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 [4] [5] (reasoning with "common sense") can be mathematically modeled and constructed algorithmically to assist computer science researchers in thinking about artificial intelligence. But 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 [7] 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

Rational Closure is a non-monotic algorithm for reasoning about facts in a knowledge base. Non-monotonic meaning that new knowledge are taken into account when reasoning about old knowledge, essentially using "common sense". Defeasible reasoning is reasoning non-monotonically, unlike classical reasoning which is monotonic, thus Rational Closure is a good representation thereof.

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 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 into how to build reasoning machines; therefore, it is crucial 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 [7] make up the GUI for our expert users, as well as Propositional Logic and Defeasible Reasoning [12] that are used to formulate Rational Closure.

3.1 Propositional Logic

Propositional logic is a formalism for reasoning about knowledge or information, abstracted away from natural language representation into formal language. It is essentially defined by a set of connectives along with a set of statements, such that more complicated statements may be constructed by combining statements using the connectives; the truth of the composite statement is then solely reliant on the truth of the base statements and the interpretation of the connectives used [1, 6].

Propositional logic was chosen as the base logic as it was the foundational logic chosen by Kraus, Lehmann and Magidor to initially define the KLM framework [9, 10].

Syntax

Propositional logic is built up from propositional atoms, also called variables or symbols [33, 63]. These are the basic building blocks of logic and are used to represent statements. Atoms are combined with a set of connectives $\{\neg, , , \rightarrow, \}$ to form more complex statements known as formulas.

Semantics

The semantics of logic defines the meaning of truth and falsity of a statement. In propositional logic, the semantics is defined by a set of truth assignments, which are mappings from the set of atoms $\{a, b, c, d$, etc.$\}$ to the set of truth values {T, F}. A truth assignment is said to satisfy a formula if the formula evaluates to T under that assignment. A formula is said to be satisfiable if there exists a truth assignment that satisfies it and is unsatisfiable if no truth assignment satisfies it.

Valuations, also known as worlds or interpretations [1, 9], assign truth to propositional atoms [11]. A valuation is a function from the set of atoms to the set of truth values {T, F}. A valuation is said to satisfy a formula if the formula evaluates to T under that valuation. A formula is said to be satisfiable if there exists a valuation that satisfies it and is unsatisfiable if no valuation satisfies it.

One formula (b) is said to be the *logical consequence* ($\vDash$) [8] of another formula (a) if the first formula is satisfied by all valuations that satisfy the second formula (hence $a \vDash b$). Similarly, they are logically equivalent if they are both logical consequences of each other.

A *Knowledge Base* ($\mathcal{K}$) is a finite set of formulas.

3.2 Classical and Defeasible Reasoning

Classical Reasoning

Classical logic is a formal system used in mathematics and computer science [2] to reason about truth and falsity. In classical logic, a proposition is either true or false. The truth value of a proposition is not affected by any other propositions. For example, if a proposition is false, adding any other proposition to it does not change its truth value. In classical logic, the truth value of a proposition is not affected by any other proposition. For example, if a proposition is false, adding any other proposition to it does not change its truth value.

Defeasible Reasoning

In defeasible logic [12], a formula is either true or false. The truth value of a formula is affected by other formulas. For example, if a formula is false, adding a true formula to it changes its truth value to true in the case of defeasible reasoning. In classical reasoning, this is not true, as the value of one formula does not affect another.

Difference between classical and defeasible reasoning

Consider the following two propositions:

Q: The sun is shining and it is raining.
P: The sun is shining.

In classical reasoning, the truth value of Q does not depend on the truth value of P. If P is false, then Q is still false. If P is true, then Q is still false. So, Q is not defeasible. In defeasible reasoning, the truth value of Q depends on the truth value of P. If P is false, then Q is false. If P is true, then Q is true. So, Q is defeasible.

3.3 Algorithms

Algorithm 1: BaseRank [7]

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**
    $E_{i+1} := \{\alpha \rightarrow \beta \in E_i \mid E_i \vDash \neg\alpha\}$;
$R_i := E_i \setminus E_{i+1}$;
$i := i + 1$;

$R_\infty := E_{i-1}$; **if** $E_{i-1} = \varnothing$ **then**
    $n := i - 1$; **else**
    $n := i$;

return $(R_0, ..., R_{n-1}, R_\infty, n)$

BaseRank Algorithm

The BaseRank algorithm works as follows:

(1) Start with a knowledge base, a set of formulas
(2) For each formula in the knowledge base, compute the set of all valuations that satisfy the formula
(3) For each valuation, compute the set of all formulas that are satisfied by that valuation
(4) For each formula, compute its rank as the number of valuations that satisfy the formula
(5) For each valuation, compute its rank as the number of formulas that are satisfied by that valuation

The BaseRank algorithm is an algorithm that computes the rank of each formula in a knowledge base. It is a recursive algorithm, in that it computes the rank of a formula by computing the rank of each valuation that satisfies that formula.

The *rank* (R_0 to R_∞) of a formula is the number of valuations that satisfy that formula. The rank of a valuation is the number of formulas that are satisfied by that valuation.

Basically, the Base Rank algorithm is a simple algorithm to calculate the base rank of a set of statements. It is based on the following rules:

- If a statement is a fact, its base rank is 1.
- If a statement is a logical consequence of other statements, its base rank is the sum of the base ranks of the statements that it is a consequence of.
- If a statement is a logical consequence of other statements, and the base ranks of the statements that it is a consequence of are equal, its base rank is the sum of the base ranks of the statements that it is a consequence of plus 1.

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.

Exceptional statements are statements that are logically inconsistent. This is a problem because it means that the knowledge base is incomplete. Rational closure is a method of filling in the gaps in the knowledge base by making assumptions about the missing information. It is a way of closing the gaps in the knowledge base that are left by exceptional statements. It is a way of adding information to the knowledge base that is not explicitly stated.

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 [7]

Input: A knowledge base $\mathcal{K}$, and a defeasible implication $\alpha \mid\!\sim \beta$
Output: **true**, if $\mathcal{K}\alpha \mid\!\sim \beta$, 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** $R_\infty \cup R \vDash \neg\alpha$ *and* $R \neq \varnothing$ **do**
    $R := R \setminus R_i$;
$i := i + 1$;

return $R_\infty \cup R \vDash \alpha \rightarrow \beta$;

Rational Closure Algorithm

The Rational Closure Algorithm works as follows:

(1) Given a knowledge base K and a formula , construct a new knowledge base K' by adding to K
(2) If K' is satisfiable, then K is consistent, and is a logical consequence of K. Thus, is added to K.
(3) If K' is unsatisfiable, then K is inconsistent, and is not a logical consequence of K. Thus, is not added to K.

The Rational Closure Algorithm is a sound and complete algorithm [3] for reasoning about knowledge bases. It is sound because it will never infer a false statement, and it is complete because it will always infer a true statement.

The algorithm terminates when a tautology is added to the knowledge base that makes it satisfiable.

A *tautology* is when a formula, *a*, is always true.

It has been shown by Freund [?] that `RationalClosure`, given a defeasible knowledge base and a query $\alpha \mid\!\sim \beta$, returns true if and only if $\alpha \mid\!\sim \beta$ is in the rational closure of :

Theorem 3.1. *Given a knowledge base $\mathcal{K}$ and a query $\alpha \mid\!\sim \beta$ as input,* `RationalClosure` *returns **true** if and only if $\mathcal{K}_{RC}\alpha \mid\!\sim \beta$.*

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 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 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.

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 10 and 9).

- 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] Ben-Ari, M. *Mathematical logic for computer science.* Springer Science & Business Media, 2012.
[2] Ben-Eliyahu, R., and Dechter, R. Default reasoning using classical logic. *Artificial Intelligence 84*, 1-2 (1996), 113–150.
[3] Blanchette, J. C., Popescu, A., and Traytel, D. Soundness and completeness proofs by coinductive methods. *Journal of Automated Reasoning 58*, 1 (2017), 149–179.
[4] Bonatti, P. A. Rational closure for all description logics. *Artificial Intelligence 274* (2019), 197–223.
[5] Brewka, G., Niemelä, I., and Truszczyński, M. Nonmonotonic reasoning. *Foundations of Artificial Intelligence 3* (2008), 239–284.
[6] Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., and Varzinczak, I. Rational defeasible reasoning for description logics. Tech. rep., University of Cape Town, 2018.
[7] 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.
[8] Etchemendy, J. The concept of logical consequence.
[9] Kraus, S., Lehmann, D., and Magidor, M. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence 44*, 1-2 (1990), 167–207.
[10] Lehmann, D., and Magidor, M. What does a conditional knowledge base entail? *Artificial intelligence 55*, 1 (1992), 1–60.
[11] Magnus, P. D. *Forall X: An Introduction to Formal Logic.* 2005.
[12] Nute, D. Defeasible logic. In *International Conference on Applications of Prolog* (2001), Springer, pp. 151–169.

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 distractive. 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

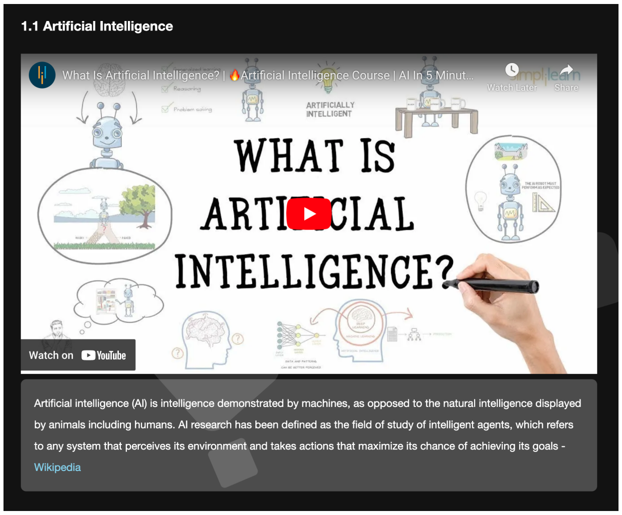

Figure 2: Introduction to Artificial Intelligence

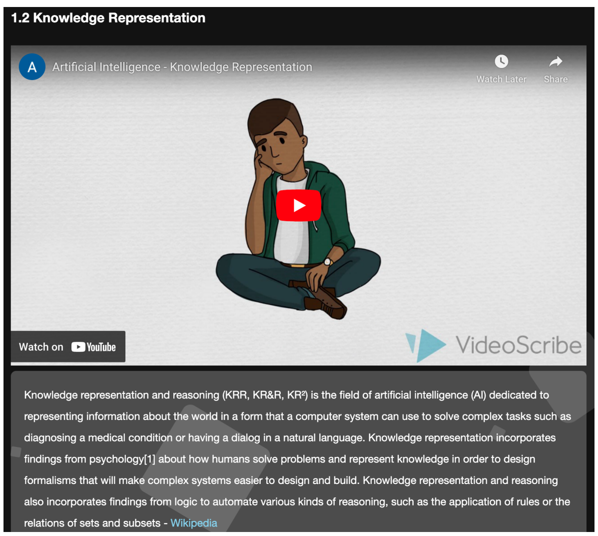

Figure 3: Introduction to Knowledge Representation

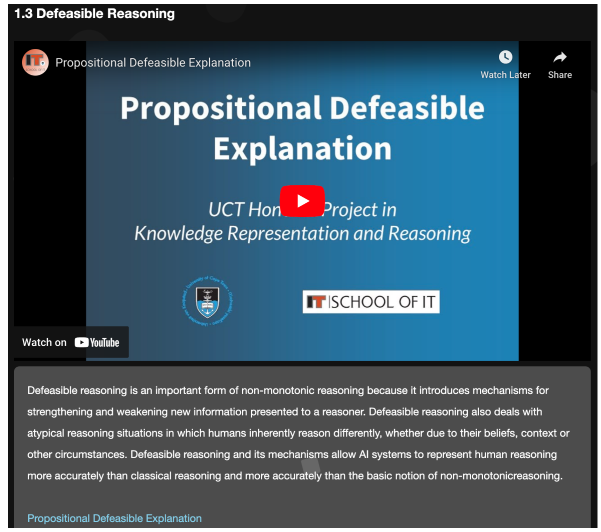

Figure 4: Introduction to Defeasible Reasoning

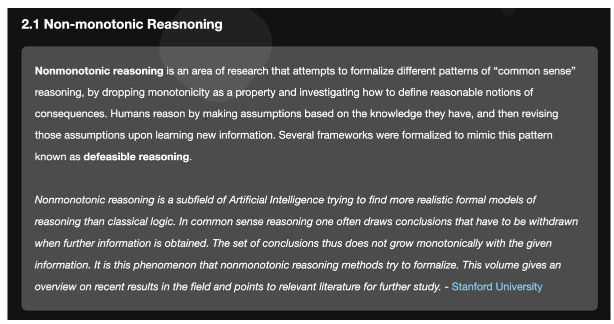

Figure 5: Non-monotonic explanation

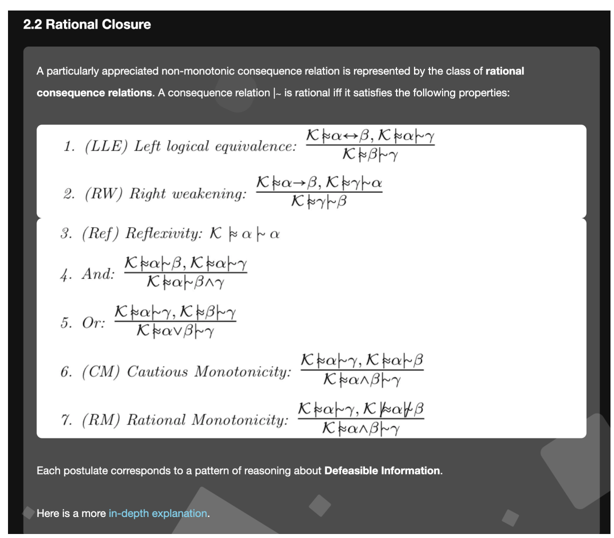

Figure 6: Rational Closure explanation

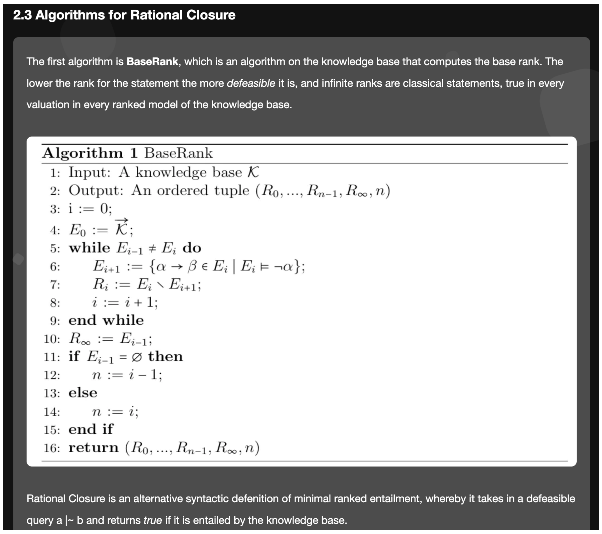

Figure 7: Rational Closure Algorithm explanation

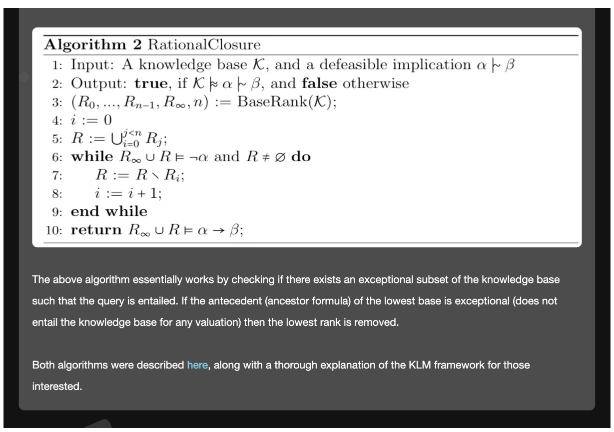

Figure 8: Base Rank Algorithm explanation

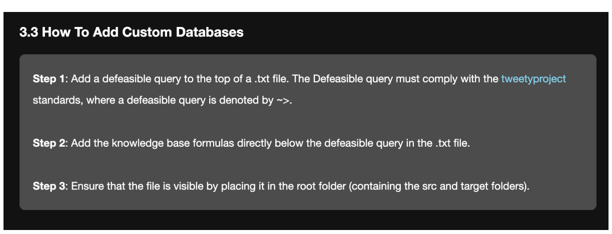

Figure 9: Step-by-step instructions to create custom knowledge bases for a tool to use

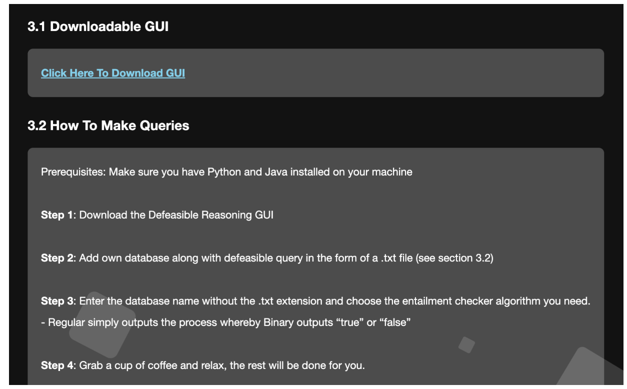

Figure 10: Step-by-step instructions to make queries with tool

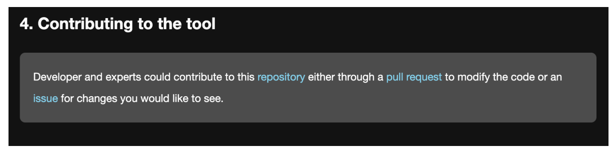

Figure 11: Contribution Guidelines

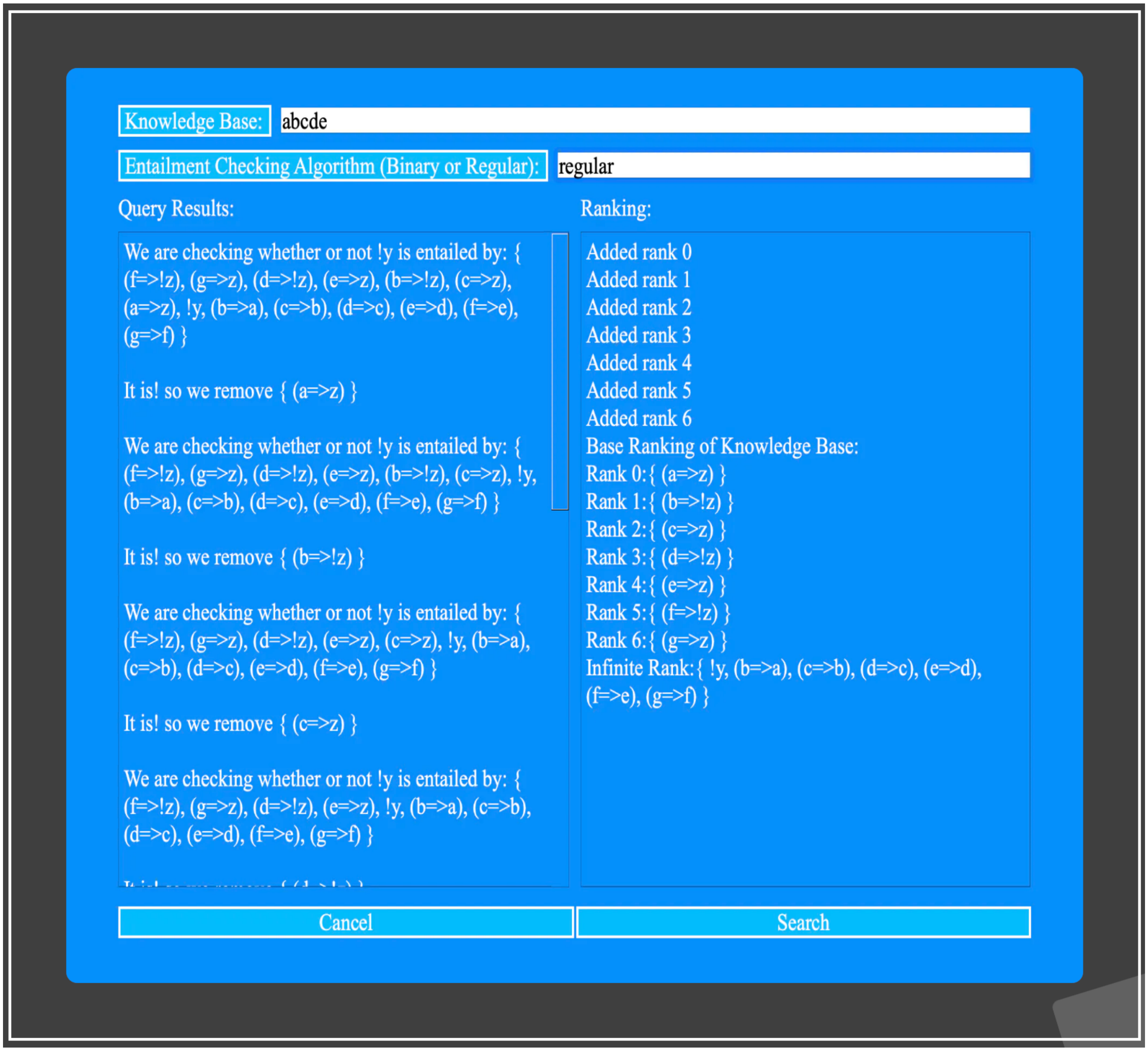

Figure 12: Graphical User Interface