Quo Vadis KLM-style Defeasible Reasoning?

Adam Kaliski and Thomas Meyer

[1] University of Cape Town
[2] Centre for Artificial Intelligence Research

Abstract. The field of defeasible reasoning has a variety of frameworks, all of which are constructed with the view of codifying the patterns of common-sense reasoning inherent to human reasoning. One of these frameworks was first described by Kraus, Lehmann and Magidor, and is accordingly referred to as the KLM framework. Initially defined in propositional logic, it has since been imported into description and modal logics, and implemented into many defeasible reasoning engines. However, there are many ways in which this framework may be advanced theoretically, and many opportunities for it to be applied. This paper covers some of the most prominent areas of future work and possible applications of this framework, with the intention that anyone who has recently familiarized themselves with this approach may then have an understanding of the kind of work in which they could engage.

Keywords: Artificial Intelligence · Knowledge Representation and Reasoning · Defeasible Reasoning · Nonmonotonic logic

1 Introduction

The KLM framework [40,43] has been the subject of extensions since it was initially defined [12,23,34], as well as been implemented in defeasible reasoning engines [52]. The focus that it has received compared to other defeasible reasoning formalisms [2,48,60] is justified by the KLM framework having three core features: a well defined set of postulates, a preferential semantics, and relative computational efficiency. This enables a large degree of flexibility, as postulates may be dropped, or additional ones enforced.

However, there are both theoretical challenges and potential applications for this form of defeasible reasoning. This paper will attempt to compile some current areas for future work. The various areas and problems covered are not intended to be exhaustive, but is rather a selection of the most prominent fields for future work. The intention of this paper is to provide a brief overview of a selection of problems in this field for those who have a foundational understanding of the field, without necessarily knowing the main candidate areas for novel work.

This paper will first define the base propositional language and concepts in defeasible reasoning, and then describe popular frameworks in the field before describing the focus of this paper: the KLM framework. Chapter 3 will outline some applications for defeasible reasoning and then chapter 4 will outline theoretical work in defeasible reasoning with an eye towards the previously described applications.

2 Background

Although the core principles of this paper are independent of a specific language, in the spirit of the initial definition of the KLM framework the base language of this paper is the propositional logic $\mathcal{L}$, which is formed from a finite set of propositional atoms P, denoted with small Latin letters $p, q, r, ...$, along with the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ to form a set of well formed formulas in the usual way, denoted with small Greek letters: α, β, $\gamma ... \in \mathcal{L}$. A knowledge base, $\mathcal{K} \subseteq \mathcal{L}$, is a finite set of well formed formulas. Classical logical consequence, generated by Tarskian semantic entailment, will be denoted with $\models$. A consequence operator over some set of statements $\mathcal{K}$, will be denoted $\mathcal{C}n(\mathcal{K}) \subseteq \mathcal{L}$, such that $\mathcal{C}n(.)$ in general is a set of formulas in the language.

Classical entailment, $\models$, is informed by the regular semantics for propositional logic. Let $\mathcal{U}$ be the set of all valuations, which are denoted u, v... where each valuation is a function: $\mathcal{L} \mapsto \{T, F\}$ where T and F refer to true and false, respectively. For any formula $\alpha \in \mathcal{L}$, if it is the case that $u(\alpha) = T$, then it is the case that $u \Vdash \alpha$, read "u satisfies α". This can be extended to sets of formulas, such that for some $\mathcal{K}$, then $u \Vdash \mathcal{K}$ if and only if for every $\alpha \in \mathcal{K}$, $u \Vdash \alpha$. Satisfaction then defines classical entailment such that for some $\mathcal{K} \subseteq \mathcal{L}$ and for some $\alpha \in \mathcal{L}$, then $\mathcal{K} \models \alpha$ if and only if for every $u \in \mathcal{U}$ such that $u \Vdash \mathcal{K}$, it is the case that $u \Vdash \alpha$.

In classical logic, reasoning is patterned along Tarskian notions of entailment [54,63]. Broadly, Tarski identified three main properties for a reasonable operation for a consequence operator $\mathcal{C}n$:

1. Inclusion: $\mathcal{K} \in \mathcal{C}n(\mathcal{K})$.
2. Idempotence: $\mathcal{C}n(\mathcal{K}) = \mathcal{C}n(\mathcal{C}n(\mathcal{K}))$.
3. Monotonicity: if $\mathcal{K} \subseteq \mathcal{K}'$ then $\mathcal{C}n(\mathcal{K}) \subseteq \mathcal{C}n(\mathcal{K}')$.

Any consequence operator that satisfies the above three properties is referred to as a Tarskian operator. Inclusion simply enforces that everything explicitly stated to be the case is in fact entailed, and idempotence states that the entailment relation should derive all possible inferences given a set of statements. These two properties are relatively uncontroversial, however monotonicity warrants discussion.

Formally, monotonicity states that for any two sets of statements $\mathcal{K}$ and $\mathcal{K}'$ such that $\mathcal{K} \subseteq \mathcal{K}'$, then the set of inferences derivable from $\mathcal{K}$ must also be subsumed by the set of inferences derivable from $\mathcal{K}'$. The intuition is that adding information will never retract a conclusion. If an agent could draw an inference based on their knowledge at some point, then there is no information they could learn that would invalidate that inference. Classical, monotonic reasoning accurately models mathematical reasoning, as there is no defeasibility in mathematics: knowledge is built upon knowledge and there is no provable mathematical theorem that invalidates a previously proven mathematical theorem. However, it is relatively well established that this is not the case with human reasoning [59], as humans frequently revise their beliefs about the world according to new information.

Therefore, monotonic reasoning is necessarily incompatible with defeasibility. Defining a defeasible logic therefore requires dropping the property of monotonicity, and the construction of new properties and axioms so as to accurately model common sense reasoning patterns. These properties and axioms will then inform either a semantics or a proof theory for defining defeasible reasoning.

2.1 Defeasible reasoning

In general, defeasible reasoning is defining reasonable notions of logical consequence about knowledge that corresponds somewhat to the various ways that humans reason on a day to day basis. Handling exceptions in information in a sensible way, while still maintaining an intuitive method of modelling the information in question. The seminal example to illustrate the goal of defeasible reasoning is the following "birds" knowledge base:

1. `Bird` → `Flies`
2. `Penguin` → `Bird`
3. `Penguin` → ¬ `Flies`

Under classical logics and notions of consequence, the above set of formulas entails ¬`Penguin`. If `Penguin` were asserted, then the knowledge base is rendered inconsistent and therefore everything is entailed. Defeasible reasoning formalizes what it means for something to *usually* be the case, and what it means to draw a conclusion from given knowledge that is treated as somewhat speculative, and subject to retraction upon learning more information. If only points 1 and 2 were known in the example above, it would be desirable to draw the inference that penguins fly. However, when point 3 is added, it is then desirable that such an inference is retracted. This mechanism of defeasibility is not present in classical reasoning.

There have been a number of formalisms to capture the patterns of defeasible reasoning. Some of the most popular or well-known formalisms will be briefly covered before describing the framework that is the focus of this paper.

Belief revision Belief revision, first defined by Alchourron, Gärdenfors and Makinson (AGM) [1,2], models an agent's set of beliefs about the world by encoding them as a formal set of statements, referred to as a *belief set*, and defines operations that model the agent adjusting the belief set on receiving new information: revision and contraction. The revision operator models being told that a given statement, perhaps not currently derivable in the belief set, is true, and modifies the belief set such that it incorporates the new statement in a satisfiable way. Contraction is the inverse operation, where a statement is provided with the information that it is not inferred from the knowledge base, and the belief set is modified such that the statement is no longer entailed.

Circumscription Circumscription is a well-known formalism for defeasible reasoning, and one of the first such nonmonotonic logics described. First defined by McCarthy [48] and then revised by McCarthy again [49], it is one of the most expressive defeasible logics. Circumscription defines a predicate over the language that states whether or not a particular individual is normal or abnormal, and furthermore states how, or how not, it is abnormal exactly. This is how circumscription achieves its expressivity: a statement as imprecise as "besides x, there is something else abnormal about y". A drawback is that this places a burden on the modelling process to precisely capture the domain knowledge, including choosing which predicates are atypical, and in what way.

Default logic Default logic represents information as defaults with the intended meaning of "most x's are y's", or "typically a's are b's". It was first described by Reiter [60], and then revised by Reiter and Criscuolo [61]. It was originally devised to enrich first-order logic, by adding the notion that there are default states, assumptions that can be drawn as inferences in the absence of information to the contrary, as a solution to the same core problem of nonmonotonic reasoning: how to most effectively model information containing exceptions. Default logic does so by choosing to address a problem arising from modelling around the exceptions: the problem of inheritance for non-exceptional subclasses. Default logic was defined by Reiter proof-theoretically, but did not have a corresponding semantics. The lack of a model theory means that it can be difficult to choose between different extensions, having to instead rely on intuition about what kind of reasoning is suitable for a given domain [62]. However, Delgrande et al. [30] defined a semantics for default logic, along with a number of extensions.

2.2 KLM approach

The KLM framework was initially defined in propositional logic, and encoded defeasibility in an object-level binary connective, $\mid\!\sim$, that is intended to be read as "is typically" and forms a *defeasible implication*, e.g. the defeasible implication $\alpha \mid\!\sim \beta$ is to be read as "α typically implies β" [40,43]. Therefore, the concept that most birds fly, with some exceptions, can be reasonably represented using the statement $\texttt{bird} \mid\!\sim \texttt{flies}$, which conveys that birds typically fly, and so allows for the possibility that there are birds that do not fly. Contrast to the corresponding statement in classical logics, $\texttt{bird} \rightarrow \texttt{flies}$ which will directly contradict with any flightless birds present in the knowledge base. The inclusion of $\mid\!\sim$ in the language then requires the definition of a new language, $\mathcal{L}_D \coloneqq \mathcal{L} \cup \{\alpha \mid\!\sim \beta | \alpha, \beta \in \mathcal{L}\}$, which is the language created by extending $\mathcal{L}$ with $\mid\!\sim$ such that any two formulas in $\mathcal{L}$ can form a defeasible implication, but $\mid\!\sim$ may not be nested.

A corresponding notion of defeasible inference is also defined and denoted $\mid\!\approx$, and can be read as "defeasibly entails". Any such defeasible entailment relation satisfying the following set of properties referred to as the KLM properties [43,23], presented as follows in $\mathcal{L}_D$, is referred to as LM-rational [23]:

1. (LLE) Left logical equivalence: $\dfrac{\mathcal{K} \mid\!\approx \alpha \leftrightarrow \beta,\ \mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \beta \mid\!\sim \gamma}$

2. (RW) Right weakening: $\frac{\mathcal{K} \approx \alpha \rightarrow \beta,\, \mathcal{K} \approx \gamma \mid\sim \alpha}{\mathcal{K} \approx \gamma \mid\sim \beta}$
3. (Ref) Reflexivity: $\mathcal{K} \approx \alpha \mid\sim \alpha$
4. And: $\frac{\mathcal{K} \approx \alpha \mid\sim \beta,\, \mathcal{K} \approx \alpha \mid\sim \gamma}{\mathcal{K} \approx \alpha \mid\sim \beta \wedge \gamma}$
5. Or: $\frac{\mathcal{K} \approx \alpha \mid\sim \gamma,\, \mathcal{K} \approx \beta \mid\sim \gamma}{\mathcal{K} \approx \alpha \vee \beta \mid\sim \gamma}$
6. (CM) Cautious Monotonicity: $\frac{\mathcal{K} \approx \alpha \mid\sim \gamma,\, \mathcal{K} \approx \alpha \mid\sim \beta}{\mathcal{K} \approx \alpha \wedge \beta \mid\sim \gamma}$
7. (RM) Rational Monotonicity: $\frac{\mathcal{K} \approx \alpha \mid\sim \gamma,\, \mathcal{K} \not\approx \alpha \mid\sim \neg \beta}{\mathcal{K} \approx \alpha \wedge \beta \mid\sim \gamma}$

Each of the above properties is an encoding of a pattern of reasoning that is reasonable in a defeasible context. *LLE* essentially states that two statements that are classically equivalent should have the same defeasible consequences. *RW* continues what *LLE* started by stating that there is a weak form of transitivity when there is a classical logical dependency: if a statement α is a logical consequence of a defeasible consequence of β, then α is also a defeasible consequence of β. *Reflexivity* is a self-explanatory property, and simply enforces that every statement is a defeasible consequence of itself. *And* and *Or* govern how conjunction and disjunction interacts with defeasible consequence: *And* states that if two different statements are defeasible consequences from the same premises, then the conjunction of the two is a defeasible consequence from those premises. *Or* states that if a statement is a defeasible consequence of two different premises, then it is a defeasible consequence of the disjunction of those premises.

CM and *RM* are defeasible counterparts to classical monotonicity. Monotonicity states that *any* new information will never result in a retraction of an inference. *CM* is a modification of monotonicity for a defeasible language, and states that strengthening the premises α of a defeasible implication with a statement β will never result in the retraction of a defeasible conclusion of α provided that β was one of the defeasible conclusions of α to begin with. This change essentially leaves the door open for novel information, knowledge that was not previously derivable from current facts, to result in an inference being withdrawn. This weakening of monotonicity allows for defeasible statements. However, *CM* is still too weak [43], as it does not allow for certain, intuitive statements to be derived. In the case where the new information has nothing to do with an existing inference, it is possible that the existing inference may not be derived in the presence of the new information, even though there is no reason for it to be withdrawn. This argument motivates the addition of *RM*, which states that any new information that does not conflict with any existing knowledge or inferences will not result in a retraction. This is a stronger property than *CM*, and, in fact, in the presence of *RM*, *CM* is superfluous.

The KLM framework is based on a preferential semantics, where a preferential interpretation, $\mathcal{P} \coloneqq \langle S, \prec, l \rangle$ is defined as a set S of states, a partial order $\prec$ over S, and a mapping $l : S \mapsto \mathcal{U}$ that assigns to every state a valuation in $\mathcal{U}$ [43]. The class of preferential interpretations that correspond to the KLM properties defined above are referred to as *ranked* interpretations, and have the property that the partial order $\prec$ is modular, meaning that $\prec$ forms a total pre-order, and essentially generates a number of "tiers" populated by members of S [43,23]. Therefore, ranked interpretations are often instead characterized in the following way [23]: a ranked interpretation $\mathcal{R} : \mathcal{U} \mapsto \mathcal{N} \cup \{\infty\}$ is a function from the set of valuations $\mathcal{U}$ to the natural numbers with infinity, such that $\mathcal{R}(u) = 0$ for some $u \in \mathcal{U}$, satisfying the convexity property: for every $i \in \mathcal{N}$ such that $\mathcal{R}(u) = i$ for some $u \in \mathcal{U}$, then it is the case that for every $0 \leq j < i$ there is a $v \in \mathcal{U}$ such that $\mathcal{R}(v) = j$. The rank of a valuation in $\mathcal{R}$ essentially encodes how that ranked interpretation credits that valuation. The lower the rank of a valuation, the more normal, or typical, the ranked interpretation views the situation corresponding to the valuation, while valuations with rank ∞ represent impossible situations. A ranked interpretation $\mathcal{R}$ satisfies a defeasible implication $\alpha \mathrel{|\!\sim} \beta$ whenever for every $u \in \mathcal{U}$ such that $u \Vdash \alpha$ and there is no $v \in \mathcal{U}$ such that $v \Vdash \alpha$ and $\mathcal{R}(v) < \mathcal{R}(u)$, then $u \Vdash \beta$. That is, $\mathcal{R} \Vdash \alpha \mathrel{|\!\sim} \beta$ if and only if every valuation that is minimal in $\mathcal{R}$ that satisfies α also satisfies β. This can naturally be extended to knowledge bases: a ranked interpretation $\mathcal{R}$ satisfies a defeasible knowledge base $\mathcal{K}$ if and only if it satisfies every defeasible implication in $\mathcal{K}$. Note that this semantics allows classical formulas to be represented as defeasible implications as well, since for some ranked interpretation $\mathcal{R}$ then: $\mathcal{R} \Vdash \neg\alpha \mathrel{|\!\sim} \bot$ for some $\alpha \in \mathcal{L}$ if and only if $u \Vdash \alpha$ for every $u \in \mathcal{U}$ such that $\mathcal{R}(u) \in \mathcal{N}$. Ranked interpretations are linked to the KLM properties via a representation theorem [43], and every ranked interpretation, $\mathcal{R}$, generates a corresponding defeasible entailment relation $\mathrel{|\!\approx}_{\mathcal{R}}$ that satisfies all the KLM properties such that for some knowledge base $\mathcal{K}$ for which $\mathcal{R} \Vdash \mathcal{K}$ then $\mathcal{K} \mathrel{|\!\approx}_{\mathcal{R}} \alpha \mathrel{|\!\sim} \beta$ if and only $\mathcal{R} \Vdash \alpha \mathrel{|\!\sim} \beta$.

As an example, given the propositional logic over the set of propositions $P = \{p, q, r\}$, then the following is a possible ranked interpretation, $\mathcal{R}$, of some knowledge base $\mathcal{K}$, where valuations are represented as propositions in typewriter text, with a bar over a proposition indicating that it is not satisfied by the valuation:

∞	$\overline{\mathtt{pqr}}$ $\mathtt{p}\overline{\mathtt{qr}}$
1	$\mathtt{pq}\overline{\mathtt{r}}$ $\mathtt{p}\overline{\mathtt{q}}\mathtt{r}$ $\overline{\mathtt{p}}\mathtt{q}\overline{\mathtt{r}}$
0	$\mathtt{pqr}$ $\overline{\mathtt{pq}}\mathtt{r}$ $\overline{\mathtt{p}}\mathtt{qr}$

Then, by way of example, the above ranked interpretation forms a defeasible entailment, $\mathrel{|\!\approx}_{\mathcal{R}}$, such that $\mathcal{K} \mathrel{|\!\approx}_{\mathcal{R}} p \mathrel{|\!\sim} q$ and $\mathcal{K} \mathrel{|\!\approx}_{\mathcal{R}} q \mathrel{|\!\sim} r$.

3 Applications

3.1 Legal reasoning

Legal informatics formalizes laws and regulations such that artificial intelligence and data driven techniques can be used to analyse legal systems [28,37,58]. In particular, much attention has been focused on modelling a set of laws and regulations as a normative system: a set of ordered pairs of the form $\langle condition, consequence\rangle$ [37]. One of the main formalisms for describing a normative system is input/output logics [47], a logic that is built up of ordered pairs as the language. Defeasibility in input/output logics has not been widely studied, however there are other logics used for reasoning about laws that have been enriched with defeasible concepts.

One of the main languages that has been used to reason about laws and regulations is deontic logic [37]. Deontic logic is a type of modal logic where the modal operators, $\Box$ and $\Diamond$, are interpreted as "it is obligatory", or "it is permitted" [64]. This, naturally, is a useful framework for analysing legal problems that are concerned with the distinction between what *is* the case, and what *ought* to be the case. There are two main varieties of deontic logic: standard deontic logic (SDL), and dyadic standard deontic logic (DSDL) [33,58]. Standard deontic logic is made up statements of the form $\Box(\alpha)$, and $\Diamond(\alpha)$ with the intended reading of "α is obligatory" and "α is permitted", respectively. Dyadic standard deontic logic, however, is more expressive by being able to also express statements of the form $\Box(\alpha|\beta)$ and $\Diamond(\alpha|\beta)$ with the reading of "given β then α is obligatory" and "given β then α is permitted", respectively. DSDL is therefore useful for reasoning about scenarios where an agent has acted contrary-to-duty [58], by allowing for reasoning about what ought to be the case even in the case where another norm was violated, which implies a level of defeasibility already baked into DSDL.

The legal domain is a great candidate for defeasible reasoning, as laws and regulations are inherently defeasible. Grossi and Rotolo [37] identify three main areas of defeasibility in the law:

1. Conflicts
2. Exclusionary norms
3. Contributory factors

Conflicts arise where two legal norms both apply and lead to contradictory conclusions. These conflicts can themselves be categorized into three different scenarios: [37]

1. One norm is an exception to the other. This is resolved by *lex specialis* which gives priority to the more specific norm, the exception.
2. There exists a ranking between the norms, for example they could be from different authorities. In this case the conflict can be resolved by *lex superior* which gives priority to the higher ranked norm.
3. The norms could have been enacted at different times. In this case the principle of *lex posterior* will resolve the conflict by giving priority to the norm enacted most recently.

Exclusionary norms are legal norms that provide explicit conditions or methods to make other norms invalid, for example fulfilling criteria to make certain evidence inapplicable.

Contributory factors refer to the set of factors that help decide whether or not a norm is applicable. This is a product of the difficulty of precisely describing what criteria need to be met for some legal issue. For example, determining whether the use of a copyrighted piece of work falls under fair use depends on a number of loosely defined factors [37].

More generally, defeasibility is baked into the legal domain [37]. One possible reason is that legality is driven by human cognition, which is inherently defeasible [59]. Law is also a dialectical exercise where conclusions that may be heavily supported by current norms may be rejected.

Given that the legal sphere inherently contains such defeasibility, it then makes sense to use defeasible reasoning in legal informatics. Deontic logic enriched with defeasibility has been shown to solve well-known paradoxes [28], which suggests that defeasibility can be successfully used to enhance reasoning techniques.

3.2 Programming frameworks

Writing programs based on formal logic has been researched since the 1970s [3,46]. Logic programs are a set of rules that form a theory that corresponds to a knowledge base in a formal language, with the goal of computation rather than theorem proving.

One of the earliest formalisms for logic programming, Datalog was originally a database querying language, but has found far more general applicability [31]. Datalog syntax is reflective of database facts and schemas, and has been successfully applied as a declarative programming language, with use in a variety of fields.

Defeasible datalog [38,55] is an extension to disjunctive datalog [31] in which the the KLM postulates [43] may be expressed. It has been shown [38,55] that defeasible entailment relations may be algorithmically defined, and computed, in defeasible datalog. Specifically, algorithms to compute the rational closure [43], lexicographic closure [42], and relevant closure [22] of a defeasible knowledge base given in an extended defeasible datalog was given by Morris et al. [55], while a general defeasible reasoning system using datalog was defined by Harrison and Meyer [38]. Both of the above were defined syntactically, as algorithms over the statements in the knowledge bases themselves. A useful continuation of this work would be provide semantic characterizations of defeasible entailment relations for datalog, which would provide a platform for comparisons in other formalisms, such as in description logics, and allow for importing work already done into datalog.

Datalog itself has found applicability in artificial intelligence projects, specifically DLV [44] and RDFox [57]. RDFox is a RDF data-store that supports datalog reasoning services. Defeasible datalog implies that defeasible implementations of both RDFox and DLV are possible and worth investigation.

Another extension of datalog is that of datalog ± [20,21], an extension that adds quantification to rule heads while restricting syntax in various ways to improve the complexity. Of particular interest is the application of datalog ± to ontologies, as it is strictly more expressive than the description logic DL-Lite, and the potential for datalog ± to be applied to RDF stores [21]. As a direct result of these, datalog ± has the potential to be applied to the semantic web and other RDF systems such as RDFox [57]. The significance of the above in the context of this paper is the potential to therefore investigate enriching datalog ± with defeasible concepts in the vein of Morris et al. [55] and Harrison and Meyer [38]. Extending defeasible datalog to defeasible datalog ± is a natural theoretical continuation that could yield interesting results.

Another logic programming framework that has close ties to defeasible reasoning is answer set programming (ASP) [9,45]. Built around the concept of answer sets: consistent sets of formulas satisfying the constraints defined by the program, ASP has found many applications, from robotics, to planning, and to bioinformatics [32]. ASP has a fundamental link to defeasible reasoning, as answer sets are essentially consistent extensions to a knowledge base, in much the same way that default logic defines consistent extensions to default theories [9]. ASP programs are, in fact, fragments of default logic [60], and can also be represented as theories in nonmonotonic modal logics [50,51]. These features make ASP a promising formalism in which to translate work done in nonmonotonic logics, such as enriching ASP with object-level defeasibility.

4 Future work

4.1 Syntax Sensitivity

Syntactic entailment relations are defined by directly using the statements in a knowledge base, contrasted with entailment relations generated from a semantics. Syntactic methods introduce the property of *syntax sensitivity*. In short, syntax sensitivity is the property that, given two classically logically equivalent sets of statements that have differing syntax, a defeasible entailment relation will draw a particular inference from only one and not the other [7]. This introduces unpredictability: given the same defeasible entailment relation and logically equivalent, under classical semantics, knowledge bases, then the same inferences would be expected to hold. Therefore, investigating the causes of syntax sensitivity, and ways to avoid it is an interesting area of research.

Baral et al. [5] noted that defeasible reasoning can introduce syntactic sensitivity. The prototypical example of syntax sensitive reasoning is the difference between the two knowledge bases: $\mathcal{K} \coloneqq \{\alpha, \beta\}$, $\mathcal{K}' \coloneqq \{\alpha \wedge \beta\}$. Both $\mathcal{K}$ and $\mathcal{K}'$ are logically equivalent under classical semantics. However, there exists many defeasible entailment relations $\approx$ such that $\mathcal{K} \cup \{\neg\alpha\} \approx \beta$, but $\mathcal{K}' \cup \{\neg\alpha\} \not\approx \beta$ [5,6]. There are possible explanations for this feature to be not undesirable: perhaps α and β are observations from different sources, whereas $\alpha \wedge \beta$ is a single observation. Such a reading of the knowledge base may actually validate syntax sensitivity as a feature, not a bug - the form of the knowledge may be a significant aspect of the reasoning.

Consider the following knowledge bases: $\mathcal{K} \coloneqq \{\texttt{Penguin} \wedge \neg\texttt{Bird} \mid\!\sim \bot, \texttt{Bird} \mid\!\sim \texttt{Flies}, \texttt{Bird} \mid\!\sim \texttt{Wings}, \texttt{Penguin} \mid\!\sim \neg\texttt{Flies}\}$ and $\mathcal{K}' \coloneqq \{\texttt{Penguin}\wedge\neg\texttt{Bird} \mid\!\sim \bot, \texttt{Bird} \mid\!\sim$

$\texttt{Flies} \wedge \texttt{Wings}, \texttt{Penguin} \mathrel{|\!\sim} \neg\texttt{Flies}\}$. Note that under classical semantics, the *materializations* - defined as $\overrightarrow{\mathcal{K}} \coloneqq \{\alpha \to \beta \mid \alpha \mathrel{|\!\sim} \beta \in \mathcal{K}\}$, that is the set of classical counterparts for every defeasible implication in a knowledge base - of $\mathcal{K}$ and $\mathcal{K}'$ are logically equivalent, i.e. they are both satisfied by the same set of valuations. Given the lexicographic closure [42] as the defeasible entailment relation, $\approx_{LC}$, then it is the case that $\mathcal{K} \approx_{LC} \texttt{Penguin} \mathrel{|\!\sim} \texttt{Wings}$ and $\mathcal{K}' \not\approx_{LC} \texttt{Penguin} \mathrel{|\!\sim} \texttt{Wings}$. Therefore, the syntax of the knowledge base, while not having any effect under classical semantics, can change what a presumptive defeasible entailment relation will or will not infer.

In the context of KLM-style defeasible reasoning, defeasible entailment relations have a syntactic definition [23]. The more presumptive a defeasible entailment, the more syntactic sensitivity is introduced. This is an issue, as it places a burden on the modelling process to represent the information in such a way as to guarantee the correct inferences. Rather, having consistent behaviour for a particular entailment relation would be more desirable for both implementation and theoretical analysis of an entailment. Such a consistent entailment relation is referred as syntax insensitive, and investigating how to ensure that defeasible syntactic entailments are syntax insensitive is an ongoing, significant area of work.

Syntactic methods are very useful for defining algorithms to perform reasoning tasks, and as such this question has direct consequences on defining defeasible reasoning for logic programming. Specifically, the algorithms presented by Morris et al. [55] and Harrison and Meyer [38] use syntactic methods to define algorithms in datalog for computing defeasible entailment relations. Defining syntax insensitive entailment relations will therefore allow for various presumptive defeasible entailment relations that represent common-sense patterns of reasoning relevant to many domains to be defined in an identical manner, and behave reliably and predictably.

The primary research focuses suggested here for syntax sensitivity would be: is syntax sensitivity a significant property of an entailment relation such that it encodes a meaningful reading of the knowledge base, and is there a corresponding syntax insensitive defeasible entailment for any syntax sensitive defeasible entailment?

4.2 Explanation

Explanation in artificial intelligence is a growing area of interest [41], in part because of the opacity of machine learning techniques. However, it is also well established in logic based techniques [39], where the primary goal is to provide, for each inference, a justification: some minimal subset of the knowledge base from which the inference follows. The goals of explainability in knowledge representation are, broadly speaking, to understand entailments that are not obviously inferred by the knowledge, to fix a possibly bugged, or inconsistent, knowledge base, and to gain some better understanding of a knowledge base with which the user may not have prior experience [39]. So far, the majority of work in explainable AI has been in the context of classical reasoning [41], however there is foundational work on extending explainability to the realm of defeasibility [10,27].

Given a classical propositional knowledge base $\mathcal{K} \coloneqq \{\texttt{Bird} \rightarrow \texttt{Flies}, \texttt{Bird} \rightarrow \texttt{Wings}, \texttt{Penguin} \rightarrow \texttt{Bird}, \texttt{Penguin} \rightarrow \neg\texttt{Flies}, \texttt{Robin} \rightarrow \texttt{Bird}\}$, then any classical reasoning engine will claim that $\mathcal{K} \models \texttt{Robin} \rightarrow \texttt{Wings}$. A justification based explanation system will be able to go further and produce a minimal subset of $\mathcal{K}$ satisfying the inference, in this case the set [26]:

- $\texttt{Robin} \rightarrow \texttt{Bird}$
- $\texttt{Bird} \rightarrow \texttt{Wings}$

The above example may be simple, but for knowledge bases on the scale of tens or hundreds of thousands of statements, finding the reasoning behind a given entailment without an explanation engine, can be an excruciating task, if at all feasible [39].

In the context of a defeasible logic, the task of explanation is complicated by the inherent nonmonotonicity: a subset of a defeasible knowledge base may well entail an inference that the whole knowledge base does not. As yet, there is only preliminary work in explanation for defeasible reasoning [10,27], and so therefore it is ripe area for research.

An important aspect to consider for explanation is providing useful justifications in a natural language that considers the intended users [53]. Miller [53] provides a selection of features for a useful, or successful, explanation. Computing justifications for a defeasible inference should take such features into account, as the defeasible nature of an inference can prove both a significant aspect that is worth conveying to the users, while also being challenging to accurately convey. The difference between an inference that is actually classical (and therefore will not be retracted) and an inference that is defeasible, and so is a speculative entailment, may well be important information to deal with in a defeasible explanation engine.

Some key areas of research for defeasible explanation are: implementing and generalizing the work of Chama [26], and comparable work for defeasible logic programming formalisms. Chama [26] provided an algorithm for computing justifications for inferences in the rational closure of a defeasible knowledge base [43], and so a natural follow up would be implementing the algorithm in question in an application. Furthermore, the algorithm in question is designed for justifications of inferences entailed by the rational closure, and so an important project would either be generalizing the algorithm to function for any defeasible entailment, or at least for other specific defeasible entailments such as lexicographic closure [42].

4.3 Expressive logics

The KLM framework was first described using a propositional language [40,43], but there has been much work in implementing KLM-style connectives and semantics to modal logics [17,19] and description logics [11,12,14,15,16,18,24,25,35].

Importing nonmonotonic formalisms into more expressive logics is a natural progression of such work, as defeasibility is a different axis of expressivity on which to enrich description logics and modal logics.

Description logics [4] have a correspondence to the Web Ontology Language (OWL) [18,56] which is used to build various ontologies, such as the Gene Ontology [29], and is also the language that defines the semantic web [8,36]. Therefore, progress made introducing defeasibility in description logics has a direct path to enriching the semantic web with defeasible, common-sense patterns of reasoning, along with ontologies used to compile domain knowledge in general.

While defeasible TBox statements in description logics have been defined with respect to representation and semantics [13,15], an ongoing area of research is that of defeasible ABox reasoning [11,14]. While reasoning with a classical ABox has been defined [11], defining reasoning with respect to a defeasible ABox is an open question [12,35]. Additionally, there is also the opportunity to compile the various ways KLM-style defeasibility has been incorporated in description logics and provide an overview paper.

Investigating defeasible modal logics [16,17,19] has relevance to legal reasoning. As stated in section 3.1, the legal domain contains inherent defeasibility when modelling laws and regulations as they are presented. Since deontic logic is a major language for modelling the legal domain, it seems intuitive that a defeasible deontic logic is worth exploring for its ability to resolve at least some conflicts that arise between factual and normative detachment [28,58]. Some primary research areas for defeasible modal logics include investigating nonmonotonic entailment relations and enriching various specific modal logics with defeasibility [19].

5 Conclusion

This paper is intended to be an overview of the various open sub-fields and research questions regarding the KLM framework for defeasible reasoning. Primary theoretical sub-fields covered are: syntax sensitivity, explanation, and theoretical advancements for more expressive languages such as description and modal logics, with the view towards applications such as legal informatics, and logic programming projects such as RDFox and DLV, and the semantic web.

Work in syntax sensitivity has applications in logic programming projects, as they allow for syntactic formulations of defeasible reasoning to be implemented, with expected behaviours. Explanation has applications in any implementation of defeasible reasoning, by providing justifications for inferences allowing for understanding entailments and repairing defeasible knowledge bases. Defeasible reasoning for description logics has many possible applications, the most obvious being to enrich OWL with defeasibility, which has impact on many projects, including the semantic web. Similarly, defeasible modal logics is an impactful field of work, one application of many would be in legal informatics: enriching deontic logic with defeasibility to resolve well-known paradoxes.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. The journal of symbolic logic **50**(2), 510–530 (1985)
2. Alchourrón, C.E., Makinson, D.: On the logic of theory change: Contraction functions and their associated revision functions. Theoria **48**(1), 14–37 (1982)
3. Apt, K.R.: Logic programming. Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B) **1990**, 493–574 (1990)
4. Baader, F., Calvanese, D., Mcguinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation, and Applications (01 2007)
5. Baral, C., Kraus, S., Minker, J., Subrahmanian, V.S.: Combining knowledge bases consisting of first-order theories. Computational intelligence **8**(1), 45–71 (1992)
6. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: IJCAI. vol. 93, pp. 640–645 (1993)
7. Benferhat, S., Dubois, D., Prade, H.: How to infer from inconsisent beliefs without revising? In: IJCAI. vol. 95, pp. 1449–1455. Citeseer (1995)
8. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific american **284**(5), 34–43 (2001)
9. Brewka, G., Eiter, T., Truszczyński, M.: Answer set programming at a glance. Communications of the ACM **54**(12), 92–103 (2011)
10. Brewka, G., Ulbricht, M.: Strong Explanations for Nonmonotonic Reasoning, pp. 135–146. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-22102-7_6, `https://doi.org/10.1007/978-3-030-22102-7_6`
11. Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Rational defeasible reasoning for description logics. Tech. rep., University of Cape Town (2018)
12. Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Theoretical foundations of defeasible description logics. arXiv preprint arXiv:1904.07559 (2019)
13. Britz, K., Casini, G., Meyer, T., Moodley, K., Varzinczak, I.: Ordered interpretations and entailment for defeasible description logics. Tech. rep., Technical report, CAIR, CSIR Meraka and UKZN, South Africa (2013)
14. Britz, K., Casini, G., Meyer, T., Varzinczak, I.: A klm perspective on defeasible reasoning for description logics. In: Description Logic, Theory Combination, and All That, pp. 147–173. Springer (2019)
15. Britz, K., Meyer, T., Varzinczak, I.: Semantic foundation for preferential description logics. In: Australasian Joint Conference on Artificial Intelligence. pp. 491–500. Springer (2011)
16. Britz, K., Meyer, T., Varzinczak, I.: Normal modal preferential consequence. In: Australasian Joint Conference on Artificial Intelligence. pp. 505–516. Springer (2012)
17. Britz, K., Varzinczak, I.: Defeasible modalities. arXiv preprint arXiv:1310.6409 (2013)
18. Britz, K., Varzinczak, I.: Introducing role defeasibility in description logics. In: European Conference on Logics in Artificial Intelligence. pp. 174–189. Springer (2016)

19. Britz, K., Varzinczak, I.: From klm-style conditionals to defeasible modalities, and back. Journal of Applied Non-Classical Logics **28**(1), 92–121 (2018)
20. Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. Journal of Web Semantics **14**, 57–83 (2012)
21. Cali, A., Gottlob, G., Lukasiewicz, T., Marnette, B., Pieris, A.: Datalog+/-: A family of logical knowledge representation and query languages for new applications. In: 2010 25th Annual IEEE Symposium on Logic in Computer Science. pp. 228–242. IEEE (2010)
22. Casini, G., Meyer, T., Moodley, K., Nortjé, R.: Relevant closure: A new form of defeasible reasoning for description logics. In: Fermé, Eduardoand Leite, J. (ed.) Logics in Artificial Intelligence. pp. 92–106. Springer International Publishing, Cham (2014)
23. Casini, G., Meyer, T., Varzinczak, I.: Taking defeasible entailment beyond rational closure. In: European Conference on Logics in Artificial Intelligence. pp. 182–197. Springer (2019)
24. Casini, G., Straccia, U.: Rational closure for defeasible description logics. In: European Workshop on Logics in Artificial Intelligence. pp. 77–90. Springer (2010)
25. Casini, G., Straccia, U.: Lexicographic closure for defeasible description logics. In: Proc. of Australasian Ontology Workshop. vol. 969, pp. 28–39. Citeseer (2012)
26. Chama, V.: Explanation for defeasible entailment. Master's thesis, Faculty of Science (2020)
27. Chama, V., Meyer, T.: Explanation for defeasible entailment (2019)
28. Chingoma, J., Meyer, T.: Forrester's paradox using typicality (2019)
29. Consortium, G.O.: The gene ontology resource: 20 years and still going strong. Nucleic acids research **47**(D1), D330–D338 (2019)
30. Delgrande, J.P., Schaub, T., Jackson], W.K.: Alternative approaches to default logic. Artificial Intelligence **70**(1), 167 – 237 (1994). https://doi.org/https://doi.org/10.1016/0004-3702(94)90106-6, `http://www.sciencedirect.com/science/article/pii/0004370294901066`
31. Eiter, T., Gottlob, G., Mannila, H.: Disjunctive datalog. ACM Transactions on Database Systems (TODS) **22**(3), 364–418 (1997)
32. Erdem, E., Gelfond, M., Leone, N.: Applications of answer set programming. AI Magazine **37**(3), 53–68 (2016)
33. Gabbay, D., Horty, J., Parent, X., van der Meyden, Ron van der Torre, L.: Handbook of Deontic Logic and Normative Systems (01 2013)
34. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: Semantic characterization of rational closure: From propositional logic to description logics. Artificial Intelligence **226**, 1–33 (2015)
35. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L., et al.: Preferential vs rational description logics: which one for reasoning about typicality?. In: ECAI. pp. 1069–1070 (2010)
36. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: Owl 2: The next step for owl. Journal of Web Semantics **6**(4), 309–322 (2008)
37. Grossi, D., Rotolo, A., et al.: Logic in the law: A concise overview. Logic and Philosophy Today. College Publications (2011)
38. Harrision, M., Meyer, T.: Rational preferential reasoning for datalog (2019)
39. Horridge, M.: Justification based explanation in ontologies. The University of Manchester (United Kingdom) (2011)
40. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial intelligence **44**(1-2), 167–207 (1990)

41. Krötzsch, M., Stepanova, D.: Reasoning Web. Explainable Artificial Intelligence: 15th International Summer School 2019, Bolzano, Italy, September 20–24, 2019, Tutorial Lectures, vol. 11810. Springer Nature (2019)
42. Lehmann, D.: Another perspective on default reasoning. Annals of Mathematics and Artificial Intelligence **15**(1), 61–82 (1995)
43. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial intelligence **55**(1), 1–60 (1992)
44. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The dlv system for knowledge representation and reasoning. ACM Trans. Comput. Logic **7**(3), 499–562 (Jul 2006). https://doi.org/10.1145/1149114.1149117, `https://doi.org/10.1145/1149114.1149117`
45. Lifschitz, V.: Answer set programming. Springer (2019)
46. Lloyd, J.W.: Foundations of logic programming. Springer Science & Business Media (2012)
47. Makinson, D., Van Der Torre, L.: Input/output logics. Journal of philosophical logic **29**(4), 383–408 (2000)
48. McCarthy, J.: Circumscription—a form of non-monotonic reasoning. Artificial intelligence **13**(1-2), 27–39 (1980)
49. McCarthy, J.: Applications of circumscription to formalizing common sense knowledge. Artificial Intelligence **28**, 89–116 (1986)
50. McDermott, D., Doyle, J.: Nonmonotonic logic 1. Artificial Intelligence **13**, 41–72 (1980)
51. McDermott, D.: Nonmonotonic logic ii: Nonmonotonic modal theories. Journal of the ACM (JACM) **29**(1), 33–57 (1982)
52. Meyer, T., Moodley, K., Sattler, U.: Dip: A defeasible-inference platform for owl ontologies. CEUR Workshop Proceedings (2014)
53. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. CoRR **abs/1706.07269** (2017), `http://arxiv.org/abs/1706.07269`
54. Monk, J.D.: The contributions of alfred tarski to algebraic logic. The Journal of Symbolic Logic **51**(4), 899–906 (1986), `http://www.jstor.org/stable/2273903`
55. Morris, M., Ross, T., Meyer, T.: Defeasible disjunctive datalog. pp. 208–219. CEUR (03/12-06/12 2019), `http://ceur-ws.org/Vol-2540/FAIR2019_paper_38.pdf`
56. Motik, B., Patel-Schneider, P.F., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., et al.: Owl 2 web ontology language: Structural specification and functional-style syntax. W3C recommendation **27**(65), 159 (2009)
57. Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., Banerjee, J.: Rdfox: A highly-scalable rdf store. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Staab, S. (eds.) The Semantic Web - ISWC 2015. pp. 3–20. Springer International Publishing, Cham (2015)
58. Parent, X., van der Torre, L.: Detachment in normative systems: Examples, inference patterns, properties. IfCoLog Journal of Logics and their Applications **4**(9), 2295–3039 (2017)
59. Pollock, J.L.: Cognitive carpentry: A blueprint for how to build a person. Mit Press (1995)
60. Reiter, R.: A logic for default reasoning. Artificial intelligence **13**(1-2), 81–132 (1980)
61. Reiter, R., Criscuolo, G.: Some representational issues in default reasoning. Computers & Mathematics with Applications **9**(1), 15–27 (1983)

62. Shoham, Y.: A Semantical Approach to Nonmonotonic Logics, p. 227–250. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1987)
63. Tarski, A.: On the concept of logical consequence. Logic, semantics, metamathematics **2**, 1–11 (1936)
64. Von Wright, G.H.: Deontic logic. Mind **60**(237), 1–15 (1951)