we use equational extensions of FOL. The computational nature of KBs and logic programs is lost in this FOL treatment of rules and facts. KBs with non-Horn rules/facts and partial predicates and functions, some of which are evaluable, will be called evaluable KBs.

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. In author's opinion,

Sakharov, A.
A Logical Characterization of Evaluable Knowledge Bases.
DOI: 10.5220/0010886700003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 681-688
ISBN: 978-989-758-547-0; ISSN: 2184-433X

the legitimacy of RAA is questionable in the presence of partial predicates or functions. It seems faulty to apply RAA when the truth value of the hypothesis is undefined because the hypothesis complement is also undefined. For example, should $P(a)$ be derivable from fact $R(a)$ and rules $P(x) \Leftarrow R(x) \wedge Q(x)$, $P(x) \Leftarrow \neg Q(x)$ if the computation of $Q(a)$ does not terminate? Derivations without RAA are often called direct.

The aim of this paper is to specify model and proof theories for inference from evaluable KBs. Inference without using RAA, i.e. without reasoning by contradiction, seems more adequate for evaluable KBs than FOL. There exists a variety of methods that implement KB inference without reasoning by contradiction (Sakharov, 2021). These methods include an adaptation of highly efficient ordered resolution.

In section 3, the semantics of inference from evaluable KBs is specified by constrained 3-valued models. The third value represents undefined truth values of partial predicates and atoms with undefined arguments. In section 4, KB inference without RAA is characterized by sequent calculi whose logical rules are the negation rules and whose structural rules exclude the contraction rule (Szabo, 1969). KB facts and rules are non-logical axioms of these calculi.

2 PRELIMINARIES

We consider inference of sets of literals, which are called goal lists, from evaluable KBs. We assume that evaluable functions are defined by means that are external to KB rules. These functions could be defined as recursive functions in a functional programming language or as algorithms in a procedural programming language. The same assumption applies to evaluable predicates. The latter are boolean functions. Combining logic and functional programming has been extensively investigated (Hanus, 2013; Casas et al., 2006; Rodríguez-Hortalá and Sánchez-Hernández, 2008).

It is possible to use other systems for specifying evaluable functions/predicates. For example, database query results can be interpreted as function values or predicate truth values. Function/predicate arguments are parameters of the queries. Some results could be interpreted as undefined values. Evaluable predicates could also be specified by a neural network (Dong et al., 2019; Serafini and d'Avila Garcez, 2016), which is used to approximate the truth values of atoms of these predicates with constant arguments. Truth values are usually approximated by real numbers from interval $[0,1]$. Numbers close to 1 are equated with true, numbers close to 0 are equated with false. Otherwise, the truth value is undefined.

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.

Non-evaluable predicates in non-Horn KBs should be considered partial by default. First, rule sets are often incomplete by design. As a result, some atoms with constant arguments are not derivable in FOL or another formal system from KB rules and facts, and their negations are not derivable either. Second, some non-evaluable predicates are inherently partial from the proof-theoretical perspective.

It is well-known that any partial recursive function can be represented by Horn facts and rules (Voronkov, 1995). Consider function f whose domain is recursively enumerable but not recursive (not decidable) (Rogers, 1967) and whose range is a known finite set. Let non-evaluable predicate P represent f, that is $P(x,y)$ is derivable if and only if $f(x) = y$. If P were total, then the derivation of either $P(a,b)$ or $\neg P(a,b)$ would succeed for any constants a,b provided that the inference procedure is complete. This contradicts our assumption about the domain of f, and thus, P is partial.

Let $-A$ denote the complement of A, i.e. it is the negation of atom A, and the atom of negative literal A. This notation will also be used for sets of literals. As usual, it is assumed that premises of inference rules have disjoint variables. For any KB rule $A \Leftarrow A_1 \wedge ... \wedge A_k$, the following implications $-A_i \Leftarrow -A \wedge A_1 \wedge ...A_{i-1} \wedge A_{i+1} ... \wedge A_n$ are called contrapositives to this rule (Stickel, 1992). KB rules entail their contrapositives in classical FOL.

Both forward and backward chaining are based on Generalized Modus Ponens (GMP) as the sole inference rule (Russell and Norvig, 2009). If $A'_1...A'_k$ are derived literals, $A \Leftarrow A_1 \wedge ... \wedge A_k$ is a KB rule, substitution θ is a unifier of literals A'_i and A_i, i.e. $A'_i\theta = A_i\theta$, for $i = 1...k$, then $A\theta$ is derivable. For facts, $k = 0$. A forward chaining step derives $A\theta$ given that $A'_1,...,A'_k$ are derived literals. Given goal list $L = \{...G...\}$ and such substitution θ that $G\theta = A\theta$, every step of backward chaining replaces goal G with $A_1\theta,...,A_k\theta$ in L and also applies θ to the other goals in L.

Forward and backward chaining have the same inference power. Forward and backward chaining are usually considered in the context of Horn KBs but they are also applicable to non-Horn KBs. For non-Horn KBs, chaining methods can be extended by adding contrapositives to KB rules. Forward and backward chaining with both rules and contrapositives is called extended chaining. As shown in (Sakharov, 2021), extended chaining gives a procedural characterization of KB inference without RAA. Because of this, extended chaining will be used to assess models and calculi for such inference.

3 CONSTRAINED 3-VALUED MODELS

Determining whether a recursive function is partial or total is undecidable. The problem of finding whether a recursive function yields a definite value (i.e true or false) for given arguments is also undecidable (Rogers, 1967). Hence, it is not possible to effectively separate ground atoms with definite truth values from the rest. As it was first noted by Kleene (Kleene, 1952), classical 2-valued FOL models are not adequate in the presence of evaluable functions and predicates. Models for evaluable KBs should have at least three values: -1 (false), 1 (true), 0 (undefined). These values are assigned to all ground atoms. The meaning of 0 is that the atom predicate or a computable function occurring in the atom does not yield a value. Fortunately, three values are sufficient for models that are relevant to KB inference without reasoning by contradiction.

Multi-valued models are usually defined by truth tables (or functions) for logical connectives so that the truth values of ground formulas can be calculated. Let $|A|$ denote the truth value of ground literal A. The following equation defines the truth values for negation: $|\neg A| = -|A|$. This definition complies with classical 2-valued FOL models for definite values and gives a reasonable choice for the negation of atoms whose truth values are undefined. This is how negation truth values are defined in natural 3-valued logics (Avron, 1991). Inference for multi-valued logics is usually specified via nonstandard proof theories (Gottwald, 2001; Takano, 1998; Malinowski, 2014).

No other formulas than literals are produced during KB derivations. Because of this, legitimate models for KB inference without RAA can be defined by the above negation truth function and by constraints on truth values in ground instances of facts and rules as opposed to truth tables for other connectives.

Definition 1. *An assignment of truth values $-1, 0, 1$ to ground literals is a $\mathcal{M}_3$ model if $|\neg A| = -|A|$ for any ground atom A and the following constraints are satisfied:*

1. *A is a ground fact instance: $|A| = 1$*
2. *$A_0 \Leftarrow A_1 \wedge ... \wedge A_k$ is a ground rule instance: If $|A_i| = 1$ for $i = 1...k$, then $|A_0| = 1$.*
3. *$A_0 \Leftarrow A_1 \wedge ... \wedge A_k$ is a ground rule instance: If $|A_0| = -1$ and $|A_i| = 1$ for $i = 1...j-1$ and $i = j+1...k$, then $|A_j| = -1$.*

Literal A is valid regarding $\mathcal{M}_3$ if $|A'| = 1$ for all its groundings A' in all $\mathcal{M}_3$ models.

Theorem 1. *Extended chaining is sound and complete with respect to $\mathcal{M}_3$ models for consistent KBs.*

Proof. Soundness. Consider an arbitrary grounding of an extended forward chaining derivation. If ground literal B is the conclusion of GMP and ground literals $B_1, ..., B_k$ along with rule or contrapositive $B' \Leftarrow B'_1 \wedge ... \wedge B'_k$ are the premises, then the model constraints guarantee that $|B| = 1$ provided that $|B_i| = 1$ for $i = 1...k$. By induction on the depth of derivations, extended forward chaining is sound with respect to $\mathcal{M}_3$ models.

Completeness. Suppose A is a ground literal, $|A| = 1$ in all $\mathcal{M}_3$ models, and A is not derivable by chaining from KB facts and rules. Let us look at model M in which $|B| = 1$ for every ground literal B that is derivable by chaining from KB facts and rules, $|C| = -1$ for every such ground literal C that $\neg C$ is derivable, and $|D| = 0$ for every other ground literal D. Such model M exists for any consistent KB, and $|A| \neq 1$ in M.

Constraint 1 holds for M because ground instances of a facts are derivable. If constraint 2 is violated for ground rule instance $A_0 \Leftarrow A_1 \wedge ... \wedge A_k$, then all A_i for $i = 1...k$ are derivable by forward chaining. Hence, A_0 is derivable from the latter by applying GMP. If constraint 3 is violated for ground rule instance $A_0 \Leftarrow A_1 \wedge ...A_j... \wedge A_k$, then all A_i for $i = 1...j-1$ and $i = j+1...k$ are derivable by forward chaining. $\neg A_0$ is also derivable. Hence, $\neg A_j$ is derivable from $\neg A_0, A_1, ...A_{j-1}, A_{j+1}, ..., A_k$ by applying GMP to contrapositive $\neg A_j \Leftarrow \neg A_0 \wedge A_1 \wedge ...A_{j-1} \wedge A_{j+1} ... \wedge A_n$. Therefore M is a $\mathcal{M}_3$ model and the assumption about A cannot be true. □

Theorem 1 shows that $\mathcal{M}_3$ models characterize KB inference without RAA. $\mathcal{M}_3$ models differ from Kleene and Lukasiewicz 3-valued logics (Avron, 1991). $|P| = 1$ in all Kleene models for which rules $P \Leftarrow Q$ and $P \Leftarrow \neg Q$ are true. $|P|$ could be 0 in the respective $\mathcal{M}_3$ models. $|\neg R| = 1$ in all Lukasiewicz models for which fact $\neg S$ and rules $S \Leftarrow P \wedge R$, $P \Leftarrow Q$, $P \Leftarrow \neg Q$ are true. $|\neg R|$ could be 0 in the respective $\mathcal{M}_3$ models.

4 SEQUENT CALCULI

We rely on sequent calculi with non-logical axioms as an instrument for logical characterization of inference from evaluable KBs. A sequent is $\Gamma \vdash \Pi$ where Γ is an antecedent and Π is a succedent. Consider a variant of Gentzen's *LK* (Szabo, 1969) in which antecedents and succedents are multisets of formulas instead of sequences. This variant does not require the exchange rule. *LK* has one logical axiom $A \vdash A$. Let us exclude the contraction rule. The remaining structural rules are cut and weakening:

$$\frac{\Gamma \vdash A,\Delta \quad A,\Pi \vdash \Sigma}{\Gamma,\Pi \vdash \Delta,\Sigma}\ \mathit{cut}$$

$$\frac{\Gamma \vdash \Pi}{A,\Gamma \vdash \Pi}\ \mathit{LW} \qquad\qquad \frac{\Gamma \vdash \Pi}{\Gamma \vdash A,\Pi}\ \mathit{RW}$$

KB inference and logic programming are concerned about the derivation of literals, i.e. sequents of the form $\vdash A$ in the terminology of sequent calculi, A is a literal. KB facts and rules can be treated as non-logical axioms (Negri and Von Plato, 2001). 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 terms in instances of these axioms.

Standard cut elimination techniques can be applied even in the presence of non-logical axioms (Negri and Von Plato, 2001). Cut instances are moved upward so that one premise of every cut is a non-logical axiom. Therefore, the subformula property (Negri and Von Plato, 2001) holds for formulas of the forms $A \wedge B$, $A \vee B$, $A \Rightarrow B$, $\forall x A(x)$, $\exists x A(x)$, and $\neg C$ where C is not an atom. Hence, the logical rules for connectives $\wedge, \vee, \Rightarrow$ and for quantifiers are admissible in derivations of literals and so are all formulas except for literals. The only logical rules that are necessary in these derivations are two negation rules:

$$\frac{\Gamma \vdash A,\Pi}{\neg A,\Gamma \vdash \Pi}\ L\neg \qquad\qquad \frac{A,\Gamma \vdash \Pi}{\Gamma \vdash \neg A,\Pi}\ R\neg$$

Definition 2. *LK_{-c} is the set of sequent calculi in which formulas are literals, the structural rules are cut, LW, RW, the logical rules are $L\neg, R\neg$ restricted to atoms as the principal formulas, the logical axiom is $A \vdash A$, and non-logical axioms represent KB rules and facts.*

Theorem 2. *Any LK_{-c} derivation of a literal can be transformed into an extended chaining derivation and vice versa provided that the KB is consistent.*

Proof. Consider a backward chaining derivation. Every step of this derivation involving a KB rule is replaced with the cut rule whose premises are the respective goal list and rule instance. Any contrapositive instance $\neg A_i \Leftarrow \neg A \wedge A_1 \wedge ... A_{i-1} \wedge A_{i+1} ... \wedge A_n$ is obtained from rule instance $A \Leftarrow A_1 \wedge ... \wedge A_k$ by applying two negation rules if A and A_i are atoms. If A is negative, then cut is applied to the rule instance and $A, \neg A \vdash$. If A_i is negative, then cut is applied to the rule instance and $\vdash A_i, \neg A_i$. Every step of the chaining derivation involving a contrapositive is replaced with the cut rule applied to the respective goal list and contrapositive instance. Thus the entire chaining derivation is transformed into a LK_{-c} derivation.

Consider a LK_{-c} derivation with the endsequent $\vdash G$. The applications of the weakening rules can be moved below all other rules. We present relevant permutations for the cases that the weakening formula is principal in the following cut or negation rule. The other cases are even simpler and left to the reader.

$$\dfrac{\Gamma \vdash A,\Pi \quad \dfrac{\Delta \vdash \Sigma}{A,\Delta \vdash \Sigma}}{\Gamma,\Delta \vdash \Pi,\Sigma} \;\rightarrow\; \dfrac{\Delta \vdash \Sigma}{\overset{\cdots}{\Gamma,\Delta \vdash \Pi,\Sigma}}$$

$$\dfrac{\dfrac{\Gamma \vdash \Pi}{\Gamma \vdash A,\Pi} \quad A,\Delta \vdash \Sigma}{\Gamma,\Delta \vdash \Pi,\Sigma} \;\rightarrow\; \dfrac{\Gamma \vdash \Pi}{\overset{\cdots}{\Gamma,\Delta \vdash \Pi,\Sigma}}$$

$$\dfrac{\dfrac{\Gamma \vdash \Pi}{\Gamma \vdash \Pi,A}}{\Gamma,\neg A \vdash \Pi} \;\rightarrow\; \dfrac{\Gamma \vdash \Pi}{\Gamma,\neg A \vdash \Pi}$$

$$\dfrac{\dfrac{\Gamma \vdash \Pi}{A,\Gamma \vdash \Pi}}{\Gamma \vdash \neg A,\Pi} \;\rightarrow\; \dfrac{\Gamma \vdash \Pi}{\Gamma \vdash \neg A,\Pi}$$

Weakening cannot be the last rule in a derivation of $\vdash G$ because sequent $\vdash$ is not derivable from consistent KBs. Hence, weakening can be eliminated from derivations of sequents like this. Now let us move negation rules upward. The following permutations accomplish this.

$$\dfrac{\dfrac{\Gamma \vdash A,\Delta \quad A,\Pi \vdash B,\Sigma}{\Gamma,\Pi \vdash B,\Delta,\Sigma}}{\neg B,\Gamma,\Pi \vdash \Delta,\Sigma} \;\rightarrow\; \dfrac{\Gamma \vdash A,\Delta \quad \dfrac{A,\Pi \vdash B,\Sigma}{A,\neg B,\Pi \vdash \Sigma}}{\neg B,\Gamma,\Pi \vdash \Delta,\Sigma}$$

$$\dfrac{\dfrac{\Gamma \vdash A,\Delta,B \quad A,\Pi \vdash \Sigma}{\Gamma,\Pi \vdash B,\Delta,\Sigma}}{\neg B,\Gamma,\Pi \vdash \Delta,\Sigma} \;\rightarrow\; \dfrac{\dfrac{\Gamma \vdash A,\Delta,B}{\neg B,\Gamma \vdash A,\Delta} \quad A,\Pi \vdash \Sigma}{\neg B,\Gamma,\Pi \vdash \Delta,\Sigma}$$

$$\frac{\dfrac{\Gamma \vdash A,\Delta \quad A,\Pi,B \vdash \Sigma}{\Gamma,\Pi,B \vdash \Delta,\Sigma}}{\Gamma,\Pi \vdash \neg B,\Delta,\Sigma} \quad \to \quad \frac{\Gamma \vdash A,\Delta \quad \dfrac{A,\Pi,B \vdash \Sigma}{A,\Pi \vdash \neg B,\Sigma}}{\Gamma,\Pi \vdash \neg B,\Delta,\Sigma}$$

$$\frac{\dfrac{B,\Gamma \vdash A,\Delta \quad A,\Pi \vdash \Sigma}{B,\Gamma,\Pi \vdash \Delta,\Sigma}}{\Gamma,\Pi \vdash \neg B,\Delta,\Sigma} \quad \to \quad \frac{\dfrac{A,\Pi \vdash \Sigma}{\Pi \vdash \neg A,\Sigma} \quad \dfrac{\dfrac{B,\Gamma \vdash A,\Delta}{\neg A,B,\Gamma \vdash \Delta}}{\neg A,\Gamma \vdash \neg B,\Delta}}{\Gamma,\Pi \vdash \neg B,\Delta,\Sigma}$$

By repeatedly applying these transformations, all applications of the negation rules can be moved above all applications of cut.

If the right premise of cut contains G occurring in the endsequent and the left premise is the conclusion of another cut, then the following transformation is applied from top to bottom. If the left premise contains this G and the right premise is a conclusion of another cut, then this transformation is applied in the opposite direction.

$$\frac{\dfrac{\Gamma \vdash A,\Delta \quad A,\Pi \vdash B,\Sigma}{\Gamma,\Pi \vdash B,\Delta,\Sigma} \quad B,\Phi \vdash \Psi}{\Gamma,\Pi,\Phi \vdash \Delta,\Sigma,\Psi}$$

$$\updownarrow$$

$$\frac{\Gamma \vdash A,\Delta \quad \dfrac{A,\Pi \vdash B,\Sigma \quad B,\Phi \vdash \Psi}{A,\Pi,\Phi \vdash \Sigma,\Psi}}{\Gamma,\Pi,\Phi \vdash \Delta,\Sigma,\Psi}$$

Every such transformation reduces the total number of instances of cut not containing G. By repeatedly applying this transformation, we can receive such derivation that every sequent not containing this G is a KB fact instance, a KB rule instance, $A \vdash A$, or the endsequent of a derivation comprised of negation rules applied to the above.

Let us construct a backward chaining derivation from the transformed sequent derivation. The endsequent of any derivation comprised of negation rules with top sequent $A \vdash A$ has one of these forms: $A \vdash A$; $\neg A \vdash \neg A$; $\vdash A, \neg A$; $A, \neg A \vdash$. Applications of cut to $A \vdash A$ and $\neg A \vdash \neg A$ can be removed from any derivation. The endsequent of any derivation comprised of negation rules with a KB fact instance or a KB rule instance as the top sequent will be called KB sequent. KB sequents are rearrangements of fact/rule literals: a literal in the succedent becomes its complement in the antecedent and vice versa.

If G occurs in the right premise of cut and the left premise is a KB sequent, then (1) is interpreted as a backward chaining step using the fact $\vdash A$, the source rule of $\Gamma, -\Delta \vdash A$, or a contrapositive for goal list $A, \Pi, -\Sigma$. If G occurs in the left premise of cut and the right premise is a KB sequent, then (2) is interpreted as a backward chaining step using the fact $\vdash -A$, the source rule of $\Pi, -\Sigma \vdash -A$ or a contrapositive for goal list $-A, \Gamma, -\Delta$. In both cases, if A or $-A$ is a fact or rule head, then a KB rule is used. Otherwise, a contrapositive is used.

$$\frac{\Gamma \vdash A,\Delta \quad A,\Pi \vdash G,\Sigma}{\Gamma,\Pi \vdash G,\Delta,\Sigma} \qquad (1)$$

$$\frac{\Gamma \vdash A,G,\Delta \quad A,\Pi \vdash \Sigma}{\Gamma,\Pi \vdash G,\Delta,\Sigma} \qquad (2)$$

If G occurs in the right premise of cut, the left premise could also be $\vdash A, -A$. If G occurs in the left premise of cut, the right premise could also be $A, -A \vdash$.

$$\frac{\vdash A,-A \quad A,\Pi \vdash G,\Sigma}{\Pi \vdash -A,G,\Sigma} \qquad (3)$$

$$\frac{\Pi \vdash G,A,\Sigma \quad A,-A \vdash}{-A,\Pi \vdash G,\Sigma} \qquad (4)$$

Let us look at the topmost premise of cut containing G. It could be one of these sequents: $G \vdash G$; $\neg G \vdash \neg G$; $\vdash G, \neg G$; $G, \neg G \vdash$. In this case, the other premise of this cut specifies the rule of the first step of the backward chaining derivation. The topmost premise of cut containing G could also be a rearrangement of literals of a KB rule. In this case, this rule or its contrapositive is used in the first step of the backward chaining derivation.

Consider the sequence of cut instances in the top-down order. The premises that are not descendants of the topmost sequent with G in cut instances of the forms (1) and (2) specify the sequence of facts or rules in all the following backward chaining steps. The premises containing G specify goal lists. Cut instances of the forms (3) and (4) simply rearrange literals in sequents, they are skipped. The entire sequence maps to an extended chaining derivation of G. □

This proof shows that weakening is admissible in LK_{-c} for consistent KBs. Theorem 2 guarantees that LK_{-c} constitute a proof theory of KB inference without RAA. The constraints of $\mathcal{M}_3$ can be reformulated for non-logical axioms of LK_{-c} instead of KB rules and facts. The following corollary is basically a combination of Theorem 1 and Theorem 2.

Corollary 1. *Derivation of literals in LK_{-c} is sound and complete with respect to $\mathcal{M}_3$ models for consistent KBs.*

Clearly, Horn KBs can be characterized by LK_{-c} without the negation rules (cf. simple logic (Besnard and Hunter, 2018)). As shown earlier, logical rules other than the negation rules are admissible in derivations of literals in *LK* instances with non-logical axioms expressing KB rules and facts. Consequently, non-Horn KBs can be characterized by LK_{-c} extended with the contraction rule. All these calculi are more lucid than calculi for FOL in its entirety.

5 RELATED WORK

Logics with more limited capabilities than FOL are relevant to KB inference. Description logics (Russell and Norvig, 2009) are one example of that. Logics with limited inference capabilities also play an important role in argumentation. Simple logic has one inference rule - Modus Ponens - and no logical axioms (Besnard and Hunter, 2018). Simple logic characterizes Horn KBs. Direct derivations are defined in (Kakas et al., 2018) as natural deduction done without using RAA. These direct derivations are not limited to literals.

In addition to the FOL interpretation of KB rules, the sets of rules with the same predicate in their heads can also be treated as inductive definitions (Moschovakis, 1974). Such rules in FO(FD) (Hou et al., 2010) rely on the notation of FOL. The paper (Hou et al., 2010) considers inference in FO(FD) for finite domains only. Other formalizations of inductive definitions as extensions of FOL are investigated in (Brotherston and Simpson, 2010) and (Cohen and Rowe, 2018). Inference in these systems is more complicated than inference in FOL, it includes cyclic proofs.

The semantics of normal logic programs are based on the negation-as-failure paradigm (Denecker et al., 2017). The most researched among them are stable and well-founded semantics (Schlipf, 1995). The semantics of normal logic programs are closely related to the semantics of inductive definitions (Denecker and Vennekens, 2014). In comparison to FOL, inference procedures for stable and well-founded semantics (Shen et al., 2002; Yamasaki and Kurose, 2001) are more complicated. They have not been investigated as much as inference procedures for FOL and its fragments. Also, 3-valued models have been used for describing the semantics of logic programs (Hölldobler and Ramli, 2009; Naish, 2006).

Sequent calculi without contraction have been extensively investigated (Grishin, 1981; Dyckhoff, 1992; Ono, 2010). Sometimes they are referred to as affine logic. Direct predicate logic (Ketonen and Weyhrauch, 1984) is *LK* without contraction, the cut rule is admissible in this logic. *LK* without contraction is equivalent to its single-succedent version *LJ* without contraction and with the axiom of double negation (Ono, 2010). The latter is also known as stable logic (Negri and Von Plato, 2001). Models for *LK* without contraction are studied in (Ono, 2010; Grishin, 1981).

The difference of LK_{-c} is that its instances contain non-logical axioms, and the cut rule is essential. Usually, cut elimination is a central part of any investigation of sequent calculi. Non-logical axioms are an obstacle to the admissability of cut but cut can be pushed up to them. Non-logical axioms naturally represent KB rules and facts in sequent calculi. Properties of sequent calculi with non-logical axioms, including those in the form of so-called mathematical basic sequents, are investigated in (Negri and Von Plato, 2001). Axioms corresponding to KB rules/facts can be transformed to these sequents.

Sequent calculus derivations for Horn and hereditary Harrop formulas are researched in (Miller et al., 1991). A sequent calculus from (Harland et al., 2000) combines forward and backward chaining in linear logic. Forward and backward chaining are characterized by means of the focusing calculi in (Chaudhuri et al., 2008). The focusing calculi are used for specifying nonstandard logics such as linear intuitionistic logic. The inverse method can be adapted to these calculi. The paper (Chaudhuri et al., 2008) describes how to model forward and backward chaining in the focused inverse method.

Studies of proof systems for Lukasiewicz logic and substructural logics usually employ nonstandard logical connectives (Jeřábek, 2010; Girard, 1987). Our research is concerned about standard logical connectives as implication, negation, and conjunction occur in KB rules. Standard connectives are intuitive and well understood by software developers. In author's opinion, using nonstandard connectives in KBs that are developed and debugged by non-logicians is a recipe for problems, and those connectives should be avoided in KB rules and facts.

6 CONCLUSION

Evaluable KBs retain the expressiveness of non-Horn KBs and naturally integrate reasoning and computation including neural computing. Reasoning by contradiction is rather incompatible with evaluable KBs. Non-Horn KBs are representable in a language that is much simpler than the language of FOL.

The constrained 3-valued models specifying the semantics of evaluable KBs are quite straightforward since the constraints are projections of KB facts and rules. The simplicity of these models is due to the fact that the truth values are defined for literals only. Substructural sequent calculi LK_{-c} representing a proof theory for evaluable KBs are comprehensible due to few logical rules and to the mapping of KB rules/facts to non-logical axioms. LK_{-c} fall into the category of well-studied formal systems. It is fair to say that sequent calculi without the contraction rule can be viewed as proof theories of inference without RAA.

Intuitionists criticized classical logic because the law of excluded middle was abstracted from finite situations and extended without justification to statements about infinite collections (Brouwer and Heyting, 1975). The same argument is valid against RAA in application to evaluable KBs. Intuitionistic logic deals with this issue by redefining the semantics of implication and negation as the definition of Kripke models shows (Mints, 2000). KB inference without RAA addresses this issue by interpreting predicates as partial boolean functions while employing the classical semantics of logical connectives as regards to definite truth values.

REFERENCES

Avron, A. (1991). Natural 3-valued logics–characterization and proof theory. *The Journal of Symbolic Logic*, 56(1):276–294.

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

Brotherston, J. and Simpson, A. (2010). Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216.

Brouwer, L. and Heyting, A. (1975). *L.E.J. Brouwer Collected Works: Philosophy and foundations of mathematics. 1*. North-Holland Publishing Company.

Casas, A., Cabeza, D., and Hermenegildo, M. V. (2006). A syntactic approach to combining functional notation, lazy evaluation, and higher-order in LP systems. In *International Symposium on Functional and Logic Programming*, pages 146–162. Springer.

Chaudhuri, K., Pfenning, F., and Price, G. (2008). A logical characterization of forward and backward chaining in the inverse method. *Journal of Automated Reasoning*, 40(2-3):133–177.

Cohen, L. and Rowe, R. N. S. (2018). Uniform inductive reasoning in transitive closure logic via infinite descent. In *27th EACSL Annual Conference on Computer Science Logic*, volume 119, pages 17:1–17:16. LIPICS.

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

Denecker, M. and Vennekens, J. (2014). The well-founded semantics is the principle of inductive definition, revisited. In *14th International Conference on the Principles of Knowledge Representation and Reasoning*. AAAI.

Dong, H., Mao, J., Lin, T., Wang, C., Li, L., and Zhou, D. (2019). Neural logic machines. In *International Conference on Learning Representations*.

Dyckhoff, R. (1992). Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807.

Girard, J.-Y. (1987). Linear logic. *Theoretical computer science*, 50(1):1–101.

Gottwald, S. (2001). *A treatise on many-valued logics*. Research Studies Press.

Grishin, V. N. (1981). Predicate and set-theoretic calculi based on logic without contractions. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 45(1):47–68.

Hanus, M. (2013). Functional logic programming: From theory to Curry. In *Programming Logics*, pages 123–168. Springer.

Harland, J. A., Pym, D. J., and Winikoff, M. (2000). Forward and backward chaining in linear logic. *Electronic Notes in Theoretical Computer Science*, 37:1–16.

Hölldobler, S. and Ramli, C. D. P. K. (2009). Logic programs under three-valued Łukasiewicz semantics. In *International Conference on Logic Programming*, pages 464–478. Springer.

Hou, P., De Cat, B., and Denecker, M. (2010). FO (FD): Extending classical logic with rule-based fixpoint definitions. *Theory and Practice of Logic Programming*, 10(4-6):581–596.

Jeřábek, E. (2010). Admissible rules of Łukasiewicz logic. *Journal of Logic and Computation*, 20(2):425–447.

Kakas, A. C., Mancarella, P., and Toni, F. (2018). On argumentation logic and propositional logic. *Studia Logica*, 106(2):237–279.

Ketonen, J. and Weyhrauch, R. (1984). A decidable fragment of predicate calculus. *Theoretical Computer Science*, 32(3):297–307.

Kleene, S. C. (1952). *Introduction to metamathematics*. Van Nostrand.

Malinowski, G. (2014). Kleene logic and inference. *Bulletin of the Section of Logic*, 43(1/2):43–52.

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

Miller, D., Nadathur, G., Pfenning, F., and Scedrov, A. (1991). Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied logic*, 51(1-2):125–157.

Mints, G. (2000). *A Short Introduction to Intuitionistic Logic*. Kluwer Academic Publishers.

Moschovakis, Y. N. (1974). *Elementary induction on abstract structures*. North Holland Publiushing Co.

Naish, L. (2006). A three-valued semantics for logic programmers. *Theory Pract. Log. Program.*, 6(5):509–538.

Negri, S. and Von Plato, J. (2001). *Structural proof theory*. Cambridge University Press.

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

Rodríguez-Hortalá, J. and Sánchez-Hernández, J. (2008). Functions and lazy evaluation in Prolog. *Electronic Notes in Theoretical Computer Science*, 206:153–174.

Rogers, H. (1967). *Theory of recursive functions and effective computability*. McGraw-Hill Book Co.

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, 3rd edition.

Sakharov, A. (2021). Inference methods for evaluable knowledge bases. In *Software Engineering Application in Informatics*, Lecture Notes in Networks and Systems, pages 499–510. Springer.

Schlipf, J. S. (1995). The expressive powers of the logic programming semantics. *J. Comput. Syst. Sci.*, 51(1):64–86.

Serafini, L. and d'Avila Garcez, A. S. (2016). Logic tensor networks: Deep learning and logical reasoning from data and knowledge. In *11th International Workshop on Neural-Symbolic Learning and Reasoning*, volume 1768. CEUR-WS.org.

Shen, Y.-D., Yuan, L.-Y., and You, J.-H. (2002). SLT-resolution for the well-founded semantics. *Journal of Automated Reasoning*, 28(1):53–97.

Stickel, M. E. (1992). A Prolog technology theorem prover: a new exposition and implementation in Prolog. *Theoretical Computer Science*, 104(1):109–128.

Szabo, M. E., editor (1969). *The collected papers of Gerhard Gentzen*. North-Holland.

Takano, M. (1998). Sequent calculi for three-valued logics. *Tsukuba Journal of Mathematics*, 22(2):447–461.

Voronkov, A. (1995). On computability by logic programs. *Annals of Mathematics and Artificial Intelligence*, 15(3-4):437–456.

Yamasaki, S. and Kurose, Y. (2001). A sound and complete procedure for a general logic program in non-floundering derivations with respect to the 3-valued stable model semantics. *Theoretical Computer Science*, 266(1-2):489–512.