UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

COMPUTER SCIENCE HONOURS
FINAL PAPER
2018

Title: Defeasible Datalog: Introducing Defeasible Reasoning into the Declarative Programming Language Datalog

Author: Thomas Pownall

Project Abbreviation: DDLOG

Supervisor(s): Professor Thomas Meyer

Category	**Min**	**Max**	**Chosen**
Requirement Analysis and Design	0	20	0
Theoretical Analysis	0	25	25
Experiment Design and Execution	0	20	0
System Development and Implementation	0	15	0
Results, Findings and Conclusion	10	20	20
Aim Formulation and Background Work	10	15	15
Quality of Paper Writing and Presentation	10		10
Quality of Deliverables	10		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	
Total marks		**80**	

Defeasible Datalog: Introducing Defeasible Reasoning into the Declarative Programming Language Datalog

Thomas Pownall
University of Cape Town
Department of Computer Science
Cape Town, South Africa
pwntho001@myuct.ac.za

ABSTRACT

Formal logic systems are central to developing artificial intelligence systems that reason similarly to human logic. However, these systems are not well apt to handle exceptions in knowledge. This paper investigates extending Datalog - a declarative programming language, with defeasible reasoning in order to achieve logical reasoning that handles exceptions.

We define our extended version of Datalog, *Defeasible Datalog*, which includes the syntax and semantics for defeasible rules. We then investigate an algorithm - rational closure - that will be used as the reasoner for *Defeasible Datalog*.

We show that rational closure is a logically appropriate algorithm for reasoning with classical and defeasible rules. We do this by showing our algorithm is a rational relation as defined by Lehman et al. [11] That is our algorithm satisfies the KLM properties. We conclude that *Defeasible Datalog* is an accurate description and reasoning procedure and thus, it is a possible solution to handling exceptions in formal logic systems.

KEYWORDS

Nonmonotonic reasoning, Defeasible reasoning, Datalog

1 INTRODUCTION

We are constantly faced with analysing information and using our analysis to form conclusions and make critical decisions. The information that we are given is critical in making our conclusions. If I want to choose the optimal route to work I must consider factors such as the weather, the day of the week and the time of day. There are situations where this information may be unknown or changes, either of which could impact our previous conclusion. Interestingly, another person with the same information may form a different conclusion than my own. This demonstrates the innate complexity of forming conclusions.

Many formalised systems have been created that attempt to model human logic. These systems provide requirements to the given information as well as procedures for forming unique conclusions. These conclusions are unique in the sense that given the same information the same conclusion should be formed. Although many of these systems are impressive in their ability to analyse information and form conclusions there are situations where these systems fall short of human logic. One such situation is in handling exceptions that arise within the given information. We feel it is best to highlight such a situation with an example.

CAPETOWN'18, September 2018, Cape Town, South Africa

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

EXAMPLE 1. Consider the following given information:

- All birds can fly
- A penguin is a bird
- *Conclusion:* A penguin can fly.

The above example illustrates how using a formalised system, such as classical logic, would reasonably conclude that penguins can fly. However, if we were to introduce a new given fact - *penguins cannot fly* - this formalised system would conclude - *penguins do not exist* - a conclusion that falls short of human logic as we understand penguins to be an exception to the - *all birds can fly* - rule. The importance of handling exceptions has led to the creation of new formalised systems in the field of nonmonotonic reasoning, and in particular a system called defeasible reasoning.

We aim to take Datalog, a declarative logic programming language, and extend its logic reasoning procedure to allow for defeasible reasoning. Datalog has applications in data integration, information extraction, networking, program analysis, security and cloud computing. Thus extending it with defeasible reasoning will allow formalised exception handling to be used in real-world applications. We will herein, refer to this extension as *Defeasible Datalog*. With this aim in mind, we separate the creation of *Defeasible Datalog* into two sections - theory and implementation. This paper is focused on the theory section. That is, this paper describes how such a system should be created in terms of syntax, semantics and reasoning procedure and provides evidence of the logical soundness of the description. The implementation section is a sister project and is handled by Joshua Abraham. The implementation section provides evidence that the *Defeasible Datalog* description provided in this paper can be implemented successfully in an application environment[17].

In order to describe *Defeasible Datalog* we first decided upon a reasoning procedure. Kraus et al. [10] recorded that any reasonable nonmonotonic reasoner should define a rational relation. Further Magidor et al. [11] outlined a procedure for nonmonotonic reasoning. We define our procedure with the above in mind and from that deduce the requirements for the syntax and semantics of *Defeasible*

Datalog. We then show that our procedure represents a rational relation and conclude that *Defeasible Datalog* has logically extended Datalog with defeasible reasoning.

With the above in mind the structure of the paper is as follows. Section 2 describes the work that has been done in relation to this project and how this project differentiates itself from the related work. Section 3 describes the syntax and semantics of *Defeasible Datalog*, the reasoning procedure that will be used in *Defeasible Datalog* as well as the evidence to show that our reasoning procedure defines a rational relation. Section 4 comments on the results and makes a final conclusion. Lastly, section 5 gives thanks to those that made this paper possible.

2 BACKGROUND AND RELATED WORK

In this section we discuss the work currently done in this field, the importance of such work, how that work relates to this paper and the importance of this paper within the literature.

It has been an important problem within the field of artificial intelligence to handle retraction of information given other information. A simple illustration of the necessity of being able to retract information is when handling exceptions to general rules. Consider that mammals are viviparous, that is they give birth to living young. We find that a platypus is a mammal, however we also find that platypus are oviparous, that is they lay eggs. Clearly, we need a system for handling such exceptions. A number of systems include, negation as failure[4], circumscription[14], the modal system[15], default logic[20], autoepistemic logic[16] and inheritance systems[21] as well as others[7, 8]. These systems all fall into nonmonotonic reasoning. Interestingly, Kraus et al.[10] developed a framework for comparing such systems and described preferential models. Furthermore, Lehmann and Magidor who worked on the framework went further.[11] Their paper argued that any reasonable nonmonotonic inference procedure should define a rational relation, in particular they outlined a procedure called rational closure. We have found many papers in the literature that extend description logics with rational closure.[2, 3, 9]

Our paper is interested in a particular area of nonmonotonic reasoning called defeasible reasoning.[19] There have been attempts to implement defeasible reasoning in Prolog[18], of which Datalog is a subset, as well at investigations into the efficiency of such systems.[12] In particular, DeAgustini et al.[5, 6, 13] have investigated into defeasible reasoning within Datalog ±, a family of Datalog that is focused on online ontologies. It is important to note however that DeAgustini et al. did not use rational closure as we attempt to do.

There is no evidence known to us that demonstrates the aim of this project. That is, there is no evidence that shows Datalog extended with defeasible reasoning using rational closure as defined by Magidor et al.[11] We believe this project can fill this gap and improve the understanding of defeasible reasoning for real-world application.

3 DEFEASIBLE DATALOG

In this section we first describe Datalog and then using a similar style we describe *Defeasible Datalog*. We have done this to demonstrate to the reader the subtle changes in syntax and semantics that will allow us to reason defeasibly. We then define the rational closure procedure[11], which will be used as the reasoner for *Defeasible Datalog*. Lastly, we show that our definition of the rational closure procedure is a rational relation as defined by Kraus et al.[10]

3.1 *Defeasible Datalog* Syntax and Semantics

Datalog is a declarative logic programming language that is often used as a query language for deductive databases. Below we define the typical syntax and semantics for Datalog using a bottom-up approach.

DEFINITION 1. We start with a (finite) set of *predicates*, say $P_1 ... P_n$. Each one has an arity, which simply indicates how many variables it has.

EXAMPLE 2. A predicate of arity one has one variable i.e. $Fly(x)$. A predicate of arity two has two variables i.e. $BrotherOf(x, y)$.

DEFINITION 2. An *atom* is something of the form $P(a)$ where P is a predicate of arity m, and a is an ordered $m - tuple$ of variables.

EXAMPLE 3. A 1-tuple is simply (x), while a 2-tuple would be (x,y). $Fly(x)$ is an atom with a predicate of arity one and an ordered 1-tuple. $BrotherOf(x, y)$ is an atom with a predicate of arity two and an ordered 2-tuple.

DEFINITION 3. A *Datalog rule* is written: $A_1(x_1), ..., A_n(x_n) \rightarrow A_0(x_0)$ where $n \geq 0$, $A_0, ..., A_n$ are predicate symbols and $x_0, ..., x_n$ are tuples of varying length and the commas separating atoms represent conjunction. It should be noted that the atoms need not have the same tuple length or even common variables. Further, the variables in the tuple of the head of our atom need not occur in any of the tuples in the atoms of the body and vice versa. In this way we it is understood as a pool of variables.

Logically what Datalog rules represent is that if $A_1, ..., A_n$ is true then A_0 is also true. As a result of the conjunction between atoms we require each A_i to be true in order to say $A_1, ..., A_n$ is true and thus that A_0 is also true.

EXAMPLE 4. A Datalog rule:

$FatherOf(x, y), Brother(x, z) \rightarrow NephewOf(y, z)$

This reads, if x is a father of y and x is a brother of z then y is a nephew of z. One may want to say a similar rule for the uncle relationship:

$FatherOf(x, y), Brother(x, z) \rightarrow UncleOf(z, y)$

An alternate approach using our first rule would be:

$NephewOf(y, z) \rightarrow UncleOf(z, y)$

DEFINITION 4. A *Datalog program* is then simply a finite collection of Datalog rules.

The standard version of Datalog lacks some of the expressiveness that we require for this project. We aim to extend Datalog with defeasible reasoning using rational closure and in order to do this we require the logical connectives negation and disjunction as well as a way of representing our defeasible rules.

Defeasible Datalog is described below, again using a bottom-up approach.

As before we start with a finite set of *predicates* $P_1 ... P_n$, each with its own arity. An *atom* is as before, an expression of the form $P(a)$ where P is an atom of arity m, and a is a an ordered $m - tuple$ of variables.

DEFINITION 5. A *literal* is either an atom $P(a)$ or the negation of an atom written $\neg P(a)$.

DEFINITION 6. A *clause* is a disjunction of literals i.e. $FatherOf(x, y) \lor UncleOf(x, y)$.

DEFINITION 7. A *rule* in *Defeasible Datalog* is written: $A(x) \rightarrow B(x)$, where $A(x)$ is a conjunction of clauses and $B(x)$ is a conjunction of literals. Note that there are no disjunctions in the head of a rule - they appear only in the body.

DEFINITION 8. For our purposes, a *defeasible rule* in *Defeasible Datalog* is written: $A(x) \rightsquigarrow B(x)$, where $A(x)$ is a conjunction of clauses and $B(x)$ is a conjunction of literals. Again note that there are no disjunctions in the head of a defeasible rule - they appear only in the body. Logically what this means is if $A(x)$ is true then $B(x)$ is *typically* true.

EXAMPLE 5. A rule in *Defeasible Datalog*:

$$(A(X) \lor B(x, y)),\ C(x, z) \rightarrow D(x, y, z), E(y, z)$$

A defeasible rule in *Defeasible Datalog*:

$$(A(x) \lor B(x, y)),\ C(x, z) \rightsquigarrow D(x, y, z), E(y, z)$$

DEFINITION 9. As before, a *Defeasible Datalog program* is simply a finite collection of *Defeasible Datalog* rules (both classical and defeasible).

DEFINITION 10. A *knowledge base* made from *Defeasible Datalog* will be of the form: $\mathcal{K} =< T, D >$ where T is the set of all *Defeasible Datalog* rules and D is the set of all defeasible *Defeasible Datalog* rules.

3.2 Rational Closure for *Defeasible Datalog*

In this section we describe the procedure that will be used for reasoning with *Defeasible Datalog*. We begin with the outline first defined by Magidor et al. [11] We then adapt procedures defined in the literature [1] to suit the syntax and semantics of Datalog.

We begin by assigning base ranks to defeasible rules, this forms the basis of the procedure for computing rational closure, which can be reduced to a number of classical entailment checks.

Define the *materialisation* of a *Defeasible Datalog* knowledge base $\mathcal{K}$ as $\overrightarrow{\mathcal{K}} \equiv_{\text{def}} \{A(x) \rightarrow B(x) \mid A(x) \rightsquigarrow B(x) \in \mathcal{K}\}$. It can then be shown that a rule $A(x) \rightarrow B(x)$ is exceptional with respect to $\mathcal{K}$ if and only if $\overrightarrow{\mathcal{K}} \models \neg A(x)$. From this we can define a procedure `BaseRank` which partitions the materialisation of $\mathcal{K}$ into $n + 1$ equivalence classes according to base rank: $i = 0, \ldots n - 1, \infty$, $R_i \equiv_{\text{def}} \{A(x) \rightarrow B(x) \mid A(x) \rightsquigarrow B(x) \in \mathcal{K}, br_{\mathcal{K}}(A(x)) = i\}$.

Algorithm 1: `BaseRank`

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

$i := 0$;
$E_0 := \overrightarrow{\mathcal{K}}$;
repeat
  $E_{i+1} := \{A(x) \rightarrow B(x) \in E_i \mid E_i \models \neg A(x)\}$;
  $R_i := E_i \setminus E_{i+1}$;
  $i := i + 1$;
until $E_{i-1} = E_i$;
$R_\infty := E_{i-1}$;
if $E_{i-1} = \emptyset$ **then**
  $n := i - 1$;
else
  $n := i$;
return $(R_0, \ldots, R_{n-1}, R_\infty, n)$

We use the `BaseRank` procedure to help us define the procedure for rational closure. It takes as input a knowledge base $\mathcal{K}$ and a defeasible rule $A(x) \rightsquigarrow B(x)$, and returns **true** if and only if $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$.

Algorithm 2: `RationalClosure`

Input: A knowledge base $\mathcal{K}$ and a defeasible rule $A(x) \rightsquigarrow B(x)$
Output: true, if $\mathcal{K} \approx A(x) \rightsquigarrow B(x)$, 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 $R_\infty \cup R \models \neg A(x)$ **and** $R \neq \emptyset$ **do**
  $R := R \setminus R_i$;
  $i := i + 1$;
return $R_\infty \cup R \models A(x) \rightarrow B(x)$;

Informally, the algorithm keeps on removing (materialisations of) defeasible rules from (the materialisation of) $\mathcal{K}$, starting with the lowest base rank, and proceeding base rank by base rank, until it finds the first R which is classically consistent with $A(x)$ (and therefore $A(x)$ is not exceptional with respect to the defeasible version of R). $A(x) \rightsquigarrow B(x)$ is then taken to be in the rational closure of $\mathcal{K}$ if and only if R classically entails the materialisation of $A(x) \rightsquigarrow B(x)$.

3.3 Evidence of Rational Relation

It has been recorded in the literature that any reasonable non-monotonic reasoning procedure should define a rational relation.[11] That is a relation with the following properties:

(Ref) $\mathcal{K} \approx A(x) \rightsquigarrow A(x)$

$$\textbf{(LLE)}\ \frac{A(x) \equiv B(x),\ \mathcal{K} \approx A(x) \rightsquigarrow C(x)}{\mathcal{K} \approx B(x) \rightsquigarrow C(x)}$$

$$\textbf{(RW)}\ \frac{\mathcal{K} \approx A(x) \rightsquigarrow B(x),\ B(x) \models C(x)}{\mathcal{K} \approx A(x) \rightsquigarrow C(x)}$$

$$\textbf{(And)}\ \frac{\mathcal{K} \approx A(x) \rightsquigarrow B(x),\ \mathcal{K} \approx A(x) \rightsquigarrow C(x)}{\mathcal{K} \approx A(x) \rightsquigarrow B(x) \land C(x)}$$

$$\textbf{(Or)}\ \frac{\mathcal{K} \approx A(x) \rightsquigarrow C(x),\ \mathcal{K} \approx B(x) \rightsquigarrow C(x)}{\mathcal{K} \approx A(x) \lor B(x) \rightsquigarrow C(x)}$$

$$\textbf{(CM)}\ \frac{\mathcal{K} \approx A(x) \rightsquigarrow B(x),\ \mathcal{K} \approx A(x) \rightsquigarrow C(x)}{\mathcal{K} \approx A(x) \land B(x) \rightsquigarrow C(x)}$$

$$\textbf{(RM)}\ \frac{\mathcal{K} \approx A(x) \rightsquigarrow C(x),\ \mathcal{K} \not\approx A(x) \rightsquigarrow \neg B(x)}{\mathcal{K} \approx A(x) \land B(x) \rightsquigarrow C(x)}$$

We now show that the rational closure procedure defined in the previous section holds for all of these properties.

3.3.1 Reflexivity.

(Ref) $\mathcal{K} \approx A(x) \rightsquigarrow A(x)$

We want to show that the reflexive property is satisfied by the rational closure algorithm. That is, we want to show $A(x) \rightsquigarrow A(x)$ is in the rational closure of $\mathcal{K}$ by using the rational closure algorithm. To do this we consider two cases. It should be clear that there are no other possible cases.

- (i) We find the smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank s' smaller than s we must have $\bigcup_{i=s'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (ii) Or we find that there is no such s. That is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of s.

Consider (i), by the rational closure algorithm the next check is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to A(x)$. It follows from classical logic that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to A(x)$ regardless of what s is and regardless of what the content of each R_i or R_∞ is. Thus we have shown $\mathcal{K} \approx A(x) \rightsquigarrow A(x)$.

Consider (ii), by the rational closure algorithm the next check is $R_\infty \models A(x) \to A(x)$. It follows from classical logic that $R_\infty \models A(x) \to A(x)$ regardless of what the content of R_∞ is. Thus we have shown $\mathcal{K} \approx A(x) \rightsquigarrow A(x)$.

Since we have shown $\mathcal{K} \approx A(x) \rightsquigarrow A(x)$ for both cases outlined and no other cases exist we are done.

3.3.2 Left Logical Equivalence.

$$\textbf{(LLE)}\ \frac{A(x) \equiv B(x),\ \mathcal{K} \approx A(x) \rightsquigarrow C(x)}{\mathcal{K} \approx B(x) \rightsquigarrow C(x)}$$

We want to show that the left logical equivalence property is satisfied by the rational closure algorithm. That is given $A(x) \equiv B(x)$ and $\mathcal{K} \approx A(x) \rightsquigarrow C(x)$ we want to show $B(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ by using the rational closure algorithm.

We are given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. This means that the rational closure algorithm found either one of two cases. It should be clear that there are no other possible cases.

- (i) It found the smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank s' smaller than s we must have $\bigcup_{i=s'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (ii) Or it found that there is no such s. That is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of s.

In each case, since we are given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ia) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to C(x)$
- (iia) $R_\infty \models \neg A(x)$

Now we want to show $B(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. To do this, the rational closure algorithm first needs to find either one of two cases. It should be clear that there are no other possible cases.

- (1) The smallest rank a such that $\bigcup_{i=a}^{n} R_i \cup R_\infty \not\models \neg B(x)$. That is for any rank a' smaller than a we must have $\bigcup_{i=a'}^{n} R_i \cup R_\infty \models \neg B(x)$.
- (2) Or that there is no such a. That is $\bigcup_{i=a}^{n} R_i \cup R_\infty \models \neg B(x)$ for all possible values of a.

Consider (i), that is there is a smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then that rank s will also be the smallest rank such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg B(x)$. This follows since we know $A(x) \equiv B(x)$. Thus we have satisfied (1). We now want to show $\bigcup_{i=s}^{n} R_i \cup R_\infty \models B(x) \to C(x)$. In case (ia) we also know that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to C(x)$ since $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. Then we must also know that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models B(x) \to C(x)$. This follows since we know $A(x) \equiv B(x)$. Thus we have shown $\mathcal{K} \approx B(x) \rightsquigarrow C(x)$.

Consider (ii), that there is no such s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then it will also be the case that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg B(x)$ for all possible values of s. This follows since we know $A(x) \equiv B(x)$. Thus we have satisfied (2). We now want to show $R_\infty \models B(x) \to C(x)$. In case (iia) we also know that $R_\infty \models \neg A(x)$ since $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. Then we must also know that $R_\infty \models \neg B(x)$. This follows since we know $A(x) \equiv B(x)$. Given $R_\infty \models \neg B(x)$ it follows from classical logic that $R_\infty \models B(x) \to C(x)$. Thus we have shown $\mathcal{K} \approx B(x) \rightsquigarrow C(x)$.

Since we have shown $\mathcal{K} \approx B(x) \rightsquigarrow C(x)$ for both cases outlined and no other cases exist we are done.

3.3.3 Right Weakening.

$$\textbf{(RW)}\ \frac{\mathcal{K} \approx A(x) \rightsquigarrow B(x),\ B(x) \models C(x)}{\mathcal{K} \approx A(x) \rightsquigarrow C(x)}$$

We want to show that the right weakening property is satisfied by the rational closure algorithm. That is given $\mathcal{K} \approx A(x) \rightsquigarrow B(x)$ and

$B(x) \models C(x)$ we want to show $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ by using the rational closure algorithm.

We are given that $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$. This means that the rational closure algorithm found either one of two cases. It should be clear that there are no other possible cases.

- (i) It found the smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank s' smaller than s we must have $\bigcup_{i=s'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (ii) Or it found that there is no such s. That is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of s.

In each case, since we are given that $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ia) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow B(x)$
- (iia) $R_\infty \models \neg A(x)$

Now we want to show $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. To do this, the rational closure algorithm first needs to find either one of two cases. It should be clear that there are no other possible cases.

- (1) The smallest rank a such that $\bigcup_{i=a}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank a' smaller than a we must have $\bigcup_{i=a'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (2) Or that there is no such a. That is $\bigcup_{i=a}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of a.

Consider (i), that is there is a smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then we have satisfied (1). We now want to show $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow C(x)$. In case (ia) we also know that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow B(x)$ since $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$. Then we must also know that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow C(x)$. This follows since we know $B(x) \models C(x)$. Thus we have shown $\mathcal{K} \mid\approx A(x) \rightsquigarrow C(x)$.

Consider (ii), that there is no such s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then we have satisfied (2). We now want to show $R_\infty \models A(x) \rightarrow C(x)$. In case (iia) we also know that $R_\infty \models \neg A(x)$ since $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$. Given $R_\infty \models \neg A(x)$ it follows from classical logic that $R_\infty \models A(x) \rightarrow C(x)$. Thus we have shown $\mathcal{K} \mid\approx A(x) \rightsquigarrow C(x)$.

Since we have shown $\mathcal{K} \mid\approx A(x) \rightsquigarrow C(x)$ for both cases outlined and no other cases exist we are done.

3.3.4 And.

$$\textbf{(And)}\ \frac{\mathcal{K} \mid\approx A(x) \rightsquigarrow B(x),\ \mathcal{K} \mid\approx A(x) \rightsquigarrow C(x)}{\mathcal{K} \mid\approx A(x) \rightsquigarrow B(x) \wedge C(x)}$$

We want to show that the and property is satisfied by the rational closure algorithm. That is given $\mathcal{K} \mid\approx A(x) \rightsquigarrow B(x)$ and $\mathcal{K} \mid\approx A(x) \rightsquigarrow C(x)$ we want to show $A(x) \rightsquigarrow (B(x) \wedge C(x))$ is in the rational closure of $\mathcal{K}$ by using the rational closure algorithm.

We are given that $A(x) \rightsquigarrow B(x)$ and $A(x) \rightsquigarrow C(x)$ are in the rational closure of $\mathcal{K}$. This means that the rational closure algorithm found either one of two cases. It should be clear that there are no other possible cases.

- (i) It found the smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank s' smaller than s we must have $\bigcup_{i=s'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (ii) Or it found that there is no such s. That is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of s.

In each case, since we are given that $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ia) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow B(x)$
- (iia) $R_\infty \models \neg A(x)$

In each case, since we are also given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ib) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow C(x)$
- (iib) $R_\infty \models \neg A(x)$

Now we want to show $A(x) \rightsquigarrow (B(x) \wedge C(x))$ is in the rational closure of $\mathcal{K}$. To do this, the rational closure algorithm first needs to find either one of two cases. It should be clear that there are no other possible cases.

- (1) The smallest rank a such that $\bigcup_{i=a}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank a' smaller than a we must have $\bigcup_{i=a'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (2) Or that there is no such a. That is $\bigcup_{i=a}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of a.

Consider (i), that is there is a smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then we have satisfied (1). We now want to show $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow (B(x) \wedge C(x))$. In case (ia) we also know that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow B(x)$ and in case (ib) we know that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow C(x)$ since $A(x) \rightsquigarrow B(x)$ and $A(x) \rightsquigarrow C(x)$ are in the rational closure of $\mathcal{K}$. Then it follows by classical logic that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow (B(x) \wedge C(x))$. Thus we have shown $\mathcal{K} \mid\approx A(x) \rightsquigarrow (B(x) \wedge C(x))$.

Consider (ii), that there is no such s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then we have satisfied (2). We now want to show $R_\infty \models A(x) \rightarrow (B(x) \wedge C(x))$. In case (iia) and (iib) we also know that $R_\infty \models \neg A(x)$ since $A(x) \rightsquigarrow B(x)$ and $A(x) \rightsquigarrow C(x)$ are in the rational closure of $\mathcal{K}$. Given $R_\infty \models \neg A(x)$ it follows from classical logic that $R_\infty \models A(x) \rightarrow (B(x) \wedge C(x))$. Thus we have shown $\mathcal{K} \mid\approx A(x) \rightsquigarrow (B(x) \wedge C(x))$.

Since we have shown $\mathcal{K} \approx A(x) \rightsquigarrow (B(x) \wedge C(x))$ for both cases outlined and no other cases exist we are done.

3.3.5 Or.

$$\textbf{(Or)}\quad \frac{\mathcal{K} \approx A(x) \rightsquigarrow C(x),\ \mathcal{K} \approx B(x) \rightsquigarrow C(x)}{\mathcal{K} \approx A(x) \vee B(x) \rightsquigarrow C(x)}$$

We want to show that the or property is satisfied by the rational closure algorithm. That is given $\mathcal{K} \approx A(x) \rightsquigarrow C(x)$ and $\mathcal{K} \approx B(x) \rightsquigarrow C(x)$ we want to show $(A(x) \vee B(x)) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ by using the rational closure algorithm.

We are given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. This means that the rational closure algorithm found either one of two cases. It should be clear that there are no other possible cases.

- (i) It found the smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank s' smaller than s we must have $\bigcup_{i=s'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (ii) Or it found that there is no such s. That is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of s.

In each case, since we are given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ia) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \rightarrow C(x)$
- (iia) $R_\infty \models \neg A(x)$

We are also given that $B(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. This means that the rational closure algorithm found either one of two cases. It should be clear that there are no other possible cases.

- (iii) It found the smallest rank t such that $\bigcup_{i=t}^{n} R_i \cup R_\infty \not\models \neg B(x)$. That is for any rank t' smaller than t we must have $\bigcup_{i=t'}^{n} R_i \cup R_\infty \models \neg B(x)$.
- (iv) Or it found that there is no such t. That is $\bigcup_{i=t}^{n} R_i \cup R_\infty \models \neg B(x)$ for all possible values of t.

In each case, since we are given that $B(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (iiia) $\bigcup_{i=t}^{n} R_i \cup R_\infty \models B(x) \rightarrow C(x)$
- (iva) $R_\infty \models \neg B(x)$

Note that if found, the ranks s and t need not be the same.

Now we want to show $(A(x) \vee B(x)) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. To do this, the rational closure algorithm first needs to find either one of two cases. It should be clear that there are no other possible cases.

- (1) The smallest rank a such that $\bigcup_{i=a}^{n} R_i \cup R_\infty \not\models \neg(A(x) \vee B(x))$. That is for any rank a' smaller than a we must have $\bigcup_{i=a'}^{n} R_i \cup R_\infty \models \neg(A(x) \vee B(x))$.
- (2) Or that there is no such a. That is $\bigcup_{i=a}^{n} R_i \cup R_\infty \models \neg(A(x) \vee B(x))$ for all possible values of a.

Consider the combinations (i) and (iii), (i) and (iv), (ii) and (iii). We can reduce these to a single check. We can do this by noticing that if we have $\bigcup_{i=r}^{n} R_i \cup R_\infty \not\models \neg A(x)$ or $\bigcup_{i=r}^{n} R_i \cup R_\infty \not\models \neg B(x)$ for some rank r then it follows from classical logic that $\bigcup_{i=r}^{n} R_i \cup R_\infty \not\models \neg(A(x) \vee B(x))$. We only require one of the above to be true to get this result. Thus if we want the smallest rank r such that $\bigcup_{i=r}^{n} R_i \cup R_\infty \not\models \neg(A(x) \vee B(x))$ then we can let $r = \min(s,t)$ where s is the smallest rank such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$ and t is the smallest rank such that $\bigcup_{i=t}^{n} R_i \cup R_\infty \not\models \neg B(x)$. If one of s or t does not exist then r is simply the s or t that does exist. Then we have satisfied (1). We now want to show $\bigcup_{i=r}^{n} R_i \cup R_\infty \models (A(x) \vee B(x)) \rightarrow C(x)$. In the cases chosen we also know that either one or both of $\bigcup_{i=r}^{n} R_i \cup R_\infty \models A(x) \rightarrow C(x)$ or $\bigcup_{i=r}^{n} R_i \cup R_\infty \models B(x) \rightarrow C(x)$ holds since $A(x) \rightsquigarrow C(x)$ and $B(x) \rightsquigarrow C(x)$ are in the rational closure of $\mathcal{K}$. It follows from classical logic that we have $\bigcup_{i=r}^{n} R_i \cup R_\infty \models (A(x) \vee B(x)) \rightarrow C(x)$ from the above. Thus we have shown $\mathcal{K} \approx (A(x) \vee B(x)) \rightsquigarrow C(x)$.

Consider the last combination of (ii) and (iv). Since there is no smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$ and no smallest rank t such that $\bigcup_{i=t}^{n} R_i \cup R_\infty \not\models \neg B(x)$ we must have that there is no rank r such that $\bigcup_{i=r}^{n} R_i \cup R_\infty \not\models \neg(A(x) \vee B(x))$. Then we have satisfied (2). We now want to show $R_\infty \models (A(x) \vee B(x)) \rightarrow C(x)$. In this combination we have cases (iia) and (iva) and so we also know that $R_\infty \models \neg A(x)$ and $R_\infty \models \neg B(x)$ since $A(x) \rightsquigarrow C(x)$ and $B(x) \rightsquigarrow C(x)$ are in the rational closure of $\mathcal{K}$. Given this, it follows from classical logic that $R_\infty \models \neg(A(x) \vee B(x))$ and further that $R_\infty \models (A(x) \vee B(x)) \rightarrow C(x)$. Thus we have shown $\mathcal{K} \approx (A(x) \vee B(x)) \rightsquigarrow C(x)$.

Since we have shown $\mathcal{K} \approx (A(x) \vee B(x)) \rightsquigarrow C(x)$ for both cases outlined and no other cases exist we are done.

3.3.6 Cautious Monotonicity.

$$\textbf{(CM)}\quad \frac{\mathcal{K} \approx A(x) \rightsquigarrow B(x),\ \mathcal{K} \approx A(x) \rightsquigarrow C(x)}{\mathcal{K} \approx A(x) \wedge B(x) \rightsquigarrow C(x)}$$

We want to show that the cautious monotonicity property is satisfied by the rational closure algorithm. That is given $\mathcal{K} \approx A(x) \rightsquigarrow B(x)$ and $\mathcal{K} \approx A(x) \rightsquigarrow C(x)$ we want to show $(A(x) \wedge B(x)) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ by using the rational closure algorithm.

We are given that $A(x) \rightsquigarrow B(x)$ and $A(x) \rightsquigarrow C(x)$ are in the rational closure of $\mathcal{K}$. This means that the rational closure algorithm found either one of two cases. It should be clear that there are no other possible cases.

- (i) It found the smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank s' smaller than s we must have $\bigcup_{i=s'}^{n} R_i \cup R_\infty \models \neg A(x)$.

- (ii) Or it found that there is no such s. That is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of s.

In each case, since we are given that $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ia) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to B(x)$. This is logically equivalent to $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x) \vee B(x)$. Since it was found for case (i) that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$ we must have $\bigcup_{i=s}^{n} R_i \cup R_\infty \models B(x)$ for $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to B(x)$ to hold.
- (iia) $R_\infty \models \neg A(x)$

In each case, since we are also given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ib) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to C(x)$. This is logically equivalent to $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x) \vee C(x)$. Since it was found for case (i) that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$ we must have $\bigcup_{i=s}^{n} R_i \cup R_\infty \models C(x)$ for $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to C(x)$ to hold.
- (iib) $R_\infty \models \neg A(x)$

Now we want to show $(A(x) \wedge B(x)) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. To do this, the rational closure algorithm first needs to find either one of two cases. It should be clear that there are no other possible cases.

- (1) The smallest rank a such that $\bigcup_{i=a}^{n} R_i \cup R_\infty \not\models \neg(A(x) \wedge B(x))$. That is for any rank a' smaller than a we must have $\bigcup_{i=a'}^{n} R_i \cup R_\infty \models \neg(A(x) \wedge B(x))$.
- (2) Or that there is no such a. That is $\bigcup_{i=a}^{n} R_i \cup R_\infty \models \neg(A(x) \wedge B(x))$ for all possible values of a.

Consider (i), that is there is a smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Case (i) also gives us (ia). That is since $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$ and $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to B(x)$ we have $\bigcup_{i=s}^{n} R_i \cup R_\infty \models B(x)$. Then classical logic tells us that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg B(x)$. That is we have found a smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg(A(x) \wedge B(x))$. Then we have satisfied (1). We now want to show $\bigcup_{i=s}^{n} R_i \cup R_\infty \models (A(x) \wedge B(x)) \to C(x)$. This is logically equivalent to $\bigcup_{i=s}^{n} R_i \cup R_\infty \models (\neg A(x) \vee \neg B(x)) \vee C(x)$. From case (ib) we have $\bigcup_{i=s}^{n} R_i \cup R_\infty \models C(x)$. It follows from classical logic that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models (\neg A(x) \vee \neg B(x)) \vee C(x)$ must then hold and further that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models (A(x) \wedge B(x)) \to C(x)$ holds. Thus we have shown $\mathcal{K} \mathrel{|\!\approx} (A(x) \wedge B(x)) \rightsquigarrow C(x)$.

Consider (ii), that there is no such s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then it follows by classical logic that there is no s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x) \vee \neg B(x)$. Then we have satisfied (2). We now want to show $R_\infty \models (A(x) \wedge B(x)) \to C(x)$. This is logically equivalent to $R_\infty \models \neg A(x) \vee \neg B(x) \vee C(x)$. In cases (iia) and (iib) we also know that $R_\infty \models \neg A(x)$ since $A(x) \rightsquigarrow B(x)$ and $A(x) \rightsquigarrow C(x)$ are in the rational closure of $\mathcal{K}$. Given this, it follows from classical logic that $R_\infty \models \neg A(x) \vee \neg B(x) \vee C(x)$ holds and further that $R_\infty \models (A(x) \wedge B(x)) \to C(x)$ holds. Thus we have shown $\mathcal{K} \mathrel{|\!\approx} (A(x) \wedge B(x)) \rightsquigarrow C(x))$.

Since we have shown $\mathcal{K} \mathrel{|\!\approx} A(x) \rightsquigarrow (B(x) \wedge C(x))$ for both cases outlined and no other cases exist we are done.

3.3.7 Rational Monotonicity.

$$\textbf{(RM)}\quad \frac{\mathcal{K} \mathrel{|\!\approx} A(x) \rightsquigarrow C(x),\ \mathcal{K} \mathrel{|\!\not\approx} A(x) \rightsquigarrow \neg B(x)}{\mathcal{K} \mathrel{|\!\approx} A(x) \wedge B(x) \rightsquigarrow C(x)}$$

We want to show that the rational monotonicity property is satisfied by the rational closure algorithm. That is given $\mathcal{K} \mathrel{|\!\approx} A(x) \rightsquigarrow C(x)$ and $\mathcal{K} \mathrel{|\!\not\approx} A(x) \rightsquigarrow \neg B(x)$ we want to show $(A(x) \wedge B(x)) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ by using the rational closure algorithm.

We are given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. This means that the rational closure algorithm found either one of two cases. It should be clear that there are no other possible cases.

- (i) It found the smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. That is for any rank s' smaller than s we must have $\bigcup_{i=s'}^{n} R_i \cup R_\infty \models \neg A(x)$.
- (ii) Or it found that there is no such s. That is $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x)$ for all possible values of s.

In each case, since we are given that $A(x) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ia) $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to C(x)$. This is logically equivalent to $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x) \vee C(x)$. Since it was found for case (i) that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$ we must have $\bigcup_{i=s}^{n} R_i \cup R_\infty \models C(x)$ for $\bigcup_{i=s}^{n} R_i \cup R_\infty \models A(x) \to C(x)$ to hold.
- (iia) $R_\infty \models \neg A(x)$

In each case, since we are also given that $A(x) \rightsquigarrow \neg B(x)$ is not in the rational closure of $\mathcal{K}$ we can conclude the following from the cases above.

- (ib) $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models A(x) \to \neg B(x)$. This is logically equivalent to $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x) \vee \neg B(x)$. Since it was found for case

(i) that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$ we must also have $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg B(x)$ for $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models A(x) \to \neg B(x)$ to hold.

- (iib) $R_\infty \models \neg A(x)$. Then we must have $R_\infty \models A(x) \to \neg B(x)$. It follows from classical logic that $R_\infty \not\models \neg B(x)$.

Now we want to show $(A(x) \wedge B(x)) \rightsquigarrow C(x)$ is in the rational closure of $\mathcal{K}$. To do this, the rational closure algorithm first needs to find either one of two cases. It should be clear that there are no other possible cases.

- (1) The smallest rank a such that $\bigcup_{i=a}^{n} R_i \cup R_\infty \not\models \neg(A(x) \wedge B(x))$. That is for any rank a' smaller than a we must have $\bigcup_{i=a'}^{n} R_i \cup R_\infty \models \neg(A(x) \wedge B(x))$.
- (2) Or that there is no such a. That is $\bigcup_{i=a}^{n} R_i \cup R_\infty \models \neg(A(x) \wedge B(x))$ for all possible values of a.

Consider (i), that is there is a smallest rank s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Then we also know from (ib) that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg B(x)$. Then we have found there exists a smallest s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x) \vee \neg B(x)$. Then we have satisfied (1). We now want to show $\bigcup_{i=s}^{n} R_i \cup R_\infty \models (A(x) \wedge B(x)) \to C(x)$. This is logically equivalent to $\bigcup_{i=s}^{n} R_i \cup R_\infty \models \neg A(x) \vee \neg B(x) \vee C(x)$. From case (ia) we have $\bigcup_{i=s}^{n} R_i \cup R_\infty \models C(x)$. It then follows from classical logic that $\bigcup_{i=s}^{n} R_i \cup R_\infty \models (A(x) \wedge B(x)) \to C(x)$. Thus we have shown $\mathcal{K} \models\!\approx (A(x) \wedge B(x)) \rightsquigarrow C(x)$.

Consider (ii), that there is no such s such that $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x)$. Furthermore we can say $\bigcup_{i=s}^{n} R_i \cup R_\infty \not\models \neg A(x) \vee \neg B(x)$. Then we have satisfied (2). We now want to show $R_\infty \models (A(x) \wedge B(x)) \to C(x)$. In case (iia) we also know that $R_\infty \models \neg A(x)$ since $A(x) \rightsquigarrow B(x)$ is in the rational closure of $\mathcal{K}$. Given $R_\infty \models \neg A(x)$ it follows from classical logic that $R_\infty \models (A(x) \wedge B(x)) \to C(x)$. Thus we have shown $\mathcal{K} \models\!\approx (A(x) \wedge B(x)) \rightsquigarrow C(x)$.

Since we have shown $\mathcal{K} \models\!\approx (A(x) \wedge B(x)) \rightsquigarrow C(x)$ for both cases outlined and no other cases exist we are done.

4 RESULTS AND DISCUSSION

The aim of this project was to extend Datalog with a form of nonmonotonic logic called defeasible reasoning. We have shown it is theoretically possible to do this using a procedure known as rational closure that we have shown is a rational relation. It has been defined by Kraus et al. [10] that this is the requirement for a reasonable nonmonotonic system. It is this requirement that establishes the correctness of the theoretical analysis as utmost important. Thus, we have worked closely with Prof. Thomas Meyer in refining the theoretical analysis to be as correct as possible.

We found that some of the properties required to be a rational relation followed easily from the rational closure procedure. However, other properties were found to be quite difficult to show. With evaluation from Prof. Thomas Meyer we believe the proofs outlined provide accurate mechanisms for correctness.

Furthermore, in regards to whether the syntax, semantics and procedure defined are applicable in implementation we found promising results. The sister project to this was focused on the implementation of *Defeasible Datalog* as described in this project. The author of the sister project, Joshua Abraham, found that it is possible to implement *Defeasible Datalog* using our description and that the rational closure procedure works as expected. Further, Abraham was able to implement *Defeasible Datalog* within the RDFox framework [17] which demonstrates the application of *Defeasible Datalog* in the real-world.

5 CONCLUSION

The ability to model real-world information in elaborate and simple knowledge bases is extremely powerful. It allows us to model a variety of situations and to draw logical conclusions from them using a formalised process of reasoning. One of the most useful features of logical reasoning is drawing conclusions, otherwise known as entailment, and the ability to draw more implicit information from our knowledge base. Furthermore, it is extremely important that our reasoning does not fail when given new facts. This is often the case in real-world modelling where new facts can contradict previously entailed information. The ability to handle such contradictions makes defeasible reasoning extremely valuable for modelling real-world information. It is important to note, however, that when using defeasible reasoning we must still comply with the properties that we desire from propositional logic, otherwise it is not useful. Further, a platform for creating and performing defeasible reasoning is important in making these concepts applicable to the real-world.

In this paper we have described *Defeasible Datalog* - a language for handling defeasible reasoning in Datalog. We have outlined a procedure for reasoning such that we have modelled a rational relation - a key requirement when creating a reasonable nonmonotonic system. [10] We have shown through theoretical analysis that the outlined procedure holds the properties needed and thus confirming the ability to create *Defeasible Datalog*. Furthermore, the sister project dedicated to the implementation of *Defeasible Datalog* was successfully implemented according to the descriptions outlined in this paper. Lastly, we hope that this project encourages further work in creating nonmonotonic systems for handling exceptions and other inconsistencies.

6 ACKNOWLEDGEMENTS

I would like to thank my project partner, Joshua Abraham, for his help and support. Michael Harrison, co-supervisor, for his knowledge. Associate Professor Hussein Suleman for being the second reader. Finally, to Professor Tommie Meyer, project supervisor, for his constant guidance and encouragement.

REFERENCES

[1] Giovanni Casini, Thomas Andreas Meyer, Kodylan Moodley, Uli Sattler, and Ivan José Varzinczak. 2015. Introducing Defeasibility into OWL Ontologies. In

The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II, Marcelo Arenas, Oscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul Groth, Michael Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab (Eds.). 409–426. https://doi.org/10.1007/978-3-319-25010-6_27

[2] Giovanni Casini, Thomas Meyer, Kody Moodley, and Ivan Varzinczak. 2013. Nonmonotonic reasoning in description logics. Rational closure for the ABox. (2013).

[3] Giovanni Casini and Umberto Straccia. 2010. Rational Closure for Defeasible Description Logics. In *Logics in Artificial Intelligence*, Tomi Janhunen and Ilkka Niemelä (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 77–90.

[4] Keith L Clark. 1978. Negation as failure. In *Logic and data bases*. Springer, 293–322.

[5] Cristhian AD Deagustini, Maria Vanina Martinez, Marcelo A Falappa, and Guillermo Ricardo Simari. 2015. On the Influence of Incoherence in Inconsistency-tolerant Semantics for Datalog.. In *JOWO@ IJCAI*.

[6] Cristhian AD Deagustini, M Vanina Martinez, Marcelo A Falappa, and Guillermo R Simari. 2018. How does incoherence affect inconsistency-tolerant semantics for Datalog? *Annals of Mathematics and Artificial Intelligence* 82, 1-3 (2018), 43–68.

[7] Thomas Eiter, Nicola Leone, Cristinel Mateis, Gerald Pfeifer, and Francesco Scarcello. 1997. A deductive system for non-monotonic reasoning. In *Logic Programming And Nonmonotonic Reasoning*, Jürgen Dix, Ulrich Furbach, and Anil Nerode (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 363–374.

[8] D. M. Gabbay. 1985. Theoretical Foundations for Non-Monotonic Reasoning in Expert Systems. In *Logics and Models of Concurrent Systems*, Krzysztof R. Apt (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 439–457.

[9] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. 2015. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence* 226 (2015), 1–33.

[10] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence* 44, 1-2 (1990), 167–207.

[11] Daniel Lehmann and Menachem Magidor. 1992. What does a conditional knowledge base entail? *Artificial intelligence* 55, 1 (1992), 1–60.

[12] Michael J Maher, Andrew Rock, Grigoris Antoniou, David Billington, and Tristan Miller. 2001. Efficient defeasible reasoning systems. *International Journal on Artificial Intelligence Tools* 10, 04 (2001), 483–501.

[13] Maria Vanina Martinez, Cristhian Ariel David Deagustini, Marcelo A. Falappa, and Guillermo Ricardo Simari. 2014. Inconsistency-Tolerant Reasoning in Datalog Ontologies via an Argumentative Semantics. In *Advances in Artificial Intelligence – IBERAMIA 2014*, Ana L.C. Bazzan and Karim Pichara (Eds.). Springer International Publishing, Cham, 15–27.

[14] John McCarthy. 1980. Circumscription—A form of non-monotonic reasoning. *Artificial Intelligence* 13, 1 (1980), 27 – 39. https://doi.org/10.1016/0004-3702(80)90011-9 Special Issue on Non-Monotonic Logic.

[15] Drew McDermott. 1982. Nonmonotonic logic II: Nonmonotonic modal theories. *Journal of the ACM (JACM)* 29, 1 (1982), 33–57.

[16] Robert C Moore. 1984. *Possible-world semantics for autoepistemic logic*. Technical Report. SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER.

[17] Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. 2015. RDFox: A Highly-Scalable RDF Store. In *International Semantic Web Conference (2) (Lecture Notes in Computer Science)*, Vol. 9367. Springer, 3–20.

[18] Donald Nute. 1988. *Defeasible Reasoning: A Philosophical Analysis in Prolog*. Springer Netherlands, Dordrecht, 251–288. https://doi.org/10.1007/978-94-009-2699-8_9

[19] John L. Pollock. 1987. Defeasible reasoning. *Cognitive Science* 11, 4 (1987), 481 – 518. https://doi.org/10.1016/S0364-0213(87)80017-4

[20] Raymond Reiter. 1980. A logic for default reasoning. *Artificial intelligence* 13, 1-2 (1980), 81–132.

[21] David S Touretzky. 1986. *The mathematics of inheritance systems*. Vol. 8. Morgan Kaufmann.