

A Knowledge Representation tool to assist Rational Closure diagnosis

Anri Lombard

Supervisor: Tommie Meyer

Department of Computer Science

University of Cape Town

lmbanr001@myuct.ac.za

1 ABSTRACT

Rational Closure has shown that non-monotonic defeasible reasoning [1] [2] (reason with “common sense”) can be mathematically modeled and constructed algorithmically to assist computer science researchers in thinking about artificial intelligence, although novice researchers have been left to climb the steep learning curve alone and experts are expected to do laborious, long-winded calculations by hand. To solve both problems we set out to build both an accessible platform where novices could easily learn more about Rational Closure and experts could use a GUI to run experiments on defeasible implications of custom knowledge bases.

2 INTRODUCTION

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 reasoning better.

2.1 Problem Statement

Novice researchers are rarely exposed to Rational Closure and are 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 do arduous calculations for hours to achieve a single conclusion from a Rational Closure algorithm, wasting time on a trivial task.

Thus, the problems this paper will 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.

2.2 Motivation

The Rational Closure Algorithm has essential 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 they might want to work on. Similarly, expert researchers already devote time to building reasoning machines. Helping them cut down on trivial parts, such as getting a result from an algorithm rather than by hand, will reduce wasted time.

2.3 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 allows 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 and Python for the GUI makes it easy to customize if users decide to do so.

3 BACKGROUND

To understand the scope of this research, we need to understand the underlying theory; specifically, the algorithms Base Rank and Rational Closure which both make up the GUI for our expert users.

3.1 Algorithms

Algorithm 1: BaseRank

```
1: Input: A knowledge base  $\mathcal{K}$ 
2: Output: An ordered tuple  $(R_0, \dots, R_{n-1}, R_\infty, n)$ 
3:  $i := 0;$ 
4:  $E_0 := \vec{\mathcal{K}};$  while  $E_{i-1} \neq E_i$  do
5:
    $E_{i+1} := \{\alpha \rightarrow \beta \in E_i \mid E_i \vDash \neg\alpha\};$ 
    $R_i := E_i \setminus E_{i+1};$ 
    $i := i + 1;$ 
8:
9:  $R_\infty := E_{i-1};$  if  $E_{i-1} = \emptyset$  then
10:
    $n := i - 1;$  else
11:
    $n := i;$ 
12:
13: return  $(R_0, \dots, R_{n-1}, R_\infty, n)$ 
```

3.1.1 BaseRank. The BaseRank algorithm [3] takes in the user's knowledge base that the query needs to run through, then maps every explicit formula (a piece of knowledge) to a natural number (0, 1, ..., etc) representing its base rank.

To understand the BaseRank algorithm, we need to develop the concept of a **base rank** further:

Definition 3.1. $br_{\mathcal{K}}(\alpha)$, referred to as the base rank of α , with respect to a given defeasible knowledge base \mathcal{K} , is the smallest r such that α is not exceptional in $\mathcal{E}_r^{\mathcal{K}}$, and is defined: $br_{\mathcal{K}}(\alpha) := \min\{r \mid \mathcal{E}_r^{\mathcal{K}} / R \vdash \neg\alpha\}$

To give an intuitive overview of an algorithm, we could use the following example

Example 3.2. Let p, sp, b, f, w, ft represent penguins, special penguins, birds, fly, wings and feathers respectively. Let our knowledge base K consist of the following statements

- (1) Penguins are birds
- (2) Special penguins are penguins
- (3) Birds have feathers
- (4) Typically birds fly
- (5) Typically penguins do not fly
- (6) Typically special penguins fly
- (7) Typically birds have wings

This is represented by

$$p \rightarrow b, sp \rightarrow p, b \rightarrow ft, b \vdash f, p \vdash \neg f, sp \vdash f, b \vdash w$$

Using the BaseRank Algorithm, we obtain the following ranking of our statements once they have been returned to their defeasible form.

0	$b \vdash f, b \vdash w$
1	$p \vdash \neg f$
2	$sp \vdash f$
∞	$p \rightarrow b, sp \rightarrow p, b \rightarrow ft$

Given the exact same knowledge base in our GUI, we could see that it would return **True** if the query is either $b \vdash f$ or $b \vdash w$, meaning in this current knowledge base the most defeasible (thus least ranked) statements are "Birds can fly" and "Birds have wings."

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 implications with lower ranks are more general statements in the current knowledge base, which makes that statement more defeasible. Knowing 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.

Algorithm 2: Rational Closure Algorithm [3]

```

1: Input: A knowledge base  $\mathcal{K}$ , and a defeasible implication  $\alpha \vdash \beta$ 
2: Output: true, if  $\mathcal{K}\alpha \vdash \beta$ , and false otherwise
3:  $(R_0, \dots, R_{n-1}, R_\infty, n) := \text{BaseRank}(\mathcal{K})$ ;
4:  $i := 0$ 
5:  $R := \bigcup_{j=0}^{j < n} R_j$ ; while  $R_\infty \cup R \vDash \neg\alpha$  and  $R \neq \emptyset$  do
6:    $R := R \setminus R_i$ ;
7:    $i := i + 1$ ;
8: return  $R_\infty \cup R \vDash \alpha \rightarrow \beta$ ;

```

3.1.2 Rational Closure. Rational Closure is the main topic of this report and is a form of nonmonotonic entailment. Using this definition of base rank, a form of nonmonotonic entailment called *rational closure* [5] can be defined as such [4]:

Definition 3.3. Given a knowledge base \mathcal{K} , a defeasible implication $\alpha \vdash \beta$ is said to be in the rational closure of \mathcal{K} , and written $\mathcal{K}_{RC}\alpha \vdash \beta$, if and only if $br_{\mathcal{K}}(\alpha) < br_{\mathcal{K}}(\alpha \wedge \neg\beta)$ or $br_{\mathcal{K}}(\alpha) = \infty$.

To understand Rational Closure we need to define **minimal ranked entailment**:

Definition 3.4. Given a defeasible knowledge base \mathcal{K} , the minimal ranked interpretation satisfying \mathcal{K}_{RC} , defines an entailment relation, called minimal ranked entailment, such that for any defeasible implication $\alpha \vdash \beta$, $\alpha \vdash \beta$ if and only if $\mathcal{K}_{RC}\alpha \vdash \beta$.

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.

It has been shown by Freund [?] that RationalClosure, given a defeasible knowledge base \mathcal{K} and a query $\alpha \vdash \beta$, returns true if and only if $\alpha \vdash \beta$ is in the rational closure of \mathcal{K} :

THEOREM 3.5. Given a knowledge base \mathcal{K} and a query $\alpha \vdash \beta$ as input, RationalClosure returns **true** if and only if $\mathcal{K}_{RC}\alpha \vdash \beta$.

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.

4 METHODOLOGY

The main approach to fulfill our research objectives was to build both a website and GUI for novices and experts respectively, to have a platform that houses resources to further explore Rational Closure and a tool for users to run experiments faster. To validate

both the website and GUI we gathered informal feedback from 3 novices and 3 experts to narrow the scope of future improvements needed and the current effectiveness of the tools in assisting our target audience.

The software to run the Rational Closure Algorithm was built a former researcher, Joel Hamilton, who implemented a Java algorithm that runs both the Base Rank and Rational Closure algorithms, and with some manipulation could return 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.

4.1 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.

- (2) The second piece is the website that hosts the download 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.

5 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.

5.1 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.**

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 (figure 1).

- **Introduction.**

Users get introduced to Artificial intelligence, Knowledge Representation, and Defeasible Reasoning through YouTube videos, an 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 intelligence (figures 2, 3, 4).

- **Key Concepts.**

Besides the introduction, interested users would eventually encounter more semantic-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 (figures 5, 6, 7 and 8).

- **Tool Walk-through.**

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 through how to make queries and add custom knowledge bases to the program (figures 9 and 10).

- **Contribution Section.**

As we will discuss in the Future Research section, this project is set up well to attract developers and researchers who are eager to expand on it (figure 11).

5.2 GUI

Our Graphical User Interface (as seen in figure 12) 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 have 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 inspecting how Rational Closure was programmed if they intend to get a more thorough grasp of it.

6 EXPERIMENTS AND RESULTS

This project has the objectives of introducing novice users to Rational Closure and assisting 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 against, therefore comparison experiments 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) its usefulness, and (2) its accuracy of the information, where experts were asked, in addition to the latter questions, (3) whether the GUI gave them useful results.

6.1 Novices

Novices are an integral part of the research community since they could become future researchers in the field and contribute, but they need resources to introduce complex materials, thus focussing on their feedback will contribute to our project's goals of helping them. In all, we got feedback from 3 novices, who gave wide-ranging input on what worked for them and what did not, and gave their perspectives on what could improve in the project.

Novice 1

The first novice is a second-year university computer science student, which represents the demographic we would like to attract to the project.

"Overall, an incredible website and a job very well done... I found it very refreshing, it isn't too flashy, and very professional. just the minor changes need to be made as specified here," which was followed by an intricate explanation of details that need changing, including spelling errors and explaining necessary concepts more thoroughly, since novices would be intimidated by random concepts being casually used without proper explanation.

Novice 2

The second novice is a university graduate with a bachelor's degree. What makes this reviewer particularly interesting is his reputation as an academically strong student throughout his college career, making his perspective important since most novice researchers would have a similar background.

"I think it's quite useful, presents enough information without overwhelming the user. Minor spelling mistakes here and there but I didn't spot too many, maybe less than 5... The tutorial is clear as well."

Novice 3

The last novice has done research in Knowledge Representation some 3 years ago, but do not consider themselves a computer scientist by profession, representing a slightly more experienced demographic that is not computer scientists. With this website and GUI, we could attract a similar demographic by showing that the information could be much more interesting and digestible than previously thought.

This reviewer gave some critical feedback on the struggles of a novice that needs to be attended to: "Actually, I have found the walkthrough a bit problematic for someone that is not a computer scientist (and I am not). I had neither Java nor Python installed on my machine, and it took a while to install java (it took a while and two installations to understand that I needed Java 15 in particular) and some help in using the terminal (as I mentioned, I am not a

computer scientist)."

He also mentioned the difficulty of starting the tool, affirming the motive of improving the format for downloading the tool, especially for non-expert and non-computer-scientist users: "... it seems impossible for me to start the tool with the nice interface you show in your walkthrough."

6.2 Experts

Experts form the pinnacle of research, as they have invested considerable amounts of time into the fields of their choice, being Knowledge Representation in this case, and have opinions that must be heeded by projects aiming to ease their research efforts over time - such as the GUI we built. In total, we got intricate criticism from 5 experts, who gave detailed and thoroughly-articulated critiques on the usefulness of our website and GUI.

Expert 1

The first expert was a former student who studied Rational Closure in depth. They have an in-depth perspective on how accurate the content on both the website and GUI is, as well as a much-needed view on what we could change to improve both.

Besides spelling mistakes, the reviewer pointed out that introducing the KLM properties might not be a good introduction to Rational Closure, as it would be jarring to newcomers who have no introduction to fundamental concepts such as non-monotonic reasoning and non-monotonic consequence relationships, that are mentioned in other paragraphs. They also point out that the general idea needs more emphasis that "nitty gritty" details, which could be left to the original papers and linked for those who are eager to learn more in-depth.

Overall, they found the website appealing, "Overall you've done great work with some very hard-to-grasp concepts," but contributed much to the perspective of the author and developer to rethink the introduction of complex materials.

Expert 2

The second expert is a student from Oxford University, emphasizing their role in representing an elite audience that might be interested in learning about Rational Closure. They focused on the negatives since they thought it might help the project more than an appraisal. Their views reflect the previous experts', again confirming the important aspects that need to change in the tools.

They specifically pointed out how jarring the textual information is in contrast to a much lighter introduction the videos give, recommending that the site should focus more on bullet point information rather than spread out paragraphs that could intimidate newcomers. They also made a useful observation about optional visibility, for example showing the paragraph (with more examples) to novices, but hiding it from experts like themselves, prompting the idea that more investment into personalizing the website for each user could be a worthwhile investment.

With this feedback at hand, the tool needs to be refocused on gradually introducing difficult mathematical concepts and personalizing the experience for each user based on their experience; otherwise, both novices and experts would be repelled.

Expert 3

The last expert focused more on the positives of the project, pointing out what worked and mentioning what did not.

The reviewer made a crucial observation about the usage of the downloadable GUI, specifically mentioning that it should be in an easy-to-download format for each machine, such as a .dmg file for macs and a matching format for other machines. Their aesthetic taste also proved helpful as they encouraged us to focus on improving the fonts and style of connectives (such as replacing the connective with words that are easier to understand and use).

Their feedback points to the difficulty of downloading the tool once again, emphasizing the importance of access to tools that this project aims to contribute to.

6.3 Trends amongst both novices and experts

Both novices and experts brought up areas where this project fell short on its promise, as well as where it excelled.

Novices in particular seemed to enjoy the aesthetics and simplicity of the website. Not one, for example, found it difficult to access the site for further exploration, and their feedback reflected their general contentedness with the video-and-text format we chose.

Experts, on the other hand, gave much more rigorous and directed critiques about improving (1) the accuracy of the information, (2) the introduction of complex concepts, (3) the aesthetic of both the website and the Graphical User Interface, and (4) the experience of both video and text formats that could flow better.

7 CONCLUSIONS

The objectives of this research project were to introduce the difficult concept of Rational Closure to novice researchers through an accessible medium they are familiar with and help expert researchers with diagnosing what Rational Closure does to reach conclusions on their defeasible queries on custom databases of their choice.

We chose a wide-ranging audience of novices and experts to represent the audience we would like to assist, and found their reviews to point out specific areas where this project works well, and others where it needs improvement. Specifically, this project excels at being accessible to novices and experts alike, although the current method of using the GUI has proven to be challenging and requires further iteration to ease its use. The format of introduction through videos seems to work, although better videos and more examples in textual format will be useful for a more gradual and enjoyable introduction to the seemingly-abstract concept of Rational Closure.

Overall, this project has made the first few steps in diffusing the complex and hidden concepts in Knowledge Representation to a level where nascent researchers could explore it, which was the intended purpose.

8 FUTURE RESEARCH

Both the GUI and website were specifically chosen as they have the potential to be expanded by developers and researchers 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.

Ultimately, the website serves as a platform that displays tools that could be used for experiments and guides on learning complex concepts associated with Knowledge Representation.

9 ACKNOWLEDGEMENTS

The author would like to thank the reviewers (in no specific order) Kialan Pillay, Sibusiso Buthelezi, Matthew Morris, Giovanni Casini, Emily Morris, and Ivan Varzinczak for adding their crucial feedback to this project.

A special thanks goes to Adam Kaliski 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, as well as Tommie Meyer who contacted the important persons and guided the research effort.

REFERENCES

- [1] BONATTI, P. A. Rational closure for all description logics. *Artificial Intelligence* 274 (2019), 197–223.
- [2] BREWKA, G., NIEMELÄ, I., AND TRUSZCZYŃSKI, M. Nonmonotonic reasoning. *Foundations of Artificial Intelligence* 3 (2008), 239–284.
- [3] CASINI, G., MEYER, T., AND VARZINCZAK, I. Taking defeasible entailment beyond rational closure. In *European Conference on Logics in Artificial Intelligence* (2019), Springer, pp. 182–197.
- [4] GIORDANO, L., GLOZZI, V., OLIVETTI, N., AND POZZATO, G. L. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence* 226 (2015), 1–33.
- [5] LEHMANN, D., AND MAGIDOR, M. What does a conditional knowledge base entail? *Artificial intelligence* 55, 1 (1992), 1–60.

10 APPENDIX

10.1 Full, unaltered expert comments

Expert 1

They had 6 critiques in particular:

- (1) "Reasoning is spelled incorrectly and there's inconsistent use of '-' in 'non-monotonic'."
- (2) "I think it would be nice to have an example (the classic penguin one) is always good to capture the intuition of why we want non-monotonic reasoning. (Although I know it is in the video, may be nice to include it in the text for those just skim viewing)"
- (3) "It may be worth introducing the key concept of non-monotonic reasoning before concepts such as defeasible reasoning. It may be slightly jarring to a reader to see a concept such as non-monotonicity used but only explained further down the page"
- (4) "May be worth linking the reader to the original paper that introducing these concepts and mentioning that that rational closure can be defined both semantically and algorithmically. This section just seems slightly disjoint with the following section on the algorithm."
- (5) "I've never personally found the KLM properties to be a good place to start when trying to understand Rational Closure. In my opinion, it may just be worth mentioning that they exist and referring the reader to other papers if they wish to read about them rather than putting them in the website. If they're here, it would be nice to have some kind of intuitive explanation of them so the user is overwhelmed by the math."
- (6) "You never fully explain what a non-monotonic consequence relationship is. I would caution against throwing in too much terminology, as this makes trying to grasp the general idea more overwhelming. This is a bit of a general trend through out the website, I would suggest trying to convey a general idea on the website and referring the reader to papers for the more nitty gritty content."

Expert 2

Specifically, their feedback was:

- (1) "The videos were a nice introduction. Given how light the format was until that point, the blocks of text in 2.1 were quite off-putting."
- (2) "It would be better served by shorter bullet-point-style information and examples. 2.2 is a bit of a brutal introduction to LM-rationality, since we've barely seen any math up until this point, yet alone defeasible entailment operators or how properties are stated. This needs to be introduced more gradually and preferably with examples as well."
- (3) "Likewise, 2.3 is a brutally fast introduction for anyone who doesn't know the syntax and hasn't worked with the algorithms before like I have. Showing it to someone like me isn't helpful since I know it already, and showing it without explanation to someone who doesn't have experience isn't helpful either since there's no way for them to understand it. Your link to Adam's 100-page long thesis won't encourage or help the reader either."

- (4) "Overall it is quite sleek and well-presented: I would just have a bit more of a think about who the target market is for this and set the level of the explanations consistently at them."

Expert 3

Their specific feedback was:

- (1) "Yes, it is accurate. (But that also depends on how deep you or the users want to go into the topic.)"
- (2) "Yes, it is effective."
- (3) "I find the soundtrack in the KRR video too distracting. I enjoyed the video on Defeasible Reasoning much more than the one on KRR. Regarding the GUI, I would choose different fonts and colors. The logical connectives can be improved; maybe replacing them with natural language constructs such as "not", "and", and so on would do the job. I've downloaded the program but couldn't play with it as neither my Python nor my Java installations are set up correctly... You may consider having a .dmg Mac version that one can run within a minute after downloading it"

10.2 Screenshots

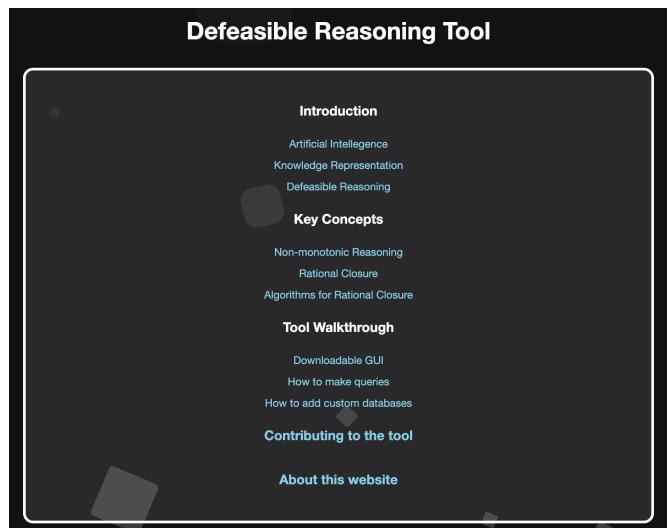


Figure 1: Contents Section

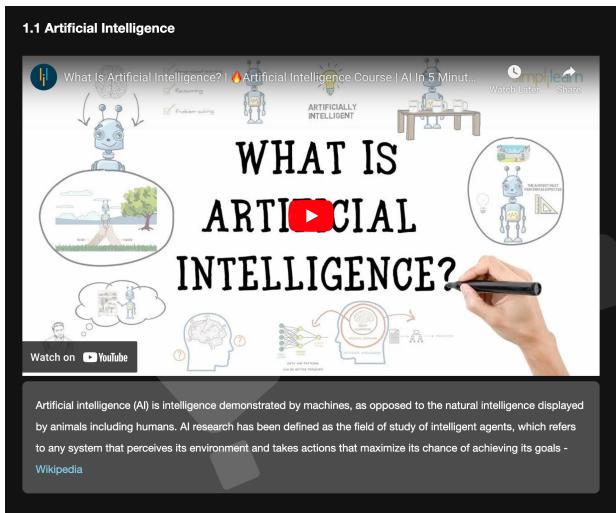


Figure 2: Introduction to Artificial Intelligence

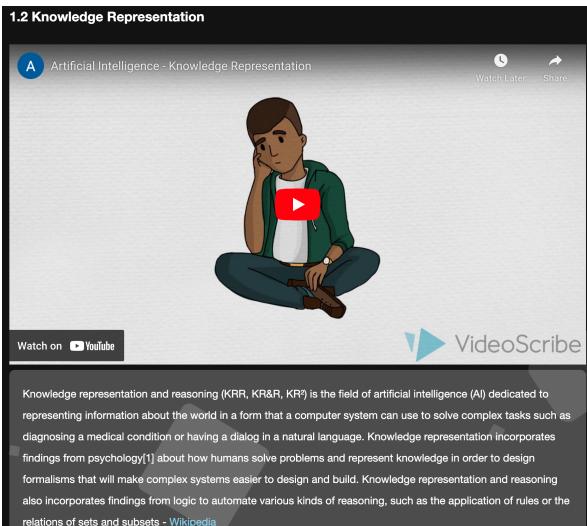


Figure 3: Introduction to Knowledge Representation

The screenshot shows a YouTube video player with the title 'Propositional Defeasible Explanation'. The video cover features a blue background with the title in white. Below the title, it says 'UCT Hon Project in Knowledge Representation and Reasoning' and 'SCHOOL OF IT'. At the bottom, there is a summary text: 'Defeasible reasoning is an important form of non-monotonic reasoning because it introduces mechanisms for strengthening and weakening new information presented to a reasoner. Defeasible reasoning also deals with atypical reasoning situations in which humans inherently reason differently, whether due to their beliefs, context or other circumstances. Defeasible reasoning and its mechanisms allow AI systems to represent human reasoning more accurately than classical reasoning and more accurately than the basic notion of non-monotonic reasoning.' There is also a link to 'Propositional Defeasible Explanation'.

Figure 4: Introduction to Defeasible Reasoning

The screenshot shows a page titled '2.1 Non-monotonic Reasoning'. It contains two sections of text. The first section, 'Nonmonotonic reasoning', describes it as an area of research that attempts to formalize different patterns of "common sense" reasoning, by dropping monotonicity as a property and investigating how to define reasonable notions of consequences. The second section, 'Nonmonotonic reasoning', describes it as a subfield of Artificial Intelligence trying to find more realistic formal models of reasoning than classical logic. It notes that common sense reasoning often draws conclusions that have to be withdrawn when further information is obtained. A quote from Stanford University follows: 'Nonmonotonic reasoning methods try to formalize this phenomenon. This volume gives an overview on recent results in the field and points to relevant literature for further study.' Below these sections, there is a link to '2.1 Non-monotonic Reasoning'.

Figure 5: Non-monotonic explanation

The screenshot shows a page titled '2.2 Rational Closure'. It starts with a definition: 'A particularly appreciated non-monotonic consequence relation is represented by the class of rational consequence relations. A consequence relation |- is rational iff it satisfies the following properties:' followed by a list of seven properties. Properties 1-5 are grouped under '(LLE) Left logical equivalence':
1. (LLE) Left logical equivalence: $\frac{\mathcal{K} \models \alpha \leftrightarrow \beta, \mathcal{K} \models \alpha \vdash \gamma}{\mathcal{K} \models \beta \vdash \gamma}$
2. (RW) Right weakening: $\frac{\mathcal{K} \models \alpha \rightarrow \beta, \mathcal{K} \models \gamma \vdash \alpha}{\mathcal{K} \models \gamma \vdash \beta}$
3. (Ref) Reflexivity: $\mathcal{K} \models \alpha \vdash \alpha$
4. And: $\frac{\mathcal{K} \models \alpha \vdash \beta, \mathcal{K} \models \beta \vdash \gamma}{\mathcal{K} \models \alpha \wedge \beta \vdash \gamma}$
5. Or: $\frac{\mathcal{K} \models \alpha \vdash \gamma, \mathcal{K} \models \beta \vdash \gamma}{\mathcal{K} \models \alpha \vee \beta \vdash \gamma}$
Properties 6-7 are grouped under '(CM) Cautious Monotonicity':
6. (CM) Cautious Monotonicity: $\frac{\mathcal{K} \models \alpha \vdash \beta, \mathcal{K} \models \beta \vdash \gamma}{\mathcal{K} \models \alpha \wedge \beta \vdash \gamma}$
7. (RM) Rational Monotonicity: $\frac{\mathcal{K} \models \alpha \vdash \beta, \mathcal{K} \models \alpha \nvdash \beta}{\mathcal{K} \models \alpha \wedge \beta \vdash \gamma}$
Below the properties, there is a note: 'Each postulate corresponds to a pattern of reasoning about Defeasible Information.' and a link to 'Here is a more in-depth explanation.'

Figure 6: Rational Closure explanation

2.3 Algorithms for Rational Closure

The first algorithm is **BaseRank**, which is an algorithm on the knowledge base that computes the base rank. The lower the rank for the statement the more *defeasible* it is, and infinite ranks are classical statements, true in every valuation in every ranked model of the knowledge base.

Algorithm 1 BaseRank

```

1: Input: A knowledge base  $\mathcal{K}$ 
2: Output: An ordered tuple  $(R_0, \dots, R_{n-1}, R_\infty, n)$ 
3:  $i := 0$ ;
4:  $E_0 := \vec{\mathcal{K}}$ ;
5: while  $E_{i-1} \neq E_i$  do
6:    $E_{i+1} := \{\alpha \rightarrow \beta \in E_i \mid E_i \vDash \neg\alpha\}$ ;
7:    $R_i := E_i \setminus E_{i+1}$ ;
8:    $i := i + 1$ ;
9: end while
10:  $R_\infty := E_{i-1}$ ;
11: if  $E_{i-1} = \emptyset$  then
12:    $n := i - 1$ ;
13: else
14:    $n := i$ ;
15: end if
16: return  $(R_0, \dots, R_{n-1}, R_\infty, n)$ 
```

Rational Closure is an alternative syntactic definition of minimal ranked entailment, whereby it takes in a defeasible query $a \vdash b$ and returns *true* if it is entailed by the knowledge base.

Figure 7: Rational Closure Algorithm explanation

Algorithm 2 RationalClosure

```

1: Input: A knowledge base  $\mathcal{K}$ , and a defeasible implication  $\alpha \vdash \beta$ 
2: Output: true, if  $\mathcal{K} \models \alpha \vdash \beta$ , and false otherwise
3:  $(R_0, \dots, R_{n-1}, R_\infty, n) := \text{BaseRank}(\mathcal{K})$ ;
4:  $i := 0$ 
5:  $R := \bigcup_{j=0}^{i-1} R_j$ ;
6: while  $R_\infty \cup R \models \neg\alpha$  and  $R \neq \emptyset$  do
7:    $R := R \setminus R_i$ ;
8:    $i := i + 1$ ;
9: end while
10: return  $R_\infty \cup R \models \alpha \rightarrow \beta$ ;
```

The above algorithm essentially works by checking if there exists an exceptional subset of the knowledge base such that the query is entailed. If the antecedent (ancestor formula) of the lowest base is exceptional (does not entail the knowledge base for any valuation) then the lowest rank is removed.

Both algorithms were described [here](#), along with a thorough explanation of the KLM framework for those interested.

Figure 8: Base Rank Algorithm explanation

3.1 Downloadable GUI

[Click Here To Download GUI](#)

3.2 How To Make Queries

Prerequisites: Make sure you have Python and Java installed on your machine

Step 1: Download the Defeasible Reasoning GUI

Step 2: Add own database along with defeasible query in the form of a .txt file (see section 3.2)

Step 3: Enter the database name without the .txt extension and choose the entailment checker algorithm you need.

- Regular simply outputs the process whereby Binary outputs "true" or "false"

Step 4: Grab a cup of coffee and relax, the rest will be done for you.

Figure 9: Step-by-step instructions to make queries with tool

3.3 How To Add Custom Databases

Step 1: Add a defeasible query to the top of a .txt file. The Defeasible query must comply with the [tweetyproject](#) standards, where a defeasible query is denoted by \rightarrow .

Step 2: Add the knowledge base formulas directly below the defeasible query in the .txt file.

Step 3: Ensure that the file is visible by placing it in the root folder (containing the src and target folders).

Figure 11: Contribution Guidelines

Figure 10: Step-by-step instructions to create custom knowledge bases for a tool to use

4. Contributing to the tool

Developer and experts could contribute to this [repository](#) either through a [pull request](#) to modify the code or an [issue](#) for changes you would like to see.

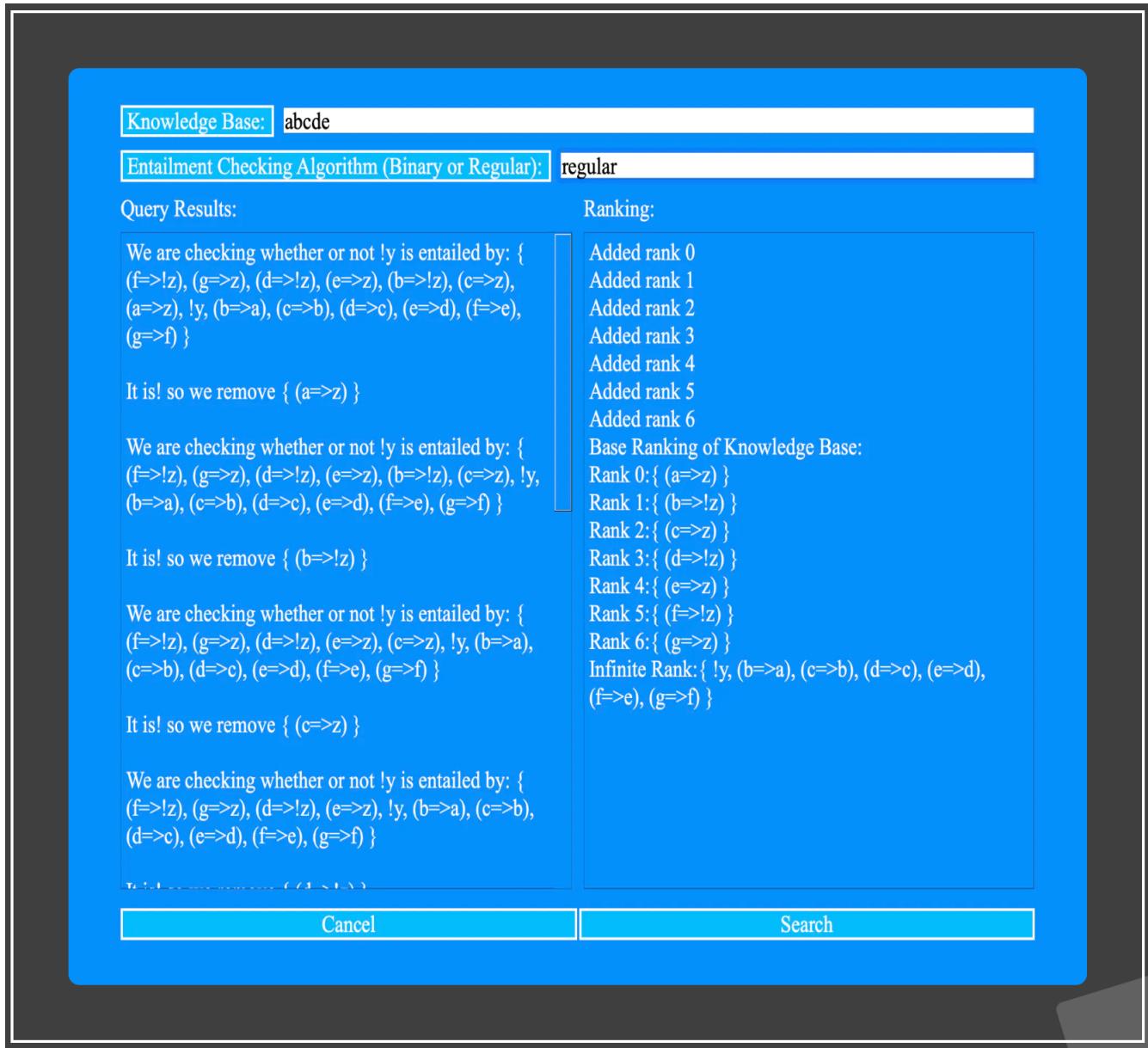


Figure 12: Graphical User Interface