A Knowledge Representation tool to assist Rational Closure diagnosis

Research proposal

Anri Lombard

Department of computer science,

University of Cape Town

lmbanr001@myuct.ac.za

Supervisor: Prof. Tommie Meyer

tommie.meyer@uct.ac.za

1. Project description and motivation

1.1. Knowledge Representation, nonmonotonic logic, defeasible reasoning, and Rational Closure

Knowledge Representation and reasoning is a field of artificial intelligence that employs formal logic to represent domain knowledge symbolically, which allows for rigorous analysis of additional knowledge inferences, implied from explicit knowledge [1,2].

Monotonicity essentially states that inferences that hold within any subset of a set of statements, have to be consequences of the entire set.

Classical **monotonic logics** lack the expressivity to explicitly represent exceptions, without challenging alterations to the formulas in the knowledge base. Kraus, Lehmann, and Magidor [3] argued that **nonmonotonic logic** should be able to explicitly state "an x is typically a y" in which we define "typically" to be read as "in the normal case, it is reasonable to conclude y, given x".

Thus, **nonmonotonic reasoning** is an area of research that attempts to formalize different patterns of "common sense" reasoning, by dropping monotonicity as property and investigating how to define reasonable notions of consequences. Humans reason by making assumptions based on the knowledge they have, and then revising those assumptions upon learning new information. Several frameworks were formalized to mimic this pattern known as **defeasible reasoning** [3, 4, 5].

Rational Closure is an alternative syntactic definition of minimal ranked entailment [5, 6], which ranks knowledge statements on bases from those that make the most sense in the current knowledge base (base 0) to those that make no sense (base infinity). Then it returns true if the query given matches the lowest possible base.

1.2. Algorithms for Rational Closure

To understand Rational Closure best the algorithms for ranking bases and doing Rational Closure are great guides.

Firstly, the BaseRank algorithm takes the knowledge base, K, and ranks the knowledge into bases with knowledge that makes the most sense (lowest base) in the current world to those that make the least sense (infinity).

Algorithm 1 BaseRank

Input: A knowledge base $\mathcal{K}$
Output: An ordered tuple $(R_0, ..., R_{n-1}, R_\infty, n)$
i := 0;
$E_0 := \overrightarrow{\mathcal{K}}$;
while $E_{i-1} \neq E_i$ **do**
    $E_{i+1} := \{\alpha \rightarrow \beta \in E_i \mid E_i \models \neg\alpha\}$;
    $R_i := E_i \setminus E_{i+1}$;
    $i := i + 1$;
end while
$R_\infty := E_{i-1}$;
if $E_{i-1} = \varnothing$ **then**
    $n := i - 1$;
else
    $n := i$;
end if
return $(R_0, ..., R_{n-1}, R_\infty, n)$

Figure 1: BaseRank algorithm

From the ranked bases, Rational Closure goes through the Knowledge base to remove exceptional formulas – the antecedent of the defeasible query of which it's negation is classically entailed by the materialisation of the full knowledge base [7].

Algorithm 2 RationalClosure

Input: A knowledge base $\mathcal{K}$, and a defeasible implication $\alpha \mid\!\sim \beta$
Output: **true**, if $\mathcal{K} \models\!\approx \alpha \mid\!\sim \beta$, and **false** otherwise
$(R_0, ..., R_{n-1}, R_\infty, n) :=$ BaseRank($\mathcal{K}$);
$i := 0$
$R := \bigcup_{i=0}^{j<n} R_j$;
while $R_\infty \cup R \models \neg\alpha$ and $R \neq \varnothing$ **do**
    $R := R \setminus R_i$;
    $i := i + 1$;
end while
return $R_\infty \cup R \models \alpha \rightarrow \beta$;

Figure 2: Rational Closure Algorithm

Rational Closure then return true if the union of all ranks is entailed by the negation of the knowledge base.

1.3. GUI and website motivation

There are many computer science research fields to explore, such as Natural Language Processing, Computer Vision, Speech Recognition, Human-Computer Interaction, High-Performance Computing, and many more. One of these sub-domains is Knowledge Representation.

KR is very dense and has many abstract and hard-to-grasp concepts that prerequisites a steep learning curve; this causes many young researchers to not even attempt learning it, or stops them in their tracks when they do express curiosity.

Our attempt is to introduce the field with Rational Closure specifically. We focus on Rational Closure since it encompasses concepts such as nonmonotonic defeasible reasoning, which is the closest form of "common sense" that would be easy to learn.

When learning about Rational Closure there is no place on the internet solely dedicated to introducing it in a learnable, enjoyable, introductory sense, with resources for further exploration if needed; nor is there a tool to study what the algorithm does – rather than depending on tedious manual calculations of what rational closure should be doing.

This project attempts to contribute solutions to these problems, thereby making it easier for beginners to explore Rational Closure through an introductory tool using the introductory tool and so more advanced researchers could reduce the amount of time wasted on menial calculation.

2. Related works

2.1. Related works

When searching “tools for knowledge representation” one will reach a website (https://www.cs.rochester.edu/research/cisd/projects/kr-tools/) that describes 2 tools, one being the “Rhet System” which is a programming language that can be used as the back-end to a user’s program to handle knowledge representation chores, and “Shocker” which just extends the previous tool with additional reasoning facilities. Both tools aim to support local research projects by expert researchers.

2.2. What differs this research from previous tools

Similarly to these works, this project will take advantage of online platforms, such as a website, so it is easily searched and accessed on the internet.

Some major differences include:

- This tool is built by Java, which is a widely used programming language that many developers can manipulate and understand. It is easy to set up a Java environment if novices are curious to use the GUI and do not force users to learn an arbitrary language needed to operate the tool lathe Rhet system enforces.
- This totalism helps novices and experts alike so that both can benefit by learning more about Rational Closure, whereas the previous tools largely serve already-expert researchers to advance in their research projects.
- This tool is free and open source for all users to use and expert developers to contribute to, unlike the previous tools which were licensed and not open source for contribution from eager developers.
- This tool directly looks at Rational Closure, where the other tools do not.

3. Research hypotheses

3.1. Hypotheses

The 2 foremost problems we would like to propose a solution to are (1) resources for learning Rational Closure is hard to come by for novices, making it difficult to learn about the concept and (2) experts waste limited time on the tedious calculation of what a defeasible query would conclude given a large customer database because they lack a tool to do it.

We hypothesize that this tool could help contribute in the following ways:

(1) Access to Rational Closure for novices will improve:

Most nascent researchers or everyday computer science students have access to the internet, especially if they are in a position to do research in a university setting, thus a website is easy to find and a convenient starting point for further exploration. This way novices could learn through familiar resources, such as linked Youtube videos, articles, and even more tedious-to-read content similar to Wikipedia pages and research papers.

(2) Experts will have a proper tool to diagnose conclusions made by Rational Closure when a query is sent to a custom database:

Almost all expert researchers in a computer science field, such as Knowledge Representation, have access to or use the internet daily. This means publishing a free and open source tool will remove any restrictions to downloading and using the tool, which makes it easier for them to experiment with different defeasible queries on custom knowledge bases.

A GUI allows researchers to take advantage of the Model-View-Controller architecture, where they could change the interface without changing the underlying functionality, making it possible to adapt to their liking, or even change the code to adapt to their algorithms (different from rational closure).

If experts are inclined, and skilled enough, to contribute to the tool, they could. Since it is open-source and free the community of researchers could make sure it is both as accurate as possible and as useful as it can be; this makes it easier to attract experts when they have trusted colleagues that contribute to its betterment.

3.2. Starting project:

As a pilot project, I built an introductory website that has a downloadable link to the GUI. The website has 2 core sections: an introductory section where theoretical concepts are explained and links to more in-depth content will be provided, as well as links to videos that aid in auditory learning as opposed to the ample research papers available for reading.

Figure 3: GUI for rational closure

The GUI has a simple 3-step process to operate:

(1) Download the GUI onto your machine and ensure you have a java environment to run it.

(2) Enter a custom knowledge base in a .txt file along with your defeasible query at the top. Then put this file into the directory including the GUI.

(3) Run the GUI and enter your knowledge base name, as well as if you want explanations along with the conclusion (regular) or simply the conclusion (binary) for fast testing.

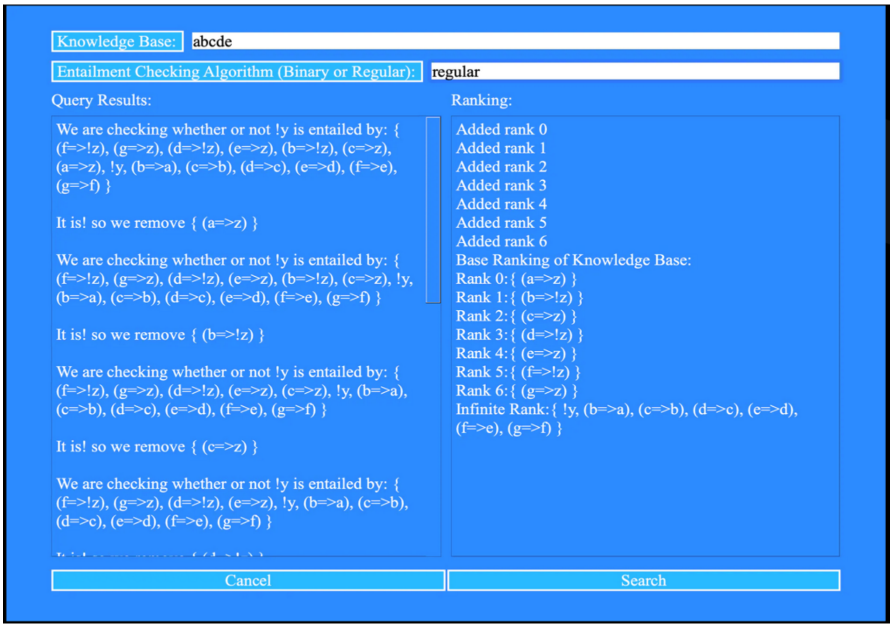

Figure 4: GUI interface with response to defeasible query

Both will be easily accessed at defeasible-knowledge.verse.app and a GitHub repository (when made public) respectively.

4. Methodology

The global penetration rate for the internet, as of 2022, is 63%, which is roughly 4.5 billion people across the world with most in the age range 25-34 years. This website and GUI would leverage that exposure to build a website including a downloadable GUI tool. Both will be designed to be easily accessed without restrictions such as paywalls or arbitrary logins

From there we will ask some users for feedback on their experience with the tool; this feedback is intended to be mostly informal as opinions about what experts think about (1) the accuracy of the information, and (2) the usefulness of the tool in their experiments. We would also ask them (3) if they like the idea of tools like this for all major KR algorithms, to gauge whether practical research like this should be pursued in the future.

This research is largely deductive, testing the hypothesis that internet tools will allow more exposure to KR for novices and experts alike and that it would improve the diagnosis of Rational Closure conclusions to defeasible queries on custom knowledge bases for experts.

4.1. Steps in methodology

(1) Build and publish a website with downloadable GUI
(2) Get feedback from both expert and novice researchers on the accuracy and usefulness of both the website and GUI, then iterate both by taking into account the feedback received

5. Resources and risks

5.1. Resources

This project, like most websites and online tools, is not resource-heavy, therefore the requirements area laptop with enough computing power to design and build both website and GUI, as well as time to iterate and improve both.

In addition to these requirements are:

- Java for the Rational Closure algorithm
- Python for GUI construction
- JavaScript, HTML, and CSS for website construction
- Access to website hosting service (vertical
- Novice and expert volunteers to give feedback on the tools

5.2. Risks

This project is laid out in such a way that we swerve past major risks such as not getting data on an algorithm that runs on large amounts of data. Our focus is much more practical and the starting project shows that the core tool is already working.

Potential Social Risks	Solutions to Social Risks	Likelihood
My supervisor may become unresponsive in an emergency.	Report to the head of the department not to contact the supervisor if possible, or plan for an alternative marker if the supervisor is not in a condition to be contacted.	Very Unlikely
Potential Academic/Technical Risks	**Solutions to Academic/Technical Risks**	**Likelihood**
The website does not get built out fully with a tool walkthrough and theoretical introduction.	Since the GUI is done, if this will be the case I will cut the theoretical section and focus on the website walkthrough.	Unlikely
Course responsibilities could cause me to miss milestones and become late on the project.	I will continuously communicate with my supervisor to make sure we keep on track with the goals.	Unlikely

6. Timeline and milestones

6.1. Timeline

The timeline for this project spans nearly 2 months after the submission of this proposal (period for background knowledge acquisition not included) and includes (1) a week for building the website, since it has already made great progress, (2) a week for user feedback from researchers as well as a chance to iterate and change the GUI and website where necessary, and (3) a month for writing and submitting the research report of our conclusions about the contribution of this project.

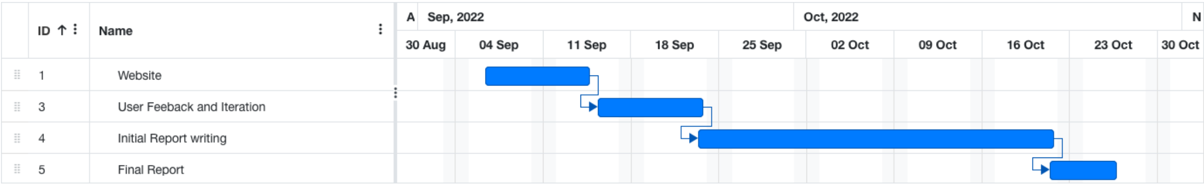

Figure 5: Timeline for research

6.2. Milestones

- 6 Sept: Website theory-based section complete
- 14 Sept: Website walk-through for tool section added and complete
- 18 September: Report layout and contents decided
- 25 September: Feedback from users
- 20 October: Report drafted
- 26 October: Report complete

7. Anticipated outcome

This research is expected to contribute toward novice researchers` ease of access to learning resources for and understanding of Rational Closure, and aid expert researchers in diagnosing the steps Rational Closure took to get to a conclusion. Both groups should be able to create custom Knowledge Bases that could be queried through the GUI.

8. References

[1] Ben-Ari, M.: Mathematical logic for computer science. Springer Science & Business Media (2012)

[2] Gallier, J.H.: Logic for computer science: foundations of automatic theorem proving. Courier Dover Publications (2015)

[3] Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence 44(1-2), 167-207 (1990)

[4] Lehmann, D.: Another perspective on default reasoning. Annals of Mathematics and Artificial Intelligence 15(1), 61-82 (1995)

[5] Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial intelligence 55(1), 1-60 (1992)

[6] Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.L.: A non-monotonic description logic for reasoning about typicality. Artificial Intellegigence 195, 165-202 (2013)

[7] Freund, M.: Preferential reasoning in the perspective of Poole default logic. Artificial Intelligence 98(1-2), 209-235 (1998)

A special acknowledgment to Adam Kaliski's MSc thesis titled "An Overview of KLM-Style Defeasible Entailment" and Joel Hamilton's Rational Closure algorithm implemented in Java.