Defeasible disjunctive datalog

Matthew Morris[1 [0000−0003−3337−7229]], Tala Ross[1 [0000−0002−4856−9410]], and Thomas Meyer[1,2 [0000−0003−2204−6969]]

[1] University of Cape Town, Cape Town, South Africa
[2] CAIR, South Africa

Abstract. Datalog is a declarative logic programming language that uses classical logical reasoning as its basic form of reasoning. Defeasible reasoning is a form of non-classical reasoning that is able to deal with exceptions to general assertions in a formal manner. The KLM approach to defeasible reasoning is an axiomatic approach based on the concept of plausible inference. Since Datalog uses classical reasoning, it is currently not able to handle defeasible implications and exceptions. We aim to extend the expressivity of Datalog by incorporating KLM-style defeasible reasoning into classical Datalog. We present a systematic approach to extending the KLM properties and a well-known form of defeasible entailment: Rational Closure. We conclude by exploring Datalog extensions of less conservative forms of defeasible entailment: Relevant and Lexicographic Closure.

1 Introduction

The KLM approach, proposed by Kraus, Lehmann and Magidor [8], is a well-known framework for defeasible reasoning. The KLM properties can be used to determine the rationality of different forms of defeasible entailment. The framework has been discussed at length in the literature for propositional logic [8,9,10] and description logics [2,3,11,14]. We present what we believe to be the first theoretical approach for extending the KLM framework to Datalog. We consider an extended form of Datalog, Disjunctive Datalog, which allows for disjunction in the head of Datalog clauses. We do not consider a semantic characterisation, in terms of a class of *ranked interpretations*, for the Datalog case. We instead provide an algorithmic definition.

There are two well-known forms of defeasible entailment satisfying the KLM properties: Rational Closure (RC) [10] and Lexicographic Closure (LC) [9]. Both are rational [4], with RC being the most conservative form of rational defeasible entailment, and LC a more permissive form. Another form of defeasible entailment, Relevant Closure (RelC) [2], has been proposed for description logics. It intuitively seems rational but does not satisfy all of the KLM properties. We provide algorithmic definitions of RC, LC and RelC, showing that RC and LC are still rational when converted to Datalog and that RelC is not.

In the next section we provide the relevant background material, after which we present our work on KLM-style defeasible entailment for the Datalog case. We conclude with a discussion of related work and suggestions for future work.

2 Background

2.1 Propositional Logic

Propositional logic [1] is a simple logic which is built up from a finite set $\mathcal{P}$ of propositional *atoms*, denoted by meta-variables $p, q, \ldots$. The language $\mathcal{L}$ of propositional logic is the set of all formulas, denoted by $\alpha, \beta, \ldots$, which are recursively defined as usual: $\alpha ::= \top \mid \bot \mid p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \rightarrow \alpha \mid \alpha \leftrightarrow \alpha$.

An *interpretation* is a function $I : \mathcal{P} \rightarrow \{T, F\}$ which assigns a single truth value to each atom. A formula $\alpha \in \mathcal{L}$ is satisfied by an interpretation I, denoted $I \Vdash \alpha$, if it can be evaluated to true by I in the usual recursive truth-functional way. We define the models of a finite set of formulas X to be $[\![X]\!] = \{I : I \Vdash \alpha, \alpha \in X\}$. We say that a set of formulas X entails a formula α, denoted by $X \models \alpha$, if $[\![X]\!] \subseteq [\![\{\alpha\}]\!]$.

2.2 KLM-style Defeasible Entailment

The *KLM approach* [8] is based on the concept of plausible inference, which is represented by defeasible implication operators of the form $\alpha \mid\!\sim \beta$. This is read as "typically, if α, then β".

Let a *knowledge base* $\mathcal{K}$ be a finite set of defeasible implications. The KLM framework answers the question: "What does it mean for a defeasible implication $\alpha \mid\!\sim \beta$ to be entailed by a knowledge base $\mathcal{K}$?". This is referred to as defeasible entailment, and denoted by $\mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta$.

Unlike classical entailment, it is well-accepted that defeasible entailment is not unique. There are multiple formalizations of defeasible entailment, such as *Rational Closure* [10], *Lexicographic Closure* [9], and *Relevant Closure* [2]. Lehmann and Magidor [10] proposed a set of rationality properties known as the *KLM properties*. They argue that if a defeasible entailment algorithm satisfies all the properties it is believed to be an acceptable form of defeasible entailment. We adopt this approach and refer to these forms of defeasible entailment as *LM-rational*. The KLM properties for propositional logic are stated below:

(Ref) $\mathcal{K} \mid\!\approx \alpha \mid\!\sim \alpha$

(LLE) $\dfrac{\alpha \equiv \beta,\ \mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \beta \mid\!\sim \gamma}$

(RW) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta,\ \beta \models \gamma}{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma}$

(And) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta,\ \mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta \wedge \gamma}$

(Or) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma,\ \mathcal{K} \mid\!\approx \beta \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \alpha \vee \beta \mid\!\sim \gamma}$

(CM) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \beta,\ \mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma}{\mathcal{K} \mid\!\approx \alpha \wedge \beta \mid\!\sim \gamma}$

(RM) $\dfrac{\mathcal{K} \mid\!\approx \alpha \mid\!\sim \gamma,\ \mathcal{K} \not\mid\!\approx \alpha \mid\!\sim \neg\beta}{\mathcal{K} \mid\!\approx \alpha \wedge \beta \mid\!\sim \gamma}$

2.3 Rational Closure

Rational closure is the most conservative form of defeasible entailment. We use the algorithmic definition [6], which we refer to as the *Rational Closure Algorithm*, as the sole definition of Rational Closure. The algorithm is split into two distinct sub-algorithms, proposed by Casini et al. [4]. The `BaseRank` algorithm is used to construct a ranking of the classical versions ($\overrightarrow{\mathcal{K}}$) of the statements in the defeasible knowledge base ($\mathcal{K}$), according to typicality of the statements.

Algorithm 1: `BaseRank`

Input: A knowledge base $\mathcal{K}$
Output: An ordered tuple $(R_0, \ldots, R_{n-1}, R_\infty, n)$

$i := 0$;
$E_0 := \overrightarrow{\mathcal{K}} := \{\alpha \to \beta \mid \alpha \mathrel{|\!\sim} \beta \in \mathcal{K}\}$;
repeat
  $E_{i+1} := \{\alpha \to \beta \in E_i \mid E_i \models \neg\alpha\}$;
  $R_i := E_i \setminus E_{i+1}$;
  $i := i + 1$;
until $E_{i-1} = E_i$;
$R_\infty := E_{i-1}$;
if $E_{i-1} = \emptyset$ **then**
  $n := i - 1$;
else
  $n := i$;
return $(R_0, \ldots, R_{n-1}, R_\infty, n)$

The `RationalClosure` algorithm is used to compute whether a defeasible implication is entailed by the knowledge base and uses the `BaseRank` algorithm.

Algorithm 2: `RationalClosure`

Input: A knowledge base $\mathcal{K}$ and a defeasible implication $\alpha \mathrel{|\!\sim} \beta$
Output: **true**, if $\mathcal{K} \mathrel{|\!\approx} \alpha \mathrel{|\!\sim} \beta$, and **false**, otherwise

$(R_0, \ldots, R_{n-1}, R_\infty, n) :=$ `BaseRank`$(\mathcal{K})$;
$i := 0$;
$R := \bigcup_{i=0}^{j<n} R_j$;
while $R_\infty \cup R \models \neg\alpha$ **and** $R \neq \emptyset$ **do**
  $R := R \setminus R_i$;
  $i := i + 1$;
return $R_\infty \cup R \models \alpha \to \beta$;

Note that while the query passed to `RationalClosure` must be expressed in terms of the defeasible implication operator, we can express any classical sentence α as a defeasible implication $\neg\alpha \mathrel{|\!\sim} \bot$ [4]. Thus, the algorithm can be used to check classical queries as well. Also note that the `RationalClosure` algorithm just reduces to a sequence of classical entailment checks.

2.4 Disjunctive Datalog

Datalog [5] is a more expressive logic than propositional logic and a popular query language for deductive databases [12,13]. Datalog is a simplified version of general logic programming. The language of Disjunctive Datalog is made up of function-free *Horn clauses* with the general form: $l_0 \wedge l_1 \wedge \cdots \wedge l_m \rightarrow l_{m+1} \vee l_{m+2} \vee \cdots \vee l_n$. Each literal l_i is either $\perp$ or is a positive atom of the form $p_i(t_0, \ldots, t_{k_i})$, where p_i is a *predicate symbol* and $t_0, \ldots, t_{k_i}$ are *terms*. A term is either a *constant* or a *variable*. In our version of Datalog, the left-hand side of the clause is referred to as the *body* and the right-hand side as the *head*. Horn clauses with a body are called *rules* and those without a body are called *facts*.

A *Herbrand Base* B^P is the set of all ground facts constructible from the symbols in a Datalog program P. A *Herbrand interpretation* assigns each constant symbol to itself and each predicate symbol to a set of predicates ranging over constant symbols, and is identified with a subset $\tau \subseteq B^P$. For any Herbrand interpretation τ, we define that $\perp$ is not in τ. A rule $l_0 \wedge l_1 \wedge \cdots \wedge l_m \rightarrow l_{m+1} \vee l_{m+2} \vee \cdots \vee l_n$ is true for Herbrand interpretation τ if and only if, for each substitution θ which replaces variables by constants, if $l_0\theta \in \tau, l_1\theta \in \tau, \ldots, l_m\theta \in \tau$ then at least one of $l_{m+1}\theta \in \tau, l_{m+2}\theta \in \tau, \ldots, l_n\theta \in \tau$ holds. A fact $l_0 \wedge l_1 \wedge \cdots \wedge l_m$ is true for Herbrand interpretation τ if and only if, for each substitution θ which replaces variables by constants, $l_0\theta \in \tau, l_1\theta \in \tau, \ldots, l_m\theta \in \tau$ all hold. A Herbrand interpretation τ is a *Herbrand model* of a set of Horn clauses X if and only if every clause in X is true for τ.

Entailment of Horn Clauses The semantics of standard Datalog only defines entailment of ground facts. However, for the purposes of this paper, we extend the semantics of Datalog to allow for classical entailment of non-ground clauses. We use the definition of entailment under Herbrand semantics for first-order logic [7]. A set of Horn clauses X entails Horn clause α, denoted by $X \models \alpha$, if and only if each Herbrand model of X is also a model of α.

Molecules as Combinations of Literals We introduce the idea of *molecules* as a shorthand for a combination of literals. A *disjunctive molecule*, denoted $\alpha^{\vee}$, is a combination of literals of the form: $l_1 \vee l_2 \vee \cdots \vee l_n$. A *conjunctive molecule*, denoted $\alpha^{\wedge}$, is a combination of literals of the form: $l_1 \wedge l_2 \wedge \cdots \wedge l_n$. A *molecule*, denoted α, is either a disjunctive molecule or a conjunctive molecule. Now a Disjunctive Datalog rule can be written as $\alpha^{\wedge} \rightarrow \beta^{\vee}$.

3 Defeasible Disjunctive Datalog

3.1 KLM-style Defeasible Rules

We represent plausible inference in Disjunctive Datalog using defeasible rules of the form: $b_1 \wedge \cdots \wedge b_m \mid\!\sim h_1 \vee \cdots \vee h_n$. This is read as "typically, if all of $b_1, \ldots, b_m$ are true, then at least one of $h_1, \ldots, h_n$ is true". We do not consider a semantic definition of defeasible rules. We will instead define defeasible rules by adapting rational defeasible entailment algorithms for Disjunctive Datalog.

3.2 Defeasible Entailment

Let knowledge base $\mathcal{K}$ be a finite set of defeasible rules. The main question of this paper is to algorithmically analyse *defeasible entailment* $\mathcal{K} \approx \alpha \mid\!\sim \beta$. That is, how do we answer the question: "*Can we conclude* $\alpha \mid\!\sim \beta$ *from a defeasible knowledge base* K*?*". When analysing different defeasible entailment algorithms, Lehmann and Magidor [10] advocate that the KLM properties be used to assess the rationality of these algorithms. We adopt this approach for Datalog and provide an extension of the KLM properties for Disjunctive Datalog.

A Motivation for Extending Disjunctive Datalog We find that, due to the restrictive nature of Datalog's syntax, none of the KLM properties can be expressed using Disjunctive Datalog without violating the syntax. However, we need to ensure that LM-rational forms of defeasible entailment satisfy all of the KLM properties. We argue that this is necessary, even though the reasoning described by some of these properties will never be computed by defeasible entailment algorithms for Disjunctive Datalog.

Let us consider an example where we can come to a conclusion that cannot be expressed in Datalog's syntax. Even though we cannot express that conclusion, we still want the algorithm to be able to compute it, otherwise the algorithm would not be *rational*. For example, we would want to be able to conclude $t(X) \mid\!\sim s(X) \wedge e(X)$ from $\{t(X) \mid\!\sim s(X),\ t(X) \mid\!\sim e(X)\}$.

Datalog+ Our proposed extension to Datalog, Datalog+, introduces the idea of compounds. Compounds, denoted by $A, B, \ldots$, are recursively defined from base literals l as follows: $A \ ::= l \mid \neg A \mid A \wedge A \mid A \vee A$. In Datalog+ a fact is a compound A and rules have the form $A \rightarrow B$.

Let τ be a Herbrand interpretation and consider some substitution θ which replaces variables by constants. We say that compound A is in τ under θ, denoted $A\theta \in \tau$, if and only if one of the following conditions holds, where B, Γ are compounds and l is a literal:

- $A = l$ and $l\theta \in \tau$
- $A = \neg B$ and $B\theta \notin \tau$
- $A = B \wedge \Gamma$, $B\theta \in \tau$ and $\Gamma\theta \in \tau$
- $A = B \vee \Gamma$ and $B\theta \in \tau$ or $\Gamma\theta \in \tau$

Herbrand interpretation τ is a model of fact A if and only if $A\theta \in \tau$ for every possible θ. Herbrand interpretation τ is a model of rule $A \rightarrow B$ if and only if, whenever $A\theta \in \tau$ for some θ, then $B\theta \in \tau$ for the same θ. A knowledge base $\mathcal{K}$ entails Datalog+ Horn clause (rule or fact) α, denoted by $\mathcal{K} \models \alpha$, if and only if each Herbrand model of $\mathcal{K}$ is also a model of α.

The KLM Properties Expressed in Datalog+ We state the KLM properties (in Datalog+) for Datalog below, where molecules α, β, γ are used as a shorthand.

$$\text{(LLE)}\ \frac{\models \alpha \to \beta, \models \beta \to \alpha,\ \mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \gamma}{\mathcal{K} \mathrel{|\!\!\approx} \beta \mathrel{|\!\sim} \gamma}$$

$$\text{(Ref)}\ \mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \alpha \qquad \text{(RW)}\ \frac{\mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \beta,\ \models \beta \to \gamma}{\mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \gamma}$$

$$\text{(And)}\ \frac{\mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \beta,\ \mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \gamma}{\mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \beta \wedge \gamma} \qquad \text{(Or)}\ \frac{\mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \gamma,\ \mathcal{K} \mathrel{|\!\!\approx} \beta \mathrel{|\!\sim} \gamma}{\mathcal{K} \mathrel{|\!\!\approx} \alpha \vee \beta \mathrel{|\!\sim} \gamma}$$

$$\text{(CM)}\ \frac{\mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \beta,\ \mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \gamma}{\mathcal{K} \mathrel{|\!\!\approx} \alpha \wedge \beta \mathrel{|\!\sim} \gamma} \qquad \text{(RM)}\ \frac{\mathcal{K} \mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \gamma,\ \mathcal{K} \not\mathrel{|\!\!\approx} \alpha \mathrel{|\!\sim} \neg\beta}{\mathcal{K} \mathrel{|\!\!\approx} \alpha \wedge \beta \mathrel{|\!\sim} \gamma}$$

4 Rational Closure for Datalog

In this section we propose a simple adaptation to the Rational Closure algorithms for the Disjunctive Datalog case.

4.1 Base Rank Algorithm

In the propositional case, we can rewrite a classical statement α as the defeasible statement $\neg\alpha \mathrel{|\!\sim} \bot$ and, hence, we can assume that all of the statements in our knowledge base are defeasible. It is not possible to rewrite classical clauses as defeasible rules for the Datalog case. Instead, the adapted version of the `BaseRank` algorithm, Algorithm 1, ranks the statements in a knowledge base $\mathcal{K} = D \cup C$, where D is the set of defeasible rules and C the set of classical clauses. It forms a ranking using only the defeasible statements by setting $E_0 := \overrightarrow{D}$ on *line 2*. Then, since the classical statements are all definite, it adds them to the the most typical level (the infinite level).

In the propositional case, a statement α is *exceptional* with respect to a set of statements X if $X \models \neg\alpha$. Datalog$^{\vee}$'s syntax does not include the negation connective $\neg$, so we use the $\bot$ literal to define a notion of falsehood, and hence exceptionality.

Proposition 1. *Let τ be a Herbrand interpretation. Then, τ is a model of $\neg\alpha$ under Datalog+ semantics iff τ is a model of $\alpha \to \bot$ under Datalog$^{\vee}$ semantics.*

The exceptionality of molecule α is now assessed using the entailment check $E_i \cup C \models \alpha \to \bot$ on *line 4*. Finally, when all the defeasible rules are ranked, `BaseRank` adds the classical clauses to the infinite level by setting $R_\infty := E_{i-1} \cup C$ on *line 8*.

4.2 Rational Closure Algorithm

As with the `BaseRank` algorithm, we choose to represent falsehood using the $\bot$ literal. The `RationalClosure` algorithm now uses the entailment check $R_\infty \cup R \models \alpha \to \bot$ on *line 4*. Under the assumption that we can compute classical entailment for Datalog$^{\vee}$, this adapted version of the `RationalClosure` algorithm can be used to check whether a rule $\alpha \mathrel{|\!\sim} \beta$ is defeasibly entailed by the knowledge base $\mathcal{K} = D \cup C$.

Proposition 2. *The adapted `RationalClosure` algorithm is LM-rational.*

The adapted algorithms and proof of LM-rationality (proofs for satisfaction of each KLM property) are provided in Appendix A and B.

5 Lexicographic Closure

It seems unnecessary for the Rational Closure algorithm to throw away an entire level of statements when there is a conflict. While it is true that a statement within the level is causing the conflict, there are other statements in the level that may have no effect on the conflict occurring. *Lexicographic closure* [9] takes a finer-grained approach to removing statements. It considers all possible subsets of worst-ranked statements and removes the smallest possible subset such that there is no longer a conflict. The semantic and algorithmic definitions of Lexicographic Closure for propositional logic are known and have been shown to be LM-rational [9]. In this section we provide an extension of Lexicographic Closure to the Datalog$^{\vee}$ case.

5.1 Lexicographic Closure for Propositional Logic

We adapt the definition of Lexicographic Closure for propositional logic provided by Casini et al.[4]. The new definition, in terms of the sub-algorithms `SubsetRank` and `LexicographicClosure`, can easily be adapted for Datalog$^{\vee}$.

The `SubsetRank` algorithm, Algorithm 3, constructs a new ranking of statements by using the base ranks $R_0, \ldots, R_{n-1}, R_\infty$ computed by the `BaseRank` algorithm. It adds new rank levels D_{i,n_i-1}, D_{i,n_i-2},..., $D_{i,1}$ in between each existing rank level R_i and R_{i+1}. Each level $D_{i,j}$ represents all the different ways of removing $|R_i| - j$ statements from R_i. The `Subsets`(X, k) function finds all possible subsets of size $k < n$ of a set X of size n.

Algorithm 3: `SubsetRank`

Input: A knowledge base $\mathcal{K}$
Output: An ordered tuple $(R_0, \ldots, R_k, R_\infty, k+1)$

$(B_0, \ldots, B_{m-1}, B_\infty, m) :=$ `BaseRank`$(\mathcal{K})$;
$i := 0$; $k := 0$;
repeat
  for $j := |B_i|$ **to** 1 **do**
    $S_{i,j} :=$ `Subsets`(R_i, j);
    $D_{i,j} := \bigvee_{X \in S_{i,j}} \bigwedge_{x \in X} x$;
    $R_k := D_{i,j}$;
    $k := k + 1$;
  $i := i + 1$;
until $i := m$;
$R_\infty := B_\infty$;
return $(R_0, \ldots, R_k, R_\infty, k+1)$

The `LexicographicClosure` algorithm ranks the statements in the input knowledge base $\mathcal{K}$ using the `SubsetRank` algorithm. It then checks whether the defeasible implication $\alpha \mid\!\sim \beta$ is defeasibly entailed by $\mathcal{K}$ in a manner equivalent to that used by the `RationalClosure` algorithm (`LexicographicClosure` is the same as `RationalClosure`, barring the use of `SubsetRank` instead of `BaseRank`.)

5.2 Lexicographic Closure for Datalog

In section 5.2 we extend the Lexicographic Closure algorithm for the propositional case to the Datalog case. We conclude the section by showing that our extended algorithm is LM-rational.

Rephrasing SubsetRank for Datalog The definition of Lexicographic Closure for the propositional case cannot directly be applied to the Datalog case. The statement $D_{i,j}$ is formed by combining statements from subset $S_{i,j}$ using $\wedge$ and $\vee$ connectives. It will violate Datalog$^{\vee}$'s syntax if $S_{i,j}$ contains multiple rules or multiple subsets of facts. However, the statement $D_{i,j}$ can be transformed into Conjunctive Normal Form (CNF) $D_{i,j} := D_1 \wedge D_2 \wedge \ldots \wedge D_n$, where:

$$\begin{aligned} D_i &:= \neg a_{i,1} \vee \ldots \vee \neg a_{i,r_i} \vee b_{i,1} \vee \ldots \vee b_{i,s_i} \\ &:= \neg(a_{i,1} \wedge \ldots \wedge a_{i,r_i}) \vee (b_{i,1} \vee \ldots \vee b_{i,s_i}) \\ &:= a_{i,1} \wedge \ldots \wedge a_{i,r_i} \rightarrow b_{i,1} \vee \ldots \vee \vee b_{i,s_i} \\ &:= \alpha_i^{\wedge} \rightarrow \beta_i^{\vee} \end{aligned}$$

Thus, $D_{i,j}$ can be rewritten as a conjunction of Disjunctive Datalog rules. Checking entailment from a conjunction of rules is equivalent to checking entailment from a set of the same rules. Hence, we can replace each statement $D_{i,j}$ with a set of Datalog$^{\vee}$ rules. On *line 7* of the `SubsetRank` algorithm, Algorithm 3, we now set $R_k :=$ `RNF`$(D_{i,j})$. The Rule Normal Form function `RNF`(Γ) takes an "extended" Disjunctive Datalog statement Γ as input and does the following:

1. Computes the Conjunctive Normal Form `CNF`(Γ).
2. Converts `CNF`(Γ) into a conjunction of clauses of the form $(\alpha_1^{\wedge} \rightarrow \beta_1^{\vee}) \wedge (\alpha_2^{\wedge} \rightarrow \beta_2^{\vee}) \wedge \ldots \wedge (\alpha_k^{\wedge} \rightarrow \beta_k^{\vee})$.
3. Converts the conjunction of clauses into a set of clauses $\{\alpha_1^{\wedge} \rightarrow \beta_1^{\vee}, \alpha_2^{\wedge} \rightarrow \beta_2^{\vee}, \ldots, \alpha_k^{\wedge} \rightarrow \beta_k^{\vee}\}$.
4. Returns the set of clauses.

Rephrasing LexicographicClosure for Datalog `LexicographicClosure` is the same as `RationalClosure` for the Datalog$^{\vee}$ case, with the exception that the adapted `SubsetRank` algorithm is used to rank statements on *line 1* instead of the `BaseRank` algorithm.

Proposition 3. *The adapted `LexicographicClosure` algorithm is LM-rational.*

The adapted algorithms and proof of LM-rationality are provided in Appendix C and D.

6 Relevant Closure for Datalog

6.1 Motivation for Relevant Closure

Here, we give the definition for Relevant Closure as provided by Casini et al. [2]. The algorithm is based on `RationalClosure`, with some slight changes. The main idea behind it is that not all statements in a level are responsible for being able to prove $R_\infty \cup R \models \neg\alpha$, given the query $\alpha \mathrel{|\!\sim} \beta$. This is the motivation for only throwing away the "relevant" statements in a level.

6.2 Algorithmic Definition

The algorithm for Relevant Closure, provided by Casini et al. [2], is defined in terms of ALC, a description logic. To make the algorithm easier to understand and convert to Datalog, we will first express it in terms of propositional logic.

Algorithm 4: `RelevantClosure`

Input: A knowledge base $\mathcal{K}$, a defeasible implication $\alpha \mathrel{|\!\sim} \beta$, and a partition $< R,\ R^- >$ of $\mathcal{K}$

Output: **true**, if $\mathcal{K} \mathrel{|\!\approx} \alpha \mathrel{|\!\sim} \beta$, and **false**, otherwise

$(R_0, \ldots, R_{n-1}, R_\infty, n) :=$ `BaseRank`$(\mathcal{K})$;
$i := 0$;
$R' := R$;
while $R_\infty \cup R^- \cup R' \models \neg\alpha$ **and** $R' \neq \emptyset$ **do**
  $R' := R' \setminus \{R_i \cap R\}$;
  $i := i + 1$;
return $R_\infty \cup R^- \cup R' \models \alpha \rightarrow \beta$;

In the partition $< R,\ R^- >$ of $\mathcal{K}$, R represents all statements *relevant* to the query $\alpha \mathrel{|\!\sim} \beta$. When throwing away statements from a level, the algorithm only considers these statements in R as eligible for removal. We say that a statement $\alpha \mathrel{|\!\sim} \beta$ is in the Relevant Closure of $\mathcal{K}$ if and only if the `RelevantClosure` algorithm returns **true** when given $\alpha \mathrel{|\!\sim} \beta$ and $\mathcal{K}$.

6.3 Defining Relevance

Now that the algorithm has been defined, the only work remaining is to define how to calculate the partition $< R,\ R^- >$ for a given query $\alpha \mathrel{|\!\sim} \beta$. Based on the ideas explored by Casini et al. [2], we would want R to contain exactly all the statements used to prove $\neg\alpha$. To formalize this, we present a sequence of definitions to gradually build up the idea of relevance.

Definition 1. *α is said to be* exceptional *for $\mathcal{K}$ if $\mathcal{K} \models \neg\alpha$.*

Definition 2. *Let $\mathcal{K}$ be a knowledge base, $\mathcal{J} \subseteq \mathcal{K}$ such that $\mathcal{J}$ only contains defeasible implications, and α a propositional sentence. Then $\mathcal{J}$ is said to be an* α-justification *w.r.t.$\mathcal{K}$ if α is exceptional for $\mathcal{J}$ and for any $\mathcal{J}' \subset \mathcal{J}$, α is not exceptional for $\mathcal{J}'$.*

Definition 3. *For a sentence* α *and knowledge base* $\mathcal{K}$, *let* $\mathcal{J}^{\mathcal{K}}(\alpha) = \{\mathcal{J} \mid \mathcal{J}$ *is an* α*-justification w.r.t.* $\mathcal{K}\}$. *Then* $\alpha \mid\sim \beta$ *is said to be in the* Basic Relevant Closure *of* $\mathcal{K}$ *if it is in the Relevant Closure of* $\mathcal{K}$ *w.r.t.* $\bigcup \mathcal{J}^{\mathcal{K}}(\alpha)$.

6.4 Minimal Relevant Closure

It could be argued that for Basic Relevant Closure, we are still considering too many statements as relevant to the query. This is because we consider *all* the statements in *all* α-justifications as relevant to proving that α is exceptional. However, we could instead consider only the statements of minimal rank from each α-justification as relevant, and still fix the exceptionality of α.

Definition 4. *For some set of justifications* $\mathcal{J} \subseteq \mathcal{K}$, *let* $\mathcal{J}^{\mathcal{K}}_{min} = \{\alpha \mid\sim \beta \mid r_{\mathcal{K}}(\alpha) \leq r_{\mathcal{K}}(\gamma)$ *for every* $\gamma \mid\sim \lambda \in \mathcal{J}\}$.

For a sentence α, *let* $\mathcal{J}^{\mathcal{K}}_{min}(\alpha) = \bigcup_{\mathcal{J} \in \mathcal{J}^{\mathcal{K}}(\alpha)} \mathcal{J}^{\mathcal{K}}_{min}$.

Then $\alpha \mid\sim \beta$ *is said to be in the* Minimal Relevant Closure *of* $\mathcal{K}$ *if it is in the Relevant Closure of* $\mathcal{K}$ *w.r.t.* $\bigcup \mathcal{J}^{\mathcal{K}}_{min}(\alpha)$.

6.5 Relevant Closure for Datalog

In terms of adapting the `RelevantClosure` algorithm for Datalog, no further work needs to be done beyond what has already been said for Rational Closure. To define a molecule α being *exceptional*, we simply need to be able to check entailment of negated molecules, which is something we already know how to do. The remainder of the definitions for both Basic and Minimal Relevant Closure only entail manipulating sets and checking the rankings of statements.

6.6 LM-Rationality

For this, we will use Minimal Relevant Closure as the definition for Relevant Closure. As shown by Casini et al. [2], Relevant Closure for propositional logic satisfies the properties *Ref*, *LLE*, *And*, and *RW*, and does not satisfy *Or*, *CM*, or *RM*. We will show that the same holds true for Relevant Closure for Datalog.

Let us consider the proofs that show that Rational Closure fulfills the KLM properties of *Ref*, *RW*, and *And*. The only difference `RelevantClosure` has from `RationalClosure` is the inclusion of the "relevance partition". Thus, the proofs can be re-used without editing, provided that the relevance partition is the same throughout the various queries.

The relevance partition is fully determined by the antecedent of the query (e.g. α in $\alpha \mid\sim \beta$), as can be seen in the definition of Minimal Relevant Closure. In the aforementioned properties, the antecedent is the same in all queries made to the algorithm. Hence, the proofs can be directly re-used to show that Relevant Closure fulfills the KLM properties of *Ref*, *RW*, and *And*.

The proof for satisfaction of the property *LLE* and the counter-examples for satisfaction of the properties *Or*, *CM*, and *RM* can be found in Appendix F. The counter-examples were adapted from the ALC case [2].

7 Conclusions

The main focus of this paper was to provide versions of defeasible reasoning for Disjunctive Datalog. To be able to express the KLM properties and the algorithm in Datalog, we motivated for extensions that would have to be made to the syntax and semantics of Datalog. We proved that Rational Closure for Datalog was LM-rational (i.e. it conforms to the KLM properties).

We introduced Relevant Closure and Lexicographic Closure as alternatives for computing defeasible entailment and adapted both of the algorithms for Datalog. We found that Lexicographic Closure is still LM-rational, but that Relevant Closure does not satisfy some of the KLM properties.

8 Future Work

Future work on this topic would most likely include finding a semantic definition of Rational Closure for Datalog, based on minimal models. Other future work could include an attempted adaptation of the Relevant Closure method for computing defeasible entailment, done in such a way that it satisfies the KLM properties, while still maintaining the basic ideas of Relevant Closure.

As another option, Casini et al. [4] showed that LM-rationality is necessary but not sufficient. The additional properties for Basic Defeasible Entailment proposed by Casini et al. [4] can be extended to Datalog. Furthermore, other properties that are specific to defeasible entailment for Datalog should be explored. Finally, there is also potential for an implementation of defeasible reasoning in Datalog. In a paper submitted to this conference, Harrison and Meyer present an implementation of a defeasible Datalog reasoner for Rational Closure.

9 Appendices

This full paper, with appendices, can be accessed online here.

References

1. Ben-Ari, M.: Mathematical Logic for Computer Science. Springer Science & Business Media, Rehovot, Israel (2012)
2. Casini, G., Meyer, T., Moodley, K., Nortje, R.: Relevant closure: A new form of defeasible reasoning for description logics. In: JELIA 2014: Logics in Artificial Intelligence. pp. 92–106. Springer, Funchal, Madeira, Portugal (2014)
3. Casini, G., Meyer, T., Moodley, K., Varzinczak, I.: Towards practical defeasible reasoning for description logics. In: Proceedings of the 26th International Workshop on Description Logics. pp. 587–599. CEUR Workshop Proceedings, Ulm, Germany (2013)
4. Casini, G., Meyer, T., Varzinczak, I.: Taking defeasible entailment beyond rational closure. In: JELIA 2019: Logics in Artificial Intelligence. pp. 182–197. Springer, Rende, Italy (2019)
5. Ceri, S., Gottlob, G., Tanca, L.: What you always wanted to know about datalog (and never dared to ask). IEEE Transactions on Knowledge and Data Engineering **1**, 146–166 (1989)
6. Freund, M.: Preferential reasoning in the perspective of Poole default logic. Artificial Intelligence **98**, 209–235 (1998)
7. Genesereth, M., Eric, K.: Introduction to Logic. Morgan & Claypool, Stanford, California (2013)
8. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44**, 167–207 (1990)
9. Lehmann, D.: Another perspective on default reasoning. Annals of Mathematics and Artificial Intelligence **15**(1), 61–82 (1995)
10. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence **55**, 1–60 (1992)
11. Moodley, K.: Practical Reasoning for Defeasible Description Logics. Ph.D. thesis, University of KwaZulu-Natal (2015). https://doi.org/10.31237/OSF.IO/DW5P2, `https://thesiscommons.org/dw5p2/`
12. Pasarella, E., Lobo, J.: A Datalog Framework for Modeling Relationship-based Access Control Policies. In: Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies - SACMAT '17 Abstracts. pp. 91–102. ACM Press, New York, New York, USA (2017). https://doi.org/10.1145/3078861.3078871
13. Shkapsky, A., Yang, M., Interlandi, M., Chiu, H., Condie, T., Zaniolo, C.: Big Data Analytics with Datalog Queries on Spark. In: Proceedings of the 2016 International Conference on Management of Data - SIGMOD '16. pp. 1135–1149. ACM Press, New York, New York, USA (2016). https://doi.org/10.1145/2882903.2915229, `http://dl.acm.org/citation.cfm?doid=2882903.2915229`
14. Straccia, U., Casini, G.: Defeasible inheritance-based description logics. Journal of Artificial Intelligence Research **48**, 415–473 (2013)