**A Knowledge Representation tool to assist Rational Closure diagnosis**

CSC2005Z Research Project

Anri Lombard (email)

Supervisor: Tommie Meyer (email)

(Remember to include many screenshots)

Abstract

* Assist novices in steep learning curve
* Assist experts in diagnosing results of defeasible query to custom knowledge bases
* Adds practical tool to theoretical research in Rational Closure

1. Introduction

* 2 problems
* Hypothesis
* Motivation to conduct research on hypothesis

Rational Closure can be used to reason about complex knowledge bases with common sense. By considering new and contradicting knowledge when searching knowledge bases, we can get accurate conclusions to defeasible queries, and build systems with more human-like reasoning abilities that represent knowledge better.

* 1. Problem statement

Novice researchers are rarely exposed to Rational Closure and warded off by articles and symbolically dense research papers. This causes them to disregard the field of Knowledge representation and miss an opportunity to learn from a different perspective on how human reasoning can be replicated for use in, say, artificially intelligent systems.

On the other hand, expert researchers scribble for hours to achieve a single conclusion from a Rational Closure algorithm, wasting time on a trivial task.

Thus, the problems this paper would focus on are (1) Novice Researchers do not have an accessible platform to learn about Rational Closure, and (2) Expert Researchers waste time on long-hand calculations to get a conclusion from the rational closure algorithm.

* 1. Motivation

The Rational Closure Algorithm has important insights on how to build reasoning machines; therefore, it is important to introduce new computer science researchers to these as they start thinking more about solutions. Similarly, expert researchers already devote time to build reasoning machines and helping them cut down on trivial parts, such as getting a result from an algorithm rather than by hand, will reduce wasted time.

* 1. Research objectives

**Build a website that is easily accessed by novices and experts alike on the internet.**

* There are no Rational-Closure-specific websites that explain the content thoroughly, nor are there Knowledge Representation resources online that emphasize Rational Closure.
* This can easily be tested by getting user feedback on how easy it was to find the website, and how useful it was for them to get an introduction to Rational Closure.

**Build a Graphical User Interface (GUI) that allow experts to diagnose and get conclusions from the Rational Closure algorithm.**

* A GUI interface allows researchers to enter necessary details for the Rational Closure algorithm to return the correct conclusions from specified knowledge bases.
* Using Java for the GUI makes it easy to customize if users decide to do so.

1. Literature review

* Adam’s work
* Rational Closure Algorithm
* Joel’s work

1. Methodology/Implementation

The software to run the Rational Closure Algorithm was built based on the work done by Joel Hamilton (reference), who implemented a Java algorithm that runs both the Base Rank and Rational Closure algorithms, and with some manipulation could return the explanations of how the algorithm got to its conclusions.

The software was built using Java, Python and the PyQt5 library, and the Next.js framework that is hosted on the Vercel platform.

* 1. Major software artifacts

The major software artifacts are divided into 2 parts:

1. The first piece we need to produce is an interface that gets input from the user, especially their defeasible query, custom knowledge base, and if they want an explanation of what the algorithm did to come to its conclusion, or just see what Boolean it returns.  
     
   **Key features:** The ability to take in user input and relay it to the algorithm, then return the conclusion to the user.

1. The second piece is the website that hosts the downloadable link for the GUI and explains unfamiliar concepts to the novice visitors.  
     
   **Key features:** The ability to convey information to users, but also be manipulated by developers and improved where necessary.
   1. Algorithm

This implementation of the algorithm was implemented by Joel Hamilton, as proposed by (insert name here).

* + 1. BaseRank

(latex code here for baserank)

The BaseRank algorithm takes in the user’s knowledge base that the query needs to run through, then maps every explicit formula (piece of knowledge) to a natural number (0, 1, …, etc) representing it’s base rank.

(definition 5.7 here)  
  
It is important to note that the reason for assigning each defeasible implication a rank is that the lower the rank, the more defeasible the statement; defeasible implicaitons with lower ranks are more general statements in the current knowledge base, which makes that statement more defeasible.

Given this, we can deduce that infinite ranks are classical statements without any defeasibility whatsoever.

Another clarification required is the meaning of “exceptional,” which simply means this statement is never true given the current knowledge base.

The BaseRank Algorithm finally outputs a tuple containing base ranks from zero to infinity. This tuple is then used as input to the Rational Closure algorithm.

* + 1. RationalClosure

(latex code here for rational closure)

Rational Closure is the main topic of this report and is a form of nonmonotonic entailment defined as:

(Defenition 5.8)  
  
To understand Rational Closure we need to define minimal ranked entailment.  
  
(Defenition 5.3)  
  
An intuitive example of what this means is the following representation:  
  
(Rep p.54)  
  
which shows the minimal ranked interpretation as R^k\_{RC}. With this knowledge at hand we are glad to announce that Rational Closure and Minimal Ranked Entailment are exactly the same, both outputting the most defeasible statement as the lowest base rank, followed by less defeasible statements.

(Theorem 5.2)  
  
(Ex p.62)  
  
After a query and knowledge base is passed through these algorithms (which are implemented by the GUI in this case) the returned boolean answers the question: “is this query valid in the current knowledge base?” The way Rational Closure answers this question is by taking knowledge and using “common sense” to validate new information and decide to keep or discard It, unlike Classical entailment that simply takes the new information in addition to previous – often contradicting – knowledge, thus representing human-like reasoning much more accurately.

* 1. Interfaces

The GUI interface is made using Python and PyQt5 library. The PyQt5 library ensures that the GUI works smoothly across devices, accommodating users across all devices.

* + 1. Website

(screenshots about website)

Each section of the website is built intentionally to be easily accessed and understood by users, especially nascent and novice researchers that want to learn about Rational Closure.

* Contents

(screenshot)

Novices could get scared off by the semantic overload in Rational Closure research, therefore this website is designed to introduce concepts through familiar mediums they are more likely to have experience with already.

As an introduction users are met with a clickable, easy-to-navigate content pane so they could jump to relevant information or get a big-picture overview of what is to come.

* Introduction

(screenshot)

Users get introduced to Artificial Intellegence, Knowledge Representation, and Defeasible Reasoning through Youtube videos, and already popular format for educational content. These 3 concepts are chosen since they build a foundational understanding of where Rational Closure fits into the impactful field of Artificial Intellegence.

* Key Concepts

(screenshot)

Besides the introduction, interested users would eventually encounter more symantic-driven concepts that take a while to digest. We chose non-monotonic reasoning, the base rank algorithm, and the Rational Closure algorithm as sufficiently intuitive concepts that could be explained through Knowledge Representation atoms and connectives.

* Tool Walkthrough

(screenshot)

Expert users most likely want to skim over the previous sections of the website to reach the Graphical User Interface (GUI) that aims to help them in experiments. This section hosts the link and walks though how to make queries and add custom knowledge bases to the program.

* Contributing Section

(screenshot)

As we will discuss in the Future Research section, this project is set up well to attract developers and researchers whom are eager to expand on it.

* + 1. GUI

(screenshots of GUI)

Our Graphical User Interface is the main tool aimed at helping expert users to run experiments with the Rational Closure algorithm.

The GUI is easy to download, set up, and use; although it requires the user to already heave a proper development environment with Python and Java installed.

The advantage of a GUI over a website is that users could modify it to their needs by altering the code and inspect how Rational Closure was programmed if they intent to get a more thorough grasp of it.

1. Experiment and results

* Feedback
* Heuristic evaluations
* Discussion of results

This project has the objectives of introducing novice users to Rational Closure and assist expert users in running their experiments; thus no time-based experimentation was feasible.

There are no matching tools already available on the internet to do comparisons agains, therefore comparison expersimients were not feasible either.

The main form of evaluation was informal feedback from novice and expert users. Both groups were asked to evaluate the website for (1) it’s usefulness, and (2) it’s accuracty of information, where experst were asked, in addition to the latter questions, (3) whether the GUI gave them useful results.

* Novices
* Expand
* Experts
* Expand

1. Conclusions

* Conclusion to hypothesis
* Contributions made by research to previous research

1. Future research

* Addition to website of more Knowledge Representation algorithms
* GUI for multiple algorithms (options to select which ones)\

Both the GUI and website were specifically chosen as they have potential to be expanded by developers and reasearchers who are capable enough to contribute.

The main future research projects we would encourage are (1) the addition of content to the website so it encapsulates a larger section of Knowledge Representation than just Rational Closure and (2) the development of similar GUIs, but for different algorithms that could assist other experts in their research.  
  
Ulitimately, the website serves as a platform that displays tools that could be used for experiments and guides on learning complex concepts associatied with Knowledge Representation.

1. Acknowledgements

* Adam
* Joel
* All reviewers (Kialan, Carl, Joel, Darshan?, Hyunmin?, Sibu?, …experts)
* Mention that Joel did previous research on the topic

The author would like to thank the participants Kialan Pillay, Sibusiso (surname), Simbarashe Mawere, Carl Combrick… for adding their feedback to this project.  
  
A special thanks goes to Adam whom wrote a dissertation breaking down Rational Closure, amongst other concepts, that was the initial spark for this project, and Joel Hamilton who implemented the Rational Closure algorithms in Java.

1. References

* Citations of papers I used (especially those mentioned in the proposal)

1. Appendix