WASHINGTON UNIVERSITY IN ST. LOUIS

McKelvey School of Engineering
Department of Computer Science & Engineering

Dissertation Examination Committee:
Yevgeniy Vorobeychik, Chair
Brendan Juba
Subbarao Kambhampati
Alvitta Ottley
William Yeoh

Explainable Decision-Making: From Formal Logic to AI Systems with Explainable Behavior

by
Stylianos Loukas Vasileiou

A dissertation presented to
the McKelvey School of Engineering
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

May 2025
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

Words will never be enough to express my feelings and gratitude to all the people that made it possible to complete my PhD research adventure. But as it is customary, I shall attempt to do so here.

First and foremost, to my advisor, Prof. William Yeoh. From the deepest foundations of my heart, thank you for everything. You may not have realized it, but accepting me as your PhD student at WashU, you gave me a chance at the life I always wanted. Your mentorship and guidance have fundamentally changed me for the better. You taught me how to navigate difficult academic and research challenges, how to conduct good research, and how to be a good researcher overall. Most importantly, you showed me how to be a better human being. Not only within academia, but also outside in the real world. You have been a legendary advisor and role model. And now, I am fortunate to have the honor to call you, my friend.

I am very grateful to my committee members: Profs. Yevgeniy Vorobeychik, Brendan Juba, Subbarao Kambhampati, and Alvitta Ottley. Our interactions throughout the years pushed me to be better and helped strengthen this thesis substantially. I would like to give a special thanks to Prof. Subbarao Kambhampati, whose pioneering work and leadership in Human-Aware AI gave life to this thesis. Thank you for supporting my work, and for being an exemplary figure in the community.

Throughout the years, I was lucky enough to collaborate with some amazing people. To all my collaborators and co-authors, thank you agreeing to work with me. I learned a great deal from you. A huge thanks goes to my wonderful YODA labmates: Ben Rachmut, Ashwin Kumar, Yinxu Tang, Jean Springsteen, Christabel Wayllace, Athena Tabakhi, and Khoi Hoang. I would also like to thank the anonymous reviewers of my papers who provided valuable feedback over the years and helped improve some of the work in this thesis. It will be remiss of me not to thank the infamous "Reviewer 2," whose negativity helped me build up resilience and perseverance.

Obviously, none of this would have been possible without the incredible support of my lovely family and friends. To Irene, Thanasis, Vicky, Rosalia, and Manos: thank you for fostering my curiosity and supporting me no matter what. To my Christina, thank you for being a wonderful partner and supporting me throughout my PhD journey, especially when I could not see the light at the end of the tunnel. Finally, to all of my good friends (you know who you are), thank you for your friendship and always being there to talk. You provided much-needed perspective and support throughout the years.

Stylianos Loukas Vasileiou

Washington University in St. Louis
May 2025

Στη μαμά μου.

ABSTRACT OF THE DISSERTATION

Explainable Decision-Making: From Formal Logic to AI Systems with Explainable Behavior

by

Stylianos Loukas Vasileiou

Doctor of Philosophy in Computer Science

Washington University in St. Louis, 2025

Professor Yevgeniy Vorobeychik, Chair

This thesis makes the claim that logic-based frameworks can serve as an *explainability layer* atop AI systems, capable of generating rigorous and flexible explanations for human users across diverse problem domains. We support this claim through a progression of novel theoretical frameworks and practical implementations, starting with a general logic-based framework for generating explanations from the knowledge bases of an AI system and a human user and showing how it can be used on a diverse set of problem domains. We then systematically extend this framework with capabilities crucial for real-world applications: probabilistic reasoning for handling uncertainty, personalization through vocabulary-based abstraction, and dynamic interaction through argumentative dialogues.

Building on these foundations, we address additional challenges by developing privacy-aware explanations for multi-agent systems and exploring explanation-guided approaches to belief revision that better align with human cognitive processes. To make our methods more accessible, we demonstrate how it can be effectively combined with large language models to generate natural language explanations while maintaining formal guarantees.

Our theoretical contributions are complemented by efficient computational methods that make these frameworks more practical, as demonstrated through extensive evaluations across diverse problem domains. Recognizing that the ultimate test of explanatory frameworks lies

in their effectiveness with real human users, we validate our approaches through several human-subject studies that show high comprehension of the explanations as well as high overall satisfaction with the explanation process, thus providing some evidence for the effectiveness of our approaches in enhancing human-AI interaction.

By showing how logic can serve as a robust explainability layer that bridges the decision-making processes of AI systems and human understanding, this work aims to contribute to the development of AI systems that are not only powerful but also understandable, trustworthy, and above all, human-aware.

Chapter 1

Prolegomena

"There has to be a mathematical explanation for how bad that tie is."
— Russel Crowe (from A Beautiful Mind)

Artificial Intelligence (AI) has emerged as one of the most transformative technologies of the 21st century. It is reshaping industries, augmenting human capabilities, and addressing some of humanity's most pressing challenges—from enhancing healthcare diagnostics to enabling autonomous vehicles, improving financial forecasting, and personalizing recommendations. These AI systems are increasingly embedded in decision-making processes that significantly affect our daily lives. The global AI market reflects this growth, projected to reach $3.68 trillion by 2034, with a compound annual growth rate of 19.1% from 2024 to 2034.[1]

However, the rapid adoption of AI in critical domains raises a pressing question: How can we ensure that these systems are *transparent*, *understandable*, and *trustworthy*? This question is at the heart of Explainable AI (XAI), an interdisciplinary field dedicated to making the decision-making processes of AI systems understandable to humans [99].

XAI has seen significant progress, particularly in machine learning (ML). Researchers have developed methods to enhance the explainability of algorithms, ranging from intrinsically interpretable models to post-hoc explanation techniques for black-box models [98, 5]. Tools like LIME (Local Interpretable Model-agnostic Explanations) [191] and SHAP (SHapley Additive exPlanations) [161] provide local explanations for individual predictions, while novel algorithms aim to balance accuracy with interpretability [64, 92, 242]. Recent developments in counterfactual explanations have proven especially effective, answering "what-if" questions central to human understanding [232]. For example, in a loan application scenario, a

[1] https://www.precedenceresearch.com/artificial-intelligence-market

counterfactual explanation might state: "Your loan would have been approved if your annual income were $5000 higher," providing some actionable insights alongside transparency [130]. All these efforts underscore a growing recognition within the ML community of the critical role that explainability plays in the deployment of AI systems [9].

Parallel to advancements in ML, the automated planning community has adopted a focused approach to generating explanations for plans produced by AI planning systems, leading to the inception of explainable AI planning (XAIP) [81]. Predominantly, XAIP research focuses on identifying explanations for plans that, when conveyed to human users, help them understand and accept the AI system's proposed actions [126, 149]. This research direction recognizes that explanations in planning are not just about describing the sequence of actions, but also about justifying why certain actions were chosen over others, how the plan achieves the specified goals, and why alternative plans were not selected. A significant thread in XAIP research is the development of *contrastive* explanations, which align closely with human cognitive processes by answering questions like "Why not action A instead of action B?" [132, 68]. Another important direction involves explaining the unsolvability of planning problems [94, 212].

A notable framework within XAIP is the *model reconciliation problem* (MRP) [37], which emphasizes aligning the AI system's (mental) model with that of the human user. The MRP framework recognizes that explanations are not one-size-fits-all; they need to be tailored to the recipient's current understanding and knowledge [158]. For instance, an explanation of a Mars rover's plan might differ significantly when given to a mission specialist versus a member of the general public. MRP provides a formal framework for generating such (personalized) explanations, considering not just the AI system's model of the world, but also its best estimate of the human's mental model.

Despite these advances, generating effective explanations remain a challenge, especially as AI systems grow more complex and their decisions more impactful. To address this challenge, we must look beyond the technical aspects of explanation generation and consider the fundamental nature of explanation itself. This requires us to draw insights from philosophy, cognitive science, and psychology, which have long grappled with questions about the goodness of explanations, how humans process and understand explanations, and how explanations contribute to knowledge growth and acquisition. Understanding the nature of explanation can guide the development of AI systems that produce not only accurate but also meaningful and actionable explanations.

1.1 The Nature of Explanation: A Brief Overview

The very etymology of the word *explain* offers an illuminating starting point for understanding its fundamental purpose. Descending from the Latin word *explanō*, which means to lay something out flat or to make something plain/clear, the term underscores the essential aim of explanation: to make something intelligible. While this etymological perspective highlights the simplicity inherent in the act of explaining, a deeper understanding requires an interdisciplinary lens that draws on insights from philosophy and psychology.

1.1.1 Philosophical View

Philosophers of science have developed several influential models of explanation, each emphasizing different aspects of what it means to explain an *explanandum* (e.g., the phenomenon to be explained). The *Deductive-Nomological Model*, proposed by Hempel and Oppenheim [109], views explanations as logical arguments where the explanandum is a logical consequence of the explanans (the explaining sentences), i.e., the truth of its premises entails truth of its conclusion. Salmon [199] developed the *Statistical Relevance Model*, focusing on the statistical relevance of different factors to the occurrence of the event being explained. Causal Models, advocated by philosophers like Woodward [239], emphasize the importance of causal relationships in explanations. *Unification Models*, proposed by Friedman [82] and Kitcher [143], suggest that explanations work by showing how apparently disparate phenomena can be unified under a common framework. Pragmatic Theories, championed by philosophers like van Fraassen [223], argue that the adequacy of an explanation depends on the context and the specific interests of those seeking the explanation.

More recently, David Deutsch has contributed to this philosophical discourse with his emphasis on explanatory power and the concept of *hard-to-vary* explanations [60]. This view emphasizes that good explanations are those which cannot be easily modified without significantly altering their meaning or implications. Deutsch also stresses the importance of explanatory reach, arguing that good explanations often explain phenomena beyond their initial scope. His approach complements existing models by highlighting the role of creativity in developing explanations and positioning the pursuit of explanation as central to scientific progress.

These philosophical perspectives collectively highlight the multifaceted nature of explanation, emphasizing aspects such as logical structure, statistical relevance, causal relationships, unifying frameworks, contextual appropriateness, and robustness.

1.1.2 Psychological View

Psychology approaches explanation from the perspective of human cognition, focusing on how people process and internalize information. Research indicates that individuals often operate with incomplete and partial explanations [137], initially constructing explanations through fast, intuitive thinking, which are subsequently refined via slower, deliberative reflective processes [217, 125]. This cognitive approach aligns with the principle of cognitive economy, where explanations should optimize the balance between minimizing cognitive effort and maximizing cognitive effect [204]. The cognitive effect contributes positively to fulfilling the individual's cognitive functions and goals.

Effective explanations are not one-size-fits-all. Psychological research emphasizes that individuals may prefer or better understand different types of explanations based on their background, expertise, and cognitive style [158]. Effective explanations facilitate understanding and knowledge change, with individuals expanding their knowledge in real time upon comprehending an explanation [137]. This suggests that explanations should be tailored to the explainee's level of understanding, aligning with Ockham's Razor principle of maintaining simplicity while ensuring relevance and comprehensibility.

Psychologists and cognitive scientists have identified several forms of explanation that align with human cognitive processes [158]. *Causal explanations* derive the explanandum through deductive arguments. *Functional explanations* describe the purpose or function of a phenomenon. *Intentional explanations*, rooted in the Theory of Mind [185],[2] account for behavior in terms of beliefs, desires, and other mental states.

[2] The Theory of Mind describes an individual's ability to attribute mental states, beliefs, and knowledge to others [185].

Miller's survey [169] synthesizes insights from multiple disciplines, offering a theoretical framework for explainable AI systems that is sensitive to human explanation processes. He identifies four key characteristics of human explanations:

- *Contrastiveness*: Explanations elucidate the cause of an event relative to alternative events that did not occur, highlighting the importance of counterfactual reasoning in human understanding.
- *Selectivity*: Humans rarely provide a "complete" cause of an event. Instead, they focus on the most relevant or salient factors, demonstrating people's cognitive tendency to prioritize information.
- *Social aspect*: Explanations are tailored to the explainee's background and expected level of understanding, emphasizing the importance of explainee awareness in effective communication.
- *Preference for causality over probability*: Statistical explanations of events are generally unsatisfying unless accompanied by causal relationships, highlighting people's preference for narrative coherence over pure statistical data.

Philosophical models provide a rigorous foundation for understanding the logical and structural dimensions of explanation, while psychological research highlights the importance of tailoring explanations to human cognitive processes. Together, these perspectives can inform the design of AI systems that generate explanations that are not only accurate but also intuitive and actionable.

The challenge, then, lies in translating these rich, multifaceted insights into practical frameworks for explainable AI systems. What formal structures can capture the nuances of logical rigor, causal relevance, and social context while remaining computationally feasible?

1.2 Thesis Statement & Organization

The technical advancements in XAI and XAIP have made significant strides towards addressing explainability in AI systems, yet a crucial gap remains between theoretical insights

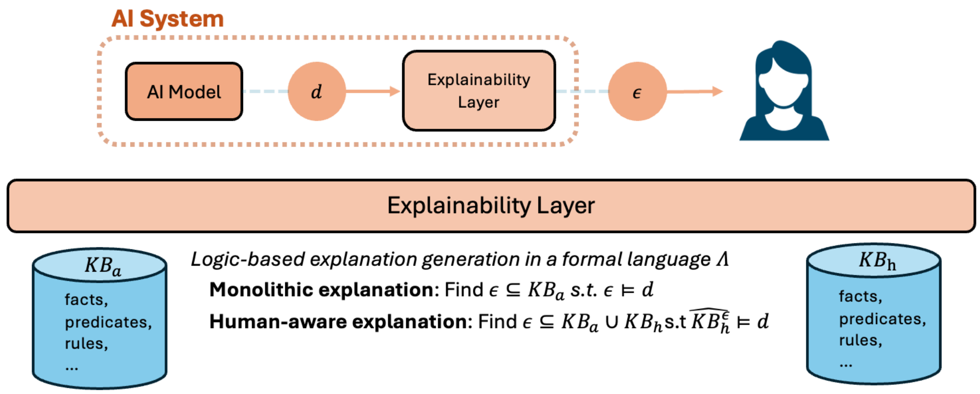


Figure 1.1: The Logic-based Explainability Layer. Given a decision d from an AI model, a *monolithic explanation* is generated with respect to the AI system's knowledge base (KB_a), while a *human-aware explanation* is generated with respect to the AI system's (KB_a) as well as the human user's knowledge base (KB_h), where $\widehat{KB_h^\epsilon}$ denotes the update of the user's knowledge base with the explanation.

and their practical implementation across various AI domains. This gap is particularly evident in the need for a unifying framework that can represent and reason about AI system decisions in a way that enables explanations that are not only correct but also aligned with human cognitive processes, adaptable to individual users, and applicable across diverse AI models and problem domains.

This thesis aims to bridge this gap by arguing that *logic-based frameworks can serve as an explanatory layer atop AI systems*. More specifically, we formulate our main thesis as follows:

> **Logic-based frameworks can serve as an explanatory representational layer for AI systems, enabling the generation of rigorous, flexible, and human-aware explanations across diverse problem domains by capturing the system's decisions in a formal logical language that supports inference and reasoning.**

This thesis proposes that formal logic provides a robust foundation for creating explanatory mechanisms by serving as an intermediate representational layer between AI systems and explanation generation. Importantly, we do not advocate replacing data-driven or other AI approaches with purely logical systems. Instead, we argue that by representing the decisions of AI systems (whether symbolic or sub-symbolic) in a logical framework, we can leverage the power of formal logic for generating explanations. As depicted in Figure 1.1, the proposed logic-based explainability layer operates by abstracting decisions from an AI system into a formal logical representation. This abstraction allows for reasoning and inference processes that are transparent, interpretable, and adaptable to the needs of different users. These features are vital to the design of AI systems that are explainable, and, importantly, *human-aware*.

We demonstrate the validity of this thesis through the development and evaluation of novel explanation generation techniques that are simultaneously rigorous, flexible, and generalizable. Our work spans multiple dimensions of explainability, from single-shot explanations to dialectical explanations, and from efficient explanation generation algorithms to effective explanation communication mediums. By maintaining a clear separation between the AI system's implementation and its logical representation for explanation purposes, our framework can be applied across diverse AI models and problem domains while preserving their underlying strengths.

A Note on Ethics

The deployment of explanation generation systems in the real world requires a careful examination of the underlying ethical principles that these systems adhere to. Ultimately, it is the developers' responsibility to be transparent about their system's morality, for example, by providing documentation describing the ethical requirements and limitations of the system. We refer the readers to the work by [213], where the authors elucidate an approach for developing ethically compliant autonomous systems based on a range of ethical frameworks. In the following, we discuss our moral motive pertaining to the construction of explanation generation systems, such as the one described in this thesis.

The moral compass of an explanation generation system should first and foremost be pointing toward truthfulness. By that, we ought to mean that the system should not allow for false explanations, nor obfuscate information, in order to attend to the user's satisfaction or persuasion. The explanations generated should be consistent with the ground truth.[3] Now, the ground truth should be carefully encoded into the AI system's model, and it is our

[3] What serves as the "ground truth" could differ among individuals, cultures, and societies, but here we take it to represent a single event, such as a fact (which we will resist the attempt to define right now).

responsibility (as developers) to ensure its accuracy to the best of our knowledge. In other words, such a system should not be able to produce lies.

Additionally, in later chapters we consider a framework for generating probabilistic explanations that, philosophically speaking, is built around a human user's subjective beliefs, which means that the matter of truthful explanations becomes increasingly more sensitive. For instance, a user's subjective beliefs could theoretically be used as a manipulation mechanism against the user, insofar as the system can craft an explanation consisting of said beliefs for the purpose of persuading the user to do, or rather believe, a certain event (e.g., exploiting confirmation bias). As one can think, applications of such systems and their misuses can be vast, especially in the current era of misinformation. Nevertheless, we proclaim that it is our duty to not allow for such actions–our framework takes advantage of a user's subjective beliefs only to generate probable explanations so long as these explanations align with the system's ground truth. That being said, such systems can omit information from the users, and as such, it is possible for the users to come to wrong conclusions on their own due to incorrect assumptions in their mental model. This is a trade-off and risk that all explanation generation systems face, and it is a phenomenon found in human-human interactions as well.

To sum up, we posit that explanation generation systems should have an intimate relationship with the pursuit of truthful explanations. This is crucial if we are to build ethically robust AI systems. In the spirit of philosopher Immanuel Kant: *lying is morally wrong and should be avoided.*[4]

1.2.1 Thesis Organization

This thesis contains content that has been previously presented in one JAIR journal article [231], one AAAI paper [225], one KR paper [224], two ECAI papers [227, 226], and one AAAI demonstration paper [229]. It also contains content from one conference paper [228] and one journal article [230] currently under review at the time of writing this thesis. The thesis is organized into chapters that progressively build the foundation, methodologies, and applications of the proposed logic-based explanatory framework:

[4] Obviously, there are cases where lying can be acceptable, such as when it is a matter of life preservation.

- **Chapter 2:** In this chapter, we provide some arguments supporting the choice of formal logic as a representational foundation for explanation generation in AI systems, emphasizing its role as an intermediate layer rather than a replacement for existing AI approaches.
- **Chapter 3 ([231])**: This chapter introduces a novel logic-based framework that extends the concept of model reconciliation to a broader class of planning problems, including classical and hybrid systems planning. This framework provides a foundation for generating logic-based, human-aware explanations in AI decision-making processes.
- **Chapter 4 ([230])**: Building upon the foundation laid in the previous chapter, this chapter extends our framework to handle probabilistic scenarios. We introduce concepts such as explanatory gain and explanatory power, addressing the challenges of generating explanations in environments characterized by uncertainty. This enables the generation of explanations in more realistic environments characterized by uncertainty.
- **Chapter 5 ([225, 230])**: This chapter presents novel algorithms for computing explanations, leveraging the duality between minimal correction sets (MCSes) and minimal unsatisfiable sets (MUSes). We demonstrate how these algorithms can be adapted to both deterministic and probabilistic settings.
- **Chapter 6 ([227])**: In this chapter, we develop an approach for generating personalized explanations by incorporating abstractions based on a user-specified vocabulary.
- **Chapter 7 ([224])**: This chapter presents the Dialectical Reconciliation via Structured Argumentative Dialogues (DR-Arg) framework, moving beyond single-shot explanations to facilitate dynamic, dialogue-based explanations between AI systems and human users.
- **Chapter 8 ([228])**: This chapter introduces a framework for explanation-guided belief revision that aligns more closely with natural human reasoning patterns. This contribution challenges the traditional assumption of minimal changes in belief revision theory, providing a more nuanced approach to updating beliefs in light of new information.
- **Chapter 9 ([226])**: Applying our developed frameworks to a practical domain, this chapter presents a logic-based approach to generating explanations for agent scheduling problems, addressing both reason-seeking and modification-seeking queries while considering privacy concerns.
- **Chapter 10 ([229])**: In this chapter, we present a demonstration system that combines parts of our logic-based explanation generation framework with large language models for generating natural language explanations in an academic course scheduling problem.

- **Chapter 11:** This chapter concludes the thesis, highlighting lessons learned and a vision for future human-aware AI systems.

Chapter 2

A Case for Formal Logic as the Foundation for Explainability in AI Systems

“ Μηδείς άγεωμέτρητος είσίτω μον την στέγην”
(Let no one ignorant of geometry enter here)
— Plato

At its core, explainability in AI systems is essentially a process of *reverse inference*. It involves backtracking through the system’s decision-making steps to support and explain the final decision. This capability is essential for understanding, validating, and trusting decisions, particularly in high-stakes domains like healthcare, law, and finance. In these contexts, the ability to explain decisions is not merely a desirable feature—it is a critical requirement. For any framework that aims to explain an AI decision, a minimum requirement would then be the ability to perform inference in reverse. Yet, not all AI approaches are equally suited to this task. This chapter explores the limitations of subsymbolic systems and makes the case for formal logic as a robust foundation for explainability in AI.

2.1 Subsymbolic Systems

Purely extensional, subsymbolic models, such as neural networks (NNs), face a significant challenge when it comes to explainability. These models, while undeniably powerful for data-intensive tasks like image recognition and natural language processing, struggle to provide meaningful explanations due to their lack of *structured semantics* and *compositionality*. To illustrate this issue, consider a simple scenario in a NN-based image classification system:

> "The input image was labeled `dog` because hidden neuron 404, which generally activates for pointy ears, had an activation of 0.44. If hidden neuron 404 had an activation of 0.8, the input image would have been labeled `fox`."

While this explanation describes the behavior of the model, it offers little in terms of semantic clarity, as it is not sufficient to simply state the activation value of a particular neuron. Even if we focus on a single neuron, there could be an infinite number of input combinations leading to that activation value. For example, if neuron 404 had only two inputs, x_1 and x_2, with weights w_1 and w_2, and used a ReLU activation function, then there are infinite combinations of x_1, x_2, w_1, and w_2 that could produce the output of 0.8. Without knowing which specific combination led to this activation, it is impossible to provide a meaningful explanation like "the image was classified as a fox because it had a certain type of ears." The hidden neuron's activation alone does not give us enough information to construct a clear, semantically meaningful explanation for the decision.

The root of this problem lies in the lack of systematicity and compositionality in neural networks.[5] NNs do not admit symbolic structures with productive syntax and corresponding semantics. The representations in NNs are distributed, correlative, and continuous numeric values that, on their own, do not correspond to anything interpretable. A hidden unit in a neural network does not, by itself, represent any conceptually meaningful object. Moreover, the composition of features in NNs is not invertible. Once vectors (e.g., tensors) are composed in a NN, their decomposition is undecidable. This irreversibility poses a significant barrier to explainability. In essence, no meaningful semantics can be captured from a neuron or collection of neurons, leading to a fundamental issue: no semantics without (invertible) compositionality, and no explainability without semantics.

It is worth noting that there have been attempts to improve the interpretability of subsymbolic models through counterfactual explanations [191, 161]. These methods, however, offer

[5] This criticism of NNs as a cognitive architecture has been raised over three decades ago by Fodor & Pylyshyn [78]. They showed why NNs cannot model systematicity, productivity and compositionality, all of which are needed for effective explainability.

post-hoc interpretations (in terms of inputs and outputs) rather than inherently explainable decision processes. They can highlight which inputs were important for a decision, but struggle to provide the kind of semantically meaningful, causal explanations that humans typically expect.

2.2 Logic-based Systems

In contrast to subsymbolic, neural-based approaches, symbolic, logic-based systems have several key features that make them particularly well-suited for explainability:

- **Structured Semantics**: Logic-based systems have well-defined compositional semantic functions that compute the meaning of a compound as a function of its constituents' meanings. This composition is invertible, allowing for the retrieval of components that produced an output. For instance, in a logical framework, we can understand how complex beliefs are formed from simpler ones, and crucially, we can decompose complex beliefs to understand their foundational elements. As we discussed above, this invertibility is fundamental to explanation–it allows us to backtrack from conclusions to premises, essentially performing inference in reverse.
- **Expressivity**: Logical languages can represent complex knowledge and reasoning processes in a form that can be traced and scrutinized. They allow for the encoding of rich, relational information about the world, including classes, hierarchies, and quantified statements. This expressivity extends to representing not just current states but also hypothetical scenarios and future possibilities. For example, in the situation calculus, we can reason about past events, current states, and potential future outcomes, providing a comprehensive framework for explaining an AI system's reasoning across time.
- **Scrutability**: Logic-based systems enable robust examination of their internal properties, both through internal verification techniques and external dialogues. This scrutability is crucial for building trust in AI systems. It allows for thorough validation of the reasoning process, ensuring that explanations are not just plausible but provably correct within the system's logical framework. Moreover, it facilitates interactive explanations, where users can query the system about its reasoning, potentially uncovering implicit assumptions or exploring alternative scenarios.

- **Augmentability**: Logic-based systems can be extended with new knowledge and support operations like composition, allowing AI systems to evolve their explanatory capabilities. This feature is particularly important for AI systems that need to adapt to new information and changing environments. The ability to seamlessly integrate new knowledge means that explanation frameworks can grow and refine over time, incorporating new concepts, rules, or even meta-level reasoning principles.
- **Abstraction**: Logic-based systems facilitate the description of decisions using high-level concepts, enhancing comprehensibility for users with varying expertise levels. This capability allows for explanations that can be tailored to the user's level of understanding, providing either high-level, conceptual explanations or detailed, step-by-step reasoning as needed. Abstraction is key to making complex AI decisions understandable to human users with varying expertise levels.
- **Meta-reasoning**: Meta-reasoning capabilities in logic-based systems allow for explanations that go beyond static justifications, providing insights into the system's learning and adaptation processes. For example, a logic-based AI system equipped with meta-reasoning could explain not just its conclusion, but also why it chose a particular inference strategy, how it resolved conflicts between different rules, or how it might revise its beliefs given new information. This could manifest in explanations like: "I concluded X because of rule Y, but I'm only 70% confident in this rule based on past performance. If we observe Z in the future, I would need to revise this belief."

Note that the rigidity, discreteness, and brittleness often attributed to logic are, in fact, a misunderstanding of its nature and potential. As Belle [8] succinctly describes: "*Logic provides a language for talking about the world and understanding what information is conveyed by expressions in that language; the language is indeed rigid and discrete, but its (possible) worlds certainly need not be.*" This distinction is key to understanding the flexibility and power of logical frameworks in modeling and explaining complex, uncertain, and dynamic real-world scenarios.

While pure neural network approaches face significant challenges in explainability, it is important to acknowledge the emergence of hybrid approaches that combine symbolic and subsymbolic methods [66, 83]. These *neuro-symbolic* AI systems aim to synergize the strengths of both paradigms: the robust learning capabilities and pattern recognition of neural networks with the interpretability and reasoning power of symbolic systems. However, the primary focus of this thesis is on leveraging formal, symbolic logic as the foundational framework for explanation generation in AI systems. We position the methods proposed in this thesis as complementary to the advancements in neuro-symbolic AI, rather than in competition with them. Our approach can be viewed as a specialized tool in the broader landscape of explainable AI, offering unique advantages in domains where rigorous, semantically meaningful explanations are paramount. Moreover, the principles and techniques developed here could potentially inform and enhance the explainability aspects of future neuro-symbolic systems.

It is important to reiterate that this thesis does not advocate for replacing state-of-the-art AI systems with purely logic-based ones. Instead, we propose using formal logic as a universal explainability layer—a framework capable of capturing and explaining the decisions of any AI system, regardless of its underlying architecture. This approach allows for seamless integration with a wide range of AI systems, including neural networks, statistical models, and other methodologies. By leveraging the strengths of formal logic—its expressivity, scrutability, and structured semantics—we aim to create an explainability layer that enhances human understanding of complex decision-making processes.

While the primary focus of this thesis is not on specific subsymbolic AI systems, it is useful to conceptualize how logic-based methods can be integrated with such systems to enhance explainability. Consider the following medical diagnosis scenario:

Example 1. *A hospital employs a sophisticated NN-based system for diagnosing skin conditions from images. Trained on a large dataset, this system swiftly classifies various skin conditions with high accuracy. In a particular case, the NN analyzes a patient's skin image and classifies it as "malignant melanoma" with 92% confidence. However, like many deep learning systems, its decision-making process is not inherently explainable.*

A typical counterfactual explanation for this scenario might be:

> If the mole were lighter in color and had a more regular shape, the image would have been classified as a benign mole instead of malignant melanoma.

While this counterfactual explanation offers some insight into the model's decision boundary, it lacks the rich contextual information and reasoning process that a logic-based approach can provide.

In contrast, integrating a logic-based layer of explainability into the system would yield a more comprehensive explanation. Such an integration could look like this:

1. *The NN classifies the image as malignant melanoma with 92% confidence.*
2. *The logic-based system maintains a knowledge base of medical facts, diagnostic criteria, and reasoning rules. For example:*

$$\forall x.\mathtt{DarkMole}(x) \land \mathtt{RapidGrowth}(x) \rightarrow \mathtt{HighRiskMelanoma}(x)$$
$$\forall x.\mathtt{HighRiskMelanoma}(x) \rightarrow \mathtt{RecommendBiopsy}(x)$$

3. *The neural network's output is translated into logical predicates. For instance, the system might assert:*

$$\textsc{ClassifiedAsMelanoma}(PatientImage123, 0.92)$$
$$\textsc{DetectedFeature}(PatientImage123, \mathtt{DarkMole})$$
$$\textsc{DetectedFeature}(PatientImage123, \mathtt{RapidGrowth})$$

4. *The logic-based framework then uses these assertions, combined with its knowledge base, to generate an explanation:*

> The system has classified this image as likely malignant melanoma (92% confidence) because it detected features consistent with high-risk melanoma, specifically a dark mole with signs of rapid growth. These features, when present together, are strong indicators of melanoma. Based on this classification, a biopsy is recommended for definitive diagnosis.

This explanation not only identifies the key features that led to the classification but also provides the reasoning behind why these features are significant, grounding the explanation in medical knowledge represented in the system.

This example demonstrates how a logic-based framework can act as an explanatory layer that complements, rather than replaces, existing AI systems. By translating subsymbolic outputs into a structured, semantically meaningful representation, logic-based systems enable detailed, causally grounded, and actionable explanations that align with human reasoning processes.

In conclusion, logic-based frameworks may offer a robust foundation for explainability in AI. By leveraging their structured semantics, precise reasoning, and adaptability, these frameworks bridge the gap between complex AI decision-making processes and human understanding. The following chapters will explore how formal logic can be operationalized to create effective, generalizable frameworks for explanation generation across diverse problem domains.

Chapter 3

A Logical Account for Explanations as Model Reconciliation

> *"You can prove anything you want by coldly logical reason—if you pick the proper postulates."*
>
> — Isaac Asimov

3.1 Introduction & Contribution

In the previous chapter, we established the potential of logical frameworks to serve as an explainability layer for explaining the decisions of AI system. We now demonstrate this potential in the domain of automated planning, namely Explainable AI Planning (XAIP).

XAIP has emerged as a crucial field in bridging the gap between theoretical planning algorithms and real-world applications. A key challenge in XAIP is the generation of explanations that account for the mental model of the human user, a concept known as the *Model Reconciliation Problem* (MRP) [37]. While existing approaches to MRP have primarily focused on automated planning techniques [34, 205, 207, 211], their applicability beyond classical planning has been unexplored.

This chapter introduces a logic-based framework for explanation generation that extends the applicability of MRP beyond classical planning scenarios. We term this approach *Logic-based Model Reconciliation Problem* (L-MRP). In particular, given a knowledge base KB_a (of an agent) and a knowledge base KB_h (of a human user), each encoding their knowledge of a planning problem, and that KB_a entails a query q (e.g., that a proposed plan of the agent is valid or that the proposed plan is optimal), the goal is to identify an explanation $\epsilon \subseteq \mathrm{KB}_a$ such that when it is used to update KB_h, then the updated KB_h also entails q. We then demonstrate that our approach can be applied not only to classical planning problems but also hybrid systems planning problems with durative actions, processes, and events.

The main contributions of this chapter are:

1. We formally define L-MRP, providing a logic-based foundation for XAIP that allows us to express planning problems and explanations in terms of logical knowledge bases.
2. We present algorithms for computing L-MRP explanations in both classical and hybrid systems planning problems, extending the applicability of model reconciliation.
3. We empirically evaluate our L-MRP approach against the current state of the art, demonstrating improved performance in certain scenarios and efficiency in handling hybrid systems planning problems.
4. We conduct a human-subject study using visualizations to explore effective methods of communicating L-MRP explanations to human users.

By representing planning problems and their solutions in formal logic through L-MRP, our framework not only improves scalability for certain types of classical planning problems but also provides a unified way to generate explanations across diverse planning domains. Importantly, this work demonstrates how a logical representational layer can enhance explainability while preserving the strengths of existing planning systems.

3.2 Explanations as Model Reconciliation

The *theory of mind* (ToM) [185] provides a fundamental framework for understanding how humans reason about others' mental states and behaviors in social and collaborative (or even adversarial) scenarios. Just as we discussed the importance of representing AI decisions in human-understandable terms in Chapter 2, ToM represents the ability to attribute mental models to others while recognizing that these models may differ from one's own. These mental models, which comprise mental states such as beliefs, knowledge, intentions, etc.

(in other words, a full range of goal and epistemic states), allow one to infer future mental states (i.e., the behavior) of others. However, social interactions can be quite convoluted, and misinterpretations may even yield frantic results.

Social interactions, however, can be quite convoluted, and misinterpretations may even yield frantic results. Nonetheless, being able to attribute mental models to other people, e.g., ideas about what other people are thinking or know about certain situations, would make social interactions placid and seamless, at least to some reasonable extent. For instance, building shared plans or goals between two people requires the very essence of ToM. Both parties must recognize the intentions of one another and subsequently work out how to mesh their actions with each other in order to achieve a common goal. However, note that in order to verbalize and intentionally communicate any differences in mental states (e.g., differences between actions), such as to provide *explanations* intending to update the receivers knowledge, it is normally assumed that the parties involved in the interaction share some common language and vocabulary (i.e., their mental models are expressed in common terms). ToM, therefore, is viewed as a vital socio-cognitive skill, inherent in the human nature, that we tend to highly use in an intuitive and natural way when interacting with other people. For a comprehensive description on the evolution and significance of ToM, we refer the interested reader to the work by Baron-

The *model reconciliation problem* (MRP) [37] has gained a lot of success due to the fact that it is rooted in the understanding of the importance of ToM. To be more precise, in the context of planning and MRP, a mental model consists simply of a PDDL expression that characterizes a planning problem (i.e., the model comprise all the fluents, predicates, objects, and actions that are allowed to be used in the particular problem). Important to note here are the assumptions that the agent possesses the human user's model a-priori,[6] the agent's model is correct and complete, and only the human user's model may contain flaws or missing information.[7]

In a typical MRP scenario, explanation generation is requested when a plan that is optimal (e.g., a shortest plan) in the agent's model is inexplicable (e.g., infeasible or suboptimal) in the human user's model, because the human user is, say, missing some preconditions from some actions in their model that are necessary for the optimal solution of the planning

[6] However, there has been some interest in relaxing this assumption [208].

[7] By correct and complete model, we mean that the agent believes that its model represents the objective and absolute truth about the specific planning problem.

problem. Then, the agent, by taking into account its own model as well as the human user's model, attempts to "reconcile" their differences by providing information from its own model (e.g., the missing preconditions) such that when this information is used by the human user to update their model (i.e., by adding the preconditions to the respective actions in their model), they can compute the optimal plan and, hence, understand its optimality.

As we can see, a key point to note in MRP is that the agent recognizes that the human user may have their own model of the planning problem, and that if there exists a discrepancy between their models such that the agent's plan is inexplicable to the human user, explanations will be couched in terms of model differences. Therefore, explanations as model reconciliation have the potential to play a significant role in explanation generation settings, mostly because of their natural consideration of how humans interact in social settings such as those that require intensionally communicating information with one another.

Unsurprisingly, researchers have empirically demonstrated explanations in the form of model reconciliation constitute a natural and effective way of explaining classical planning problems to human users [35, 240]. Specifically, they showed, using map visualizations of a planning problem, that human users not only understand explanations in the form of model reconciliation, but also believe that such explanations are necessary to explain (classical planning) plans. This empirical algorithm-agnostic assessment provides some supporting evidence for the real-world applicability of our proposed explanation generation framework for classical planning problems. Nevertheless, the applicability of explanations as model reconciliation for hybrid systems planning problems remains suspect, to the best of our knowledge. As such, in Section 3.6.2, we investigate, through a user study, to what extend explanations as model reconciliation are effective for hybrid systems planning problems.

A Logical Representation for Model Reconciliation

Having painted a small picture about the usefulness of explanations as model reconciliation, our interest in this work lies in extending and generalizing MRP. Building on our thesis of logic as an explainability layer, we are interested in laying the the theoretical and algorithmic foundations for a logic-based explanation generation framework. Succinctly, the mental models in our approach are in essence knowledge bases consisting of formulae expressed in some type of logic that fully describe a planning problem. For example, it is well known that a classical planning problem can be encoded as a propositional satisfiability instance (SAT) consisting of formulae that represent the initial state, goal state, and the action dynamics for n time steps, where n is an upper bound on the horizon of the problem, and is typically the length of the plan that can be found in the knowledge base [136]. In a similar fashion, a hybrid system planning problem can be expressed in first-order logic interpreted in the quantifier-free linear real arithmetic theory [30].

This logical representation of MRP provides several key advantages that we identified in Chapter 2, i.e., it provides a unified framework for representing and reasoning about different types of planning problems, it enables formal verification of explanations through logical inference, and it supports abstraction and adaptation of explanations to different user needs. Crucially, our approach does not replace existing planning systems but rather provides a representational layer for explaining their decisions. For example, a state-of-the-art search-based planner like FastDownward [108] can first find a plan, which we then represent in our logical framework for explanation generation. This integration preserves the efficiency of specialized planning systems while adding powerful explanation capabilities.

By using logic to represent planning problems and their solutions, we create a bridge between the computational efficiency of modern planning systems and the human need for clear, verifiable explanations. This aligns perfectly with our broader thesis about logic serving as an explanatory layer atop AI systems, demonstrating its practical application in planning domains.

3.3 Essential Background

3.3.1 Logic

A *logic* L is a tuple $\langle \mathrm{KB}_L, BS_L, ACC_L \rangle$, where KB_L is the set of well-formed knowledge bases (or theories) of L – each being a set of formulae. BS_L is the set of possible belief sets; each element of BS_L is a set of syntactic elements representing the beliefs L may adopt. $ACC_L : \mathrm{KB}_L \rightarrow 2^{BS_L}$ describes the *"semantics"* of L by assigning to each element of KB_L a set of acceptable sets of beliefs. For each $\mathrm{KB} \in \mathrm{KB}_L$ and $B \in ACC_L(\mathrm{KB})$, we say that B is a *model* of KB. A logic is monotonic if $\mathrm{KB} \subseteq \mathrm{KB}'$ implies $ACC_L(\mathrm{KB}') \subseteq ACC_L(\mathrm{KB})$.

Example 2. *Assume that L refers to the propositional logic over an alphabet P. Then,* KB_L *is the set of propositional theories over P, $BS_L = 2^P$, and ACC_L maps each theory* KB *into the set of its models in the usual sense.*

Definition 1 (Skeptical Entailment)**.** *A formula φ in the logic L is* skeptically entailed *by* KB*, denoted by* $\mathrm{KB}\models^s_L\varphi$*, if $ACC_L(\mathrm{KB}) \neq \emptyset$ and $\varphi \in B$ for every $B \in ACC_L(\mathrm{KB})$.*

Definition 2 (Credulous Entailment)**.** *A formula φ in the logic L is* credulously entailed *by* KB*, denoted by* $\mathrm{KB}\models^c_L\varphi$*, if $ACC_L(\mathrm{KB}) \neq \emptyset$ and $\varphi \in B$ for some $B \in ACC_L(\mathrm{KB})$.*

Definition 3 (Consistent Knowledge Base)**.** *A* KB *is* consistent *iff $ACC_L(\mathrm{KB}) \neq \emptyset$ or, equivalently, iff* KB *does not skeptically entail false.*

We will assume that a negation operator $\neg$ over formulae exists. Additionally, φ and $\neg\varphi$ are contradictory with each other in the sense that, for any KB and $B \in ACC_L(\mathrm{KB})$, if $\varphi \in B$, then $\neg\varphi \notin B$; and if $\neg\varphi \in B$, then $\varphi \notin B$. Therefore, if $\{\varphi, \neg\varphi\} \subseteq \mathrm{KB}$, then KB is inconsistent, i.e., $ACC_L(\mathrm{KB}) = \emptyset$. $\epsilon \subseteq \mathrm{KB}$ is called a *sub-theory* of KB. A theory KB *subsumes* a theory KB$'$, denoted by $\mathrm{KB} \lhd \mathrm{KB}'$, if $ACC_L(\mathrm{KB}) \subset ACC_L(\mathrm{KB}')$.

Boolean Satisfiability

Boolean Satisfiability (SAT) [42] is the problem of finding an assignment of truth values to variables in order to make a set of propositional formulae true. The problem can be stated as follows: Given a Boolean expression ψ with variables $V = \{v_1, \ldots, v_n\}$, find an assignment to the variables V that satisfies ψ or prove that one does not exist. For example,

$$\psi = (v_1 \vee v_2) \wedge (\neg v_2 \vee v_3) \wedge \neg v_1 \tag{3.1}$$

is satisfiable with respect to the truth assignment $M = \{v_1 = F, v_2 = T, v_3 = T\}$.

Satisfiability Modulo Theories

Satisfiability Modulo Theories (SMT) [7] is the problem of deciding the satisfiability of a first-order formula expressed in a given theory. The problem can be stated as follows: Given a first order formula ψ with variables $V = \{v_1, \ldots, v_n\}$ and a set of constraints over those variables, find an assignment to the variables V that satisfies ψ or prove that one does not exist. In contrast to the SAT problem, the variables are not restricted to Boolean values, but depend upon a theory, and the constraints are expressed with respect to a background logic. The theory and logic are critical elements of an SMT problem. Theories exist for Boolean propositions, bit-vectors, arrays, integers, reals, and so on. For example, an SMT problem in the quantifier-free linear real arithmetic theory is:

$$\psi = (v_1 + 3 \leq 2v_2) \vee (v_3 + 4 \geq 2) \vee (v_1 + v_2 + v_3 \geq 1), \tag{3.2}$$

which is satisfiable with respect to the assignment $M = \{v_1 = 1, v_2 = 1, v_3 = 1\}$.

3.3.2 Planning Problems

We now describe two general planning problem formulations, *classical planning problems* and *hybrid systems planning problems*.

Classical Planning

A classical planning problem, typically represented in PDDL [91], is a tuple $\Pi = \langle D, I, G \rangle$, which consists of the domain $D = \langle F, A \rangle$ – where F is a finite set of fluents representing the world states ($s \in F$) and A a set of actions – and the initial and goal states $I, G \subseteq F$. An action a is a tuple $\langle pre_a, \mathit{eff}_a \rangle$, where pre_a are the preconditions of a – conditions that must hold for the action to be applied; and $\mathit{eff}_a = \langle \mathit{eff}_a^+, \mathit{eff}_a^- \rangle$ are the addition (eff_a^+) and deletion (eff_a^-) effects of a – conditions that must hold after the action is applied. More formally, using $\delta_\Pi : 2^F \times A \rightarrow 2^F$ to denote the transition function of problem Π, if $s \not\models pre_a$, then $\delta_\Pi(s, a) \models \perp$; otherwise, $\delta_\Pi(s, a) \models s \cup \mathit{eff}_a^+ \setminus \mathit{eff}_a^-$. The solution to a planning problem Π is a plan $\pi = \langle a_1, \ldots, a_n \rangle$ such that $\delta_\Pi(I, \pi) \models G$, where $\delta_\Pi(s, \pi) = \delta_\Pi(\delta_\Pi(s, a_1), \pi')$ with $\pi' = \langle a_2, \ldots, a_n \rangle$. The cost of a plan π is given by $C(\pi, \Pi) = |\pi|$. Finally, the cost-minimal plan $\pi^* = \operatorname{argmin}_{\pi \in \{\pi' | \delta_\Pi(I, \pi') \models G\}} C(\pi, \Pi)$ is called the optimal plan.

Encoding Classical Planning Problems as Boolean Satisfiability: A classical planning problem can be encoded as a SAT problem [136, 135]. The basic idea is the following: Given a planning problem P, find a solution for P of length n by creating a propositional formula that represents the initial state, goal state, and the action dynamics for n time steps. This is referred to as the *bounded planning problem* (P, n), and we define the formula for (P, n) such that: *Any* model of the formula represents a solution to (P, n) and if (P, n) has a solution, then the formula is satisfiable.

We encode (P, n) as a formula Φ involving one variable for each action $a \in A$ at each timestep $t \in \{0, \ldots, n-1\}$ and one variable for each fluent $f \in F$ at each timestep $t \in \{0, \ldots, n\}$. We denote the variable representing action a in timestep t using subscript a_t, and similarly for facts. The formula Φ is constructed such that $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ is a solution for (P, n) if and only if Φ can be satisfied in a way that makes the fluents $a_0, a_1, \ldots, a_{n-1}$ true. The formula Φ is a conjunction of the following formulae:

- **Initial state:** Let F and I be the sets of fluents and initial states, respectively, in the planning problem:

$$\bigwedge_{f \in I} f_0 \wedge \bigwedge_{f \in F \setminus \{I\}} \neg f_0 \tag{3.3}$$

- **Goal state:** Let G be the set of goal states:

$$\bigwedge_{f \in G} f_n \tag{3.4}$$

- **Action scheme:** Formulae enforcing the preconditions and effects of each action a at time step t:

$$a_t \Rightarrow \bigwedge_{f \in pre_a} f_t \tag{3.5}$$

$$a_t \Rightarrow \bigwedge_{f \in \mathit{eff}_a^+} f_{t+1} \tag{3.6}$$

$$a_t \Rightarrow \bigwedge_{f \in \mathit{eff}_a^-} \neg f_{t+1} \tag{3.7}$$

- **Explanatory frame axioms:** Formulae enforcing that facts do not change between subsequent time steps t and $t + 1$ unless they are effects of actions that are executed at time step t:

$$\neg f_t \wedge f_{t+1} \Rightarrow \bigvee \{a_t \mid f \in \mathit{eff}_a^+\} \tag{3.8}$$

$$f_t \wedge \neg f_{t+1} \Rightarrow \bigvee \{a_t \mid f \in \mathit{eff}_a^-\} \tag{3.9}$$

- **Action exclusion axioms:** Formulae enforcing that only one action can occur at each time step t:

$$\bigwedge_{a \in A} \bigwedge_{a' \in A | a \neq a'} (\neg a_t \vee \neg a'_t) \tag{3.10}$$

  where A is the set of actions in the planning problem.

Finally, we can *extract* a plan by finding an assignment of truth values that satisfies Φ (i.e., for all time steps $t = 0, \ldots, n-1$, there will be exactly one action a such that $a_t = \mathit{True}$). This could be easily done by using a satisfiability algorithm, such as the well-known DPLL algorithm [52].

It is worth mentioning that planning as SAT has gathered a lot of traction, as there is a significant number of works which have been devoted to formalizing and improving the encodings of planning problems using propositional logic [193, 63, 29].

Hybrid Systems Planning

A hybrid system planning problem, hereinafter simply *hybrid planning*, typically represented in PDDL+ [80], is a tuple $\Pi+ = \langle P, V, A, Ps, E, I, G \rangle$, in which P is a set of propositions; V is a vector of real variables (fluents); A is a set of durative and instantaneous actions; Ps is a set of processes; E is a set of events; and I and G are the initial and goal states, respectively. A durative action $a \in A$ is defined by a tuple $\langle \mathit{pre}_a, \mathit{eff}_a, \mathit{dur}_a \rangle$, where pre_a is the precondition, eff_a is the effect, and dur_a is a duration constraint – a conjunction of numeric constraints corresponding to the duration of action a.

In contrast to classical planning, the precondition $\mathit{pre}_a = \langle \mathit{pre}_{\vdash a}, \mathit{pre}_{\leftrightarrow a}, \mathit{pre}_{\dashv a} \rangle$ of a durative action a consists of three disjoint subsets, where each subset represents conditions that must hold at the start of the action, throughout its execution, and at the end of the action, respectively. In turn, the effect $\mathit{eff}_a = \langle \mathit{eff}^{\pm}_{\vdash a}, \mathit{eff}^{num}_{\vdash a}, \mathit{eff}^{\pm}_{\dashv a}, \mathit{eff}^{num}_{\dashv a}, \mathit{eff}_{\leftrightarrow a} \rangle$ of an action a consists of five disjoint subsets, where $\mathit{eff}^{\pm}_{\vdash a}$ is the set of instantaneous effects of adding/removing propositions at the start of the action, $\mathit{eff}^{num}_{\vdash a}$ is the set of instantaneous numeric effects at the start of the action, $\mathit{eff}^{\pm}_{\dashv a}$ is the set of instantaneous effects of adding/removing propositions at the end of the action, $\mathit{eff}^{num}_{\dashv a}$ is the set of instantaneous numeric effects at the end of the action, and $\mathit{eff}_{\leftrightarrow a}$ is a conjunction of numeric effects which are applied continuously while the action is executing. Note that the values of instantaneous effects can be exploited to support other actions only after a small amount of time ϵ, which is referred to as epsilon separation [79].

Each process $ps \in PS$, defined by a tuple $\langle pre_{ps}, \mathit{eff}_{ps} \rangle$, is similar to a durative action, except that it does not have a set duration but is instead active when their preconditions are satisfied (without any epsilon separation) and inactive when their preconditions are not satisfied. Consequently, unlike durative actions, processes do not have durative constraints. In addition, a process's precondition consists of a single condition, whereas its effect consists of a single continuous numeric effect. Each event $e \in E$, defined by a tuple $\langle pre_e, \mathit{eff}_e \rangle$, is analogous to an instantaneous action and, thus, also does not have a duration constraint. Further, an event comprises of a single triggering precondition and an instantaneous effect. Note that events can make immediate use of effects (without any epsilon separation). If the effect of an action, process, or other event make true the condition of an event, then it occurs immediately and simultaneously with that effect. In general, processes and events are used to model exogenous events in the world [90]. Therefore, they are not under the direct control of the planner and are triggered immediately when their preconditions are satisfied (see Bogomolov *et al.* [19] for more details).

Moreover, the cost of a PDDL+ plan depends upon a specified plan metric. Plan metrics assert, for the benefit of the planner, how a plan will be evaluated for a particular problem.[8] For instance, the same initial and goal states might yield entirely different optimal plans given different plan metrics. Examples of plan metrics include the makespan of the plan (i.e., the sum of the duration of each action in the plan), optimizing a specific quantity in the domain, etc.

[8] Metrics are specified in the problem description, allowing a planner to easily explore the effect of different metrics in the construction of solutions to problems for the same domain.

Finally, it is important to mention that, to the best of our knowledge, there does not exist any general optimal PDDL+ planners. As a matter of fact, it has been shown that even finding the existence of PDDL+ plans is undecidable [107]. The reason is that PDDL+ domains have an established relationship with the *reachability* problem for hybrid automata [80], where it is known to be undecidable [110]. In the context of planning, the reachability problem corresponds to the problem of solving plan existence. Nonetheless, there exist PDDL+ fragments where plan existence is decidable [80].

Encoding Hybrid Planning Problems as Satisfiability Modulo Theory: A hybrid planning problem $\Pi+$ can be encoded as an SMT formula [30] with bound n in the theory of quantifier-free (non-linear) real arithmetic with n copies of the set of variables x, where x is called a *happening*. A happening encodes the change in the state at a particular time point due to effects of actions, processes, or events happening at that time point:

$$x = \langle t, \hat{E}, \hat{Ps}, \hat{P}, \hat{V}, A, P^+, V^+, flow_V, dur_{Ps} \rangle \tag{3.11}$$

where t is the current time point, $\hat{E} = \{E_0, \ldots, E_B\}$ is the chain of events triggered at t, B is a bound on the length of the causal chain of events at each time point, $\hat{Ps} = \{Ps_0, \ldots, Ps_B\}$ is the chain of active processes at t, $\hat{P} = \{P_0, \ldots, P_B\}$ is the causal change in the propositional state variables at t, $\hat{V} = \{V_0, \ldots, V_B\}$ is the causal change in the real state variables at t, A is a set of durative and instantaneous actions, P^+ and V^+ are the values of the propositional and real state variables, respectively, at time $t + \epsilon$, $flow_V = \{flow_v \mid v \in V\}$ is a numerical expression that represents the change in value of v from a time point to the next, and $dur_{Ps} = \{dur_{ps} \mid ps \in Ps\}$ is the remaining duration of each process ps. Note that the dur_{Ps} variable enforces the duration constraint of a durative action, not that of a process, as we highlight in the next paragraph.

Following the same manner as in the encoding of classical planning, the SMT formula is comprised of a conjunction of formulae that represent the dynamics of the given $\Pi+$ problem. Then, a plan for $\Pi+$ with length n would correspond to the action variables with true assignments in any proof of the SMT formula of $\Pi+$. In order to encode a durative action in SMT, we split it into two instantaneous actions representing the start and end of the action respectively, and a process representing the action's durative portion. The start and end actions are constructed in a straightforward way, with the addition of a new effect, whose only purpose is to activate the process. The effect of the process is the continuous effect of the durative action, plus a continuous decrement of some timer variable. The instant end action uses that timer variable and the durative action's duration inequality in its precondition. This ensures that the start and end of the action are the correct distance apart in the timeline. It is only this kind of process, representing a durative action, that has a duration associated with it.

Below, we describe the SMT formulae that characterize a $\Pi+$ problem (for a more thorough description, we refer the reader to Cashmore *et al.* [31]). Formulae (12) to (25) encode the constraints for each happening $x_0, \ldots, x_n$, and formulae (26) to (36) are additional constraints needed in the SMT formula $\Pi+$:

- **Proposition and real variable support:** Formulae ensuring that the values of propositions and real variables remain consistent from $P_0 \cup V_0$ to $P_B \cup V_B$:

$$\bigwedge_{i=0}^{B-1} \bigwedge_{p \in P} p_{i+1} \rightarrow (p_i \bigvee_{e|p \in \mathit{eff}_e^+} e_i) \tag{3.12}$$

$$\bigwedge_{i=0}^{B-1} \bigwedge_{p \in P} \neg p_{i+1} \rightarrow (\neg p_i \bigvee_{e|p \in \mathit{eff}_e^-} e_i) \tag{3.13}$$

$$\bigwedge_{i=0}^{B-1} \bigwedge_{v \in V} (\bigwedge_{e|v \in \mathit{eff}_e^{num}} \neg v_i) \rightarrow (v_{i+1} = v_i) \tag{3.14}$$

- **Event preconditions and effects:** Formulae enforcing that an event is triggered if and only if its preconditions hold, and that if an event is triggered, its effects are present in the next time step:

$$\bigwedge_{i=0}^{B-1} \bigwedge_{e \in E} e_i \rightarrow (pre_e)_i \tag{3.15}$$

$$\bigwedge_{i=0}^{B-1} \bigwedge_{e \in E} e_i \rightarrow (\mathit{eff}_e)_{i+1} \tag{3.16}$$

- **Action preconditions and effects:** Formulae enforcing that an action's preconditions must hold in $P_B \cup V_B$ and their effects are enforced in $P^+ \cup V^+$:

$$\bigwedge_{a \in A} a \rightarrow (pre_a)_B \tag{3.17}$$

$$\bigwedge_{a \in A} a \rightarrow (\mathit{eff}_a)^+ \tag{3.18}$$

- **Support across epsilon separation:** Formulae ensuring that the values of propositions and real variables remain consistent from $P_B \cup V_B$ to $P^+ \cup V^+$:

$$\bigwedge_{p \in P} p^+ \rightarrow (p_B \bigvee_{a | p \in \mathit{eff}_a^+} a) \tag{3.19}$$

$$\bigwedge_{p \in P} \neg p^+ \rightarrow (\neg p_B \bigvee_{a | p \in \mathit{eff}_a^-} a) \tag{3.20}$$

$$\bigwedge_{v \in V} (\bigwedge_{a | v \in \mathit{eff}_a^{num}} \neg a) \rightarrow (v^+ = v_B) \tag{3.21}$$

- **Process triggering:** Formulae enforcing that a process is active if and only if its preconditions are satisfied in each set $P_0 \cup V_0$ to $P_B \cup V_B$, and ensuring that a process cannot finish outside of a happening:

$$\bigwedge_{i=0}^{B} \bigwedge_{ps \in Ps} ps_i \leftrightarrow (pre_{ps})_i \tag{3.22}$$

$$\bigwedge_{ps \in Ps} dur_{ps} \geq 0 \tag{3.23}$$

$$\bigwedge_{ps \in Ps} ps_B \leftrightarrow (dur_{ps} > 0) \tag{3.24}$$

- **Action exclusion axioms:** Formulae enforcing that only one action can occur at each time step:

$$\bigwedge_{a \in A} \bigwedge_{a' \in A | a \neq a'} (\neg a \vee \neg a') \tag{3.25}$$

- **Instance description:** Formulae enforcing that the initial state holds in the first happening and the goal holds in the final happening:

$$I_0 \tag{3.26}$$

$$G_n \tag{3.27}$$

$$t_0 = 0 \tag{3.28}$$

$$\bigwedge_{i=1}^{n} t_i \geq t_{i-1} + \epsilon \tag{3.29}$$

- **Proposition support:** Formulae ensuring that the discrete state variables P do not change between happenings:

$$\bigwedge_{i=1}^{n} \bigwedge_{p \in P} (p_0)_i \rightarrow (p^+)_{i-1} \tag{3.30}$$

$$\bigwedge_{i=1}^{n} \bigwedge_{p \in P} \neg(p_0)_i \rightarrow \neg(p^+)_{i-1} \tag{3.31}$$

- **Invariants:** Formulae ensuring that the continuous numeric change between happenings is valid:

$$\bigwedge_{i=1}^{n} \bigwedge_{ps \in Ps} (ps_B)_{i-1} \rightarrow ((dur_{ps})_i) = (dur_{ps})_{i-1} + t_i - t_{i+1}) \tag{3.32}$$

$$\bigwedge_{i=0}^{n-1} \bigwedge_{ps \in Ps} (ps_B)_i \leftrightarrow (pre_{\leftrightarrow ps})_i \tag{3.33}$$

$$\bigwedge_{i=0}^{n-1} \bigwedge_{e \in E} \neg(pre_{\leftrightarrow e})_i \tag{3.34}$$

- **Continuous change on real variables:** Formulae enforcing the continuous change on real variables:

$$\bigwedge_{i=0}^{n-1} \bigwedge_{v \in V} (flow_v)_i = \int_{t_i}^{t_{i+1}} \bigcup_{ps \in Ps} (eff^{num}_{\leftrightarrow ps}[v])_i \; dt \tag{3.35}$$

$$\bigwedge_{i=1}^{n} \bigwedge_{v \in V} ((v_0)_i = (v^+)_{i-1} + (flow_v)_{i-1}) \tag{3.36}$$

As mentioned earlier, there does not exist any optimal PDDL+ planners. Consequently, SMT solvers fall within this category as well. For example, in a temporal setting, a lower number of happenings does not mean a higher-quality plan. It could be that by adding happenings, a plan of shorter duration, or with better cost, can be found. As an example, consider a domain with a car, actions to increase and decrease acceleration by one step, and a goal to move the car a given distance. The optimal plan in terms of duration will be to accelerate as many times as possible until the half-way point, and then to decelerate until the car stops at the specified distance. A plan with the fewest steps/happening only accelerates and decelerates once each, and takes much longer. Therefore, a solution found by an SMT solver is not guaranteed to be optimal with respect to time.

3.3.3 The Model Reconciliation Problem

The *Model Reconciliation Problem* (MRP), as introduced by Chakraborti *et al.* [37], highlights the critical need for aligning the planning models of a human user and an agent to facilitate effective collaboration and understanding. This alignment becomes especially pertinent in scenarios where the agent's plan deviates from human expectations, necessitating a mechanism to reconcile these differences through explanations. In this approach, the (planning) agent must have knowledge of the human's model in order to contemplate their goals and foresee how its plan will be perceived by them. When there exist differences between the models of the agent and the human such that the agent's plan diverges from the human's expectations, the agent provides a minimal set of model differences, namely an *explanation*, to the human.

More formally, MRP is defined by the tuple $\Psi = \langle \Phi, \pi \rangle$, where $\Phi = \langle M^R, M^R_H \rangle$ is a tuple of the agent's model $M^R = \langle D^R, I^R, G^R \rangle$ and the agent's approximation of the human's model $M^R_H = \langle D^R_H, I^R_H, G^R_H \rangle$, and π is the optimal plan in M^R. A solution to an MRP is an explanation ϵ such that when it is used to update the human's model M^R_H to $\widehat{M}^{R,\epsilon}_H$, the plan π is optimal in both the agent's model M^R and the updated human model $\widehat{M}^{R,\epsilon}_H$. The goal is to find a cost-minimal explanation, where the cost of an explanation is defined as the length of the explanation [37].

It is important to highlight that, in order to effectively solve MRP, the following (implicit) assumptions typically hold:

1. The *agent model represents the ground truth* or, in other words, the agent model is the "correct" encoding of the domain. This assumption is predicated on the notion that the explanation is generated from the agent's perspective, thereby rendering it reasonable to assume that the agent "thinks" that its model is accurate or correct.

2. The *agent has access to the human model*, which is an approximation of the actual human model. In the worst case, it can be empty; but, practically, it can be approximated based on past interactions [205, 124].
3. *Both models are assumed to be deterministic*, and they thus are able to represent only deterministic domains.

3.4 Logic-based Model Reconciliation Problem

We introduce the concept of *logic-based model reconciliation problem* (L-MRP) in the following setting, where, for brevity, we use the term $\models^x_L$ for $x \in \{s, c\}$ to refer to skeptical (s) or credulous (c) entailment:

> **Logic-based Model Reconciliation Problem:** Given two knowledge bases KB_a and KB_h and a formula φ in a logic L, where $\mathrm{KB}_a \models^x_L \varphi$ and $\mathrm{KB}_h \not\models^x_L \varphi$, the goal is to identify an explanation (i.e., a set of formulae) $\epsilon \subseteq \mathrm{KB}_a$ such that when it is used to *update* KB_h to $\widehat{\mathrm{KB}}^\epsilon_h$, the updated $\widehat{\mathrm{KB}}^\epsilon_h \models^x_L \varphi$.

When updating a knowledge base KB with an explanation ϵ, the updated knowledge base $\mathrm{KB} \cup \epsilon$ may be inconsistent as there may be contradictory formulae in KB and ϵ. As such, to make the knowledge base consistent again, one needs to remove this set of contradictory formulae $\gamma \subseteq \mathrm{KB}$ from KB. More formally:

Definition 4 (Knowledge Base Update)**.** *Given a knowledge base* KB *and an explanation* ϵ*, the updated knowledge base is* $\widehat{\mathrm{KB}}^\epsilon = \mathrm{KB} \cup \epsilon \setminus \gamma$*, where* $\gamma \subseteq \mathrm{KB} \setminus \epsilon$ *is a set of formulae that must be removed from* KB *such that the updated* $\widehat{\mathrm{KB}}^\epsilon$ *is consistent.*[9]

We now define the notion of a *support* of a formula w.r.t. a knowledge base KB before defining the notion of *explanations*.

Definition 5 (Support)**.** *Given a knowledge base* KB *and a formula* φ *in a logic* L*, where* $\mathrm{KB} \models^x_L \varphi$*,* $\epsilon \subseteq \mathrm{KB}$ *is a* support *of* φ *w.r.t.* KB *if* $\epsilon \models^x_L \varphi$*. Assume that* ϵ *is a support of* φ

[9] Intuitively, one should prefer the set of formula γ that is removed to be as small as possible, though we chose to not require such a restriction here.

w.r.t. KB. *We say that* $\epsilon \subseteq$ KB *is a* $\subseteq$-minimal support *of* φ *if no proper sub-theory of* ϵ *is a support of* φ*. Furthermore,* ϵ *is a* $\lhd$-general support *of* φ *if there is no support* ϵ' *of* φ *w.r.t.* KB *such that* ϵ *subsumes* ϵ'*.*

Definition 6 (Explanation)**.** *Given two knowledge bases* KB_a *and* KB_h *and a formula* φ *in a logic* L*, where* $\mathrm{KB}_a \models^x_L \varphi$ *and* $\mathrm{KB}_h \not\models^x_L \varphi$*, an* explanation for φ from KB_a for KB_h *is a support* ϵ *w.r.t.* KB_a *for* φ *such that the updated knowledge base* $\widehat{\mathrm{KB}}^{\epsilon}_h \models^x_L \varphi$*, where* $\widehat{\mathrm{KB}}^{\epsilon}_h$ *is updated according to Definition 4.*

Example 3. *Consider propositional logic theories over the set of propositions* $\{a, b, c\}$ *with the usual definition of models, satisfaction, etc. Assume* $\mathrm{KB}_a = \{a, b, a \rightarrow c, a \wedge b \rightarrow c\}$ *and* $KB_{h_1} = \{a\}$*. We have that* $\epsilon_1 = \{a, a \rightarrow c\}$ *and* $\epsilon_2 = \{a, b, a \wedge b \rightarrow c\}$ *are two* $\subseteq$*-minimal supports of* c *w.r.t.* KB_a*. Only* ϵ_1 *is a* $\lhd$*-general support of* c *w.r.t.* KB_a *since* $\epsilon_2 \lhd \epsilon_1$*. Both* ϵ_1 *and* ϵ_2 *can serve as explanations for* c *from* KB_a *for* KB_{h_1}*. Of course,* KB_a *is itself an explanation for* c *from* KB_a *for* KB_{h_1}*.*

Now consider $KB_{h_2} = \{a, \neg b\}$*. In this case, both* ϵ_1 *and* ϵ_2 *are possible explanations for* c *from* KB_a *for* KB_{h_2}*, but if* ϵ_2 *is chosen, then* $\neg b$ *will need to be removed from* KB_{h_2} *so that it is consistent according to Definition 4.*

3.4.1 Preferred Explanations

When considering explanatory systems, a natural question that potentially arises would be: *Are all explanations equal*? For example, one would want to differentiate between *trivial* and *non-trivial* explanations. While it might be acceptable in some cases, trivial explanations,[10] which are akin to a parent providing the explanation "because I said so" when asked "why?" by their child, are not preferred in most cases.

Besides computing an explanation ϵ, the agent also needs to present that explanation to the user or, in other words, describe the content of the explanation ϵ to the user. Given knowledge bases KB_a and KB_h and a formula φ, there might be several explanations for φ from KB_a for KB_h. Therefore, an agent might prefer an explanation that requires the least amount of effort[11] in presenting explanation ϵ to the human. One way to characterize the

[10] There might be cases where we need to explain an assumption or a fact that is missing from a KB, and therefore, trivial explanations will be succinct and acceptable.

[11] By "effort," we could use either the effort needed by the robot to present the explanation, the effort needed by the human to understand the explanation, or a combination of both. For example, the length of

effort of the agent when presenting an explanation is to associate a cost to the elements of explanation ϵ. For example, one might prefer a subset-minimal explanation or a shortest length explanation over others. Next, we quantify the cost of an explanation, which is then used in to define a general preference relation over explanations.

We assume a cost function $\mathcal{C}_L$ that maps knowledge bases and sets of explanations to non-negative real values:

$$\mathcal{C}_L : KB_L \times \Omega \to \mathcal{R}^{\geq 0} \tag{3.37}$$

where Ω is the set of explanations and $\mathcal{R}^{\geq 0}$ denotes the set of non-negative real numbers. Intuitively, this function can be used to characterize different complexity measurements of an explanation. A cost function $\mathcal{C}_L$ is *monotonic* if for any two explanations $\epsilon_1 \subseteq \epsilon_2$, $\mathcal{C}_L(\mathrm{KB}, \epsilon_1) \leq \mathcal{C}_L(\mathrm{KB}, \epsilon_2)$. $\mathcal{C}_L$ induces a preference relation $\prec_{\mathrm{KB}}$ over explanations as follows.

Definition 7 (Preferred Explanation)**.** *Given a cost function $\mathcal{C}_L$, a knowledge base KB_h, and two explanations ϵ_1 and ϵ_2 for KB_h, explanation ϵ_1 is* preferred *over explanation ϵ_2 w.r.t. KB_h (denoted by $\epsilon_1 \preceq_{\mathrm{KB}_h} \epsilon_2$) iff*

$$\mathcal{C}_L(\mathrm{KB}_h, \epsilon_1) \leq \mathcal{C}_L(\mathrm{KB}_h, \epsilon_2) \tag{3.38}$$

and ϵ_1 is strictly preferred *over ϵ_2 w.r.t. KB_h (denoted by $\epsilon_1 \prec_{\mathrm{KB}_h} \epsilon_2$) if*

$$\mathcal{C}_L(\mathrm{KB}_h, \epsilon_1) < \mathcal{C}_L(\mathrm{KB}_h, \epsilon_2) \tag{3.39}$$

This allows us to compare explanations as follows.

Definition 8 (Most Preferred Explanation)**.** *Given a cost function $\mathcal{C}_L$ and a knowledge base KB_h, an explanation ϵ is a* most preferred explanation *w.r.t. KB_h if there exists no other explanation ϵ' such that $\epsilon' \prec_{\mathrm{KB}_h} \epsilon$.*

There are several natural monotonic cost functions. For example:

- $C_L^1(\mathrm{KB}_h, \epsilon) = |\epsilon|$, the cardinality of ϵ, indicates the number of formulae that need to be explained;

the explanation can be used to represent both the effort needed by the robot to explain as well as the effort needed by the human to understand.

- $C_L^2(\mathrm{KB}_h, \epsilon) = |\epsilon \setminus \mathrm{KB}_h|$, the cardinality of $\epsilon \setminus \mathrm{KB}_h$, indicates the number of *new* formulae that need to be explained;
- $C_L^3(\mathrm{KB}_h, \epsilon) = length(\epsilon)$ indicates the number of literals in ϵ that need to be explained.

3.4.2 Explanations in Planning Problems

As discussed in Section 3.3.2, classical and hybrid planning problems can be encoded using the logic-based SAT and SMT problems, respectively. As such, our logic-based notions of explanations proposed in the previous section can be applied to explainable planning, particularly the model reconciliation problem, in the context of explaining classical and hybrid planning problems. Nonetheless, recall that a model reconciliation problem is strictly defined for explaining optimal plans (see Section 3.3.3). We can, however, relax this definition and generalize it for arbitrary, valid plans. The reasoning behind this relaxation is that, even if optimality cannot be guaranteed, the user may have doubts about the validity of a plan (i.e., whether the plan is sound and can be executed to achieve the goal). Therefore, valid plan explanations are crucial for engendering trust in the user. Note that, in the case of hybrid planning problems, we restrict ourselves to only explaining valid plans, as guaranteeing optimality is often infeasible for such tasks.

Hereinafter, we focus on the following two problems: (1) Explaining the *validity* of a plan to the user, and (2) Explaining the *optimality* of a plan to the user, where we define them using logical notations. From now on, we use KB_a and KB_h to denote the knowledge bases encoding the planning problem of the planning agent and the human user, respectively.

Plan Validity

Assume π is a valid plan with respect to KB_a but not KB_h. In other words, it is not possible to execute π to achieve the goal with respect to KB_h. For example, an action in the plan cannot be executed because its precondition is not satisfied, an action in the plan does not exist, or the goal is not reached after the last action in the plan is executed. From the perspective of logic, a plan is valid if there exists at least one model in KB_h in which the plan can be executed and the goal is reached:

Definition 9 (Plan Validity). *Given a planning problem* Π*, a plan* π *of* Π*, where* α_t *is an action of the plan at time step* t*, and a knowledge base* KB_h *encoding* Π*,* π *is a* valid *plan in* KB_h *if* $\mathrm{KB}_h \models^c_L \pi \wedge g_n$*, where* g_n *is the fact corresponding to the goal of the planning problem at time step* n*.*

Plan Optimality

Assume that π^* is an optimal plan in a model of KB_a. To explain the optimality of π^* to KB_h, we need to prove that no shorter (optimal) plan exists in KB_h. Thus, we need to prove that no shorter plan exists in *all* models of KB_h. This can be easily done by using the notion of skeptical entailment.

Definition 10 (Plan Optimality). *Given a planning problem* Π*, a plan* π *of* Π *with length* n*, and a knowledge base* KB_h *encoding* Π*, the plan* π *is* optimal *in* KB_h *if and only if* $\mathrm{KB}_h \models^c_L \pi \wedge g_n$ and $\mathrm{KB}_h \models^s_L \phi$*, where* $\phi = \bigwedge_{t=0}^{n-1} \neg g_t$ *and* g_t *is the fact corresponding to the goal of the planning problem at time step* t*.*

In essence, the query ϕ in the above definition is that no plan of lengths 1 to $n-1$ exists. Therefore, when combined with the fact that a plan π of length n that achieves the goal state exists, then that plan must be an optimal plan.

Note that the above definition applies only to classical planning problems and not hybrid planning problems. The reason is because the cost of a hybrid plan depends on a user-specified plan metric, and this cost is not explicitly encoded by SMT encodings of hybrid plans. Nonetheless, we do not view this as a significant loss since finding optimal hybrid plans is often highly intractable.

Illustrative Example

For the purpose of clarity and ease of understanding, we have constructed a simplified, classical planning version of the Generator domain[12] as our working example, and use this to demonstrate different concepts explained throughout this chapter. We intentionally kept

[12]The domain is originally defined for hybrid planning problems and it can be found here: https://github.com/KCL-Planning/SMTPlan/tree/master/benchmarks

the example very simple so that the explanations are succinct and are easy to present and understand. We refer the interested reader to the Appendix in [231] where we present an example on the hybrid planning version of the Generator domain.

The domain and problem files are shown in Listings 3.1 and 3.2, respectively. The domain consists of two actions gen_on and gen_off. Action gen_on consists of one precondition $(= fuel\_full)$, one addition effect $(= gen\_running)$, and one deletion effect $(= \neg fuel\_full)$. Action gen_off has one precondition $(= gen\_running)$ and one addition effect $(= gen\_ran)$. In this particular problem, the initial state asserts that the fuel is full $(= fuel\_full)$ and the goal state is that the generator has been ran $(= gen\_ran)$. The optimal plan for this problem is thus to first execute action gen_on such that $gen\_running$ is true, and then action gen_off such that gen_ran is true. In other words, $\pi^* = [gen\_on_0, gen\_off_1]$.

Listing 3.1: Domain File of Simple Generator

```
(define (domain Simple-Generator)
(:requirements :strips)
(:predicates (fuel_full)
             (gen_ran)
             (gen_running))
(:action gen_on
    :precondition (and (fuel_full))
    :effect (and (not (fuel_full))
            (gen_running)))

(:action gen_off
    :precondition (and (gen_running))
    :effect (and (gen_ran))))
```

Listing 3.2: Problem File of Simple Generator

```
(define (problem Sample_Problem)
(:domain Simple-Generator)
(:init (fuel_full))
(:goal (and (gen_ran))))
```

Given the domain and problem specifications, the knowledge base of the agent KB_a, encoded in propositional logic in the fashion of Kautz *et al.* [135], consists of the following set of formulae:

- **Initial states:**

$$fuel\_full_0 \tag{3.40}$$

$$\neg gen\_ran_0 \tag{3.41}$$

$$\neg gen\_running_0 \tag{3.42}$$

- **Goal state:**

$$gen\_ran_2 \tag{3.43}$$

- **Action gen_on preconditions and effects:**

$$gen\_on_i \rightarrow fuel\_full_i \tag{3.44}$$

$$gen\_on_i \rightarrow \neg fuel\_full_{i+1} \tag{3.45}$$

$$gen\_on_i \rightarrow gen\_running_{i+1} \tag{3.46}$$

$$fuel\_full_i \wedge \neg fuel\_full_{i+1} \rightarrow gen\_on_i \tag{3.47}$$

$$\neg gen\_running_i \wedge gen\_running_{i+1} \rightarrow gen\_on_i \tag{3.48}$$

- **Action gen_off preconditions and effects:**

$$gen\_off_i \rightarrow gen\_running_i \tag{3.49}$$

$$gen\_off_i \rightarrow gen\_ran_{i+1} \tag{3.50}$$

$$\neg gen\_ran_i \wedge gen\_ran_{i+1} \rightarrow gen\_off_i \tag{3.51}$$

- **Action exclusions:**

$$\neg gen\_on_i \vee \neg gen\_off_i \tag{3.52}$$

  for $i = \{0, 1\}$.

Now, let's assume the following knowledge base KB_h for a human user:

- **Initial states:**

$$fuel\_full_0 \tag{3.53}$$
$$\neg gen\_ran_0 \tag{3.54}$$
$$\neg gen\_running_0 \tag{3.55}$$

- **Goal state:**

$$gen\_ran_2 \tag{3.56}$$

- **Action gen_on preconditions and effects:**

$$gen\_on_i \rightarrow fuel\_full_i \tag{3.57}$$
$$gen\_on_i \rightarrow \neg fuel\_full_{i+1} \tag{3.58}$$
$$gen\_on_i \rightarrow gen\_running_{i+1} \tag{3.59}$$
$$fuel\_full_i \wedge \neg fuel\_full_{i+1} \rightarrow gen\_on_i \tag{3.60}$$
$$\neg gen\_running_i \wedge gen\_running_{i+1} \rightarrow gen\_on_i \tag{3.61}$$

for $i = \{0, 1\}$. That is, the human user is missing the set of formulae that represent action gen_off (Lines 49-51) and the action exclusion of the two actions (Line 52), from her knowledge base.

Due to the omission of that set of formulae, the human user is unaware of the action gen_off and the plan $\pi^* = [gen\_on_0, gen\_off_1]$ is thus not only suboptimal but is also invalid to the human user (specifically, with respect to the user's knowledge base). Therefore, the goal is to explain the validity and/or optimality of π^* to the human user by reconciling the two knowledge bases. Couched in terms of propositional logic, we have to find an explanation ϵ such that $\widehat{\mathrm{KB}}_h^{\epsilon} \models^c_L \pi^*$ when explaining the validity of the plan (see Definition 9) and/or $\widehat{\mathrm{KB}}_h^{\epsilon} \models^s_L (\neg gen\_ran_0 \wedge \neg gen\_ran_1)$ when explaining the optimality of the plan (see Definition 10).

Now, the set of formulae that yield an explanation for both plan validity and optimality is the set of formulae consisting of action gen_off and the action exclusion axiom of actions

Formula Type	Notation	Template
Initial state	f_0	$\{f\}.name$ must be part of the initial specification of the problem.
Goal state	f_n	$\{f\}.name$ must be part of the goal specification of the problem.
Action Precondition	$a_t \Rightarrow f_t$	Action $\{a\}.name$ requires precondition $\{f\}.name$.
Action Addition effect	$a_t \Rightarrow f_{t+1}$	Action $\{a\}.name$ requires addition effect $\{f\}.name$.
Action Deletion effect	$a_t \Rightarrow \neg f_{t+1}$	Action $\{a\}.name$ requires deletion effect $\{f\}.name$.
Action Duration	$a_{t\_\mathrm{sta}} \Rightarrow$ $a_{t\_\mathrm{dur}} \leq duration \wedge$ $a_{t\_\mathrm{dur}} \geq duration$	Action $\{a\}.name$ has a duration of $\{duration\}$.
Process precondition	$ps_t \Leftrightarrow f_t$	Process $\{ps\}.name$ requires precondition $\{f\}.name$.

Table 3.1: Various formula types and their mapping onto pre-defined natural language templates.

gen_on and gen_off, i.e.,

$$\epsilon = [gen\_off_i \rightarrow gen\_running_i, gen\_off_i \rightarrow gen\_ran_{i+1}, \tag{3.62}$$

$$\neg gen\_ran_i \wedge gen\_ran_{i+1} \rightarrow gen\_off_i, \neg gen\_on_i \vee \neg gen\_off_i] \tag{3.63}$$

for $i = \{0, 1\}$.

Then, after updating KB_h with ϵ using Definition 4 to get the updated $\widehat{\mathrm{KB}}_h^{\epsilon} = (\mathrm{KB}_h \cup \epsilon) \setminus \emptyset$, the plan π^* is valid (i.e., $\widehat{\mathrm{KB}}_h^{\epsilon} \models_L^c \pi^*$) and optimal (i.e., $\widehat{\mathrm{KB}}_h^{\epsilon} \models_L^s (\neg gen\_ran_0 \wedge \neg gen\_ran_1)$) to the human user.

3.4.3 Mapping Logic-based Explanations to Natural Language Templates

We now describe a simple method for transforming logic-based explanations from our framework into a human-understandable format. To do that, we leverage the expressivity and symbolic nature of logic. Notice that a knowledge base encoding a planning problem contains logical formulae that represent various phenomena of the problem. These formulae are of a specific type, i.e., there are formulae encoding the initial and goal states, the action dynamics of the problem, and so on (see Section 3.4.2). Each formula is grounded on propositional variables, with each variable "symbolizing" a planning element such as an action or a predicate. For example, $a_0 \rightarrow p_0$ is a formula characterizing that action a_0 has precondition p_0. As such, given an explanation consisting of a set of logical formulae, each formula's variables can be extracted and, depending on the type of the formula, be mapped onto pre-defined, natural language templates.[13] To offer some more concrete examples, Table 3.1 shows this method for different types of formulae that may arise in an explanation.

3.5 Computing Explanations in Planning Problems

Since finding optimal plans for hybrid planning problems is often infeasible, we focus on describing algorithms that compute explanations for explaining the validity and optimality of classical plans to users in this section. Nonetheless, these algorithms can be trivially adapted to compute explanations for explaining only the validity (but not optimality) of both classical and hybrid plans, which we will describe in Section 3.5.2.[14]

Our core algorithmic engine is Algorithm 3.1, which is a general search algorithm that searches through the space of explanations in a best-first manner to find one that is optimal with respect to a given cost function. It takes as inputs two knowledge bases KB_a and KB_h of a logic L, two formulae φ_s and φ_c, and a cost function $\mathcal{C}_L$. The algorithm will output an explanation ϵ such that when it is used to update KB_h to $\widehat{\mathrm{KB}}_h^\epsilon$ according to Definition 4, the resulting updated knowledge base will credulously entail φ_c (i.e., $\widehat{\mathrm{KB}}_h^\epsilon \models^c_L \varphi_c$) and skeptically entail φ_s (i.e., $\widehat{\mathrm{KB}}_h^\epsilon \models^s_L \varphi_s$).

The algorithm makes three assumptions: (1) First, it assumes that KB_a and KB_h encode the version of the same planning problem of the planning agent and human user, respectively.

[13]Note that within our framework, this becomes relatively straightforward, as our explanations consist of macro-formulae (see end of Section 3.5.1), i.e., there will be no formulae repeated across multiple time steps in our explanations.

[14]For a theoretical analysis of the algorithms in this section, please see Appendix ??.

(2) It assumes that KB_a is *correct* and *complete*,[15] and only KB_h can contain errors or omissions. (3) It assumes that $\mathrm{KB}_a \models^c_L \varphi_c$ and $\mathrm{KB}_a \models^s_L \varphi_s$. The first assumption is reasonable and follows closely the definition of a model reconciliation planning problem (see Section 3.3.3). The last two assumptions stem from the fact that the explaining agent bases its explanations on the view (or model) of the specific problem [169]. Therefore, the agent should believe that its model KB_a is correct and complete, and that its model correctly and appropriately entails the queries φ_c and φ_s. Together, these three assumptions imply that each erroneous formula in KB_h will have a corresponding correct formula in KB_a, or, more formally:

Definition 11 (Corresponding Formula)**.** *Given two knowledge bases* KB_a *and* KB_h*, where each knowledge base encodes the same planning problem* Π*, a formula* φ_h *in* KB_h *has a corresponding formula* φ_a *in* KB_a *if both formulae characterize the same action (or state) axiom of* Π.

This is an important property that our algorithm, which we will explain later, will exploit to improve efficiency.

Finally, our algorithm also relies on the existence of an algorithm for checking credulous and skeptical entailment between knowledge bases and formulae (Line 1). For example, one can use the DPLL algorithm for the SAT encoding for classical planning. If KB_h already credulously entail φ_c and skeptically entail φ_s, then there is no need to compute an explanation (Lines 1-2). Otherwise, it goes into a search for an explanation, which is described below.

The key data structures in the algorithm is a priority queue q, initialized to only include the empty set, of potential explanations ordered by their costs (Line 4) and a set *checked* of invalid explanations that have been considered thus far (Line 5). The algorithm repeatedly loops the following steps:

- Move the explanation ϵ with the smallest cost from the priority queue q to *checked* (Lines 7-8).
- Create a copy of KB_h updated with ϵ according to Definition 4 (Line 9).
- Check if the copy $\widehat{\mathrm{KB}}_h^{\epsilon}$ credulously entails φ_c and skeptically entails φ_s. If so, return the explanation ϵ (Lines 10-11).

[15] By *complete* we mean that the KB encodes the the full planning problem as specified in the PDDL domain.

Algorithm 3.1: `most-preferred`$(L, \mathrm{KB}_a, \mathrm{KB}_h, \varphi_s, \varphi_c, \mathcal{C}_L)$

Input: Logic L, KBs KB_a and KB_h, formulae φ_s and φ_c, cost function $\mathcal{C}_L$

Output: A most-preferred explanation w.r.t. $\mathcal{C}_L$ from KB_a to KB_h to skeptically entail φ_s and credulously entail φ_c

1 **if** $\mathrm{KB}_h \models^c_L \varphi_c$ *and* $\mathrm{KB}_h \models^s_L \varphi_s$ **then**
2   **return** $\emptyset$
3 **else**
4     $q = \emptyset$ `// priority queue of potential explanations`
5     $checked = \emptyset$ `// a set of sets of elements in` KB_a `considered`
6   **repeat**
7         $\epsilon = dequeue(q)$
8   insert ϵ into *checked*
9         $\widehat{\mathrm{KB}}_h^{\epsilon} = \mathrm{KB}_h$ updated with ϵ according to Definition 4
10   **if** $\widehat{\mathrm{KB}}_h^{\epsilon} \models^c_L \varphi_c$ *and* $\widehat{\mathrm{KB}}_h^{\epsilon} \models^s_L \varphi_s$ **then**
11   **return** ϵ
12   **else**
13   **for** $a \in \mathrm{KB}_a \setminus \mathrm{KB}_h$ **do**
14   **if** $\epsilon \cup \{a\} \notin checked$ **then**
15                     $v = \mathcal{C}_L(\mathrm{KB}_h, \epsilon \cup \{a\})$
16                     $q = enqueue(\epsilon \cup \{a\})$ `// use` v `as key`
17   **until** *q is empty*

- If not, extend the explanation by 1 (with each formula from $\mathrm{KB}_a \setminus \mathrm{KB}_h$) and insert the extended explanations into the priority queue q (Lines 12-16). Only the formulae in $\mathrm{KB}_a \setminus (\mathrm{KB}_h \cap \mathrm{KB}_a)$ are considered since formulae that are already in KB_h will not help in the entailment process.

This search process continues until an explanation is found. It is impossible to exhaust all potential explanations and not find a valid explanation. The reason is that, in the worst case, the entire KB_a will serve as an explanation since KB_a credulously and skeptically entail φ_c and φ_s, respectively.

Note that this algorithm is defined in terms of logic and is agnostic to the underlying planning application domain. However, it can be used to find explanations for the MRP problem in explainable planning by setting $\varphi_c = \pi^* \wedge g_n$ to the optimal plan π^* of length n that needs to be explained and that the goal state is achieved at time step n; and by setting $\varphi_s = \bigwedge_{t=0}^{n-1} \neg g_t$ to the negation of the goal being reached at all time steps before time step n. Then, the algorithm will return a most-preferred explanation, with respect to $\mathcal{C}_L$, that explains that the plan π^* is both valid and optimal.

Constructing the Search Space: Finally, we would like to emphasize on a strategy that can be employed in Algorithm 3.1 to speed up its search procedure. Notice that the formulae in an encoded knowledge base are often repeated across all time steps (see Section 3.3.2).

For instance, consider the example in Section 3.4.2. The precondition ($= fuel\_full$) of the action gen_on in KB_a is repeated across time steps $i = 0, 1$, e.g., $\{gen\_on_0 \rightarrow fuel\_full_0, gen\_on_1 \rightarrow fuel\_full_1\} \in \text{KB}_a$. It is straightforward to see that if the knowledge base is encoded at a larger horizon, the search space of the algorithm (i.e., $\text{KB}_a \setminus \text{KB}_h$, Line 13) will increase in size analogously. However, as the same phenomena should hold across all time steps in a knowledge base encoding a given planning problem, a reasonable and intuitive strategy would be to construct the search space by aggregating the formulae with respect to the time steps, akin to the lifted representation of action dynamics in a PDDL domain. Specifically, the search space can now consist of *macro-formulae*, e.g., $Pre(gen\_on, fuel\_full) = \{gen\_on_0 \rightarrow fuel\_full_0, gen\_on_1 \rightarrow fuel\_full_1\}$, instead of formulae representing the same action dynamic at each time step.[16]

3.5.1 Pre-Processing Approximation Algorithm

While Algorithm 3.1 can compute a most-preferred explanation, it is straightforward to see that the complexity of the problem is at least NP-hard since finding a most-preferred explanation is a combinatorial problem. As the intended use of the algorithm is to provide an explanation that a particular optimal plan π^* is both valid and optimal, we now exploit this assumption and introduce a pre-processing algorithm that can be used to modify KB_h. At a high level, this algorithmic approach can be thought of as "reforming" the knowledge base of the human user in order to make the agent's plan valid.

Before describing the algorithm, we make an observation that there exists only a single model in a knowledge base that is encoding classical and hybrid planning problems that is consistent with a plan π^* of the problem. We formalize this as Proposition 1 below.

[16]This simple idea can be used during the encoding of a planning problem into a logical knowledge base, i.e., by creating a hash table mapping macro-formulae to the associated (repeated) formulae.

Proposition 1. *Given a plan π^* and a knowledge base* KB_a *encoding the classical or hybrid planning problem, there exists only a* single *model in* KB_a *that is consistent with* π^**, i.e.,* $|ACC_L(\mathrm{KB}_a \wedge \pi^*)| = 1$.

PROOF. First, observe that both classical and hybrid planning problems are deterministic planning problems without parallel action execution. Consequently, for classical planning problems, the transition between states is encoded by the frame axioms, which enforce that a state literal becomes TRUE (resp. FALSE) if and only if it was an addition (resp. deletion) effect of an action. The same reasoning can be extended to hybrid planning problems, where state transitions are described by happenings, which encode the causal chain of events, processes and instantaneous actions. Further, due to the action exclusions axioms, only one action can be TRUE at each time step, and hence only the action literals supported by π^* can yield an assignment of TRUE. Therefore, it follows logically that there exists **only one** model consistent with the plan π^*, i.e., $|ACC_L(\mathrm{KB}_a \wedge \pi^*)| = 1$. □

Using Proposition 1, one can then generalize that the formulae in KB_h that are false according to this model must be erroneous with respect to KB_a. A trivial approach would be to replace these formulae with their corresponding (correct) formulae from KB_a (see Definition 11) before running Algorithm 3.1 with this modified KB_h instead of the original KB_h as the input. The (correct) formulae from KB_a as well as the output from Algorithm 3.1 would then serve as the explanation to KB_h.

However, it is important to note that not all formulae need to be corrected in order for KB_h to credulously entail an optimal plan π^* of length n that reaches the goal at time step n and skeptically entail that the goal cannot be reached before time step n. For example, there may be actions that are not used in the optimal plan with wrong preconditions or effects that need not be corrected. Therefore, we only use a *partial model* – we only extract the truth value assignments for literals that are *directly needed for the optimal plan* π^*, i.e., the literals corresponding to the initial and goal states, to the states that are in the precondition of any action in the plan as well as the literals corresponding to all the actions in KB_a. We call this the *relevant literals* with respect to π^*. States that are not preconditions of actions in the plan are not extracted. Function 1 describes the pseudocode of this procedure. First, it initializes the partial model μ as an empty set (Line 1) and extracts the satisfying model M from KB_a that credulously entails the optimal plan π^* of length n and that the goal state is reached only at time step n and not before, where each element of M is a tuple consisting

Function 1: `extract-partial-model`$(L, \mathrm{KB}_a, \pi^*)$

Input: Logic L, KB_a, and optimal plan π^*
Output: Partial model from KB_a w.r.t. π^*

1 $\mu = \emptyset$
2 $M =$ `get-SAT-model`(KB_a)
3 $\Lambda =$ `extract-relevant-literals`(KB_a, π^*)
4 **for** $(l, t) \in M$ **do**
5   **if** $l \in \Lambda$ **then**
6         $\mu = \mu \cup \{(l, t)\}$
7 **return** μ

Algorithm 3.2: `pre-processing`$(L, \mathrm{KB}_a, \mathrm{KB}_h, \pi^*)$

Input: Logic L, KBs KB_a and KB_h, and optimal plan π^*
Output: *Approximated* explanations from KB_a to KB_h

$\epsilon = \emptyset$
$M =$ `extract-partial-model`$(L, \mathrm{KB}_a, \pi^*)$
for $k_h \in \mathrm{KB}_h$ **do**
    if $\neg$`evaluate`(L, M, k_h) **then**
        $\epsilon = \epsilon \cup$ corresponding formula from KB_a
return ϵ

Algorithm 3.3: `approximate`$(L, \mathrm{KB}_a, \mathrm{KB}_h, \pi^*, \mathcal{C}_L)$

Input: Logic L, KBs KB_a and KB_h, plan π^* with length n, cost-function $\mathcal{C}_L$
Output: Explanation from KB_a to KB_h to credulously entail π^* and the goal state is reached at time step n, and skeptically entail that the goal state is not reachable before time step n

$\phi = \bigwedge_{t=0}^{n-1} \neg g_t$
$\epsilon =$ `pre-processing`$(L, \mathrm{KB}_a, \mathrm{KB}_h, \pi^*)$
$\widehat{\mathrm{KB}}_h = \mathrm{KB}_h$ with formulae corresponding to ϵ (if any) replaced with ϵ
$\epsilon' =$ `most-preferred`$(L, \mathrm{KB}_a, \widehat{\mathrm{KB}}_h, \phi, \pi^* \wedge g_n, \mathcal{C}_L)$
return $\epsilon \cup \epsilon'$

of a literal and its respective truth value (Line 2). Then, it extracts the relevant literals Λ with respect to π^* (Line 3) before it loops through all elements (l, t) in M and adds them to partial model μ if the literal l is a relevant literal (Lines 4-6). Finally, the partial model μ is returned after the loop (Line 7).

It is straightforward to see how this can help speed up the search in Algorithm 3.1. The pre-processing step will increase the number of formulae that are in both KBs. Consequently, since the search space of Algorithm 3.1 is the power set of formulae in KB_a that are not in KB_h (see Line 13), our pre-processing step will reduce the search space and runtime of the algorithm. Finally, as long as atomic explanations are formulae in KB_a, the pre-processing step will provide *approximated* explanations, that is, the formulae that are replaced approximate the formulae that are needed for the entailment of the optimal plan. As this method may yield superfluous information, in so far as the explanations may contain formulae not needed for the entailment of the plan, we frame this method as an approximation technique.

Algorithm 3.2 describes the pseudocode for this pre-processing algorithm. First, it initializes its set of explanations ϵ as an empty set (Line 1) and extracts a partial model of KB_a that credulously entails the optimal plan π^* of length n and that the goal state is reached only at time step n and not before (Line 2). Then, the algorithm loops through all formulae of KB_h (Line 3) and checks if each formula evaluates to false according to the partial model (Line 4). If it is false, then that formula is replaced with the corresponding formula from KB_a and it is added to the explanation set ϵ (Line 5). The set of *approximated* explanations is returned after the whole loop (Line 6). To illustrate the utility of this procedure, consider the following example.

Example 4. *Assume a version of the Generator domain that consists of two actions gen_on = {precondition: $fuel\_full$, effect: $gen\_running$} and gen_on_alt = {precondition: $fuel\_mid$, effect: $gen\_running$} with initial and goal states $fuel\_full$ and $gen\_running$, respectively, and a plan $\pi^* = [gen\_on]$. Also, assume that the human user is not aware that action gen_on has effect $gen\_running$. Then, the knowledge bases encoding the models of the agent and the human are respectively:*

$$\mathrm{KB}_a = [fuel\_full_0, \neg fuel\_mid_0, \neg gen\_running_0, gen\_running_1, \tag{3.64}$$

$$gen\_on_0 \rightarrow fuel\_full_0, gen\_on_0 \rightarrow gen\_running_1, \tag{3.65}$$

$$gen\_on\_alt_0 \rightarrow fuel\_mid_0, gen\_on\_alt_0 \rightarrow gen\_running_1, \tag{3.66}$$

$$\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0 \vee gen\_on\_alt_0, \tag{3.67}$$

$$\neg gen\_on_0 \vee \neg gen\_on\_alt_0] \tag{3.68}$$

$$\mathrm{KB}_h = [fuel\_full_0, \neg fuel\_mid_0, \neg gen\_running_0, gen\_running_1, \tag{3.69}$$

$$gen\_on_0 \rightarrow fuel\_full_0, gen\_on\_alt_0 \rightarrow fuel\_mid_0, \tag{3.70}$$

$$gen\_on\_alt_0 \rightarrow gen\_running_1, \tag{3.71}$$

$$\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on\_alt_0, \neg gen\_on_0 \vee \neg gen\_on\_alt_0] \tag{3.72}$$

Now, the partial model we extract from KB_a *with respect to* π^* *is:*

$$M = \{fuel\_full_0 = T, fuel\_mid_0 = F, gen\_running_0 = F, gen\_on_0 = T, \tag{3.73}$$

$$gen\_on\_alt_0 = F, gen\_running_1 = T\} \tag{3.74}$$

Then, we can see that according to M*, the formula* $\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on\_alt_0$ *from* KB_h *evaluates to false. As such, it would be replaced by the corresponding formula from* KB_a*, namely* $\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0 \vee gen\_on\_alt_0$*.*

Algorithm 3.3 describes the complete algorithm that uses Algorithm 3.2 as a pre-processing step. After running the pre-processing algorithm and getting the preliminary set of *approximated* explanations ϵ (Line 2), it creates a copy $\widehat{\mathrm{KB}}_h$ with formulae corresponding to ϵ replaced with ϵ (Line 3). Then, it runs Algorithm 3.1 to find the remaining set of explanations ϵ' (Line 4) and returns the union of those both sets (Line 5). The key observation here is that the input to Algorithm 3.1 is $\widehat{\mathrm{KB}}_h$ and not KB_h. Since $\widehat{\mathrm{KB}}_h$ is more similar to KB_a, the search space of Algorithm 3.1 will be smaller and it is thus more efficient. It is important to emphasize again that this an *approximation* technique for finding most-preferred explanations of minimal cardinality.

3.5.2 Modifications for Plan Validity Explanations

Recall that the algorithms described above compute explanations for explaining both the validity and optimality of classical plans to users. However, they can be easily adapted to compute explanations for explaining only the validity (but not optimality) of both classical and hybrid plans. Specifically, the changes are the following:

- For Algorithm 3.1, one needs to only omit φ_s from the pseudocode. Specifically, φ_s need not be passed in as an argument, and the checks on Lines 1 and 10 need to be changed to only check for the credulous entailment of φ_c.

- For Algorithm 3.2, no changes are necessary, except that the input plan π^* corresponds to the plan whose validity needs to be explained. It does not have to be an optimal plan.
- For Algorithm 3.3, similar to the case above, the input plan π^* corresponds to the plan whose validity needs to be explained. Additionally, ϕ need not be passed in as an argument on Line 4 and, consequently, Line 1 can be omitted.

3.5.3 Theoretical Analysis

We first prove the completeness and correctness of the general Algorithm 3.1 in Theorems 1 and 2, respectively, in the context of finding explanations for optimal (or valid) plans in model reconciliation problems. We then prove the completeness of our pre-processing Algorithm 3.2 in Theorem 3, before proving the completeness of the combined Algorithm 3.3 in Theorem 4.

Theorem 1. *Algorithm 3.1 is guaranteed to terminate with a solution when one exists.*

Proof. First, a solution will always exist because, in the worst case, the entire $\mathrm{KB}_a \setminus \mathrm{KB}_h$ will serve as an explanation to KB_h since KB_a credulously entails φ_c and skeptically entails φ_s. We now prove that it is guaranteed to terminate with a solution. As Algorithm 3.1 iteratively adds sets of formulae of increasing size from $\mathrm{KB}_a \setminus \mathrm{KB}_h$ into its priority queue q (Lines 7 and 13-16), it will eventually add the entire power set of $\mathrm{KB}_a \setminus \mathrm{KB}_h$ into the queue. Since each element in the queue is only evaluated exactly once (Lines 8 and 14), the set of formulae in $\mathrm{KB}_a \setminus \mathrm{KB}_h$ will eventually be evaluated, and when it is used to update KB_h (Line 9), the updated $\widehat{\mathrm{KB}}_h^{\epsilon}$ will credulously entail φ_c and skeptically entail φ_s (Line 10). As a result, $\epsilon = \mathrm{KB}_a \setminus \mathrm{KB}_h$ will be returned as a solution upon termination (Line 11). □

Theorem 2. *Algorithm 3.1 is guaranteed to return a most-preferred explanation if the cost function is monotonic.*

Proof. If the cost function $\mathcal{C}_L$ used by Algorithm 3.1 is monotonic, then for any two explanations $\epsilon_1 \subseteq \epsilon_2$, $\mathcal{C}_L(\mathrm{KB}_h, \epsilon_1) \leq \mathcal{C}_L(\mathrm{KB}_h, \epsilon_2)$.

We now prove by contradiction that it is not possible for Algorithm 3.1 to return an explanation ϵ that is less preferred than a most-preferred explanation ϵ^* (i.e., $\mathcal{C}_L(\mathrm{KB}_h, \epsilon) > \mathcal{C}_L(\mathrm{KB}_h, \epsilon^*)$). Assume that the algorithm does return such an explanation ϵ. Since potential explanations are popped off the priority queue according to their costs (Line 7 and 16), it means that when the algorithm popped off explanation ϵ, the explanation ϵ^* is not in the priority queue since $\mathcal{C}_L(\mathrm{KB}_h, \epsilon) > \mathcal{C}_L(\mathrm{KB}_h, \epsilon^*)$. There are the following two cases:

- Explanation ϵ^* is not in the queue because it was already popped off earlier. In this case, the algorithm would have terminated and returned the explanation ϵ^*, which contradicts our assumption that the algorithm returned explanation ϵ.
- Explanation ϵ^* is not in the queue because it hasn't yet been added into the queue. This means that there exists some subset $\epsilon' \subset \epsilon^*$ that is in the queue and is not yet evaluated. Further, it must be the case that $\mathcal{C}_L(\mathrm{KB}_h, \epsilon) \leq \mathcal{C}_L(\mathrm{KB}_h, \epsilon')$ because, otherwise, ϵ' would have been popped off the queue and evaluated. Additionally, since the cost function is monotonic, $\mathcal{C}_L(\mathrm{KB}_h, \epsilon') \leq \mathcal{C}_L(\mathrm{KB}_h, \epsilon^*)$. Combining these two inequalities, we get $\mathcal{C}_L(\mathrm{KB}_h, \epsilon) \leq \mathcal{C}_L(\mathrm{KB}_h, \epsilon^*)$, which contradicts our assumption that $\mathcal{C}_L(\mathrm{KB}_h, \epsilon) > \mathcal{C}_L(\mathrm{KB}_h, \epsilon^*)$.

Therefore, it is not possible for Algorithm 3.1 to return an explanation ϵ that is less preferred than a most-preferred explanation ϵ^*. □

Recall that each formula in KB_h will have a corresponding formula in KB_a since KB_a is assumed to be complete.[17] However, it may not be true that each formula in KB_a will have a corresponding formula in KB_h since KB_h can be incomplete. We formalize this statement in the following postulate, and then use it to prove properties of our pre-processing Algorithm 3.2.

Postulate 1. *For a model reconciliation problem, assume that* KB_a *and* KB_h *encode the SAT (or SMT) instances of the planning agent and human user, respectively. Then, each formula in* KB_h *will have a corresponding formula in* KB_a.

Theorem 3. *Algorithm 3.2 is guaranteed to terminate with a solution when one exists.*

Proof. Algorithm 3.2 iteratively evaluates all formulae in KB_h with respect to KB_a's partial model (Line 20). If any formulae evaluate to false with respect to KB_a's partial model, the algorithm will replace them with the corresponding ones from KB_a (Lines 21-22). Since the number of formulae in KB_h is finite, the algorithm will eventually complete evaluating all the formulae and return the set of formulae from KB_a that correspond to the

[17] Assuming that $\mathrm{KB}_h \neq \emptyset$.

set of formulae in KB_h that evaluates to false (Line 23). In other words, if a solution exists, Algorithm 3.2 is guaranteed to return it. $\square$

Theorem 4. *Algorithm 3.3 is guaranteed to terminate with a solution when one exists.*

Proof. As Algorithm 3.3 comprises of Algorithms 3.1 and 3.2, which are guaranteed to terminate with a solution when one exists (Theorems 1 and 3), the algorithm is also guaranteed to terminate with a solution when one exists. $\square$

We now describe the worst-case time complexities of the algorithms.

Theorem 5. *The time complexity of Algorithm 3.1 is $O(2^{|\mathrm{KB}_a|} + 2^{|\mathrm{KB}_a \setminus \mathrm{KB}_h|+m})$, where m is the maximum number of variables in $\widehat{\mathrm{KB}}_h^{\epsilon}$ over all candidate explanations ϵ.*

Proof. On the basic operations, the runtimes for inserting elements into sets (Line 8), checking for set memberships (Line 14), and computing costs of potential explanations (Line 15) are all $O(1)$; and the runtimes of inserting and removing elements into priority queues (Line 7 and 16) are $O(\log(n))$, where n is the size of the queue. On the entailment checks on Lines 1 and 10, their runtimes are $O(2^m)$, where m is the number of variables in the knowledge base, because that is the number of models in the knowledge base. The number of times the algorithm has to loop through Lines 6 to 17 is the size of the power set $\mathrm{KB}_a \setminus \mathrm{KB}_h$, which is $O(2^{|\mathrm{KB}_a \setminus \mathrm{KB}_h|})$, since there are that many unique subsets of potential explanations to consider. Within the loop, Line 9 takes $O(2^{|\mathrm{KB}_h|})$ time because it has to iterate through the power set of KB_h to find the minimal set of formulae to remove according to Definition 4, and the number of times the algorithm has to loop through Lines 13 to 16 is $|\mathrm{KB}_a \setminus \mathrm{KB}_h|$.

Therefore, in total, the runtime of the algorithm is $O(2^m)$ (Line 1) $+$ $O(2^n)$ (Lines 6 and 17) $\cdot\big[O(\log(n))$ (Lines 7-8) $+$ $O(2^{|\mathrm{KB}_h|})$ (Line 9) $+$ $O(2^m)$ (Line 10) $+$ $O(n)$ (Line 13) $\cdot O(\log(n))$ (Line 16) $\big]$ $= O(2^n \cdot (2^{|\mathrm{KB}_h|} + 2^m + n\log(n))) = O(2^{n+|\mathrm{KB}_h|} + 2^{n+m}) = O(2^{|\mathrm{KB}_a|} + 2^{n+m})$, where $n = |\mathrm{KB}_a \setminus \mathrm{KB}_h|$ and m is the maximum number of variables in $\widehat{\mathrm{KB}}_h^{\epsilon}$ over all candidate explanations ϵ. $\square$

Theorem 6. *The time complexity of Algorithm 3.2 is $O(2^n + |\mathrm{KB}_h| \cdot |\mathrm{KB}_a|)$, where n is the number of variables in KB_a.*

Proof. The time complexity of extracting the partial model (Function 1) on Line 19 is as follows. Finding a satisfying model M from KB_a for π^* (Line 25) takes $O(2^n)$, where

n is the number of variables in KB_a, since it is a Boolean satisfiability problem, which is NP-complete [42]. Extracting the relevant literals (Line 26) takes $O(l)$ time, where l is the maximum number of relevant literals. Finally, Lines 27-29 take $O(|M|)$. Therefore, the runtime for Line 19 is $O(2^n) + O(l) + O(|M|) = O(2^n)$.

The number of times the algorithm has to loop through Lines 20 to 22 is $O(|\text{KB}_h|)$. Within the loop, Line 21 takes $O(k)$ time, where k is the length of the longest formula evaluated,[18] and Line 22 takes $O(|\text{KB}_a|)$ time since it needs to loop through the entire KB_a in the worst case to find the corresponding formula.[19]

Therefore, in total, the runtime of the algorithm is $O(2^n)$ (Line 19) + $O(|\text{KB}_h|)$ (Line 20) $\cdot\big[O(k)$ (Line 21) + $O(|\text{KB}_a|)$ (Line 22) $\big] = O(2^n + |\text{KB}_h| \cdot (k + |\text{KB}_a|)) = O(2^n + |\text{KB}_h| \cdot |\text{KB}_a|)$, where n is the number of variables in KB_a. □

Theorem 7. *The time complexity of Algorithm 3.3 is* $O(2^{|\text{KB}_a|} + 2^{|\text{KB}_a \setminus \text{KB}_h| + m})$, *where m is the maximum number of variables in $\widehat{\text{KB}}_h^{\epsilon}$ over all candidate explanations ϵ.*

Proof. The pre-processing call on Line 32 takes $O(2^n + |\text{KB}_h| \cdot |\text{KB}_a|)$, where n is the number of variables in KB_a (Theorem 6).

To implement the update of the knowledge base KB_h with the explanation ϵ to $\widehat{\text{KB}}_h^{\epsilon}$ on Line 33, we loop through the explanation ϵ and, for each formula in the explanation, we loop through KB_h to find the corresponding formula. Once found, we replace the formula in KB_h with the formula from the explanation. Therefore, the runtime for this step is $O(|\epsilon| \cdot |\text{KB}_h|)$.

Finally, the runtime for Line 34 is $O(2^{|\text{KB}_a|} + 2^{|\text{KB}_a \setminus \widehat{\text{KB}}_h| + m})$, where n is the maximum number of variables in $\widehat{\text{KB}}_h^{\epsilon'}$ over all candidate explanations ϵ' (Theorem 5). Assuming $|\text{KB}_h| = |\widehat{\text{KB}}_h|$, then the runtime is $O(2^{|\text{KB}_a|} + 2^{|\text{KB}_a \setminus \text{KB}_h| + m})$.

Therefore, in total, the runtime of the algorithm is $O(2^m + |\text{KB}_h| \cdot |\text{KB}_a|)$ (Line 32) + $O(|\epsilon| \cdot |\text{KB}_h|)$ (Line 33) + $O(2^{|\text{KB}_a|} + 2^{|\text{KB}_a \setminus \widehat{\text{KB}}_h| + m})$ (Line 34) $= O(2^m + |\text{KB}_h| \cdot |\text{KB}_a| + |\epsilon| \cdot |\text{KB}_h| + 2^{|\text{KB}_a|} + 2^{|\text{KB}_a \setminus \widehat{\text{KB}}_h| + m}) = O(2^{|\text{KB}_a|} + 2^{|\text{KB}_a \setminus \text{KB}_h| + m})$, where m is the maximum number of variables in $\widehat{\text{KB}}_h^{\epsilon}$ over all candidate explanations ϵ.

□

[18] We assume the formula is represented in conjunctive normal form.

[19] This upper bound can be tightened by using hash functions, but we consider naive implementations here.

3.6 Empirical Evaluations

In this section, we present results from computational experiments as well as human-subject experiments.

3.6.1 Computational Experiments

We now describe the empirical evaluations of our algorithms presented in the previous section for finding explanations on classical and hybrid planning problems, encoded as SAT and SMT problems, respectively.

Setup and Prototype Implementation: We ran our experiments on a Macbook Pro comprising an Intel Core i7 2.6GHz processor with 16GB of memory. We implemented our algorithms in Python and integrated the well known *z3* solver [54] for satisfiability and entailment checking, which was accessed through the PyZ3 framework.[20] The knowledge bases representing the planning problems were each encoded up to the time step that the optimal (or valid) plan was found. To encode the knowledge bases for classical planning problems, we used our own implementation of the encoding in [135], whereas for hybrid planning problems we used the encoding provided in SMTPlan [30]. Further, note that each knowledge base was encoded as a hash table (see Footnote 16). The time limit for all experiments was set to 1500s. We have also made our source code available in a publicly-accessible repository.[21]

Our empirical evaluations were tailored around the following three questions:

- **Question 1:** What is the advantage of using the pre-processing method described in Section 3.5.1?
- **Question 2:** What is the efficacy of our algorithms on classical planning problems?
- **Question 3:** What is the efficacy of our algorithms on hybrid planning problems?

[20] https://github.com/Z3Prover/z3.

[21] https://github.com/YODA-Lab/Explanation-Generation-for-Planning-Problems.

Question 1: Advantage of Pre-Processing Approach

With this question, we wanted to examine if there is any advantage in using the pre-processing approach introduced in Section 3.5.1 for finding most-preferred explanations. To do this, we evaluated Algorithm 3.1 (referred to as ALG1), which does not use the pre-processing method, and Algorithm 3.3 (referred to as ALG3), which does use the method, to find most-preferred explanations for plan validity and optimality on classical planning benchmarks from the International Planning Competition (IPC).[22] We used the explanation length $|\epsilon|$ as the cost function of the algorithms and incorporated the knowledge base update on Line 9 in Algorithm 3.1 by using a simple linear search algorithm (See the Appendix in [231] for more details).

We used the actual IPC instances as the model of the agent (i.e., KB_a), and tweaked that model and assigned it to be the model of the human user (i.e., KB_h). In order to make a more comprehensive analysis, we considered five different ways to tweak the models, resulting in the following five scenarios.

- **Scenario 1:** We removed one random precondition from every action in the human's model.
- **Scenario 2:** We removed one random effect from every action in the human's model.
- **Scenario 3:** We removed one random precondition and one random effect from every action in the human's model.
- **Scenario 4:** We removed (on average) fifteen random preconditions and effects from every action in the human's model.
- **Scenario 5:** We removed (on average) ten random predicates from the initial states in the human's model.

Table 3.2 tabulates the length of the explanations $|\epsilon|$ and the runtimes of the two algorithms as well as a third algorithm called CSZK, which we will describe in the next section. We will focus on the comparisons between ALG1 and ALG3 in this section.

[22] https://github.com/potassco/pddl-instances.

Prob.		Scenario 1				Scenario 2				Scenario 3				Scenario 4				Scenario 5			
		$\|\epsilon\|$	CSZK	ALG1	ALG3	$\|\epsilon\|$	CSZK	ALG1	ALG3	$\|\epsilon\|$	CSZK	ALG1	ALG3	$\|\epsilon\|$	CSZK	ALG1	ALG3	$\|\epsilon\|$	CSZK	ALG1	ALG3
Blocks-world	4	1	**0.5s**	2.5s	3.0s	2	**0.5s**	1.0s	0.7s	3	2.0s	3.0s	**1.5s**	3	32.0s	29.0s	**16.0s**	2	–	1.5s	**0.7s**
	5	2	**2.5s**	8.0s	8.5s	3	**2.0s**	4.0s	2.5s	5	17.0s	6.0s	**6.0s**	7	–	–	**194.0s**	4	–	4.0s	**2.0s**
	6	1	**1.0s**	25.0s	25.0s	2	**0.5s**	5.5s	5.5s	3	**3.0s**	6.0s	6.0s	4	213.0s	–	**120.0s**	5	–	8.0s	**5.0s**
	8	3	**62.0s**	296.5s	297.0s	3	**1.0s**	30.0s	29.5s	6	869.0s	101.0s	**30.0s**	7	–	–	**203.0s**	5	–	40.0s	**27.0s**
Elev-ator	1	1	0.5s	**0.1s**	**0.1s**	2	1.0s	1.5.s	**0.1s**	2	0.5s	1.0s	**0.1s**	2	1.0s	5.0s	**0.1s**	1	–	**0.1s**	**0.1s**
	10	2	1.5s	0.8s	**0.7s**	2	**0.5s**	1.0s	**0.5s**	3	3.0s	2.5s	**0.4s**	4	57.0s	30.0s	**2.5s**	6	–	5.0s	**0.2s**
	15	2	**1.5s**	2.5s	3.0s	2	**1.0s**	15.0s	13.0s	3	3.0s	10.0s	**2.0s**	4	57.0s	61.0s	**10.0s**	6	–	14.0s	**1.2s**
	19	2	**2.0s**	7.5s	8.0s	2	**0.5s**	25.0s	25.0s	3	**2.5s**	50.0s	10.0s	4	49.0s	123.0s	**20.0s**	14	–	40.0s	**5.0s**
Rover	1	1	**0.5s**	6.5s	6.0s	2	**0.5s**	10.0s	7.0s	4	33.0s	10.0s	**5.0s**	6	–	29.0s	**5.0s**	4	–	**1.5**	**1.5s**
	2	1	**1.0s**	4.0s	4.0s	1	**0.5s**	12.0s	4.0s	4	39.0s	6.0s	**4.5s**	6	–	125.0s	**4.5s**	4	–	2.0s	**1.3s**
	3	1	**0.5s**	7.5s	7.0s	2	**0.5s**	22.0s	7.5s	4	35.0s	16.0s	**7.0s**	6	–	173.0s	**10.0s**	6	–	2.0s	**1.5s**
	4	1	**0.5s**	4.0s	4.0s	1	**0.5s**	4.0s	4.0s	2	**1.5s**	4.5s	4.5s	4	–	33.0s	**4.5s**	10	–	15.0s	**5.5s**
Gripper	1	1	**0.5s**	2.0s	1.5s	2	**0.3s**	3.0s	3.0s	3	**1.5s**	41.0s	40.0s	5	70.0s	201.0s	**45.0s**	4	–	3.0s	**2.0s**
	2	1	**0.5s**	5.0s	5.0s	2	**0.8s**	19.0s	7.0s	3	**2.0s**	122.0s	45.0s	5	73.0s	349.0s	**49.0s**	5	–	15.0s	**6.0s**
	3	1	**0.7s**	5.5s	5.0s	2	**1.0s**	22.0s	7.0s	3	**2.5s**	46.0s	45.0s	5	163.0s	555.0s	**60.0s**	8	–	20.0s	**15.0s**
	4	1	**1.5s**	37.5s	38.0s	2	**3.0s**	224.0s	50.0s	3	**5.0s**	149.0s	50.0s	5	–	700.0s	**80.0s**	11	–	66.0	**28.0s**
Hanoi	1	1	**0.3s**	0.5s	0.5s	1	0.2s	**0.1s**	**0.1s**	1	0.3s	0.5s	**0.1s**	1	0.2s	**0.1s**	**0.1s**	4	–	**0.1s**	**0.1s**
	2	1	**0.3s**	0.5s	0.5s	1	**0.3s**	2.0s	2.0s	1	**0.3s**	5.5s	1.5s	5	7.0s	13.0s	**3.0s**	6	–	2.5s	**0.3s**
	3	1	**0.3s**	3.5s	3.0s	1	**0.4s**	7.0s	4.0s	2	**0.4s**	11.5s	1.5s	6	**10.0s**	31.0s	14.0s	8	–	6.0s	**1.5s**
	4	1	**0.3s**	19.5s	20.0s	1	**0.3s**	21.0s	20.0s	2	**0.4s**	44.0s	15.0s	6	**10.0s**	230.0s	30.0s	10	–	30.0s	**10.0s**
TPP	1	1	0.7s	**0.5s**	**0.5s**	2	0.2s	**0.1s**	**0.1s**	6	23.0s	30.0s	**3.5s**	11	–	100.0s	**20.0s**	2	–	**0.1s**	**0.1s**
	2	1	0.7s	**0.5s**	**0.5s**	2	**0.3s**	3.0s	2.0s	6	24.0s	45.0s	**5.5s**	11	–	404.0s	**27.0s**	4	–	3.0s	**0.3s**
	3	1	**0.7s**	3.5s	3.0s	2	**0.4s**	11.0s	4.0s	6	25.5s	95.0s	**7.0s**	11	–	1200.s	**70.0s**	6	–	9.0s	**1.0s**
	4	1	**0.7s**	21.0s	20.0s	2	**0.3s**	71.0s	20.0s	6	25.0s	124.0s	**15.0s**	11	–	–	**111.0s**	8	–	22.0s	**8.0s**
Driver Log	1	1	**1.0s**	2.0s	2.0s	4	2.5s	2.0s	**1.5s**	2	**2.5s**	11.0s	4.0s	5	–	66.0s	**6.0s**	3	–	5.0s	**2.0s**
	2	2	24.0s	14.0s	**13.0s**	5	**5.0s**	7.5s	7.0s	4	–	239.0s	**63.0s**	7	–	–	**90.0s**	5	–	14.0s	**5.0s**
	3	2	**5.0s**	10.0s	10.0s	3	**0.5s**	11.0s	11.0s	5	–	481.0s	**70.0s**	6	–	–	**120.0s**	7	–	19.0s	**9.5s**
	4	2	**7.0s**	11.0s	11.5s	5	**4.0s**	13.0s	12.5s	5	–	355.0s	**55.0s**	9	–	–	**237.0s**	9	–	41.0s	**22.5s**
Logi-istics	1	2	**1.5s**	2.0s	2.0s	3	**1.0s**	4.0s	2.5s	4	30.0s	15.0s	**5.0s**	4	168.0s	250.0s	**12.5s**	2	–	**1.5s**	**1.5s**
	2	2	**2.0s**	3.0s	**2.0s**	3	**1.5s**	2.0s	2.0s	4	29.0s	**5.5s**	**5.5s**	4	169.5s	240.5s	**13.0s**	3	–	3.5s	**1.5s**
	3	2	**2.0s**	2.5s	2.5s	3	**1.0s**	2.5s	2.0s	4	31.0s	**5.5s**	**5.5s**	4	167.0s	400.0s	**13.0s**	4	–	5.0s	**2.0s**
	4	2	**2.0s**	**2.5s**	**2.0s**	3	**0.9s**	4.0s	3.0s	4	30.0s	28.0s	**6.0s**	4	168.5s	103.0s	**12.5s**	5	–	12.0s	**2.0s**
Zeno Travel	1	1	**0.5s**	0.8s	0.7s	1	**0.5s**	**0.5s**	**0.5s**	1	**0.5s**	1.0s	**0.5s**	1	**0.3s**	3.0s	0.4s	3	–	**0.5s**	**0.5s**
	2	4	10.0s	**5.0s**	**5.0s**	1	0.8s	**0.5s**	**0.5s**	4	40.0s	12.0s	**5.5s**	5	–	60.0s	**10.0s**	5	–	4.0s	**2.0s**
	3	3	8.5s	5.0s	**4.5s**	3	1.0s	**0.5s**	**0.5s**	5	30.5s	45.0s	**5.0s**	6	–	100.0s	**20.0s**	7	–	13.0s	**3.0s**
	4	3	10.0s	**5.0s**	**5.0s**	3	1.5s	1.0s	**0.7s**	5	31.0s	50.0s	**6.0s**	6	–	633.0s	**50.0s**	9	–	25.0s	**4.5s**
Storage	1	1	0.5s	0.5s	**0.3s**	1	**0.3s**	1.0s	0.5s	2	0.5s	0.5s	**0.3s**	3	647.0s	450.0s	**10.0s**	2	–	0.5s	**0.2s**
	2	1	0.5s	0.5s	**0.3s**	1	0.5s	0.5s	**0.4s**	3	**4.0s**	14.0s	5.0s	5	–	400.0s	**11.5s**	4	–	3.0s	**0.6s**
	3	1	0.7s	0.5s	**0.3s**	1	**0.5s**	0.5s	0.6s	3	**5.0s**	25.0s	**5.0s**	4	–	504.0s	**21.0s**	6	–	22.0s	**1.5s**
	4	5	**11.0s**	12.0s	12.0s	4	**4.0s**	22.0s	10.0s	6	**41.0s**	222.0s	51.0s	4	–	823.0s	**46.0s**	8	–	41.0s	**5.0s**

Table 3.2: Evaluation of ALG1, ALG3 and CSZK on Varying PDDL Domain and Scenario.

Notice that ALG1 and ALG3 yielded comparable runtimes in most instances of Scenarios 1 and 2. This is due to the fact that the tweaked KB_h produced from those scenarios is consistent with all candidate explanations from $\mathrm{KB}_a \setminus \mathrm{KB}_h$ (i.e., the search space of potential explanations). In other words, there is no need to restore the consistency of KB_h during an update with a candidate explanation. In that case, ALG1 and ALG3 are virtually similar, as they both follow the same general search procedure.[23] Nevertheless, the difference in runtime becomes quite substantial in Scenarios 3, 4, and 5, highlighting the strength of

[23]Recall that the main advantage of ALG3 is the ability to identify (and replace) formulae in KB_h that evaluate to false with respect to KB_a's partial model.

the pre-processing approach. In fact, these are scenarios in which the tweaked KB_h is either inconsistent (i.e., there is no valid plan in KB_h) or becomes inconsistent with potential explanations from KB_a. There are two main reasons as to why ALG3 outperformed ALG1 in those scenarios. First, the search space of ALG3 may be smaller than that of ALG1 as ALG3 employs the pre-processing technique before forming the search space of potential explanations. Furthermore, ALG1 may need to restore the consistency of KB_h multiple times throughout its execution, in the worst case with each potential explanation, which consequently may increase the total runtime of the algorithm. Note that ALG1 performs an uninformed search over what formulae to remove when updating KB_h with each potential explanation. In contrast, the pre-processing technique of ALG3 guarantees that KB_h will be consistent with all potential explanations by performing an informed search over what formulae to remove (i.e., it removes only the necessary set of formulae that evaluate to false with respect to the partial model of KB_a). In conclusion, these results shows that there is a clear advantage in employing the pre-processing approach for finding most-preferred explanations within our framework, especially for problems where the consistency of KB_h needs to be restored during an update.

Question 2: Efficacy on Classical Planning Problems

This question concerns how well our algorithms perform on classical planning problems. To address this, we compared our algorithms against the current planning-based state-of-the-art algorithm by Chakraborti *et al* [37], referred to as CSZK – the initials of the last names of the authors.[24] In what follows, we will only discuss the results of our best performing algorithm ALG3 and CSZK.

Table 3.2 tabulates the length of the explanations $|\epsilon|$ as well as the runtimes of the algorithms. In general, CSZK outperformed ALG3 in a majority of cases, except for Scenarios 3 and 4 in all domains. These cases also happen to be the cases where the explanation length $|\epsilon|$ is larger. We did not report runtimes of CSZK for Scenario 5 as the available implementation could not handle that scenario.

To verify if such correlations exist, we conducted an additional experiment where we varied the explanation length $|\epsilon|$ as well as the optimal plan length in the BLOCKSWORLD domain.

[24] We used the implementation of the authors, which is publicly available at https://github.com/TathagataChakraborti/mmp.

Optimal Plan Length	Explanation Length $\|\epsilon\|$									
	2		4		6		8		10	
	CSZK	ALG3	CSZK	ALG3	CSZK	ALG3	CSZK	ALG3	CSZK	ALG3
6	**0.5s**	1.0s	2.0s	**0.9s**	9.5s	**0.8s**	300.0s	**1.0s**	500s	**1.0s**
10	**0.5s**	3.0s	**2.5s**	**2.5s**	9.5s	**3.0s**	300.0s	**3.5s**	500s	**2.5s**
12	**0.5s**	4.5s	**2.0s**	**5.0s**	9.0s	**6.5s**	305.0s	**4.5s**	505s	**7.0s**
16	**1.0s**	28.0s	**2.0s**	27.0s	**10.0s**	26.0s	309.0s	**27.0s**	502s	**31.0s**
26	**1.0s**	70.0s	**8.0s**	75.0s	**20.0s**	73.0s	312.5s	**80.0s**	–	**85.0s**

Table 3.3: Varying Explanation and Plan Lengths for the BLOCKSWORLD PDDL Domain.

Table 3.3 tabulates the results. It shows a clear trend that the runtimes of CSZK increases as the explanation lengths increase. The reason is that CSZK needs to search over a larger search space as the explanation length increases. As such, its runtime also increases. In contrast, the runtimes of ALG3 remain relatively unchanged with varying explanation lengths. The reason is that the runtimes of ALG3 are dominated by the size of the encoded knowledge bases, which are independent of the explanation lengths.

The results also show that the runtimes of CSZK remain relatively unchanged with varying optimal plan lengths. The reason is that the runtimes of CSZK are dominated by its search for explanations. CSZK runs an A* search over the explanation search space and as long as the explanation length remains unchanged, the runtime complexity of the search, which is exponential in the explanation length, remains relatively unchanged as well. However, the runtimes of ALG3 increase as the optimal plan lengths increase. We attribute this to the following two reasons. First, longer plans means that there are more combinations of actions to consider in the explanation search space, consequently increasing the runtime of the algorithm. Additionally, the size of the knowledge bases increases as the plan length increases. Thus, there is an increasing number of formulae, which likely results in an increase in the runtime of the underlying SAT solver.

Upon closer inspection of the instances generated the experiments thus far, we observed that in almost all of these instances, the shortest plan in the human's model is at least as long as the optimal plan in the agent's model. Therefore, the experiments thus far were strongly biased in favor of explanations for *plan validity* instead of *plan optimality*. We thus conducted an additional experiment where we varied the number of invalid shorter-than-optimal plans in the human's model. These plans are invalid because they are comprised of some actions with wrong or missing preconditions and/or effects and these actions enable the goal state to be reached earlier than using a plan that is optimal in the agent's model. Table 3.4 tabulates the results. As expected, the results show that the runtime of both CSZK and ALG3 increases

Prob.	# of Invalid Shorter-than-Optimal Plans														
	2			4			6			8			10		
	$\lvert\epsilon\rvert$	CSZK	ALG3	$\lvert\epsilon\rvert$	CSZK	ALG3	$\lvert\epsilon\rvert$	CSZK	ALG3	$\lvert\epsilon\rvert$	CSZK	ALG3	$\lvert\epsilon\rvert$	CSZK	ALG3
1	2	2.5s	**0.2s**	4	50.0s	**1.0s**	6	1231.0s	**11.0s**	8	–	**104.0s**	10	–	**840.5s**
2	2	3.0s	**0.5s**	4	57.0s	**1.0s**	6	1225.0s	**12.5s**	8	–	**105.5s**	10	–	**842.0s**
3	2	3.0s	**2.0s**	4	62.0s	**2.5s**	6	1240.0s	**11.0s**	8	–	**107.0s**	10	–	**845.0s**
4	2	**2.5s**	3.0s	4	60.0s	**1.5s**	6	1235.0s	**13.0s**	8	–	**111.0s**	10	–	**850.0s**

Table 3.4: Varying Invalid Shorter-than-Optimal Plans.

as the number of invalid plans increases. However, interestingly, the runtime of CSZK grows faster than that of ALG3, and ALG3 is up to 2 orders of magnitude faster than CSZK (when there are six invalid plans).

All of these observations result in the following three conclusions that highlight the different situations for when one should use one algorithm over the other: (1) ALG3 outperforms CSZK when explanations are long, and vice versa when explanations are short; (2) ALG3 outperforms CSZK when optimal plans are short, and vice versa when the optimal plans are long; and (3) ALG3 outperforms CSZK when there are many invalid shorter-than-optimal plans.

Question 3: Efficacy on Hybrid Planning Problems

In this final question, we wanted to investigate the generality of our approach on problems beyond classical planning, particularly on hybrid planning problems. As such, we now provide results on a number of PDDL+ problems.

Recall that finding optimal hybrid plans is often infeasible, and as such, there is no SAT/SMT planner that can prove optimality for general PDDL+ problems (see Section 3.3.2). Therefore, it is not feasible to find optimal plans to explain in our experiments. However, given that optimality cannot be guaranteed, a user may have doubts about the validity of a plan (i.e., whether the plan is sound and can be executed to achieve the goal). Thus, we limited ourselves to experiments for *plan validity* only.

As Algorithm 3.3, referred to as ALG3, was designed to find explanations that explain both plan validity and optimality, we tweaked it to only check for credulous entailment since that is sufficient for finding explanations for plan validity (see Section 3.5.2). We did not

Prob.		Scenario 1		Scenario 2		Scenario 3		Scenario 4		Scenario 5		Scenario 6		Scenario 7		Scenario 8	
		$\|\epsilon\|$	ALG3	$\|\epsilon\|$	ALG3	$\|\epsilon\|$	ALG3	$\|\epsilon\|$	ALG3	$\|\epsilon\|$	ALG3	$\|\epsilon\|$	ALG3	$\|\epsilon\|$	ALG3	$\|\epsilon\|$	ALG3
LINEAR GENER.	1	0	0.1s	2	0.1s	2	0.2s	1	0.1s	2	0.1s	1	0.1s	3	0.2s	0	–
	3	0	0.1s	2	0.2s	2	0.8s	1	0.2s	2	0.2s	1	0.1s	3	0.8s	0	–
	5	0	0.2s	2	0.2s	2	2.0s	1	0.4s	2	0.4s	1	0.3s	3	3.0s	0	–
	7	0	0.3s	2	0.5s	2	4.0s	1	1.0s	2	0.6s	1	0.5s	3	14.0s	0	–
TORI-CELLI	1	1	0.2s	2	0.3s	2	0.4s	3	0.6s	4	0.2s	2	0.2s	4	0.2s	0	–
	2	1	0.4s	2	1.3s	2	2.0s	3	1.1s	5	0.9s	2	0.4s	4	0.7s	0	–
	3	1	0.5s	2	5.0s	2	11.0s	3	2.8s	7	3.6s	2	0.5s	4	2.0s	0	–
	4	1	1.0s	2	16.0s	2	38.0s	3	5.8s	5	1.1s	2	1.0s	4	4.0s	0	–
GENER. EVENTS	1	1	0.2s	2	0.2s	3	2.5s	3	0.2s	2	0.2s	1	0.1s	4	0.2s	1	0.2s
	2	1	0.3s	2	0.5s	3	5.0s	3	0.5s	3	0.2s	2	0.7s	5	0.7s	2	0.4s
	3	2	0.8s	2	1.3s	3	10.0s	3	1.5s	4	0.7s	2	0.6s	5	2.0s	2	1.5s
	4	1	1.3s	2	2.0s	3	26.0s	3	2.5s	6	0.9s	1	1.0s	4	4.5s	3	4.5s
CAR NO DRAG	1	2	0.2s	1	0.3s	3	0.3s	3	0.3s	2	0.2s	1	0.4s	3	0.4s	2	0.5s
	2	2	0.2s	1	0.2s	2	0.4s	3	0.3s	3	0.3s	2	0.5s	3	0.3s	2	0.4s
	3	2	0.3s	1	0.3s	2	0.2s	3	0.4s	1	0.1s	2	0.3s	3	0.4s	1	0.6s
	4	2	0.2s	1	0.2s	3	0.3s	3	0.3s	2	0.2s	1	0.2s	4	0.3s	1	1.0s
NONLIN. GENER	1	1	0.2s	2	0.1s	2	0.3s	1	0.2s	2	0.1s	1	0.1s	3	0.2s	0	–
	2	1	0.2s	2	0.2s	1	0.5s	1	0.4s	2	0.2s	1	0.2s	3	0.7s	0	–
	3	1	0.3s	2	0.2s	2	1.0s	1	0.3s	2	0.3s	1	0.4s	3	2.0s	0	–
	4	1	1.2s	2	0.5s	2	5.0s	1	0.5s	2	0.5s	1	0.5s	3	5.5s	0	–
SOLAR ROVER	1	1	0.5s	1	0.4s	2	0.4s	3	0.7s	2	0.2s	1	0.1s	3	0.2s	1	0.4s
	2	1	0.4s	1	0.5s	1	0.6s	2	1.4s	4	0.2s	1	0.2s	3	0.5s	1	0.4s
	3	1	0.8s	1	0.9s	2	1.0s	2	2.0s	6	0.3s	1	0.4s	3	5.0s	1	0.6s
	4	1	1.5s	1	2.0s	1	2.0s	3	3.5s	8	0.4s	1	0.5s	5	10.0s	1	1.5.s
POWERED DESCENT	1	1	0.2s	1	0.4s	2	0.4s	3	1.0s	2	0.3s	1	0.2s	3	0.8s	1	0.6s
	2	1	0.3s	1	0.6s	1	0.5s	2	2.5s	4	0.4s	1	0.1s	3	1.5s	1	0.7s
	3	1	0.3s	1	1.0s	2	1.5s	2	3.0s	6	0.3s	1	0.3s	3	3.0s	1	1.0s
	4	1	0.7s	1	3.0s	1	3.5s	3	5.5s	8	0.4s	1	0.2s	4	8.0s	1	3.0s
NONLIN SOLAR ROVER	1	1	0.5s	2	0.4s	1	0.6s	3	1.0s	2	0.3s	1	0.3s	3	0.8s	1	0.5s
	2	2	1.0s	2	1.0s	2	1.5s	4	3.5s	4	0.4s	2	0.4s	5	1.5s	2	1.5s
	3	2	4.0s	3	4.0s	2	2.5s	4	4.0s	6	0.3s	2	0.4s	5	3.0s	2	1.0s
	4	3	5.0s	4	4.0s	1	2.0s	6	7.5s	8	0.4s	1	0.3s	4	6.0s	2	1.0s

Table 3.5: Evaluation of ALG3 on Varying PDDL+ Domain and Scenario.

compare it with any other algorithm since to the best of our knowledge existing explanation generation algorithms are limited to classical planning problems.

Similar to our first experiment for classical planning, we used the actual domain instances as the model of the agent and tweaked it for the model of the human user. We considered the same five scenarios earlier as well as the following three additional scenarios:

- **Scenario 6:** We changed the duration of all the durative actions in the human's model.
- **Scenario 7:** We removed (on average) five random preconditions and effects as well as changed the duration of all durative actions in the human's model.

- **Scenario 8:** We removed (on average) two events and processes in the human's model.

Table 3.5 tabulates the results. We did not report runtimes for the LINEAR, TORICELLI, and NON-LINEAR GENERATOR domains for Scenario 8 as these domains do not contain events and/or processes. In general, ALG3 was able to maintain small runtimes of less than 1s in the majority of instances. The reason is that the size of the encoded knowledge bases are relatively small because SMTPLAN uses the iterative encoding facility of the $z3$ solver. Specifically, the encoding of each layer consists of the following steps: Adding the new variables and constraints for the next happening, adding the goal constraints to the new constraint set, pushing the constraint set onto the stack, solving, and popping the goal constraint set off the stack. As such, at each step in the iterative deepening with $z3$, only the latest layer needs to be encoded.

In conclusion, these results demonstrate that our approach can be generalized beyond classical planning to hybrid planning, improving the applicability of explainable planning approaches.

3.6.2 Human-Subject Experiments

While there has been some supporting evidence suggesting that explanations in the form of model reconciliation are well understood by human users when explaining classical plans to them [35, 240], to our knowledge, their applicability to hybrid planning problems has not been investigated thus far. Therefore, in what follows we present a human-subject study on a hybrid planning variant of the Logistics domain [167], where we use a combination of visualizations and text for presenting explanations to human users.

Study Design

Recall that the model reconciliation problem requires that the explaining agent knows both its model and the model of the human user receiving the explanation. To enforce this assumption, we used variations of the Logistics domain [167] as the model of the explaining agent, tweaked that model, and assigned this tweaked model to participating human users. To ensure the human users understood their assigned model, we presented it to them at the start

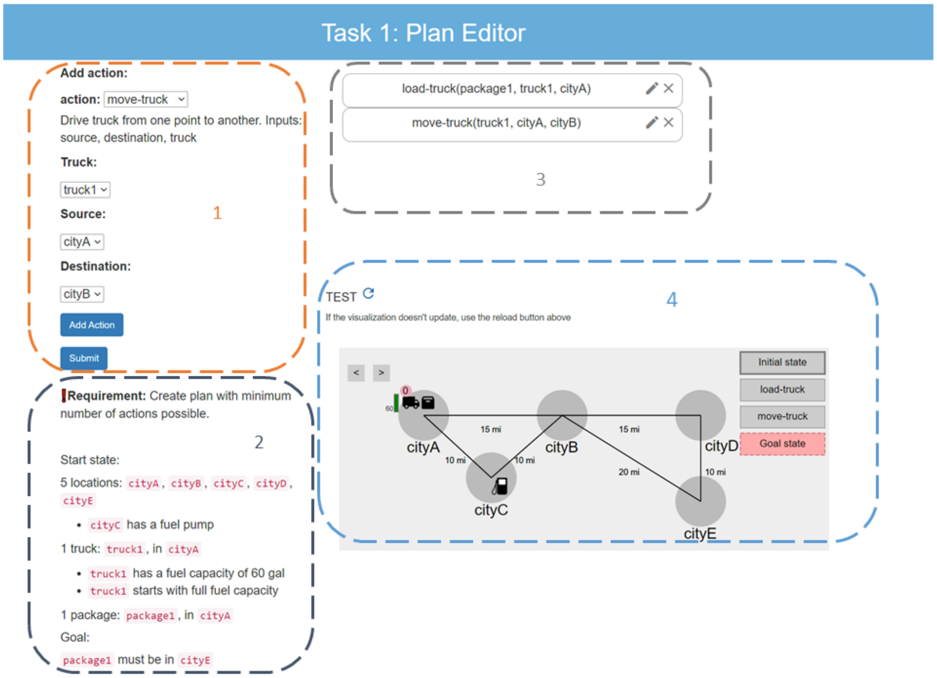


Figure 3.1: A view of the plan editor for the human user study. (1) Action selection; (2) The initial and goal states; (3) User's current plan; (4) Test visualization showing validity of the plan.

of the study and asked them to create a plan given that model. We removed participants who created plans inconsistent with the tweaked model provided to them. Then, human users were provided an explanation in the form of model reconciliation (e.g., from our algorithms) and were asked to answer a series of questions (see Appendix A.1) as well as correct their plans based on their understanding of the explanation. The answers to those questions as well as the human users' ability to correct their plans reflect their understanding of the explanations provided. In this study, we hypothesize that *explanations in the form of model reconciliation are effective for explaining hybrid planning plans to human users.*

The user study was conducted on the online crowdsourcing platform Prolific [175]. We created two tasks for each participant as follows:

- **Task 1:** For this task, participants were asked to create a (shortest) plan based on the modified model provided to them using the visualization system coupled with a simple plan editing interface, as shown in Figure 3.1.[25] This interface also allows users to evaluate their plans, which can subsequently provide information about any errors in their plans due to their misunderstanding of the provided information in their models.

A participant succeeds in Task 1 if they create and submit a valid plan given their model. Participants that succeeded in Task 1 continued to Task 2, and participants that failed in Task 1 were filtered out and ignored. This is important due to the assumption (in model reconciliation) that the human user's model is known a-priori. In addition, participants whose plans were valid in their model but did not require an explanation (e.g., their plans were also valid in the agent's model) were also filtered out.

- **Task 2:** For this task, we informed the participants that the initial model provided to them contained errors, and presented an explanation for those errors using visualizations as well as text. They were then asked a series of questions to evaluate their understanding of the explanation provided (**Task 2a**). The exact questions we asked the participants can be found in Appendix D. Then, they were shown the plan editor again and asked to correct their plan, this time without the ability to evaluate their plans for correctness (**Task 2b**). A participant succeeded in Task 2b if their corrected plan was valid in the agent's model.

To incentivize participants to provide answers to the best of their ability, we provided a bonus reward to participants who succeeded in Task 1, and an additional bonus to participants who also succeeded in Task 2b. Furthermore, we also included two questions for attention checks in the study, where participants were asked to type a particular string or select a particular answer in a multiple choice question. Participants who wrongly answered both of these questions were filtered out of the study.

Additionally, we performed a control experiment, where participants would receive the exact same tasks, but in Task 2a and Task 2b, they would not be shown the explanations. Instead, they are expected to correct their plan with just the knowledge of which actions were wrong. This allows us to see whether providing the explanation has any benefit, or if just the knowledge of the wrong actions is enough to help users in correcting the plan.

Finally, each participant had the following interactions in the study: (1) They arrive at the webpage following the link from Prolific, where they enter their demographics and some

[25] To eliminate any learning effects, participants were shown the visualization system in Task 1 to ensure that they are familiar with the system before receiving an explanation using that interface.

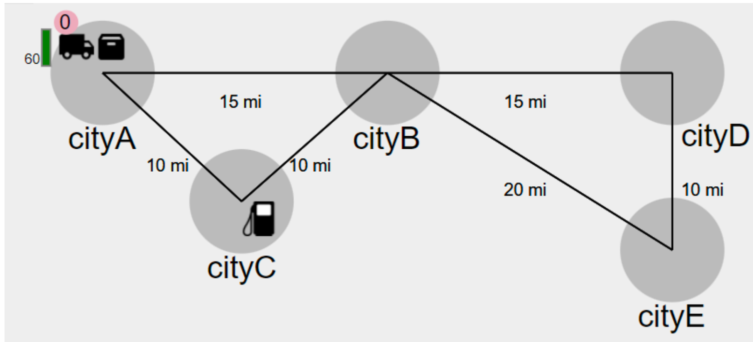


Figure 3.2: The initial state of the modified Logistics problem. The goal is to transport the package from `cityA` to `cityE`. A shortest plan here would have the truck load the package at `cityA` and move the truck to `cityC` to refuel before moving it to (the goal location) `cityE`.

information on their educational background. (2) To ensure that they have the background necessary to solve the tasks, they are given tutorials on automated planning, the Logistics domain, and the plan editing interface. (3) Following the tutorials, they are asked to complete Task 1. (4) If they succeeded in Task 1, they are asked to complete Tasks 2a and 2b. (5) All participants, including those who failed Task 1, are then asked to provide feedback on the system's usability [112] and are informed of their payments before being redirected back to Prolific.

Domain and Problem

As mentioned, our choice of domain was a variant of the Logistics domain [167], which we augmented to contain elements of hybrid planning problems. The particular Logistics problem involves the transportation of packages across cities via trucks. To make the domain a hybrid planning problem, we included a *process* by adding a fuel bar to the truck which is depleted by the execution of the action `move-truck`, and an *event* to stall the truck if it runs out of fuel midway to a city. The truck has a specified fuel consumption rate in gallons per mile (gal/mi), that is, it burns 2gal/mi. Additionally, we added the action `refuel` to refuel the truck, and a predicate `is-fuelpump` to note locations where the truck can be refueled. We created a simple problem with five cities and one fuel pump. Figure 3.2 shows the initial state for this problem. There is one truck available and one package that needs to be transported to the goal city without stalling the truck.

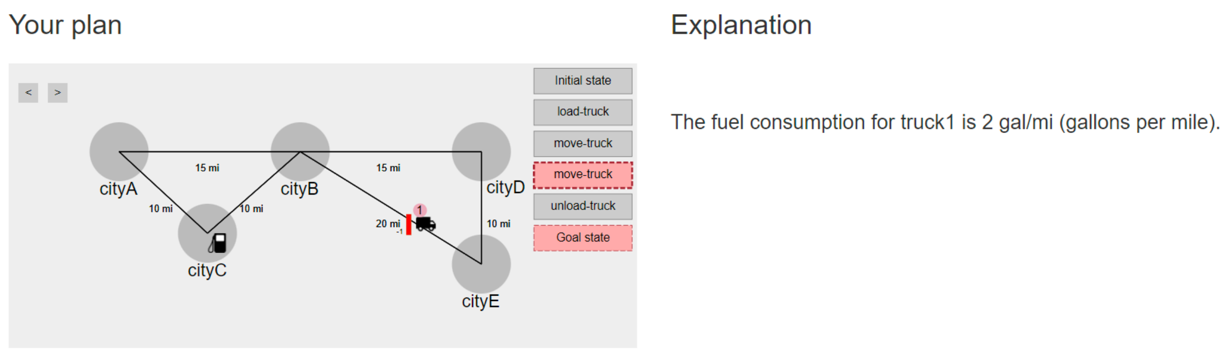


Figure 3.3: A view of the explanation phase for the modified Logistics problem. We used a pulsing border on the fuel bar to indicate that the truck has run out of fuel, and thus stalled. The explanation is also presented in text to the right of the visualization.

We considered the following change for the modified model of the human user: We modified the fuel consumption rate to be 1gal/mi instead of the actual 2gal/mi. This allowed users to create a shorter plan in the modified domain by directly moving the truck to the destination instead of going to `cityC` and refueling. Thus, an explanation is needed to explain why their plan is invalid, and how to compute a (shortest) valid plan.

Prototype Implementation

We created a visualization system that combines the action-space and state-space information of the planning problem and can visualize plans and their execution as well as present explanations to human users. In [146] we describe how to create such a system in more detail. The inclusion of the action-space information also conveniently presents a simple way for human users to select and view different states. Actions whose preconditions are not met in the human user's plan are highlighted as red. his helps human users while creating their plan, ensuring they understand the preconditions specified in the provided model. Moreover, the state-space shows the "current" state of the world after the execution of the action selected by the human user in their plan, including information such as the positions of the truck and package, and the truck's fuel level. In addition, we used animations showing the movement of the truck between each city as well as the truck's fuel consumption.

During the explanation phase (Task 2a), actions are highlighted by using red highlights to indicate which actions are wrong, with the state-space visualization providing more details about what exactly went wrong in the human user's plan. For presenting explanations within the state-space visualization, we do the following: For each state in the execution of the plan, starting from the initial state, we display the current state with respect to the actions that are executed in the human user's plan, using the agent's domain. Figure 3.3 shows the explanation phase.

Note that for the purpose of this study, the text-based explanation (as seen in Figure 3.3) was handcrafted. Nonetheless, there may be systematic ways for translating logic-based explanations from our framework to natural language text as we highlight in Section 3.4.3.

Evaluation

The study was conducted with 100 participating users (50 in the main group and 50 in the control group). All 100 participants have at least some undergraduate education in any field. Out of all the participants, only data from 72 users was used (35 in the main group and 37 in the control group), as the rest either failed Task 1 or had created plans that did not need explanations (e.g., their plans included going to City C to refuel before going to Cities B and E).

As we mentioned earlier, the goal of this study is to identify if explanations in the form of model reconciliation can convey to humans the validity of hybrid planning plans. To evaluate our hypothesis, we used the following measures:

- **Comprehension Score:** Number of questions users answered correctly in Task 2a.
- **Correction Ratio:** Proportion of users who succeeded in Task 1 (plan creation phase) that also succeeded in Task 2b (plan correction phase).

Table 3.6 summarizes the results of our analysis, where we report the population size and the two measures used for evaluation. In general, the results seem to suggest that the vast majority of users in the main study population understood the explanations communicated to them. In particular, we first scored participants based on the number of correct answers they gave for the questions asked in Task 2a (comprehension score). With a maximum possible score of 8, we managed to obtain an overall mean score of 6, which indicates that the presented explanations had a positive cognitive effect, as most users used the explanation

	Population Size	Comprehension Score	Correction Ratio
Main Study	35	6 (out of 8)	91% (32/35)
Control	37	N/A	24% (9/37)

Table 3.6: Human User Study Results for the Modified Logistics Domains.

to reason through the questions and successfully answer most of them. We then measured how many users were able to accurately correct their plans in Task 2b when shown the explanation (correction ratio), i.e., the users' ability to reflect the explanation presented to them into creating the (agent's) correct plan. The percentage of users that succeeded was 91%, showing that most users indeed made use of the explanation by adding its information to their models and creating the correct plan. From the control group, we can observe that the explanation does indeed have an effect of how well the participants were able to correct their plans, as users who did not see the explanations had a much lower correction ratio (24%), as expected.

We would also like to address the potential issue of bias in the results from Task 2b. Since we removed users who failed to complete Task 1, we might have filtered out participants who did not understand planning and their assigned models or had difficulty creating plans in general, thus creating a selection bias for users who are better at understanding and solving planning problems. As mentioned earlier, it was necessary to filter out users who did not understand their assigned models, as the explanation provided is contingent on the human model as per the assumptions made by model reconciliation. However, we believe that the ability to observe errors during plan creation in Task 1 mitigates this issue to some extent, as all users could observe any errors in their plans during this phase, and as such, potentially correct them before submitting. Further, since we observe a large difference in correction ratio for the control group with a similar number of people who completed Task 1, we believe that this potential bias does not have a major effect.

In conclusion, one can see that these findings corroborate our hypothesis, that is, explanations in the form of model reconciliation are effective for explaining hybrid planning plans to human users. The results of this study, complemented by the results obtained for classical planning problems by Chakraborti *et al.* [35] and Zahedi *et al.* [240], demonstrate the real-world efficacy of explanations as model reconciliation for planning problems beyond classical planning.

3.7 Related Work

We now provide a discussion of related work from the knowledge representation and reasoning (KR) and planning literature. We focus on these two areas as our logic-based approach bears some similarity to other logic-based approaches in KR and our application domain of explainable planning has been solved by other planning approaches.

3.7.1 KR Literature

As our explanation generation framework bears some similarities with the theory of belief change and abductive explanations, we start by describing their underlying theory. We then provide two examples that illustrate the differences between these approaches. Finally, we discuss some further related work from the KR community.

Belief Change

Belief change is a kind of change that can occur in a knowledge base. Depending on how beliefs are represented and what kinds of inputs are accepted, different typologies of belief changes are possible. In the most common case, when beliefs are represented by logical formulae, one can distinguish three main kinds of belief changes, namely, *expansion*, *revision*, and *contraction*. In the following, we formally describe the aforementioned notions.

Expansion: An expansion operator of a knowledge base can be formulated in a logical and set-theoretic notation:

Definition 12 (Expansion Operator). *Given a knowledge base* KB *and a formula* ϕ*,* $+_e$ *is an expansion operator if it expands* KB *by* ϕ *as* $\mathrm{KB} +_e \phi = \{\psi : \mathrm{KB} \cup \phi \vdash \psi\}$.

It is trivial to see that $KB +_e \phi$ will be consistent when ϕ is consistent with KB, and that $KB +_e \phi$ will be closed under logical consequences.

Revision: A belief revision occurs when we want to add new information into a knowledge base in such a way that, if the new information is inconsistent with the knowledge base, then the resulting knowledge base is a new consistent knowledge base. Alchourrón, Gärdenfors, and Makinson conducted foundational work on knowledge base revision, where they proposed a set of rationality postulates, called *AGM postulates*, and argued that every revision operator must satisfy them [2, 86, 89]. Although revision cannot be defined in a set-theoretic manner closed under logical consequences (like expansion), it can be defined as follows:

Definition 13 (Revision Operator)**.** *Given a knowledge base* KB *and a formula* ϕ*,* $+_r$ *is a revision operator if it satisfies the AGM postulates for revision and modifies* KB *w.r.t.* ϕ *such that the resulting* KB *is consistent.*

The underlying motivation behind the AGM postulates is that when we change our beliefs, we want to retain as much as possible the information from the old beliefs. Thus, when incorporating new information in the knowledge base, the heuristic criterion should be the criterion of *information economy* (i.e., minimal changes to the knowledge base is preferred). As such, a model-theoretic characterization of minimal change has been introduced by Katsuno *et al.* [134], where minimality is defined as selecting the models of ϕ that are "closest" to the models of KB.

However, the AGM rationality postulates will not be adequate for every application. Katsuno *et al.* [133] proposed a new type of belief revision called *update*. The fundamental distinction between the two kinds of belief revision in a knowledge base, namely *revision* and *update*, is that the former consists of incorporating information about a static world, while the latter consists of inserting information to the knowledge base when the world described by it changes. As such, they claim that the *AGM postulates describe only revision* and showed that *update can be characterized by a different set of postulates called KM postulates*.

Definition 14 (Update Operator)**.** *Given a knowledge base* KB *and a formula* ϕ*,* $+_u$ *is an update operator if it satisfies the KM postulates for update and modifies* KB *w.r.t.* ϕ *such that the updated* KB *incorporates the change in the world introduced by* ϕ*.*

From a model-theoretic view, the difference between revision and update, although marginal at first glance, can be described as follows: Procedures for revising KB by ϕ are those that satisfy the AGM postulates and select the models of ϕ that are "closest" to the models of KB. In contrast, update methods are exactly those that satisfy the KM postulates and select, for each model I of KB, the set of models of ϕ that are closest to I. Then, the updated KB will be characterized by the union of these models.[26]

It is worth mentioning that, on a high level, the key difference between update and revision is a temporal one: Update incorporates into the knowledge base the fact that the world described by it has changed, while revision is a change to our world description of a world that has not itself changed. We refer the reader to KAtsuno *et al.* [133] for a comprehensive description as well as an intuitive meaning between revision and update.

Contraction: Similarly to the AGM postulates for revision, Alchourrón *et al.* [2] proposed a set of axioms that any contraction operator must satisfy. Therefore, a contraction operator is defined by:

Definition 15 (Contraction Operator)**.** *Given a knowledge base* KB *and a formula ϕ, $-_c$ is a contraction operator if it satisfies the AGM postulates for contraction, and contracts* KB *w.r.t. ϕ by retracting formulae in* KB *without adding of new ones.*

We can see that, as in the case of revision, it is not possible to define contraction in a set-theoretic manner closed under logical consequences. We illustrate this with the following example:

Example 5. *Consider the Generator domain from Section 3.4.2. Assume a subset of the original* KB_a*, i.e.,* $\mathrm{KB}_a = [\neg gen\_running_0, gen\_running_1, \neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0]$ *and* $\phi = [gen\_on_0]$ *that we wish to contract. Then, in order to maintain consistency in* KB_a*, one of* $\neg gen\_running_0, gen\_running_1$*, or* $\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0$ *must be retracted. But which one? There is no logical reason for making one choice rather than the other.*

Interestingly, it has been shown that the problems of revision and contraction are closely related [87]. Despite the fact that the postulates that characterize revision and contraction are "independent,"[27] revision can be defined in terms of contraction (and vice versa). This is referred to as the Levi identity [153].

[26]This approach is called the *possible models approach* [238].

[27]In the sense that the postulates for revision do not refer to contraction and vice versa.

Definition 16 (Levi Identity)**.** *Assume a knowledge base* KB*, a formula ϕ, and operators $+_r$ and $-_c$ that satisfy the AGM postulates for revision and contraction, respectively. Then,* $\text{KB} +_r \phi = (\text{KB} -_c \neg\phi) +_e \phi$.

Hence, a revision of a knowledge base can be viewed as contracting KB with respect to $\neg\phi$ and then expanding $(\text{KB} -_c \neg\phi)$ by ϕ.

Abductive Explanations

Explanations in knowledge base systems were first introduced by Levesque [152] in terms of abductive reasoning, that is, given a knowledge base and a formula that we do not believe at all, what would it take for us to believe that formula? A more formal definition is provided below.

Definition 17 (Abductive Explanation)**.** *Given a knowledge base* KB *and a query q to be explained, α is an explanation of q w.r.t. to* KB *iff* $\text{KB} \cup \{\alpha\}$ *is consistent and* $\text{KB} \cup \{\alpha\} \models^s_L q$.

Usually, such explanations are phrased in terms of a hypothesis set H (set of atomic sentences – also called abducibles), and, generally, is an intuitive methodology for deriving root causes.

Two Illustrative Examples

To illustrate the differences between our approach and the KR approaches described in this section, we discuss below how they operate on two simplifications of the Generator domain example in Section 3.4.2.

Problem 1

Assume a simplified version of the Generator domain with only one action gen_on = {precondition: $fuel\_full$, effect: $gen\_running$}, and initial and goal states $fuel\_full$ and $gen\_running$, respectively. Clearly, the plan for this problem is $\pi^* = [gen\_on]$. Also, assume that the human user is not aware that action gen_on has effect $gen\_running$. Now, the knowledge bases encoding the models of the agent and the human, are respectively:

$$\text{KB}_a = [fuel\_full_0, \neg gen\_running_0, gen\_running_1, \tag{3.75}$$

$$gen\_on_0 \rightarrow fuel\_full_0, gen\_on_0 \rightarrow gen\_running_1, \tag{3.76}$$

$$\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0] \tag{3.77}$$

$$\text{KB}_h = [fuel\_full_0, \neg gen\_running_0, gen\_running_1, gen\_on_0 \rightarrow fuel\_full_0] \tag{3.78}$$

Further, without loss of generality, suppose that the explanation needed to explain π^* to KB_h is $\epsilon = [gen\_on_0 \rightarrow gen\_running_1, \neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0]$.

Abductive Explanations: Abductive explanation cannot be applied in this setting because KB_h does not contain any causal rules that can be used to abduce the query.

Revision: Since the union of ϵ and KB_h is consistent, the revision operator will yield a trivial update according to the second AGM axiom: $KB_h +_r \epsilon = KB_h \cup \epsilon$.

Update: To use the update operator, we first need to find the models of KB_h and ϵ:

- $ACC_L(KB_h)$:
  $I_1 = \{fuel\_full_0, gen\_running_1, gen\_on_0\}$,
  $I_2 = \{fuel\_full_0, gen\_running_1\}$.
- $ACC_L(\epsilon)$:
  $J_1 = \{gen\_on_0, gen\_running_1, fuel\_full_0\}$,
  $J_2 = \{gen\_on_0, gen\_running_1\}$,
  $J_3 = \{gen\_on_0, gen\_running_1, gen\_running_0, fuel\_full_0\}$,
  $J_4 = \{gen\_on_0, gen\_running_1, gen\_running_0\}$,
  $J_5 = \{gen\_running_1, gen\_running_0, fuel\_full_0\}$,
  $J_6 = \{gen\_running_1, gen\_running_0\}$,
  $J_7 = \{gen\_running_0, fuel\_full_0\}$,
  $J_8 = \{gen\_running_0\}$,
  $J_9 = \{fuel\_full_0\}$,
  $J_{10} = \{\}$.

Now, according to the KM postulates, we need to find the models of ϵ that are closest to I_1 and I_2. Then, the updated KB will be the disjunction of the conjunction of the variables in each model. Now, let the function $\mathit{Diff}(m_1, m_2)$ denote the set of propositional letters with different truth values in models m_1 and m_2.

Then, for I_1, it is easy to see that the closest model is J_1 because $\mathit{Diff}(I_1, J_1) = \emptyset < \mathit{Diff}(I_1, J_k)$ for all k. So, J_1 is selected. For I_2, we need to calculate the difference for every model of ϵ:

$$
\begin{aligned}
&\underline{\mathit{Diff}(I_2, J_1) = \{gen\_on_0\}},\\
&\mathit{Diff}(I_2, J_2) = \{gen\_on_0, fuel\_full_0\},\\
&\mathit{Diff}(I_2, J_3) = \{gen\_on_0, gen\_running_0\},\\
&\mathit{Diff}(I_2, J_4) = \{gen\_on_0, gen\_running_0, fuel\_full_0\},\\
&\underline{\mathit{Diff}(I_2, J_5) = \{gen\_running_0\}},\\
&\mathit{Diff}(I_2, J_6) = \{fuel\_full_0, gen\_running_0\},\\
&\mathit{Diff}(I_2, J_7) = \{gen\_running_0, gen\_running_1\},\\
&\mathit{Diff}(I_2, J_8) = \{gen\_running_0, gen\_running_1, fuel\_full_0\},\\
&\underline{\mathit{Diff}(I_2, J_9) = \{gen\_running_1\}},\\
&\mathit{Diff}(I_2, J_{10}) = \{fuel\_full_0, gen\_running_1\}
\end{aligned}
$$

where sets with the minimal elements are underlined. So, J_1, J_5, and J_9 are selected and the final result is the union of all selected models, that is, $ACC_L(KB_h +_u \epsilon) = \{J_1, J_5, J_9\}$. Thus, the resulting KB must satisfy all three models, yielding the following: $KB_h +_u \epsilon = [\big(gen\_on_0 \wedge gen\_running_1 \wedge fuel\_full_0 \wedge \neg gen\_running_0\big) \vee \big(gen\_running_1 \wedge gen\_running_0 \wedge fuel\_full_0 \wedge \neg gen\_on_0\big) \vee \big(fuel\_full_0 \wedge \neg gen\_running_1 \wedge \neg gen\_on_0 \wedge \neg gen\_running_0\big)]$.

Our Approach: As a first step, our method will first check if KB_h is consistent with the model of KB_a. Since it is, it will simply insert ϵ to KB_h, yielding $\widehat{\mathrm{KB}}_h^{\epsilon} = KB_h \cup \epsilon$ just like revision.

In conclusion, this problem demonstrates that it is possible for *belief revision* to yield the same update as our approach, which is when $KB_h \cup \epsilon$ is consistent (per AGM postulates). It also highlights why *belief update* is not applicable for explainable planning, namely that the updated knowledge base $\mathrm{KB}_h +_u \epsilon$ violates the action dynamics of classical planning problems [135].

Problem 2

Now assume a version of the Generator domain which consists of two actions gen_on = {precondition: $fuel\_full$, effect: $gen\_running$} and gen_on_alt = {precondition: $fuel\_mid$, effect: $gen\_running$} with initial and goal states $fuel\_full$ and $gen\_running$, respectively, and a plan $\pi^* = [gen\_on]$. Also, assume that the human user is not aware that action gen_on has effect $gen\_running$. Then, the knowledge bases encoding the models of the agent and the human are respectively:

$$
\begin{aligned}
KB_a = [&fuel\_full_0, \neg fuel\_mid_0, \neg gen\_running_0, gen\_running_1, && (3.79)\\
&gen\_on_0 \rightarrow fuel\_full_0, gen\_on_0 \rightarrow gen\_running_1, && (3.80)\\
&gen\_on\_alt_0 \rightarrow fuel\_mid_0, gen\_on\_alt_0 \rightarrow gen\_running_1, && (3.81)\\
&\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0 \vee gen\_on\_alt_0, && (3.82)\\
&\neg gen\_on_0 \vee \neg gen\_on\_alt_0] && (3.83)
\end{aligned}
$$

$$
\begin{aligned}
KB_h = [&fuel\_full_0, \neg fuel\_mid_0, \neg gen\_running_0, gen\_running_1, && (3.84)\\
&gen\_on_0 \rightarrow fuel\_full_0, gen\_on\_alt_0 \rightarrow fuel\_mid_0, && (3.85)\\
&gen\_on\_alt_0 \rightarrow gen\_running_1, && (3.86)\\
&\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on\_alt_0, \neg gen\_on_0 \vee \neg gen\_on\_alt_0] && (3.87)
\end{aligned}
$$

As in the previous problem, we now assume that the explanation needed is $\epsilon = [gen\_on_0 \rightarrow gen\_running_1, \neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0 \vee gen\_on\_alt_0]$.

Abductive Explanations: The method of abductive explanations will fail in this setting due to the fact that KB_h is inconsistent. Further, even if KB_h was consistent, we would still not be able to find any abductive explanations due to the lack of causal rules in KB_h.

Revision: Following AGM postulates, revision cannot be applied because KB_h is individually inconsistent.

Update: Again, as KB_h is inconsistent, and according to KM update postulates, it cannot be repaired using update.

Our Approach: As $KB_h \cup \epsilon$ is inconsistent, our approach will identify the erroneous formula $\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on\_alt_0$ and replace it with the corresponding correct formula $\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0 \vee gen\_on\_alt_0$ from KB_a, thereby restoring consistency. The updated knowledge base will be:

$$\widehat{\mathrm{KB}}_h^{\epsilon} = [fuel\_full_0, \neg fuel\_mid_0, \neg gen\_running_0, gen\_running_1, \tag{3.88}$$

$$gen\_on_0 \rightarrow fuel\_full_0, gen\_on_0 \rightarrow gen\_running_1, \tag{3.89}$$

$$gen\_on\_alt_0 \rightarrow fuel\_mid_0, gen\_on\_alt_0 \rightarrow gen\_running_1, \tag{3.90}$$

$$\neg gen\_running_0 \wedge gen\_running_1 \rightarrow gen\_on_0 \vee gen\_on\_alt_0, \tag{3.91}$$

$$\neg gen\_on_0 \vee \neg gen\_on\_alt_0] \tag{3.92}$$

In summary, this problem demonstrates that when KB_h is inconsistent, abductive explanations, revision, and update cannot be applied but our approach can be applied.

Therefore, a key distinction between the previous approaches and our approach is that, historically, belief change refers to a *single agent* revising its belief after receiving a new piece of information that is in conflict with its current beliefs; so, there is a temporal dimension in belief change and a requirement that it should maintain as much as possible the belief of the agent, per AGM postulates. Our notion of explanation is done with respect to *two knowledge bases* and there is no such requirement (with respect to KB_h). For example, if the agent believes that block A is on block B, the human believes that block B is on block A, and the explanation does not remove this fact from the human's KB, then the agent and the human will still have some conflicting knowledge about the positions of blocks A and B after the update. Thus, the previous notions of belief change cannot accurately capture and characterize the MRP problem.

Some Further Discussion

Similar to belief change, explanation differs from other similar notions, such as diagnosis [189]. In general, a diagnosis is defined with respect to a knowledge base KB, a set of components H, and a set of observations O. Given that $\mathrm{KB} \cup O \cup \{\neg ab(c) \mid c \in H\}$ is inconsistent, a diagnosis is a subset S of H such that $\mathrm{KB} \cup O \cup \{ab(c) \mid c \in S\} \cup \{\neg ab(c) \mid c \in H \setminus S\}$ is consistent. Here, $ab(c)$ denotes that the component c is faulty. Generalizing this view, the inconsistency condition could be interpreted as the query q and $\mathrm{KB} \cup O \models^s_L \neg q$. Then a diagnosis is a set $S \subseteq H$ such that $\mathrm{KB} \cup O \cup S \models^s_L q$. An explanation for q from KB_a to KB_h is, on the other hand, a pair (e^+, e^-) such that $(\mathrm{KB}_h \setminus e^-) \cup e^+ \models^s_L q$. Thus, the key difference is that an explanation might require the removal of some knowledge of KB_h while a diagnosis does not.

Another earlier research direction that is closely related to the proposed notion of explanation is that of developing explanation capabilities of knowledge-based systems and decision support systems, which resulted in different notions of explanation such as trace, strategic, deep, or reasoning explanations (see review by Moulin *et al.* [170]). All of these types of explanations focus on answering why certain rules in a knowledge base are used and how a conclusion is derived, which is not our focus in this work. The present development differs from earlier proposals in that explanations are identified with the aim of explaining a given formula to a second theory. Furthermore, the notion of a cost-optimal explanation with respect to the second theory is proposed.

There have been attempts to using argumentation for explanation [46, 48] because of the close relation between argumentation and explanation. For example, argumentation was used by Cyras *et al.* [48] to answer questions such as why a schedule does (does not) satisfy a criteria (e.g., feasibility, efficiency, etc.); the approach was to develop for each type of inquiry, an abstract *argumentation framework* (AF) that helps explain the situation by extracting the attacks (non-attacks) from the corresponding AF.

The problem of restoring consistency in a knowledge base in our framework is similar in spirit to the notion of minimal repairs/diagnoses studied by Ulbricht t [219]. However, they consider an AF as an agent's knowledge base, whereas we consider knowledge bases encoding planning problems. In addition, restoring consistency intersects with the problem of finding *minimally unsatisfiable sets* (MUSes) and *minimal correction sets* (MCSes) [164, 165].

However, most MUS and MCS algorithms are specialized for propositional logic and, to the best of our knowledge, are with respect to a single theory. In contrast, our notion of retracting unsatisfiable sets is with respect to two theories (i.e., KB_a and KB_h).

It is worth to point out that the problem of computing a most preferred explanation for φ from KB_a to KB_h might look similar to the problem of computing a weakest sufficient condition of φ on KB_a under KB_h as described by Lin *et al.* [157]. As it turns out, the two notions are quite different. Given that $\mathrm{KB}_a = \{p, q\}$ and $\mathrm{KB}_h = \{p\}$. It is easy to see that q is the unique explanation for q from KB_a to KB_h. On the other hand, the weakest sufficient condition of q on KB_a under KB_h is $\perp$ (Proposition 8, [157]).

A recent research direction that is closely related to the proposed notion of explanation is that by Shvo *et al.* [201], where they propose a general belief-based framework for generating explanations that employs epistemic state theory to capture the models of the explainer (agent in this work) and the explainee (human user in this work), and incorporates a belief revision operator to assimilate explanations into the explainee's epistemic states. A main difference with our proposed framework is that our framework restricts knowledge to be stored in logical formulae, while theirs considers epistemic states that can characterize different types of problems and have no such restriction.

Finally, in another line of our work, we laid the theoretical foundations and emphasized the knowledge representation aspects of model reconciliation to problems that can be formulated as logic programs [202]. Additionally, we have also developed a dedicated ASP-based solver [172] for solving these problems. These prior work, combined with our current work, reflect that knowledge representation and reasoning can provide a fertile ground for explanation generation in model reconciliation problems and explainable planning.

3.7.2 Planning Literature

As the main theme of this chapter falls under the general umbrella of *explainable AI planning* (XAIP), it is important to provide a general overview of XAIP and discuss current trends as well as situate our contributions within the related work in this area.

While there is some work on adapting planning algorithms to find easily explainable plans (i.e., plans that are easily understood and accepted by a human user) [243], most work has focused on the *explanation generation problem* (i.e., the problem of identifying explanations of plans found by planning agents that, when presented to users, will allow them to understand and accept the proposed plan) [126, 149]. Within this context, researchers have tackled the problem where the model of the human user (1) must be learned [243]; and (2) is of a different form or abstraction than that of the planning agent [218, 210]. However, when designing explanatory planning systems, one of the main considerations is taking into account the personality of the explainee [150]. Currently in the literature, there are three personalities considered:

- **Domain designer:** The person working in acquiring the model that the system works with;
- **Algorithm designer:** The developer of the algorithms of the planning system; and
- **End user:** The person interacting/collaborating with the system in the form of a user.

Naturally, these different personas will require different kinds of explanations. These explanations fall under two primary classes of explanations: *Algorithm-based explanations* and *model-based explanations* (see the survey by Chakraborti *et al.* [36] for a comprehensive discussion).

Algorithm-based explanations generally target expert users (i.e., algorithm designers) and attempt to explain the inner workings of the underlying planning algorithm. For example, Magnaguagno *et al.* [163] developed a state-space search visualization that represents how the distance to the goal state changes during the search procedure by defining a heuristic gradient using heat maps. The gradient colors the states based on their estimated distance to the goal state and can be used to highlight that the estimated distance can be different according to the heuristic used in the algorithm. One interesting aspect of this work is that it can visualize failed planning instances, which can be practical for debugging purposes.

In contrast, model-based explanations are considered in a very large number of XAIP papers. This category consists of algorithm-agnostic explanations that can be evaluated independently of the method used to come up with. For instance, Borgo *et al.* [23] and Cashmore *et al.* [28] created a service that allows users to hold a dialogue with the system by means of specifying contrastive questions about the plan. Essentially, they assume questions specified by users can be best understood as constraints on the plans they are expecting (i.e., a certain action to be included or excluded in the plan). The explanation is then to identify an exemplary plan that satisfies those constraints and, thus, demonstrating how the computed plan is better. Such kinds of explanations fall under the broad umbrella of *contrastive explanations*, where explanations answer questions of the form "Why not A (instead of B)?", where A is an alternative (or foil) suggested by the human to decision B taken by the agent. An explanation can explain why A is suboptimal or why the agent's decisions are better than the foil. There are multiple forms this contrastive question can take, i.e., having the user present an entire plan or specific actions as foils, and it can also take on the more general form of "Why B?", where the implicit comparison is to all possible alternatives. Contrastive explanations have been gathering interest in literature, with applications in linear temporal logic systems to answer factual questions [132], and in oversubscription planning to explain goal subsets [68] being recent examples. There has also been work towards user interfaces for decision support that gives users suggestions in response to foils provided to AI generated plans in an interactive manner [131].

On a similar thread, Gobelbecker *et al.* [94] have considered the case of explaining why a planning problem is unsolvable. In particular, they transform an unsolvable planning problem Π into a new problem Π' by adding predicates or objects to initial states, which they refer to as *excuses*, such that Π' becomes solvable. It is interesting to note that a logic-based method, like our proposed framework, could also be used to explain the unsolvability of planning problems; Notice that an unsolvable planning problem Π translates to an unsatisfiable knowledge base KB encoding Π. Specifically, for a given time horizon h, KB is unsatisfiable if and only if there exists no plan of length h. As such, one could find the reason for the unsatisfiability of KB by computing a *minimal unsatisfiable set* (MUS) [117] over KB, which would then serve as an explanation. Additionally, one could also compute a *maximal satisfiable set* (MSS) [117] over the KB in order to find potential subproblems of Π that are solvable, and which may provide useful information for Π. We leave this interesting problem for future work.

Nevertheless, the explanations considered above do not capture directly the user's knowledge of the given planning problem and are thus not a realistic inception of a true explanatory system targeting non-expert human users. It is widely accepted that human users often come with their own preconceived notions and/or expectations of the system [27] and, as such, human users might evaluate plans on their own models, which may disagree with the system's outcome or quality.

A key paper that considers the mental models of human users is by Chakraborti *et al.* [37], who introduced the *model reconciliation problem* that we are tackling in this paper. The high-level difference between our two approaches is that our approach is based on KR while theirs is based on automated planning and heuristic search techniques. However, both approaches share a lot of similarities. Particularly, explanations generated in both approaches can be characterized according to two properties defined in [37]: *Completeness*, that is, the plan is valid (or optimal) in the updated human user's model; and *Monotonicity*, that is, there are no model differences in the agent's and human user's models that can change the completeness of an explantion (i.e., no further model updates can invalidate an explanation). In consequence, both approaches share similar types of explanations that can be found. For example, the $\subseteq$-minimal support in Definition 5 is equivalent to *minimally complete explanations* (MCEs) (the shortest explanation that is complete), while the $\lhd$-general support can be viewed as similar to the *minimally monotonic explanations* (MMEs) (the shortest explanation that is complete and monotonic). Additionally, *model patch explanations* (MPEs) (includes all the model updates) are trivial explanations and are equivalent to our definition that the entire KB_a itself serves as an explanation for KB_h. Note that, in our approach (as also in the original model reconciliation problem), we allow for explanations on "mistaken" expectations in the human model (Algorithm 2). However, a similar property can be seen if the mental model is not known and, therefore, by taking an "empty" model as the starting point, explanations can only add to the human's understanding but not mend mistaken ones.

Although the model reconciliation problem is a good stepping stone towards creating good explanatory planning systems, it makes a strong assumption that the system has knowledge of the human user's mental model. An alternative approach that relaxes this requirement is called *model-free model reconciliation*, which predicts how model information can affect the expectation of the human user by learning a model that characterizes the human user's expectation and using it to drive the search to determine what information should be exposed to the human user [208]. However, there might be caveats in going model-free. The explanations generated by such systems might be purposely false in order to satisfy the human user. For example, in the model reconciliation problem the explanation was guaranteed to be consistent with the ground truth. Researchers have showed that this guarantee can be relaxed in such a way that allows the model reconciliation process to generate erroneous explanations and, hence, create ethical quandaries that would need further investigation [33, 34].

Another popular theme in XAIP is that of *plan summarization*, where the main interest is in presenting a long plan to a single human user [171], or to multiple human users (e.g., human teams) [140]. One possible way to approach this would be to use the model reconciliation process with an empty model of the human user and compute the minimal explanation (e.g., causal links) necessary to explain every action in the plan. Another possibility would be to approach this issue through a process called *verbalization of plans*, that is, paths taken by an agent along different levels of abstraction [195]. Interestingly, there have also been efforts on approaching plan summarization with contrastive explanations. For example, Kim *et al.* [141] proposed on a Bayesian approach to infer contrastive linear temporal logic specifications aimed at explaining how two sets of plan traces differ.

Finally, it is worth mentioning that our plan validity check bears some similarity with the validity check that is provided by VAL [113]. The key difference is that, in case of an invalid plan, VAL identifies the first action in the plan with unsatisfied preconditions, and identifies the precondition that is not satisfied. In contrast, the validity check that is provided here identifies all the differences in the model that prevents the plan to be valid.

3.8 Concluding Remarks

The design of explanatory systems often raises fundamental questions about the identification, representation, and provision of explanations. Logic-based systems have long been considered well-equipped to serve as an explainability layer for AI systems, with examples such as decision trees producing explanations directly from their logical models [148, 116]. This chapter has examined and evaluated this potential through the lens of the Model Reconciliation Problem (MRP) in explainable AI planning (XAIP), demonstrating how logic can serve as an effective explainability layer for planning systems.

We introduced the *Logic-based Model Reconciliation Problem* (L-MRP), a novel framework that extends the applicability of MRP beyond classical planning to hybrid planning scenarios. L-MRP addresses situations where an AI agent's plan is inexplicable to a human user due to discrepancies in their respective models of the problem, using logic as an explainability layer to bridge this gap. Our key contributions are

1. We reformulated model reconciliation from a knowledge representation and reasoning perspective, defining the notion of logic-based explanations explanations for plan validity and optimality.

2. We proposed complexity cost functions to capture preferences between explanations, allowing for preferred explanation generation.

3. We developed algorithms to compute explanations for both classical and hybrid systems planning problems.

4. We empirically demonstrated that our L-MRP approach complements and extends the current state of the art, generalizing beyond classical planning to hybrid planning scenarios. Specifically, our empirical results demonstrate that, on classical planning problems, our approach is faster than the state of the art when the explanations are long or when the size of the knowledge base is small (e.g., the plans to be explained are short). They also demonstrate that our approach is efficient for hybrid systems planning problems.

5. Through human-subject studies, we demonstrated the real-world efficacy of explanations as model reconciliation for planning problems beyond classical planning.

The L-MRP framework presented in this chapter advances both the theoretical foundations of XAIP and our broader thesis about logic serving as an explainability layer for AI systems. L-MRP offers a general approach that can adapt to various planning scenarios while preserving the underlying strengths of different planning systems. Importantly, while our focus in this chapter was on planning problems, L-MRP's applicability as an explainability layer can extend beyond planning to various problem types, so long as these problems can be encoded in a logical formalism for which satisfiability of sets can be checked. The following chapter will illustrate this generality of L-MRP by exploring its application to probabilistic settings.

Chapter 4

Explanation Generation under Uncertainty

> *"We may have knowledge of the past but cannot control it; we may control the future but have no knowledge of it."*
>
> — *Claude Shannon*

4.1 Introduction & Contribution

Previously, we demonstrated how logic can serve as an explainability layer for deterministic planning scenarios through the Logic-based Model Reconciliation Problem (L-MRP). While L-MRP showed the effectiveness of logic-based methods in generating explanations, it relies on assumptions of deterministic knowledge that may not hold in many real-world applications. As AI systems are increasingly deployed in complex environments, they often must make decisions based on incomplete or uncertain information, requiring us to extend our framework to handle uncertainty.

The motivation for extending our work to handle uncertainty stems from the following observations:

- **Inherent Uncertainty in Real-World Scenarios:** Many real-world decision-making processes involve uncertainty. For instance, in medical diagnosis, a doctor's knowledge base includes probabilistic relationships between symptoms and diseases. Traditional deterministic explanations, including those generated by L-MRP, may fail to capture the nuanced reasoning required in such scenarios.

- **Limitations of Deterministic Model Reconciliation:** While our previous work on L-MRP provided valuable insights, it assumed that the AI agent had perfect knowledge of the human's model. In practice, an AI system's understanding of human knowledge is often incomplete or uncertain, necessitating a more flexible approach to explanation generation.
- **Need for Explanations in Single-Agent Scenarios:** Not all explanation scenarios involve model reconciliation between two agents. In many cases, an AI system needs to explain its reasoning based on its own uncertain knowledge. This motivates our exploration of probabilistic *monolithic explanations*, which can provide insights into an agent's decision-making process under uncertainty.

These observations indicate several challenges in generating explanations under uncertainty. First, there is the issue of intrinsic uncertainty, where AI agents often operate with probabilistic knowledge bases rather than deterministic rules. This uncertainty in the agent's knowledge adds complexity to the process of generating effective explanations. Second, the presence of multiple competing hypotheses presents a challenge. Explanations may need to consider alternative hypotheses and their relative likelihoods, requiring a more comprehensive approach to explanation generation. Third, the quantification of uncertainty becomes important in these scenarios. Explanations should provide ways to express uncertainty through quantitative or qualitative measures, allowing users to understand the degree of confidence associated with different aspects of the explanation. Lastly, in model reconciliation scenarios, we encounter the challenge of uncertain human models. The AI agent's understanding of human knowledge may itself be uncertain, adding another factor to consider in the task of generating effective explanations. These challenges collectively suggest the need for refined approaches to explanation generation in probabilistic environments.

To address these challenges, this chapter considers two distinct types of explanations and makes the following contributions:

1. **Probabilistic Monolithic Explanations:** We present a framework for generating explanations from a single, probabilistic knowledge base. This approach is suited for scenarios where an AI system needs to explain its own reasoning without reference to a human model. Given a probabilistic knowledge base $\mathcal{B}$ and an explanandum φ,

we aim to find explanations that increase the probability of φ being true. We introduce the concepts of *explanatory gain* and *explanatory power* to quantify explanation effectiveness.

2. **Probabilistic Model Reconciling Explanations:** We extend L-MRP to scenarios with uncertain human models, addressing situations where explanations need to bridge the gap between agent and human knowledge. Given an agent's knowledge base $\mathcal{KB}_a$, an explanandum φ entailed by $\mathcal{KB}_a$, and a human probabilistic knowledge base $\mathcal{B}_h$, we develop methods to find explanations that increase φ's probability for $\mathcal{B}_h$ while minimizing conflicts between the explanation and $\mathcal{B}_h$.

By addressing both monolithic and model reconciling explanations in probabilistic settings, our framework provides a comprehensive and flexible approach to explainability under uncertainty. This work aims to contribute to the development of AI systems capable of providing explanations in domains characterized by uncertainty, whether explaining their own reasoning or reconciling their knowledge with that of human users.

4.2 Essential Background

We will adopt a propositional language $\mathcal{L}$ built from a finite set of atomic variables $\mathcal{V} = \{a, b, c, \ldots\}$. A *possible world* is a truth-value assignment to each variable $\omega : \mathcal{V} \mapsto \{T, F\}$, where T and F denote truth and falsity respectively. The set of all possible worlds of $\mathcal{L}$ is denoted by Ω. The simplest formulae in $\mathcal{L}$ are atoms: Individual variables that may be true or false in a given possible world. More complex formulae are recursively constructed from atoms using the classical logical connectives. A *model* of a formula is a possible world in which the formula is satisfied (i.e., evaluates to true). A knowledge base KB is a set of formulae. If there exists at least one possible world ω that satisfies all formulae in KB, then KB is *consistent*, otherwise we say that KB is *inconsistent*. We use $\models$ to denote the classical entailment relation and say that a (consistent) KB entails a formula φ, expressed as KB $\models \varphi$, if and only if every model of KB is also a model of φ, or equivalently, if KB $\cup \{\neg\varphi\}$

is inconsistent. Unless stated otherwise, it is assumed that all formulae are expressed in *conjunctive normal form* (CNF).[28]

Given a knowledge base KB and a formula φ, called the *explanandum* such that $\mathrm{KB} \models \varphi$, we define a *monolithic explanation* for φ from KB as a minimal set of formulae that entails φ:

Definition 18. *(Monolithic Explanation) Let* KB *be a knowledge base and* φ *an explanandum such that* $\mathrm{KB} \models \varphi$*. We say that* $\epsilon \subseteq \mathrm{KB}$ *is a monolithic explanation for* φ *from* KB *if and only if: (i)* $\epsilon \models \varphi$*; and (ii)* $\nexists \epsilon' \subset \epsilon$ *such that* $\epsilon' \models \varphi$*.*[29]

Example 6. *Consider the knowledge base* $\mathrm{KB} = \{p, \neg p \vee q, \neg p \vee r\}$ *build up from* $\mathcal{V} = \{p, q, r\}$*. Notice that* $\mathrm{KB} \models q$*. Then,* $\epsilon = \{p, \neg p \vee q\}$ *is a monolithic explanation for* q *from* KB*.*

Note that in this work we do not consider formulae $\epsilon \equiv \varphi$ as monolithic explanations. These trivial explanations, which are of the form *"why φ, because φ"*, are uninformative pertaining the explanandum.

Building upon the foundation laid in Chapter 3, we define a *model reconciling explanation*, which takes into account both the knowledge base KB_α of the agent providing an explanation as well as the knowledge base KB_h of the human receiving the explanation:

Definition 19 (Model Reconciling Explanation)**.** *Given the knowledge bases of an agent* KB_α *and a human user* KB_h *as well as an explanandum* φ*, such that* $\mathrm{KB}_\alpha \models \varphi$ *and* $\mathrm{KB}_h \not\models \varphi$*,* $\mathcal{E} = \langle \epsilon^+, \epsilon^- \rangle$ *is a model reconciling explanation if and only if* $\epsilon^+ \subseteq \mathrm{KB}_\alpha$*,* $\epsilon^- \subseteq \mathrm{KB}_h$*, and* $(\mathrm{KB}_h \cup \epsilon^+) \setminus \epsilon^- \models \varphi$*.*

When KB_h is *updated* with a model reconciling explanation $\mathcal{E} = \langle \epsilon^+, \epsilon^- \rangle$, new formulae ϵ^+ from KB_α are added to KB_h and formulae ϵ^- from KB_h are retracted to ensure consistency. Note that since a model reconciling explanation is from the perspective of the agent's knowledge base KB_α, we implicitly assume that if a formula in KB_h is inconsistent with KB_α, then that formula is "false" from the perspective of the agent.

[28] A CNF formula is a conjunction of clauses, where each clause is a disjunction of literals. A literal is either an atom or its negation. This is not a restrictive requirement, since any propositional formula can be transformed into a CNF representation.

[29] This definition is similar to the definition of support defined in the previous chapter.

Example 7. *Let* $\mathrm{KB}_\alpha = \{a, \neg a \vee b, \neg a \vee c\}$ *and* $\mathrm{KB}_h = \{\neg a, \neg a \vee b\}$ *be the knowledge bases of an agent and a human user, respectively, where* $\mathrm{KB}_\alpha \models b$ *and* $\mathrm{KB}_h \not\models b$*. A model reconciling explanation is then* $\mathcal{E} = \langle \{a\}, \{\neg a\} \rangle$*, where* $(\mathrm{KB}_h \cup \{a\}) \setminus \{\neg a\} = \{a, \neg a \vee b\} \models b$*.*

Modeling Uncertainty in Propositional Logic

Building on a propositional language $\mathcal{L}$, we can model the uncertainty of propositional formulae using a *probability distribution* over the possible worlds Ω of $\mathcal{L}$. Formally,

Definition 20 (Probability Distribution)**.** *Let* Ω *be the set of possible worlds of the language* $\mathcal{L}$*. A probability distribution* P *on* Ω *is a function* $P : \Omega \mapsto [0, 1]$ *such that* $\sum_{\omega \in \Omega} P(\omega) = 1$*.*

In essence, a probability distribution over possible worlds creates a *ranking* between those worlds with respect to how likely they are to be true. This then allows us to quantify the uncertainty in a formula as follows:

Definition 21 (Degree of Belief)**.** *Let* Ω *be the set of possible worlds and* P *a probability distribution over* Ω*. The degree of belief in a formula* $\varphi \in \mathcal{L}$ *is* $P(\varphi) = \sum_{\omega \models \varphi} P(\omega)$*.*

We may refer to $P(\varphi)$ as degree of belief or probability of φ interchangeably. Note that the possible worlds approach to probabilities is essentially equivalent to probabilities assigned directly to the formulae [6].

Now, the probability distribution on Ω can be induced from a weighted knowledge base, referred to as a *belief base*:

$$\mathcal{B} = \{(\phi_1, w_1), \dots, (\phi_n, w_n)\} \tag{4.1}$$

where each formula $\phi_i \in \mathcal{L}$ is associated with a weight $w_i \in \mathbb{R}^+$.[30]

Intuitively, the weights serve as meta-information and reflect the certainty about the truth of the corresponding formulae – the higher the weight, the more certain the formula is. In that sense, formulae with higher weights are prioritized for satisfaction, effectively capturing the certainty of the particular formulae. This mechanism is especially useful for handling

[30] We assume, without loss of generality, that all weights are non-negative because a formula with a negative weight w can be replaced by its negation with weight $-w$.

inconsistency and non-monotonic reasoning patterns, thus capturing a broader spectrum of problems.[31] Further, we will denote with $\mathcal{B}^{\downarrow w}$ the *classical projection* of $\mathcal{B}$, that is, $\mathcal{B}^{\downarrow w} = \{\phi_i \mid (\phi_i, w_i) \in \mathcal{B}\}$.

Given a belief base $\mathcal{B}$, one way to induce a probability distribution is the following:

$$\forall \omega \in \Omega,\ P_{\mathcal{B}}(\omega) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} w_i \cdot \mathbb{I}(\omega, \phi_i)\right) \tag{4.2}$$

where $\mathbb{I}(\omega, \phi) = 1$ if $\omega \models \phi$ and 0 otherwise, and $Z = \sum_{\omega \in \Omega} \exp\left(\sum_{i=1}^{n} w_i \cdot \mathbb{I}(\omega, \phi_i)\right)$ is the normalization factor.

The induced probability distribution quantifies the likelihood that a given (possible) world is the actual world. Higher formula weights amplify the (log-) probability difference between a world that satisfies the formula and one that does not, other things being equal. Consequently, worlds that violate fewer formulas are deemed more probable. Note that a belief base $\mathcal{B}$ is essentially a *log-linear model* [15], from which a *joint probability distribution* of the set of variables of $\mathcal{L}$ is induced. Interestingly, log-linear models are special cases of Markov Logic Networks and can represent any positive distribution [192]. When taken from context, we will simply use P to denote the distribution induced from $\mathcal{B}$.

Entailment in a belief base KB becomes *graded*, that is, we now say that $\mathcal{B}$ entails a formula ϕ with degree of belief $P(\phi)$. However, when all weights are equal and tend to infinity, a belief base represents a uniform distribution over the worlds that satisfy it and, as such, entailment of a formula can be answered by computing the probability of the formula and checking whether it is 1. In other words, entailment under belief bases collapses to classical entailment under knowledge bases. See [192] for a proof.

Finally, the weighted formulae in a belief base $\mathcal{B}$ can be viewed as *soft* constraints, i.e., formulae that need not to be satisfied. In contrast, *hard* constraints can be imposed as formulae with "infinite" weights.[32]

[31] For example, the notion of inconsistency is relaxed as follows: Given two inconsistent formulae ϕ and $\neg\phi$, if $P(\phi) = 0.9$, then from the axioms of probability we have that $P(\neg\phi) = 0.1$. This then means that the worlds where $\neg\phi$ is true are more unlikely than the worlds where ϕ is true, but not impossible.

[32] In practice, infinite weights can be replaced with $\sum_{i=1}^{n} w_i + 1$.

4.3 A Framework for Probabilistic Explanation Generation

In this section, we outline a framework designed to extend the classical concepts of monolithic explanation, as defined by Definition 18, and model reconciling explanation, as defined by Definition 19, into probabilistic contexts.

4.3.1 Probabilistic Monolithic Explanations

Building on the classical notion of monolithic explanation presented in Definition 18, we introduce the concept of a *probabilistic monolithic explanation.* This concept aims to account for the uncertainty inherent in knowledge bases, providing a framework for explanations that not only identify contributing factors for an explanandum but also quantify the uncertainty in these factors. Throughout this section, we assume a belief base $\mathcal{B}$ and its induced probability distribution P.

Formally, a probabilistic monolithic explanation for an explanandum φ from belief base $\mathcal{B}$ is defined as follows:

Definition 22 (Probabilistic Monolithic Explanation)**.** *Let $\mathcal{B}$ be a belief base, $\mathcal{B}^{\downarrow w}$ its classical projection, and φ an explanandum. We say that $\tilde{\epsilon} \subseteq \mathcal{B}^{\downarrow w}$ is a probabilistic monolithic explanation for φ from $\mathcal{B}$ if and only if $P(\varphi \mid \tilde{\epsilon}) > P(\varphi)$.*

Intuitively, a probabilistic monolithic explanation $\tilde{\epsilon}$ seeks to increase the degree of belief in the explanandum φ. If $P(\varphi \mid \tilde{\epsilon}) > P(\varphi)$, this then represents the case where $\tilde{\epsilon}$ increases the degree of belief in φ and the greater the value of $P(\varphi \mid \tilde{\epsilon})$ the greater the degree of belief in φ.

Example 8. *Consider the belief base $\mathcal{B} = \{(a, 1), (\neg a \vee b, 2)\}$ and the explanandum b. The probability of the explanandum is $P(b) = 0.73$. Then, $\tilde{\epsilon}_1 = \{a\}$ and $\tilde{\epsilon}_2 = \{\neg a \vee b\}$ are two probabilistic monolithic explanations for b from $\mathcal{B}$, that is, $P(b \mid \tilde{\epsilon}_1) = 0.88 > P(b)$ and $P(b \mid \tilde{\epsilon}_2) = 0.78 > P(b)$.*

It is important to note that Definition 22 can be extended to the case where the formulae $\tilde{\epsilon}$ do not necessarily come from $\mathcal{B}$, but rather from the language $\mathcal{L}$. However, we restrict our attention only to formulae from $\mathcal{B}$ in order to be compatible with the classical notion of monolithic explanations (see Definition 18) and the algorithms that we will present in Chapter 5. For brevity, and until the end of this section, we will refer to probabilistic monolithic explanations as monolithic explanations.

Looking at Example 8, we can see that monolithic explanations will typically vary in their capacity to increase the degree of belief in the explanandum. In other words, each monolithic explanation provides us with some *explanatory gain* for the explanandum. Following Good [95, 96], explanatory gain is defined as follows:[33]

Definition 23 (Explanatory Gain of Monolithic Explanations)**.** *Let $\tilde{\epsilon}$ be a monolithic explanation for explanandum φ from belief base $\mathcal{B}$. The explanatory gain of $\tilde{\epsilon}$ for φ is defined as $G(\tilde{\epsilon}, \varphi) = \log\left(\frac{P(\varphi \mid \tilde{\epsilon})}{P(\varphi)}\right)$.*[34,35]

In essence, the explanatory gain can be thought of as a measure that quantifies how well the monolithic explanation $\tilde{\epsilon}$ *explains* the explanandum φ or, equivalently, the degree to which $\tilde{\epsilon}$ entails φ. The greater the value of $G(\tilde{\epsilon}, \varphi)$, the more substantial the explanatory gain and, hence, the more effective $\tilde{\epsilon}$ is at explaining φ.

It is essential to recognize that this measure, while initially introduced to assess the weak explanatory power of hypotheses in light of evidence [95], it is used here to evaluate monolithic explanations. By quantifying the extent to which a monolithic explanation explains an explanandum, we can systematically identify the most informative monolithic explanations within a probabilistic framework.

Example 9. *Continuing from Example 8, consider the monolithic explanations $\tilde{\epsilon}_1 = \{a\}$, $\tilde{\epsilon}_2 = \{\neg a \vee b\}$, and $\tilde{\epsilon}_3 = \{a, \neg a \vee b\}$ for explanandum b. The explanatory gains of $\tilde{\epsilon}_1$, $\tilde{\epsilon}_2$, and $\tilde{\epsilon}_3$ for b are $G(\tilde{\epsilon}_1, b) = \log\left(\frac{P(b \mid \tilde{\epsilon}_1)}{P(b)}\right) = \log\left(\frac{0.88}{0.73}\right) = 0.27$, $G(\tilde{\epsilon}_2, b) = \log\left(\frac{P(b \mid \tilde{\epsilon}_2)}{P(b)}\right) = \log\left(\frac{0.78}{0.73}\right) = 0.11$, and $G(\tilde{\epsilon}_3, b) = \log\left(\frac{P(b \mid \tilde{\epsilon}_3)}{P(b)}\right) = \log\left(\frac{1}{0.73}\right) = 0.45$, respectively.*

Now, a natural course of action when seeking monolithic explanations for an explanandum is to seek the one with the highest explanatory gain. While it is tempting to do this, it is

[33] Good [95] originally introduced this measure to quantify the *(weak) explanatory power* of a hypothesis with respect to evidence, essentially evaluating how effectively the hypothesis explains the evidence.

[34] We use log with base 2 in our calculations.

[35] Note that $G(\tilde{\epsilon}, \varphi)$ is always positive due to the requirement of monolithic explanations that $P(\varphi \mid \tilde{\epsilon}) > P(\varphi)$ (Definition 22).

important to emphasize that when a monolithic explanation entails the explanandum, then the explanatory gain takes on its greatest value. For example,

Example 10. *Consider the three monolithic explanations $\tilde{\epsilon}_1$, $\tilde{\epsilon}_2$, and $\tilde{\epsilon}_3$ from Example 9. Notice that $\tilde{\epsilon}_3 = \{a, \neg a \vee b\}$ entails b (i.e., $\tilde{\epsilon}_3 \models b$) and that its explanatory gain is higher than that of $\tilde{\epsilon}_1$ and $\tilde{\epsilon}_2$. As $\tilde{\epsilon}_1$, $\tilde{\epsilon}_2$, and $\tilde{\epsilon}_3$ are the only three possible explanations for b, $G(\tilde{\epsilon}_3, b)$ is indeed the maximum achievable explanatory gain for b.*

We formalize this in the following proposition:

Proposition 2. *Given a monolithic explanation $\tilde{\epsilon}$ for an explanandum φ from belief base $\mathcal{B}$, if $\tilde{\epsilon} \models \varphi$, then $G(\tilde{\epsilon}, \varphi)$ achieves its maximal value for φ, specifically $G(\tilde{\epsilon}, \varphi) = -\log P(\varphi)$.*

Proof. If $\tilde{\epsilon} \models \varphi$, then for all possible worlds ω in which $\omega \models \tilde{\epsilon}$, it holds that $\omega \models \varphi$. That is, the worlds ω in which $\tilde{\epsilon}$ is true are subsumed by the worlds in which φ is true, which implies that also $\omega \models \varphi \wedge \tilde{\epsilon}$. Consequently, $P(\varphi|\tilde{\epsilon}) = \frac{\sum\limits_{\omega \models \varphi \wedge \tilde{\epsilon}} P(\omega)}{\sum\limits_{\omega \models \tilde{\epsilon}} P(\omega)} = \frac{\sum\limits_{\omega \models \tilde{\epsilon}} P(\omega)}{\sum\limits_{\omega \models \tilde{\epsilon}} P(\omega)} = 1$. Therefore, when $\tilde{\epsilon} \models \varphi$, the explanatory gain of $\tilde{\epsilon}$ for φ is $G(\tilde{\epsilon}, \varphi) = \log\left(\frac{P(\varphi \mid \tilde{\epsilon})}{P(\varphi)}\right) = \log\left(\frac{1}{P(\varphi)}\right) = -\log P(\varphi)$. $\square$

The following corollary follows naturally from Proposition 2:

Corollary 1. *Let $\tilde{E}(\varphi)$ denote the set of all monolithic explanations for explanandum φ from belief base $\mathcal{B}$. For any two monolithic explanations $\tilde{\epsilon}_1, \tilde{\epsilon}_2 \in \tilde{E}(\varphi)$, if $\tilde{\epsilon}_1 \models \varphi$ and $\tilde{\epsilon}_2 \models \varphi$ (resp. $\tilde{\epsilon}_2 \not\models \varphi$), then $G(\tilde{\epsilon}_1, \varphi) = G(\tilde{\epsilon}_2, \varphi)$ (resp. $G(\tilde{\epsilon}_1, \varphi) > G(\tilde{\epsilon}_2, \varphi)$).*

What Proposition 2 and Corollary 1 essentially underscore is that the exclusive focus on explanatory gain as an evaluation metric for a monolithic explanation neglects the inherent likelihood of the explanation itself. That is, the explanatory gain of a monolithic explanation for an explanandum evaluates how effectively the explanation explains the explanandum, *assuming that the explanation itself is true*. Nonetheless, this premise often lacks practical relevance because, in probabilistic contexts, each monolithic explanation is associated with a probability reflecting its likelihood for being true. Therefore, a good measure for evaluating monolithic explanations should incorporate the explanation's inherent plausibility.

Addressing this gap, Good [96] introduced the concept of *(strong) explanatory power* that integrates the monolithic explanation's explanatory gain with its probability, offering a more balanced metric for evaluating monolithic explanations.[36] Building on Good's measure of explanatory power, we adapt it to our setting and define it as follows:[37]

Definition 24 (Explanatory Power of Monolithic Explanations)**.** *Let $\tilde{\epsilon}$ be a monolithic explanation for explanandum φ from belief base $\mathcal{B}$. The explanatory power of $\tilde{\epsilon}$ for φ is defined as* $\mathrm{EP}(\tilde{\epsilon}, \varphi) = G(\tilde{\epsilon}, \varphi) + \gamma \cdot P(\tilde{\epsilon})$*, where $\gamma \in [0, 1]$ is a constant.*

This definition effectively combines the measure of how much a monolithic explanation explains the explanandum (explanatory gain) with the likelihood of the explanation itself, mediated by a parameter γ. The constant γ serves as a tuning parameter, enabling the adjustment of the relative importance of the monolithic explanation's probability in the overall assessment of explanatory power. This flexibility is important for tailoring the evaluation process to specific contexts or preferences, where the balance between the informativeness of a monolithic explanation and its plausibility may vary.

Example 11. *Consider the belief base $\mathcal{B} = \{(a, 1.5), (b, 3), (\neg a \vee c, 1), (\neg b \vee c, 1)\}$ and the explanandum c with initial probability $P(c) = 0.84$. Notice that $\tilde{\epsilon}_1 = \{a, \neg a \vee c\}$ and $\tilde{\epsilon}_2 = \{b, \neg b \vee c\}$ are two monolithic explanations for c from $\mathcal{B}$, each of which entail c (i.e., $\tilde{\epsilon}_1 \models c$ and $\tilde{\epsilon}_2 \models c$), with probabilities $P(\tilde{\epsilon}_1) = 0.68$ and $P(\tilde{\epsilon}_2) = 0.80$, respectively. This means that their explanatory gain for c is equal (Corollary 1), that is, $G(\tilde{\epsilon}_1, c) = G(\tilde{\epsilon}_2, c) = 0.25$. Now, assuming $\gamma = 0.5$, the explanatory power of $\tilde{\epsilon}_1$ and $\tilde{\epsilon}_2$ respectively is* $\mathrm{EP}(\tilde{\epsilon}_1, c) = 0.25 + 0.5 \cdot 0.68 = 0.59$ *and* $\mathrm{EP}(\tilde{\epsilon}_2, c) = 0.25 + 0.5 \cdot 0.80 = 0.65$*.*

With the introduction of explanatory power as an evaluative measure of (probabilistic) monolithic explanations, we can now define a (probabilistic) *preference relation* among monolithic explanations, which allows for a systematic approach to determining the most effective monolithic explanation for a given explanandum:

Definition 25 (Preference Relation for Monolithic Explanation)**.** *Let $\tilde{\epsilon}_1$ and $\tilde{\epsilon}_2$ be two monolithic explanations for explanandum φ from belief base $\mathcal{B}$. $\tilde{\epsilon}_1$ is preferred over $\tilde{\epsilon}_2$, denoted as $\tilde{\epsilon}_1 \succeq \tilde{\epsilon}_2$, if and only if* $\mathrm{EP}(\tilde{\epsilon}_1, \varphi) \geq \mathrm{EP}(\tilde{\epsilon}_2, \varphi)$*.*

[36] Good's measure of (strong) explanatory power is defined as $\log\left(\frac{P(\varphi \mid h) \cdot P(h)^{\gamma}}{P(\varphi)}\right)$, where h is a hypothesis and $0 < \gamma < 1$ a constant [96].

[37] For a detailed defense of Good's measure as a quantitative criterion for explanatory power, alongside a discussion of relevant properties and a comprehensive comparison with other measures, we refer the reader to the work by Glass [93].

This definition enables a quantitatively grounded approach to preference among monolithic explanations, where the preference is directly tied to the explanatory power of each explanation. It facilitates a structured way to navigate the space of potential monolithic explanations, prioritizing those that not only explain the explanandum more effectively, but also align more closely with the existing knowledge represented by the belief base $\mathcal{B}$.

Example 12. *Continuing from Example 11, the two monolithic explanations for c from $\mathcal{B}$ are $\tilde{\epsilon}_1$ and $\tilde{\epsilon}_2$ and have explanatory power* $\mathrm{EP}(\tilde{\epsilon}_1, c) = 0.59$ *and* $\mathrm{EP}(\tilde{\epsilon}_2, c) = 0.65$. *Thus, $\tilde{\epsilon}_2$ is preferred over $\tilde{\epsilon}_1$ (i.e., $\tilde{\epsilon}_2 \succeq \tilde{\epsilon}_1$).*

Finally, given the set of all monolithic explanations for an explanandum, we say that a monolithic explanation is *most preferred* if and only if it is (probabilistically) preferred over every other possible monolithic explanation for that explanandum. Formally,

Definition 26 (Most-Preferred Monolithic Explanation)**.** *Let $\tilde{E}(\varphi)$ denote the set of all monolithic explanations for explanandum φ from belief base $\mathcal{B}$. A monolithic explanation $\tilde{\epsilon}^* \in \tilde{E}(\varphi)$ is the most-preferred monolithic explanation if and only if $\tilde{\epsilon}^* \succeq \tilde{\epsilon}$ for all $\tilde{\epsilon} \in \tilde{E}(\varphi)$.*

In the next section, we consider how probabilistic monolithic explanations will look like for the model reconciliation problem.

4.3.2 Probabilistic Model Reconciling Explanations

Recall that, in the model reconciliation problem (MRP), the models of the agent and the human user diverge with respect to an explanandum, insofar as the explanandum is explicable in the agent's model but inexplicable in the human's model. The goal is then to find a model reconciling explanation (i.e., a set of model differences) such that the explanandum becomes explicable in the human's model. Three important assumptions underlying MRP typically hold: (1) the agent model is the ground truth; (2) the agent has access to the human model; and (3) both models are deterministic.

Generally, assumption (1) is reasonable since explanations are generated from the agent's perspective. In other words, the agent "thinks" that its model is correct. For assumption (2), the agent does not have access to the human's actual model, but an approximation of it. In the worst case, it can be empty; but, practically, it can be approximated based on past interactions [205, 124]. For assumption (3), we will relax the assumption that the human model is deterministic in our work, but we will still assume that the agent model is deterministic.

The motivation for moving away from deterministic human models becomes stronger when we consider two key points. First, since the agent is using an approximated human model, deterministic approximations are more likely to be inaccurate compared to probabilistic ones. Consequently, deterministic models may generate explanations that are incorrect or not meaningful for the user, thereby reducing the effectiveness of MRP. Secondly, it is likely that humans hold beliefs with varying degrees of certainty, highlighting a shortfall of deterministic models in capturing this range of uncertainties. These factors together underscore the necessity for models that incorporate probabilistic aspects, thus potentially enabling a more accurate and user-relevant application of MRP.

To that end, we will now expand the scope of MRP to cases in which the agent is uncertain about the human model. Particularly, we build on Definition 19 and extend it to the case where the human knowledge base is probabilistic (i.e., a belief base). In other words, we are now interested in *probabilistic model reconciling explanations.*

First, we show through the following example how the concepts surrounding probabilistic monolithic explanations introduced in the previous section are applicable to the case of an agent knowledge base KB_α and a human belief base $\mathcal{B}_h$.

Example 13. *Let* $\mathrm{KB}_\alpha = \{a, \neg a \vee b, c\}$ *and* $\mathcal{B}_h = \{(c, 2), (\neg c \vee \neg a, 2)\}$ *be the knowledge bases of an agent and the belief base of a human, respectively. Additionally, let* b *be the explanandum, where* $\mathrm{KB}_\alpha \models b$ *and* $P_h(b) = 0.5$*. The goal in this example would then be to find which formulae from* KB_α *increase the probability of the explanandum for* $\mathcal{B}_h$*, that is, to find a probabilistic monolithic explanation* $\tilde{\epsilon}$ *for* b *from* KB_α *for* $\mathcal{B}_h$ *such that* $P_h(b \,|\, \tilde{\epsilon}) > P_h(b)$ *(Definition 22).*

Given KB_α*, there are three possible monolithic explanations:* $\tilde{\epsilon}_1 = \{a\}$*,* $\tilde{\epsilon}_2 = \{\neg a \vee b\}$*, and* $\epsilon_3 = \{a, \neg a \vee b\}$*. Evaluating them with respect to the probability distribution induced by* $\mathcal{B}_h$*, we get* $P_h(b \mid \tilde{\epsilon}_1) = 0.5$*,* $P_h(b \mid \tilde{\epsilon}_2) = 0.55$*, and* $P_h(b \mid \tilde{\epsilon}_3) = 1$*. Notice now that only* $\tilde{\epsilon}_2$ *and* $\tilde{\epsilon}_3$ *qualify as monolithic explanations since* $P_h(b \mid \tilde{\epsilon}_2) > P_h(b)$ *and* $P_h(b \mid \tilde{\epsilon}_3) > P_h(b)$*, whilst* $\tilde{\epsilon}_1$ *does not qualify as a monolithic explanation as* $P_h(b \mid \tilde{\epsilon}_1) = P_h(b) = 0.5$*.*

Given $\tilde{\epsilon}_2$ and $\tilde{\epsilon}_3$ as the two possible monolithic explanations, we can now evaluate their effectiveness in terms of explanatory gain (Definition 23) and explanatory power (Definition 24). In terms of explanatory gain, we get $G(\tilde{\epsilon}_2, b) = 0.14$ and $G(\tilde{\epsilon}_3, b) = 1$. In terms of explanatory power (for $\gamma = 0.5$), we get $\mathrm{EP}(\tilde{\epsilon}_2, b) = 0.59$ and $\mathrm{EP}(\tilde{\epsilon}_3, b) = 1.04$. Finally, following the definition of most-preferred monolithic explanation (Definition 26), we get that $\tilde{\epsilon}_3$ is the most-preferred monolithic explanation for b from KB_α for $\mathcal{B}_h$.

On the one hand, example 13 shows that the definitions introduced in Section 4.3.1 can be directly applied to the case of an agent knowledge base KB_α and a human belief base $\mathcal{B}_h$. On the other hand, there is something important to highlight here. Despite $\tilde{\epsilon}_3$ being the most-preferred monolithic explanation (i.e., it has the highest explanatory power), notice that its probability $P_h(\tilde{\epsilon}_3) = 0.09$ is rather low, which means that its negation $\neg\tilde{\epsilon}_3$ has a much higher probability with $P_h(\neg\tilde{\epsilon}_3) = 0.91$. Logically, this is explained by the fact that $\tilde{\epsilon}_3$ is inconsistent with the formulae in $\mathcal{B}_h^{\downarrow w}$. Therefore, the probabilistic monolithic explanation $\tilde{\epsilon}_3$ may not achieve the intended "reconciliation" between the agent and the human.

Recall that a model reconciling explanation (see Definition 19) is of the form $\mathcal{E} = \langle \epsilon^+, \epsilon^- \rangle$, where ϵ^- is specifically intended to resolve the inconsistency between the agent and the human with respect to the explanandum. Intuitively, the provision of ϵ^- can be thought of as the agent's suggestion of what is "false" in the human knowledge base, at least compared to the agent knowledge base. In the case of a human belief base $\mathcal{B}_h$, we can account for ϵ^- by finding a set of formulae from $\mathcal{B}_h$ such that $P_h(\epsilon^+ \mid \neg\epsilon^-) > P_h(\epsilon^+)$. For example,

Example 14. *Let $\mathrm{KB}_\alpha = \{a, \neg a \vee b, c\}$ and $\mathcal{B}_h = \{(c, 2), (\neg c \vee \neg a, 2)\}$ from Example 13. From the perspective of KB_α, explanation $\tilde{\epsilon} = \{a, \neg a \vee c\}$ can be seen as the formulae that should be true (e.g., added) in $\mathcal{B}_h$ (i.e., $\tilde{\epsilon}^+ = \tilde{\epsilon}$). However, notice that $\tilde{\epsilon}^+$ is inconsistent with $\mathcal{B}_h^{\downarrow w} = \{c, \neg c \vee \neg a\}$. Thus, from the perspective of KB_α, some formulae from $\mathcal{B}_h^{\downarrow w}$ are false (e.g., they should be retracted). One can see that $\tilde{\epsilon}^- = \neg c \vee \neg a$ is the only formula that should be false as it is the only one that is inconsistent with KB_α. Indeed, if $\tilde{\epsilon}^-$ is assumed to be false, then the probability of $\tilde{\epsilon}^+$ increases, i.e., $P_h(\tilde{\epsilon}^+ \mid \neg\tilde{\epsilon}^-) = 0.5 > P_h(\tilde{\epsilon}^+) = 0.09$. Therefore, $\tilde{\epsilon}^+$ and $\tilde{\epsilon}^-$ can be seen as a model reconciling explanation for b from KB_α for $\mathcal{B}_h$.*

Before formally defining what constitutes a probabilistic model reconciling explanation, we state the following assumptions underlying our framework:

- **Shared Domain Language**: The agent and the human user share the same (propositional) language $\mathcal{L}$, that is, they share the same set of atomic variables $\mathcal{V}$ from which formulae specific to a domain can be constructed.
- **Agent Knowledge Base**: The agent model is represented by the (deterministic) knowledge base KB_α, encoding the ground truth of the domain.
- **Human Belief Base**: The human model is represented by the belief base $\mathcal{B}_h$ (and its associated probability distribution P_h), reflecting the agent's uncertainty, for example, its degrees of belief about the human model. The agent has access to $\mathcal{B}_h$ a-priori.[38]

We define a *probabilistic model reconciling explanation* as follows:

Definition 27 (Probabilistic Model Reconciling Explanation)**.** *Given the knowledge base* KB_α *of an agent, the belief base* $\mathcal{B}_h$ *of a human user, and an explanandum* φ *such that* $\mathrm{KB}_\alpha \models \varphi$ *and* $P_h(\varphi) < 1$*,* $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ *is a probabilistic model reconciling explanation if and only if* $\tilde{\epsilon}^+ \subseteq \mathrm{KB}_\alpha$ *and* $\tilde{\epsilon}^- \subseteq \mathcal{B}_h^{\downarrow w}$*, and* $P_h(\varphi \mid \tilde{\epsilon}^+) > P_h(\varphi)$ *and* $P_h(\tilde{\epsilon}^+ \mid \neg\tilde{\epsilon}^-) > P_h(\tilde{\epsilon}^+)$*.*

A probabilistic model reconciling explanation $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ for φ from KB_α for $\mathcal{B}_h$ is a tuple that increases the degree of belief in φ with $\tilde{\epsilon}^+$, as well as increasing the degree of belief in $\tilde{\epsilon}^+$ with $\tilde{\epsilon}^-$ if $\tilde{\epsilon}^+$ is inconsistent with $\mathcal{B}_h^{\downarrow w}$. For brevity, until the end of this section, we will refer to probabilistic model reconciling explanations $\tilde{\mathcal{E}}$ as model reconciling explanations. In this context, the notion of explanatory gain takes the following form:

Definition 28 (Explanatory Gain for Model Reconciling Explanations)**.** *Let* $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ *be a model reconciling explanation for explanandum* φ *from* KB_α *for* $\mathcal{B}_h$*. The explanatory gain of* $\tilde{\mathcal{E}}$ *for* φ *is defined as* $G(\tilde{\mathcal{E}}, \varphi) = \log\left(\frac{P(\varphi \mid \tilde{\epsilon}^+)}{P(\varphi)}\right) + \log\left(\frac{P(\tilde{\epsilon}^+ \mid \neg\tilde{\epsilon}^-)}{P(\tilde{\epsilon}^+)}\right)$*.*

In essence, the explanatory gain of $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ for φ evaluates to what extent $\tilde{\epsilon}^+$ increases the probability of φ, as well as the extent to which $\tilde{\epsilon}^-$ increases the probability of $\tilde{\epsilon}^+$, *assuming that* $\tilde{\epsilon}^-$ *is false.*

Example 15. *Let* $\tilde{\mathcal{E}} = \langle \{a, \neg a \vee b\}, \{\neg c \vee \neg a\} \rangle$ *be the model reconciling explanation for* b *from* KB_α *for* $\mathcal{B}_h$ *in Example 14. The explanatory gain of* $\tilde{\mathcal{E}}$ *for* b *is* $G(\tilde{\mathcal{E}}, b) = \log\left(\frac{1}{0.5}\right) + \log\left(\frac{0.5}{0.09}\right) = 1 + 2.47 = 3.47$.

[38] We leave the question of acquiring (or learning) the human belief base open for future work.

Similarly, the notion of explanatory power is defined in the following way:

Definition 29 (Explanatory Power for Model Reconciling Explanations). *Let $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ be a model reconciling explanation for explanandum φ from KB_α for $\mathcal{B}_h$. The explanatory power of $\tilde{\mathcal{E}}$ for φ is defined as $\mathrm{EP}(\tilde{\mathcal{E}}, \varphi) = G_h(\tilde{\mathcal{E}}, \varphi) + \gamma \cdot (P_h(\tilde{\epsilon}^+) + P_h(\tilde{\epsilon}^-))$, where $\gamma \in [0, 1]$ is a constant.*

This definition of explanatory power of $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ for φ assesses, in addition to the explanatory gain of $\tilde{\mathcal{E}}$, the likelihoods of $\tilde{\epsilon}^+$ and $\tilde{\epsilon}^-$, with γ parameterizing their relative importance in the overall assessment.

Example 16. *Continuing from Example 15, the explanatory power of $\tilde{\mathcal{E}} = \langle \{a, \neg a \vee b\}, \{\neg c \vee \neg a\} \rangle$ for b (for $\gamma = 0.5$) is $\mathrm{EP}(\tilde{\mathcal{E}}, b) = 3.47 + 0.5 \cdot (0.09 + 0.90) = 3.96$*

Finally, a preference relation and a most-preferred model reconciling explanation can be defined in the same manner as in Definition 25 and Definition 26, respectively.

Definition 30 (Preference Relation for Model Reconciling Explanation). *Let $\tilde{\mathcal{E}}_1$ and $\tilde{\mathcal{E}}_2$ be two model reconciling explanations for explanandum φ from knowledge base KB_α for belief base $\mathcal{B}_h$. $\tilde{\mathcal{E}}_1$ is preferred over $\tilde{\mathcal{E}}_2$, denoted $\tilde{\mathcal{E}}_1 \succeq \tilde{\mathcal{E}}_2$, if and only if $\mathrm{EP}(\tilde{\mathcal{E}}_1) \geq \mathrm{EP}(\tilde{\mathcal{E}}_2)$.*

Definition 31 (Most-Preferred Model Reconciling Explanation). *Let $\tilde{E}(\varphi)$ denote the set of all model reconciling explanations for explanandum φ from knowledge base KB_α for belief base $\mathcal{B}_h$. A model reconciling explanation $\tilde{\mathcal{E}}^* \in \tilde{E}(\varphi)$ is the most-preferred model reconciling explanation for φ if and only if $\tilde{\mathcal{E}}^* \succeq \tilde{\mathcal{E}}$ for all $\tilde{\mathcal{E}}^{\in} \tilde{E}(\varphi)$.*

4.4 Related Work

In this chapter, we presented a framework for probabilistic explanation generation in monolithic and model reconciliation scenarios. In the monolithic case, our definition of a probabilistic monolithic explanation (Definition 22) may appear similar to what was proposed by Gärdenfors [87]. Nonetheless, an important distinction here is that Gärdenfors is dealing with epistemic states that do not contain the explanandum, while we are dealing with belief bases that do contain the explanandum. We also define a different notion of explanatory power as well as present algorithms for computing explanations. Urszula et al [32] have also considered the problem of defining what constitutes a (monolithic) explanation in probabilistic system, however they focus on epistemic states defined over causal structures.

The notion of (monolithic) explanation has also been explored by the *probabilistic logic programming* (PLP) community [56, 76], a formalism that extends logic programming languages (i.e., Prolog) with probabilities. In PLP, explanations have been associated with possible worlds. For instance, the MPE (most probable explanation) task consists in finding the world with the highest probability given some evidence [200]. However, a world does not show the chain of inferences of a given explanandum and, moreover, it is not minimal by definition, since it usually includes a (possibly large) number of probabilistic facts whose truth value is irrelevant for the explanandum. Another alternative consists in using the proof of an explanandum as an explanation [142], where one can associate a proof with a (minimal) partial world ω' such that for all worlds $\omega \supseteq \omega'$, the explanandum is true in ω. In this case, one can easily ensure minimality, but even if the partial world contains no irrelevant facts, it is still not easy to determine the chain of inferences behind a given explanandum. Renkens et al. [190] have tackled explanation generation in PLP from the perspective of weighted model counting and knowledge compilation.

In the model reconciliation setting, we have extended our work on the logic-based model reconciliation problem from Chapter 3 to a probabilistic case (Definition 40) for capturing scenarios where the human model is uncertain. Sreedharan et al. [205] proposed a method for generating explanations in the case of uncertain human models, however, their approach is limited to planning problems, and importantly, it does not quantify the uncertainty levels of the generated explanations, that is, there is no notion of probabilistic explanation. In contrast, the application of probabilistic explanations in the context of model reconciliation that we consider in this work is, to our knowledge, novel.

4.5 Concluding Remarks

This chapter has addressed the challenge of generating explanations in environments characterized by uncertainty. Our work aims to bridge the gap between classical explanation models and the inherent uncertainty prevalent in real-world scenarios. We have made two primary contributions:

1. **Probabilistic Monolithic Explanations:** We developed a framework for generating explanations within uncertain knowledge bases. This approach is particularly relevant for scenarios where an AI system needs to explain its own reasoning process. We introduced the concepts of explanatory gain and explanatory power as quantitative measures to evaluate the effectiveness and relevance of explanations, offering a more detailed assessment of explanation quality in probabilistic settings.

2. **Probabilistic Model Reconciling Explanations:** We extended the model reconciliation problem to address situations where the human model is not known with certainty. This contribution addresses the need to reconcile model differences between an agent and a human user in scenarios where perfect knowledge of the human's understanding is unavailable.

While this work represents a step forward in explanation generation under uncertainty, it also reveals areas for future research, particularly in the model reconciliation setting. Our current approach assumes a classical knowledge base for the agent model. Future research could explore scenarios where the agent's model is also probabilistic, leading to the challenge of reconciling two belief bases. This presents significant challenges due to the complexity of probabilistic logic and the need to account for uncertainty in both models.

Reconciling two sets of beliefs, each with its own probability distributions, to achieve a coherent understanding that accurately reflects the true state of affairs or intentions, is a complex task. Future work could investigate methods to ensure that the reconciled belief base maintains a meaningful probability distribution over its assertions, considering both logical consistency and probabilistic coherence.

While this work represents a step forward in explanation generation under uncertainty, it also reveals areas for future research. One promising direction is the extension of our methods to other probabilistic logic frameworks. For instance, our approaches could potentially be adapted to work with Markov Logic Networks (MLNs) [192], which combine first-order logic with probabilistic graphical models. Such an extension could leverage the expressive power of MLNs to handle more complex uncertain knowledge representations. Furthermore, exploration of other probabilistic logic formalisms, such as Probabilistic Logic Programming [77], could yield insights into generating explanations in different types of uncertain reasoning systems. Each of these frameworks has unique characteristics that could influence the explanation generation process and potentially lead to new methods for quantifying explanatory power in these contexts.

In the model reconciliation setting, future research could explore scenarios where both the agent's and the human's models are uncertain. This presents significant challenges due to the complexity of probabilistic logic and the need to account for uncertainty in both models. Reconciling two sets of belief based, each with its own probability distributions, to achieve a coherent understanding that accurately reflects the true state of affairs or intentions, is a complex task. Future work could investigate methods to ensure that the reconciled belief base maintains a meaningful probability distribution over its assertions, considering both logical consistency and probabilistic coherence. This could involve developing new techniques for belief merging in probabilistic logics or adapting existing approaches from belief revision theory to handle probabilistic knowledge bases.

These research directions highlight the potential for further extending logic's role as an explainability layer in probabilistic settings. As AI systems continue to evolve and handle increasingly complex uncertain scenarios, the need for effective explanations becomes even more critical. Our framework contributes to the development of explainable AI systems that can handle uncertainty while maintaining the formal rigor and clarity that logic provides as an explainability layer.

However, a crucial challenge remains: how can we efficiently compute these explanations in practice? While this chapter has established the theoretical foundations for explanation generation under uncertainty, the next chapter addresses the computational aspects of our framework. We will show how the theory of hitting sets can be leveraged to develop efficient algorithms for computing both deterministic and probabilistic explanations, moving our logical explainability layer closer to real-world applications.

Chapter 5

Exploiting Hitting Sets for Efficient Explanation Computation

"Ex Falso Quodlibet." (From falsehood, anything)

— William of Soissons

5.1 Introduction & Contribution

The preceding chapters established the theoretical foundations for logic-based explanation generation. We introduced the Logic-based Model Reconciliation Problem (L-MRP) for deterministic settings and extended it to handle probabilistic knowledge bases, providing a framework for both monolithic and model reconciling explanations. However, to bridge the gap between theory and practice, we must develop efficient algorithms capable of computing these explanations in practice.

This chapter presents a general algorithmic approach for computing explanations, applicable across various domains and extending beyond our initial focus on planning problems. Our approach leverages the fundamental duality between *minimal correction sets* (MCSes) and *minimal unsatisfiable sets* (MUSes), a relationship explored in different contexts by researchers such as Reiter [189] (as diagnoses and conflicts) and Kleer *et al.* [53] (relating them to prime implicates and prime implicants in propositional logic).

By exploiting this duality, we develop algorithms that compute explanations in both deterministic and probabilistic settings. The main contributions of this chapter are:

1. Two algorithms for explanation generation:
   - An algorithm for computing classical (deterministic) monolithic explanations.
   - An algorithm for computing classical (deterministic) model reconciling explanations.
2. Adaptation techniques for these algorithms to compute probabilistic explanations, building upon the frameworks introduced in the previous chapter.
3. An evaluation of the algorithms' efficiency and scalability, demonstrating their applicability to problems beyond the planning domain.

This approach to explanation computation addresses a crucial challenge in making logic-based explainability practical: the need for efficient generation of explanations across various problem types and knowledge representations. By providing algorithms that work efficiently in both deterministic and probabilistic settings, we demonstrate how our logical explainability layer can handle the complexities and uncertainties present in various scenarios while remaining computationally tractable.

5.2 Essential Background

We adopt a propositional language language $\mathcal{L}$ and all definitions that are described in Chapter 5 (Section 4.2).

Duality of Minimal Unsatisfiable and Minimal Corrections Sets

Definition 32 (Minimal Unsatisfiable Set (MUS)). *Given an inconsistent knowledge base* KB*, a subset $\mathcal{M} \subseteq \text{KB}$ is an MUS if $\mathcal{M}$ is inconsistent and $\forall \mathcal{M}' \subset \mathcal{M}$, $\mathcal{M}'$ is consistent.*

Definition 33 (Minimal Correction Set (MCS)). *Given an inconsistent knowledge base* KB*, a subset $\mathcal{C} \subseteq \text{KB}$ is an MCS if $\text{KB} \setminus \mathcal{C}$ is consistent and $\forall \mathcal{C}' \subset \mathcal{C}$, $\text{KB} \setminus \mathcal{C}'$ is inconsistent.*

By definition, every inconsistent KB contains at least one MUS.

Definition 34 (Partial MUS). *A set of formulae* Φ *is a partial MUS of an inconsistent knowledge base* KB *if there exists at least one MUS* $\mathcal{M} \subseteq$ KB *such that* $\Phi \subseteq \mathcal{M}$.

Partial MUSes in an inconsistent knowledge base KB appear when a subset of formulae is set as *hard*, that is, formulae that must always be satisfied in a solution. Conversely, *soft* formulae may not always be satisfied. Given a formula φ, we will write φ^* with $* \in \{s, h\}$ to denote it as *soft* and *hard*, respectively.

MUSes and MCSes are related by the concept of *minimal hitting set*:

Definition 35 (Minimal Hitting Set). *Given a collection* Γ *of sets from a universe* U, *a hitting set for* Γ *is a set* $H \subseteq U$ *such that* $\forall S \in \Gamma$, $H \cap S \neq \emptyset$ *and* $\nexists H' \subset H$ *such that* $H' \cap S \neq \emptyset$.

The relationship between MUSes and MCSes is discussed by Liffiton *t al.* [156, 155], and it was firstly presented by Reiter [189], where MUSes and MCSes are referred to as (minimal) conflicts and diagnoses, respectively.

Proposition 3. *A subset of an inconsistent knowledge base* KB *is an MUS (resp. MCS) if and only if it is a minimal hitting set of the collection of all MCSes (resp. MUSes) of* KB.

It follows from the above proposition that a cardinality minimal MUS (resp. MCS) is a minimal hitting set. *Cardinality minimal MUS* are referred to as SMUS, whereas a *cardinality minimal MCS* corresponds to the complement of a MaxSAT solution [154]. We may refer to a cardinality minimal set as a minimum or smallest set.

Lemma 1. *Given a subset* $\mathcal{H}$ *of all the MCSes of knowledge base* KB, *a hitting set is an SMUS if: (1) It is a minimal hitting set* h *of* $\mathcal{H}$, *and (2) The subformula induced by* h *is inconsistent.*

See the work by Ignatiev *et al.* [117] for a proof.

Proposition 3 and Lemma 1 naturally extend to the case of partial MUS. Note that when some formulae are set as hard in an inconsistent knowledge base, the set of all MCSes is a subset of the soft formulae. In this case, every minimal hitting set on the set of all MCSes is a partial MUS.

Finally, MUSes and monolithic explanations are related by the following:

Proposition 4. *Given a knowledge base* KB*, a consistent set of formulae* $\epsilon \subseteq \mathrm{KB}$ *is a monolithic explanation for* φ *from* KB *(Definition 18) if and only if* ϵ *is a partial MUS of* $\epsilon \cup \{\neg\varphi\}$*.*

Example 17. *Let* $\mathrm{KB} = \{p, \neg p \vee q, \neg p \vee r\}$ *and* $\epsilon = \{p, \neg p \vee q\}$ *from Example 6. Notice how* $\mathcal{M} = \{p, \neg p \vee q, \neg q\}$ *is an MUS of* $\mathrm{KB} \cup \{\neg q\}$*. Then, it is easy to see that* ϵ *is a partial MUS of* $\mathcal{M}$*.*

5.3 Exploiting Hitting Sets

We now describe our algorithms for computing explanations. We first show how to exploit the hitting set duality of MUSes and MCSes for computing classical (deterministic) monolithic explanations (Definition 18) and model reconciling explanations (Definition 19), and then show how to extend it to probabilistic settings.

5.3.1 Classical Explanations

Monolithic Explanations

We consider a knowledge base KB and an explanandum φ such that $\mathrm{KB} \models \varphi$. The principal idea of this approach is to reduce the problem of computing a monolithic explanation of minimum size to the one of computing a *smallest minimal unsatisfiable set* (SMUS) over an inconsistent knowledge base [117].

In particular, notice that, by definition, we have that $\mathrm{KB} \models \varphi$ if and only if $\mathrm{KB} \cup \{\neg\varphi\}$ is inconsistent. Moreover, in Proposition 4, we have already stated the relation between a monolithic explanation and a *minimal unsatisfiable set* (MUS). This suggests that, in order to extract a monolithic explanation, we just need to run an MUS solver over the knowledge base $\mathrm{KB}^s \cup \{\neg\varphi^h\}$, where KB^s and φ^h denote that KB and φ are treated as soft and hard constraints, respectively, and then remove $\neg\varphi$ from the returned MUS.[39] The hitting set

[39] Recall that soft constraints may be removed by the MUS solver, while hard constraints will not be removed.

Algorithm 5.1: `monolithic-explanation`(KB, φ)

Input: Knowledge base KB and explanandum φ

Result: A minimum size monolithic explanation ϵ for φ from KB

1 $\mathcal{H} \leftarrow \emptyset$
2 **while** *true* **do**
3     $seed \leftarrow$ `minHS`$(\mathcal{H})$ `// compute a minimal hitting set`
4     $\epsilon \leftarrow \{c_i \mid i \in seed\}$
5   **if not** `SAT`$(\epsilon \cup \{\neg\varphi\})$ **then**
6   **return** ϵ `// minimum size monolithic explanation`
7   **else**
8         $\mathcal{C} \leftarrow$ `getMCS`$(seed, \mathrm{KB}^s \cup \{\neg\varphi^h\})$ `// compute a minimal correction set`
9     $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{C}\}$

duality relating MUSes and *minimal correction sets* (MCSes) (see Lemma 1) is a key aspect for the computation of an SMUS.

Algorithm 5.1 describes the main steps of our approach. $\mathcal{H}$ is a collection of sets, where each set corresponds to an MCS on KB. At the beginning, it is initialized with the empty set (line 1). Each MCS in $\mathcal{H}$ is represented as the set of the indexes of the formulae in it. $\mathcal{H}$ stores the MCSes computed so far. At each step, a minimal hitting set on $\mathcal{H}$ is computed (line 3). In line 4, the formulae induced by the computed minimal hitting set is stored in ϵ. Then, $\epsilon \cup \{\neg\varphi\}$ is evaluated for satisfiability (line 5). If $\epsilon \cup \{\neg\varphi\}$ is inconsistent, then ϵ is a monolithic explanation of minimum size and the algorithm returns ϵ. If instead $\epsilon \cup \{\neg\varphi\}$ is consistent, then it means that $\epsilon \not\models \varphi$ and the algorithm continues in line 8. The computation of an MCS of this kind can be performed via standard MCS procedures [165], using the set of formulae indexed by the *seed* as the starting formula to extend. Since φ is set to hard (line 8), the returned MCS $\mathcal{C}$ is guaranteed to be contained in KB. Due to the hitting set duality relation, we will also have $\epsilon \subseteq$ KB. Finally, notice that the procedure *getMCS* always reports a new MCS because, by construction, we have $seed \subseteq \mathrm{KB} \setminus \mathcal{C}$. In fact, the *seed* contains at least one formula for each previously computed MCS and, thus, $seed \cap \mathcal{C} = \emptyset$ (i.e., at least one formula for each previously computed MCS is not in $\mathcal{C}$). Example 18 shows an example trace of the algorithm.

Algorithm 5.1 is complete in the sense that eventually a monolithic explanation $\epsilon \subseteq$ KB of minimum size such that $\epsilon \models \varphi$ will be returned. This can be easily verified by observing that every time $\epsilon \cup \{\neg\varphi\}$ is satisfiable, a new MCS is computed. Eventually, all the MCSes will be computed and, from Propositions 3 and 4, it follows that a minimal hitting set on the collection of all MCSes corresponds to the smallest MUS, and as such, to a monolithic explanation of minimum size.

Note that deciding whether there exists a monolithic explanation of size less or equal to k is Σ_2^p-complete and extracting a smallest monolithic explanation is in $FP^{\Sigma_2^p}$. This follows directly from the complexity of deciding and computing an SMUS on which Algorithm 5.1 is based on [117].

Example 18. *Consider the following knowledge base* KB $=\overset{C_1}{(a \vee b)} \wedge \overset{C_2}{(\neg b \vee c)} \wedge \overset{C_3}{(\neg c)} \wedge \overset{C_4}{(\neg b \vee d)}$

We have that KB $\models a$. *The execution of Algorithm 5.1 proceeds as follows:*

1. ***Initialize:*** $\mathcal{H} \leftarrow \emptyset$
2. ***Compute*** *seed* $\leftarrow \emptyset$ *(minHS($\mathcal{H}$))*
3. ***Check*** $\emptyset \not\models a$ *(SAT($\epsilon \cup \{\neg a\}$))*
4. ***Compute*** $\mathcal{C} \leftarrow \{C_1\}$ *(MCS on $KB^s \cup \{\neg a^h\}$ with seed)*
5. ***Update*** $\mathcal{H} \leftarrow \{\{C_1\}\}$
6. ***Compute*** *seed* $\leftarrow \{C_1\}$ *(minHS($\mathcal{H}$))*
7. ***Check*** $\{a \vee b\} \not\models a$ *(SAT($\epsilon \cup \{\neg a\}$))*
8. ***Compute*** $\mathcal{C} \leftarrow \{C_2\}$ *(MCS on $KB^s \cup \{\neg a^h\}$ with seed)*
9. ***Update*** $\mathcal{H} \leftarrow \{\{C_1\}, \{C_2\}\}$
10. ***Compute*** *seed* $\leftarrow \{C_1, C_2\}$ *(minHS($\mathcal{H}$))*
11. ***Check*** $\{a \vee b, \neg b \vee c\} \not\models a$ *(SAT($\epsilon \cup \{\neg a\}$))*
12. ***Compute*** $\mathcal{C} \leftarrow \{C_3\}$ *(MCS on $KB^s \cup \{\neg a^h\}$ with seed)*
13. ***Update*** $\mathcal{H} \leftarrow \{\{C_1\}, \{C_2\}, \{C_3\}\}$
14. ***Compute*** *seed* $\leftarrow \{C_1, C_2, C_3\}$ *(minHS($\mathcal{H}$))*

Algorithm 5.2: `model-reconciling-explanation`$(\mathrm{KB}_\alpha, \mathrm{KB}_a^h, \varphi)$

Input: Knowledge bases KB_α and KB_h and explanandum φ

Result: A model reconciling explanation $\mathcal{E} = \langle \epsilon^+, \epsilon^- \rangle$ for φ from KB_α for KB_h

$\mathbb{R} \leftarrow \emptyset$
$\mathrm{KB}_\alpha^h \leftarrow \mathrm{KB}_\alpha \cap \mathrm{KB}_h$
$\mathrm{KB}_\alpha^s \leftarrow \mathrm{KB}_\alpha \setminus \mathrm{KB}_\alpha^h$
if not $\mathtt{SAT}(\mathrm{KB}_h \cup \mathrm{KB}_\alpha)$ **then**
  $E^- \leftarrow \mathtt{getMCS}((\mathrm{KB}_h \setminus \mathrm{KB}_\alpha)^s \cup \mathrm{KB}_\alpha^h)$ `// restore consistency on` KB_h
  $\mathrm{KB}_h \leftarrow \mathrm{KB}_h \setminus E^-$
while *true* **do**
  $seed \leftarrow \mathtt{minHS}(\mathbb{R})$
  $\epsilon^+ \leftarrow \{c_i \mid i \in seed\}$ `// explanation` ϵ^+ `induced by the seed`
  if not $\mathtt{SAT}(\mathrm{KB}_h \cup \epsilon^+ \cup \{\neg\varphi\})$ **then**
    $\epsilon^- \leftarrow \emptyset$
    if not $\mathtt{SAT}(\mathrm{KB}_h \cup \epsilon^+ \cup E^-)$ **then**
      $\epsilon^- \leftarrow \mathtt{getMCS}((\mathrm{KB}_h \cup \epsilon^+)^h \cup (E^-)^s)$
    return $\langle \epsilon^+, \epsilon^- \rangle$
  else
    $\mathcal{C} \leftarrow \mathtt{getMCS}(seed, \mathrm{KB}_a^h \cup \{\neg\varphi^h\} \cup \mathrm{KB}_\alpha^s)$
    $\mathbb{R} \leftarrow \mathbb{R} \cup \{\mathcal{C}\}$

15. ***Check*** $\{a \vee b, \neg b \vee c, \neg c\} \models a$ *($\neg SAT(\epsilon \cup \{\neg a\})$)*

16. ***Return*** $\{C_1, C_2, C_3\}$ *(monolithic explanation for a from* KB*)*

The algorithm terminates, returning the set $\{a \vee b, \neg b \vee c, \neg c\}$ as the minimum size support for a from KB*.*

Model Reconciling Explanations

We now show how Algorithm 5.1 can be further extended for computing model reconciling explanations $\mathcal{E} = \langle \epsilon^+, \epsilon^- \rangle$ for an explanandum φ from an agent knowledge base KB_α for a human knowledge base KB_h, where where $\epsilon^+ \subseteq \mathrm{KB}_\alpha$ and $\epsilon^- \subseteq \mathrm{KB}_h$, such that $(\mathrm{KB}_h \cup \epsilon^+) \setminus \epsilon^- \models \varphi$.

Algorithm 5.2 describes the pseudocode of our approach. At the beginning of the algorithm, we initialize $\mathbb{R}$ to the null set (line 1). $\mathbb{R}$ is used to store the MCSes, which acts as a mediator between KB_α and KB_h. Lines 2-3 are used to specify which clauses of KB_α will be treated as hard and soft constraints, respectively. We then check if $\mathrm{KB}_h \cup \mathrm{KB}_\alpha$ is inconsistent (line 4). This is important in order to avoid the possibility of finding subsets ϵ^+ that explain why $\mathrm{KB}_h \cup \mathrm{KB}_\alpha$ is inconsistent instead of the target explanandum. In case $\mathrm{KB}_h \cup \mathrm{KB}_\alpha$ is inconsistent, we preprocess KB_h by removing from $\mathrm{KB}_h \setminus \mathrm{KB}_\alpha$ a minimal set of formulae causing the conflict (i.e., an MCS) (lines 5-6), where E^- stores the set of potential formulae ϵ^- to retract. The reconciliation procedure starts in line 7. The algorithm proceeds iteratively by computing a minimal hitting set on $\mathbb{R}$ and then testing for satisfiability the formulae ϵ^+ (lines 8-10). The test checks whether adding ϵ^+ to KB_h is sufficient for entailing φ. If $\mathrm{KB}_h \cup \epsilon^+ \cup \{\neg\varphi\}$ is unsatisfiable, then $\mathrm{KB}_h \cup \epsilon^+ \models \varphi$. In that case, the algorithm then checks whether $\mathrm{KB}_h \cup \epsilon^+ \cup E^-$ is inconsistent, and if it is, it computes an MCS e^- on $(\mathrm{KB}_h \cup \epsilon^+)^h \cup (E^-)^s$ (lines 12-13). The model reconciling explanation $\langle \epsilon^+, \epsilon^- \rangle$ is then returned in line 14. Otherwise, the algorithm continues in line 16, where a new MCS is computed and added to $\mathbb{R}$. Note that the algorithm is complete as it is based on Algorithm 5.1, which is complete. Example 19 shows an example trace of the algorithm.

Example 19. *Consider the following knowledge bases* KB_α *and* KB_h*:*

$$\mathrm{KB}_\alpha = \{\overset{C_1}{(a \vee b)}, \overset{C_2}{(\neg b \vee c)}, \overset{C_3}{\neg c}, \overset{C_4}{(\neg b \vee d)}, \overset{C_5}{\neg d}\}$$
$$\mathrm{KB}_h = \{\overset{D_1}{b}, \overset{D_2}{\neg c}\}$$

We have that $\mathrm{KB}_\alpha \models a$ *and* $\mathrm{KB}_h \not\models a$*. The execution of Algorithm 5.2 proceeds as follows:*

1. ***Initialize:*** $\mathcal{R} \leftarrow \emptyset$
2. ***Compute*** $\mathrm{KB}_\alpha^h \leftarrow \mathrm{KB}_\alpha \cap \mathrm{KB}_h = \{C_3\}$
3. ***Compute*** $\mathrm{KB}_\alpha^s \leftarrow \mathrm{KB}_\alpha \setminus (\mathrm{KB}_\alpha \cap \mathrm{KB}_h) = \{C_1, C_2, C_4, C_5\}$
4. ***Compute*** $E^- \leftarrow \{D_1\}$ *(MCS on* $(\mathrm{KB}_h \setminus \mathrm{KB}_\alpha)^s \cup \mathrm{KB}_\alpha^h$*)*
5. ***Update*** $\mathrm{KB}_h \leftarrow \{D_1, D_2\} \setminus \{D_1\} = \{D_2\}$

6. ***Compute*** *seed* $\leftarrow \emptyset$ *(minHS($\mathcal{R}$))*

7. ***Check*** $\{\neg c\} \not\models a$ *(SAT($\mathrm{KB}_h \cup \epsilon^+ \cup \{\neg a\}$))*

8. ***Compute*** $\mathcal{C} \leftarrow \{C_1\}$ *(MCS on $\mathrm{KB}_a^h \cup \mathrm{KB}_\alpha^s \cup \{\neg a^h\}$)*

9. ***Update*** $\mathcal{R} \leftarrow \{\{C_1\}\}$

10. ***Compute*** *seed* $\leftarrow \{C_1\}$ *(minHS($\mathcal{R}$))*

11. ***Check*** $\{\neg c, a \vee b\} \not\models a$ *(SAT($\mathrm{KB}_h \cup \epsilon^+ \cup \{\neg a\}$))*

12. ***Compute*** $\mathcal{C} \leftarrow \{C_2, C_4\}$ *(MCS on $\mathrm{KB}_a^h \cup \mathrm{KB}_\alpha^s \cup \{\neg a^h\}$)*

13. ***Update*** $\mathcal{R} \leftarrow \{\{C_1\}, \{C_2, C_4\}\}$

14. ***Compute*** *seed* $\leftarrow \{C_1, C_4\}$ *(minHS($\mathcal{R}$))*

15. ***Check*** $\{\neg c, a \vee b, \neg b \vee d\} \not\models a$ *(SAT($\mathrm{KB}_h \cup \epsilon^+ \cup \{\neg a\}$))*

16. ***Compute*** $\mathcal{C} \leftarrow \{C_2, C_5\}$ *(MCS on $\mathrm{KB}_a^h \cup \mathrm{KB}_\alpha^s \cup \{\neg a^h\}$)*

17. ***Update*** $\mathcal{R} \leftarrow \{\{C_1\}, \{C_2, C_4\}, \{C_2, C_5\}\}$

18. ***Compute*** *seed* $\leftarrow \{C_1, C_2\}$ *(minHS($\mathcal{R}$))*

19. ***Check*** $\{\neg c, a \vee b, \neg b \vee c\} \models a$ *($\neg$SAT($\mathrm{KB}_h \cup \epsilon^+ \cup \{\neg a\}$))*

20. ***Compute*** $\epsilon^- \leftarrow \{D_1\}$ *(MCS on $(\mathrm{KB}_h \cup \epsilon^+)^h \cup (E^-)^s$)*

21. ***Return*** $\langle\{C_1, C_2\}, \{D_1\}\rangle$ *(model reconciling explanation for a from KB_α for KB_h)*

The algorithm terminates, returning the pair $\langle\{a \vee b, \neg b \vee c\}, \{b\}\rangle$ as the model reconciling explanation for a from KB_α for KB_h.

5.3.2 Probabilistic Explanations

We now show how the algorithms described in the previous section can be used for computing probabilistic monolithic explanations (Definition 22) and probabilistic model reconciling explanations (Definition 40).

Monolithic Explanations

Consider an explanandum φ and a belief base $\mathcal{B}$. First, notice that if we assume that the classical projection of $\mathcal{B}$ entails the explanandum φ, that is $\mathcal{B}^{\downarrow w} \models \varphi$, then Algorithm 5.1 can directly be applied on $\mathcal{B}^{\downarrow w}$ and φ.[40] In that case, Algorithm 5.1 guarantees to find a monolithic explanation with maximum explanatory gain, since we know from Proposition 2 that explanatory gain achieves its maximum value for φ when the monolithic explanation entails φ. Nevertheless, this does not guarantee that the monolithic explanation will be the most-preferred one, that is, the one with the highest explanatory power (Definition 26).

Obviously, a straightforward way of computing a most-preferred monolithic explanation is to use Algorithm 5.1 to enumerate all possible monolithic explanations for φ, and return the one that has the highest probability, which corresponds to the one with the highest explanatory power. But enumerating through all possible monolithic explanations and computing their probabilities can be computationally prohibited, as even extracting a smallest monolithic explanation is in $FP^{\Sigma_2^p}$ [117] and computing the probability of a formula is $\#P$-complete [196, 38]. We can, however, account for this high computational complexity by seeking for a monolithic explanation that is guaranteed to have a probability above a certain threshold.

First, the following lemma notes that for all possible monolithic explanations $\tilde{\epsilon}$ for explanandum φ, the following upper and lower probability bounds hold:

Lemma 2. *Let $\tilde{E}(\varphi)$ be the set of all monolithic explanations for explanandum φ from belief base $\mathcal{B}$, where $\tilde{\epsilon} \models \varphi$ for all $\tilde{\epsilon} \in \tilde{E}(\varphi)$, and let ω_1 be the most-probable world in which φ is true. Then, for any $\tilde{\epsilon} \in \tilde{E}(\varphi)$, it holds that $P(\omega_1) \leq P(\tilde{\epsilon}) \leq P(\varphi)$.*

Proof. For the upper probability bound, since we assume that for all $\tilde{\epsilon} \in \tilde{E}(\varphi)$, $\tilde{\epsilon} \models \varphi$, then it must hold that for all $\tilde{\epsilon} \in \tilde{E}(\varphi)$, the worlds where $\tilde{\epsilon}$ is true are subsumed by the worlds where φ is true (entailment property). This implies that for any $\tilde{\epsilon} \in \tilde{E}(\varphi)$, $P(\tilde{\epsilon}) \leq P(\varphi)$.

For the lower bound, since ω_1 is the most-probable world of φ, that is, the world where the highest number of formulae from $\mathcal{B}$ are satisfied, then all monolithic explanations for φ must be true in ω_1 (i.e., $\omega_1 \models \tilde{\epsilon}$). As such, for any $\tilde{\epsilon} \in \tilde{E}(\varphi)$, $P(\tilde{\epsilon}) \geq P(\omega_1)$. $\square$

[40] Recall that the classical projection of belief base $\mathcal{B}$ is the unweighted version of the set of formulae from $\mathcal{B}$.

However, some monolithic explanations may have a higher lower probability bound. Formally, we call such explanations *k-bounded monolithic explanations*:

Definition 36 (k-Bounded Monolithic Explanation)**.** *Let $\tilde{E}(\varphi)$ be the set of all monolithic explanations for explanandum φ from belief base $\mathcal{B}$. Let $\Omega(\varphi) = \{\omega_1, \ldots, \omega_n\}$ be the set of possible worlds in which φ is true, where $P(\omega_1) \geq P(\omega_2) \geq \ldots \geq P(\omega_n)$. Also let $I_k = \bigcap_{i=1}^{k} \{\phi \mid \phi \in \mathcal{B}^{\downarrow w}, \omega_i \models \phi\}$ be the intersection of formulae that are true in worlds ω_1 to ω_k. We say that $\tilde{\epsilon} \in \tilde{E}(\varphi)$ is a k-bounded monolithic explanation for φ from $\mathcal{B}$, with lower bound $P(\tilde{\epsilon}) \geq \sum_{i=1}^{k} P(\omega_i)$, if and only if $\tilde{\epsilon} \subseteq I_k$.*

Example 20. *Consider the belief base $\mathcal{B} = \{(a, 1), (\neg a \vee b, 3), (c, 2), (\neg c \vee b, 1)\}$ and explanandum b. The two monolithic explanations for b from $\mathcal{B}$ that entail b are $\tilde{\epsilon}_1 = \{a, \neg a \vee b\}$ and $\tilde{\epsilon}_2 = \{c, \neg c \vee b\}$, where $P(\tilde{\epsilon}_1) = 0.64$ and $P(\tilde{\epsilon}_2) = 0.77$. Notice that there are four possible worlds in which b is true: $\omega_1 = \{a = T, b = T, c = T\}$, $\omega_2 = \{a = F, b = T, c = T\}$, $\omega_3 = \{a = T, b = T, c = F\}$, and $\omega_4 = \{a = F, b = T, c = F\}$, where $P(\omega_1) = 0.57$, $P(\omega_2) = 0.20$, $P(\omega_3) = 0.07$, and $P(\omega_4) = 0.02$. The maximum number of intersections that entail b is $k = 2$ (i.e., $I_2 = \{\neg a \vee b, c, \neg c \vee b\}$). Indeed, $\tilde{\epsilon}_2 \subseteq I_2$ and $P(\tilde{\epsilon}_2) = 0.77 = P(\omega_1) + P(\omega_2)$. Finally, notice how $\tilde{\epsilon}_2$ is also the most-preferred monolithic explanation for b from $\mathcal{B}$; for $\gamma = 0.5$, $\mathrm{EP}(\tilde{\epsilon}_2, b) = 0.57 > \mathrm{EP}(\tilde{\epsilon}_1, b) = 0.50$.*

Proposition 5. *Let $\mathcal{B}$ be a belief base and φ an explanandum. A 1-bounded monolithic explanation $\tilde{\epsilon}$ for φ from $\mathcal{B}$ always exists.*

Proof. The proof follows directly from Lemma 2. □

Interestingly, there also exists a maximal k-bounded monolithic explanation.

Corollary 2. *If $I_k \models \varphi$ and $I_{k+1} \not\models \varphi$, then $\exists \tilde{\epsilon} \subseteq I_k$ with maximal lower bound $P(\tilde{\epsilon}) \geq P(I_k)$*

Proof. First, notice that if $I_k \models \varphi$ and $I_{k+1} \not\models \varphi$, then $I_{k+j} \not\models \varphi$ for all $j = 1, \ldots, n-k$. As such, k is the maximum number of intersections (from ω_1 to ω_k) such that $I_k \models \varphi$. Thus, since $\tilde{\epsilon} \models \varphi$ for all $\tilde{\epsilon} \in \tilde{E}(\varphi)$, it must be the case that there exists at least one $\tilde{\epsilon}$ such that $\tilde{\epsilon} \subseteq I_k$, from which we know that $P(I_k) \leq P(\tilde{\epsilon})$. Moreover, as I_k is the set of formulae that are true in worlds ω_1 to ω_k, its probability must be at least equal to the sum of the

Algorithm 5.3: `probabilistic-monolithic-explanation`$(\mathcal{B}, \varphi, \hat{k})$

Input: Belief base $\mathcal{B}$, explanandum φ, and user-defined parameter $\hat{k}$
Result: A k-bounded monolithic explanation $\tilde{\epsilon}$ for φ from $\mathcal{B}$ for some $k \leq \hat{k}$

1 $k \leftarrow \hat{k}$
2 $\Omega_{\varphi} \leftarrow$ `getTopKWorlds`$(\mathcal{B} \cup \{(\varphi, \infty)\}, k)$ `// find candidate set of formulae`
3 **while** *true* **do**
  `// get intersecting formulae from top` k `worlds of` φ
4   $I_k \leftarrow$ `getIntersections`$(\mathcal{B}^{\downarrow w}, \Omega_{\varphi}, k)$
5   **if not** `SAT`$(I_k \cup \{\neg\varphi\})$ **then**
6     $\tilde{\epsilon} \leftarrow$ `monolithic-explanation`(I_k, φ)
7   **return** $\tilde{\epsilon}$
8   **else**
9     $k \leftarrow k - 1$

probabilities of these worlds (i.e., $P(I_k) \geq \sum_{i=1}^{k} P(\omega_i)$). Therefore, $P(\tilde{\epsilon}) \geq P(I_k) \geq \sum_{i=1}^{k} P(\omega_i)$, meaning that the probability of $\tilde{\epsilon}$ has a maximal lower bound by the top k most-probable worlds of φ. □

The utility of a k-bounded monolithic explanation in computing probabilistic monolithic explanations can be described as follows. If we take the top k most-probable worlds in which the explanandum φ is true, then we can prune the search space of possible monolithic explanations by taking the intersection of those worlds and checking if it entails φ – if it does, then we know that at least one monolithic explanation must be true in that world with probability at least equal to the sum of the probabilities of these top k worlds. Building on this, we now present an algorithm for computing k-bounded monolithic explanations for φ from $\mathcal{B}$, where we use Algorithm 5.1 as our core monolithic explanation generation engine.

Algorithm 5.3 describes the main steps of our approach. The important factor is the user-defined parameter $\hat{k}$, which dictates the number of worlds of φ to be considered. It is an integer with range $1 \leq \hat{k} \leq |\Omega(\varphi)|$, where $\Omega(\varphi)$ is the set of all possible worlds of φ. Intuitively, the larger the $\hat{k}$, the more exhaustive the search will be as more worlds will be considered. The algorithm starts in line 1 with k taking the user-defined value $\hat{k}$, and then proceeds to line 2, where it uses a weighted MaxSAT solver to find the top k most-probable worlds of φ. Note that (φ, ∞) denotes that φ is added to the solver as a hard constraint. The main loop of the algorithm starts in line 3. In line 4, `getIntersections` extracts the set of intersecting formulae I_k from $\mathcal{B}^{\downarrow w}$ that are true in worlds ω_1 to ω_k. If $I_k \models \varphi$, then we know that a monolithic explanation is in I_k and the algorithm proceeds to use Algorithm 5.1 with I_k and φ as inputs to compute and return a monolithic explanation (lines 5-7). Otherwise, the algorithm discounts k by 1 and repeats the process until a suitable k is found.

Algorithm 5.3 is complete in the sense that, eventually, a monolithic explanation will be returned.

Theorem 8. *Algorithm 5.3 is guaranteed to terminate with a solution.*

Proof. The proof rests on the fact that, in the worst case, the parameter k will reach a value of 1. This will then correspond to the most-probable world of φ, which entails all possible monolithic explanations for φ. From Lemma 2, we know that the most-probable world of φ entails all possible monolithic explanations for φ, that is, for any $\tilde{E}(\varphi)$, $\omega_1 \models \tilde{\epsilon}$, and $\tilde{\epsilon} \subseteq I_1$. Therefore, as Algorithm 5.3 uses I_1 as an input to Algorithm 5.1, which is guaranteed to return a solution, the algorithm is also guaranteed to terminate with a solution.

□

Theorem 9. *Algorithm 5.3 is guaranteed to return a maximal k-bounded monolithic explanation if the user-defined parameter $\hat{k}$ is initialized to $|\Omega(\varphi)|$.*

Proof. First, note that if the user-defined parameter is initialized to $\hat{k} = |\Omega(\varphi)|$, then Algorithm 5.3 will perform an exhaustive and iterative search, starting from $k = |\Omega(\varphi)|$, to find I_k, such that $I_k \models \varphi$, and use it in Algorithm 5.1. Now, as the algorithm discounts k by 1 at each new iteration, eventually it will be the case that $I_k \models \varphi$ and $I_{k+1} \not\models \varphi$. From Corollary 2, we then know that $\exists \tilde{\epsilon} \subseteq I_k$ such that $P(\tilde{\epsilon}) \geq P(I_k) \geq \sum_{i=1}^{k} P(\omega_i)$, which means that $\tilde{\epsilon}$ corresponds to a k-bounded monolithic explanation. Therefore, the algorithm is guaranteed to return a maximal k-bounded monolithic explanation for φ. □

Model Reconciling Explanations

We now move on to the case of computing probabilistic model reconciling explanations $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ for an explanandum φ from an agent knowledge base KB_α for a human belief

Algorithm 5.4: `probabilistic-model-reconciling-explanation`$(\mathrm{KB}_\alpha, \mathcal{B}_h, \varphi, \hat{k})$

Input: Knowledge base KB_α, belief base $\mathcal{B}_h$, explanandum φ, and user-defined parameter $\hat{k}$

Result: A probabilistic model reconciling explanation $\tilde{\mathcal{E}} = \langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$ for φ from KB_α for $\mathcal{B}_h$

1 $k \leftarrow \hat{k}$

2 $\mathrm{KB}_\alpha^h \leftarrow \mathrm{KB}_\alpha \cap \mathcal{B}_h^{\downarrow w}$

3 $W \leftarrow \sum_{i=1}^{n} \{w_i \mid (\phi_i, w_i) \in \mathcal{B}_h\}$

4 $\mathcal{B}_\alpha \leftarrow \{(\phi, W) \mid \phi \in \mathrm{KB}_\alpha \setminus \mathrm{KB}_\alpha^h\}$

5 $\Omega_\varphi \leftarrow$ `getTopKWorlds`$(\mathcal{B}_h \cup \mathcal{B}_\alpha \cup \{(\varphi, \infty)\}, k)$

6 **while** *true* **do**

7     $I_k \leftarrow$ `getIntersections`$(\mathrm{KB}_\alpha, \Omega_\varphi, k)$

8   **if not** `SAT`$((I_k \cup \mathrm{KB}_\alpha^h \cup \{\neg\varphi\}))$ **then**

9         $\langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle \leftarrow$ `model-reconciling-explanation`$(I_k \cup \mathrm{KB}_\alpha^h, \mathcal{B}_h^{\downarrow w}, \varphi)$

10   **return** $\langle \tilde{\epsilon}^+, \tilde{\epsilon}^- \rangle$

11   **else**

12         $k \leftarrow k - 1$

base $\mathcal{B}_h$. Similarly to what we described for monolithic explanations, Algorithm 5.2 can directly be used on KB_α and $\mathcal{B}_h^{\downarrow w}$ for computing model reconciling explanations. Additionally, the concept of a k-bounded explanation (Definition 36) can also be used to guarantee a lower bound on the probability of $\tilde{\epsilon}^+$.

Algorithm 5.4 shows the pseudocode of our approach. The initial computational steps are similar to those in Algorithm 5.3, with the exception that KB_α is now also considered in the computation of the most-probable worlds of the φ. Specifically, in line 4, KB_α is converted into a belief base $\mathcal{B}_\alpha$ where each formula is given a weight that is larger than the sum of weights of $\mathcal{B}_h$. This is to enforce these formulae to be true in the worlds of the explanandum φ. Then, $\mathcal{B}_\alpha$ is used in conjunction with $\mathcal{B}_h$ to compute the top k most-probable worlds of φ (line 5). The algorithm proceeds in line 7 to extract formulae from KB_α that are true in the first k intersections of the worlds of φ. If they entail φ, the algorithm then proceeds to compute a model reconciling explanation by invoking Algorithm 5.2 (lines 8-9). Otherwise, the algorithm continues by discounting k by 1 and repeats the process.

Note that Algorithm 5.4 is complete and correct as it is based on Algorithms 5.2 and 5.3, which are complete and correct.

5.4 Computatonal Evaluations

This section presents a comprehensive evaluation of the algorithms presented in the previous section, assessing their effectiveness and efficiency across a range of scenarios.

5.4.1 Experimental Setup

Experiments were conducted on a system equipped with an M1 Max processor and 32GB of memory. The algorithms were implemented in Python, utilizing the PySAT toolkit [115] for SAT solving, MCS/MUS finding, weighted MaxSAT, and minimal hitting set computations. The time limit for all experiments was set to $500s$.

For our benchmarks, we selected a diverse set of problem instances:

- **Classical Planning Problems**: We encoded classical planning problems from the International Planning Competition (IPC) in the style of Kautz *et al.* [135], and used them as knowledge bases. The explanandum for each problem was the plan optimality query, which we constructed as described in Chapter 3 (Section 3.4.2).
- **Agent Scheduling Problems**: We encoded logic-based agent scheduling problems based on the description provided in Chapter 9, and used them as the knowledge bases. The explanandum for each problem was a set of unsatisfied agent constraints.
- **Random CNF Problems**: We generated random CNF formulae as knowledge bases using CNFgen [151]. The explanandum for each problem was a conjunction of backbone literals,[41] which we computed using the minibones algorithm proposed by Janota *et al.* [120].

Note that we created associated belief bases for each problem by simply adding a random weight to each formula in the knowledge base.

[41] The backbone literals of a propositional knowledge base are the set of literals entailed by the knowledge base.

Parameter $\hat{k}$	Planning			Scheduling			Random CNF		
	S	T/O	Runtime	S	T/O	Runtime	S	T/O	Runtime
1	28	9	82.0s	30	5	80.0s	25	5	12.4s
50	32	5	79.0s	30	5	53.8s	25	5	8.6s
100	31	6	49.6s	30	5	44.7s	25	5	5.8s
150	31	6	45.5s	30	5	38.0s	25	5	3.4s
200	31	6	45.2s	30	5	37.2s	25	5	1.6s

Table 5.1: Number of Instances Solved (S) vs. Timed Out (T/O) by ALG1 ($\hat{k} = 1$) and ALG3 ($\hat{k} = 50$, $\hat{k} = 100$, $\hat{k} = 150$, $\hat{k} = 200$).

5.4.2 Results and Discussion

We now describe and discuss our experimental results, first for monolithic explanations and then for model reconciling explanations.

Monolithic Explanations

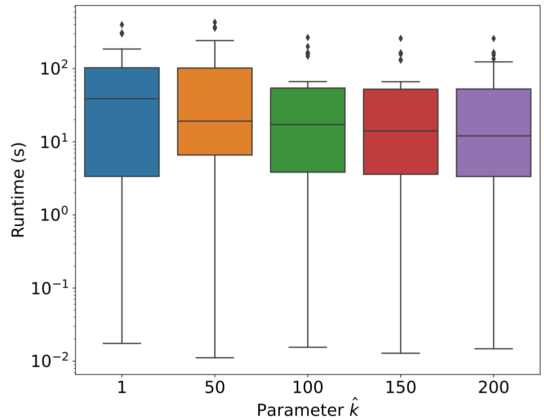
We evaluated Algorithm 5.1 and Algorithm 5.3, referred to as ALG1 and ALG3 respectively, on computing monolithic explanations. These experiments aim to answer the following questions:

Q1: What is the performance of the algorithms on computing monolithic explanations across different problem instances?

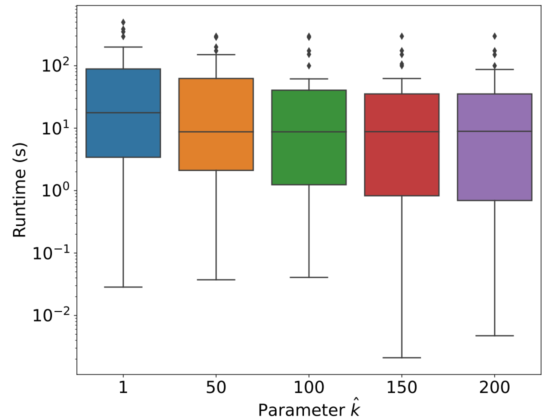
Q2: Does the efficacy of ALG3 change under different values of the user-defined parameter $\hat{k}$?

Table 5.1 tabulates the instances solved (i.e., found a monolithic explanation within the time limit) and not solved (i.e., timed out) by ALG1 ($\hat{k} = 1$) and ALG3 at $\hat{k} = \{5, 100, 150, 200\}$.[42] We observe that the algorithm managed to solve most instances across different values of $\hat{k}$. Figure 5.1 shows the runtime distributions of ALG3 across all values of $\hat{k}$ for computing a monolithic explanation. Interestingly, we observe that the runtimes decrease as $\hat{k}$ increases. This can be explained by the fact that for larger values of $\hat{k}$, ALG3 considers the intersections

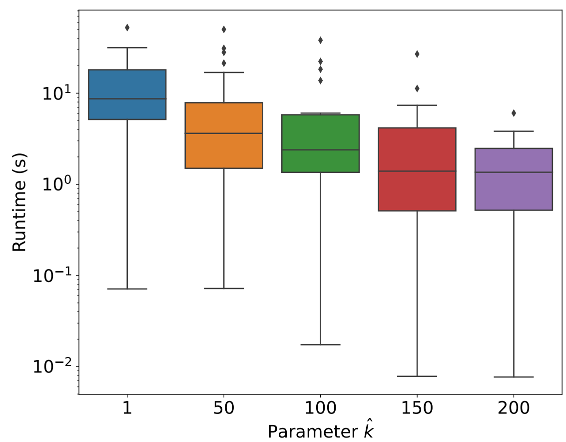
[42] ALG3 at $\hat{k} = 1$ corresponds to ALG1 because each encoded knowledge base is consistent and entails the explanandum. As such, all formulae in the knowledge base are true in the most-probable world of the explanandum (i.e., $\hat{k} = 1$), which means that ALG3 reduces to ALG1.



(a) Planning Instances.



(b) Scheduling Instances.



(c) Random Instances.

Figure 5.1: Runtime distributions of ALG1 ($\hat{k} = 1$) and ALG3 ($\hat{k} = 50$, $\hat{k} = 100$, $\hat{k} = 150$, $\hat{k} = 200$) across all planning, scheduling, and random CNF instances.

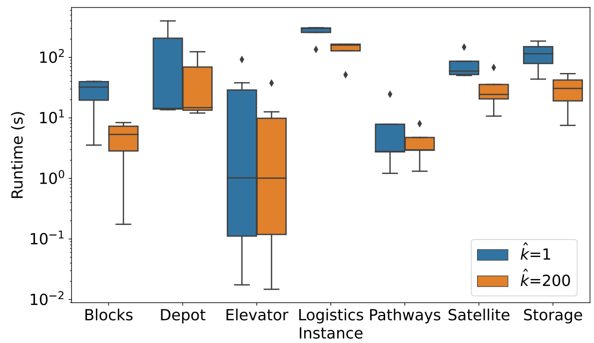
of more worlds where the explanandum is true, which means that the number of formulae that are true in these intersections decreases. As such, the overall search space of monolithic explanations decreases as well, thus resulting in a reduced runtime needed for ALG1 to compute a monolithic explanation. This can also be observed more granularly in Figure 5.2, where we can see the runtime distributions of ALG1 ($\hat{k} = 1$) and ALG3 at $\hat{k} = 200$ for each instance of the planning, scheduling, and random CNF problems. Again, the runtime of ALG3 at $\hat{k} = 200$ is smaller than that of ALG1. Moreover, and as expected, in Figure 5.3, we can observe a positive correlation between runtime and the size of the encoded knowledge bases – as the size of the knowledge base increases, the runtimes increase as well. This is due to the fact that there is an increasing number of variables and formulae that must be considered, thus increasing the computational effort needed by the WMaxSAT, MCS, and hitting set solvers.

All of these observations indicate the feasibility and practical efficacy of ALG3 across all benchmarks. In particular, from these experiments, we may conclude that the performance of ALG3 increases as the user-defined parameter $\hat{k}$ increases. To reiterate, this is mainly because the overall search space of monolithic explanations that needs to be considered by ALG1 (the main monolithic explanation generation engine) decreases. Finally, it is important to note that the performance of these algorithms lies in the effectiveness of the underlying WMaxSAT, MCS, and hitting set solvers. In other words, this also implies that any advancement in those solvers will automatically reflect in performance gains in our algorithms. Thus, future work can look at efficient and optimized solvers and examine whether there is any variability in performance.

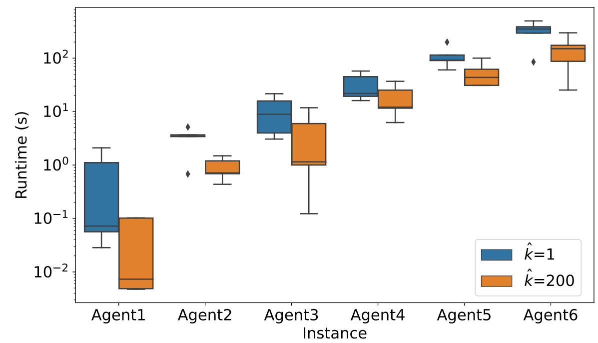
Model Reconciling Explanations

We now examine the effectiveness of Algorithm 5.2, referred to as ALG2, and Algorithm 5.4, referred to as ALG4, on computing model reconciling explanations. We chose the value of $\hat{k} = 200$ for ALG4 as it was the better performing parameter for ALG3 in our previous experiments. More specifically now, we are interested in scenarios with varying degrees of knowledge asymmetry between the agent and human models. To simulate such scenarios, we used the actual encoded knowledge bases as the model of the agent (KB_α), and tweaked that model and assigned it to be the model of the human (KB_h or $\mathcal{B}_h$). We considered the following ways to tweak the human model, resulting in the following five scenarios:

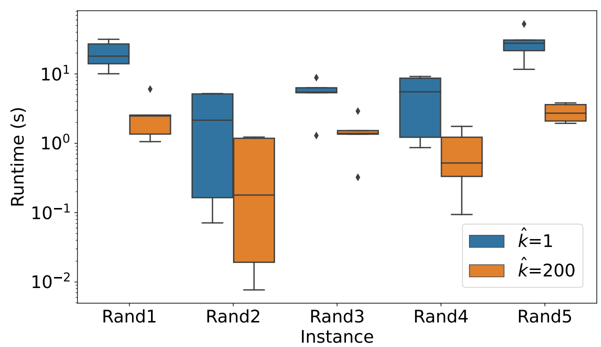
- **Scenario 1:** We randomly removed 10% of the formulae and removed 20% of literals from 10% of the total formulae in the human's model.
- **Scenario 2:** We randomly removed 20% of the formulae and removed 20% of literals from 20% of the total formulae in the human's model.
- **Scenario 3:** We randomly removed 30% of the formulae and removed 20% of literals from 30% of the total formulae in the human's model.



(a) Planning Instances.



(b) Scheduling Instances.



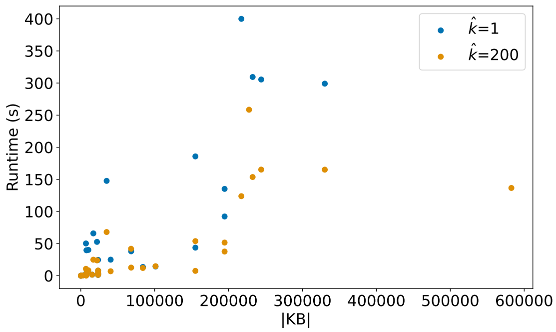
(c) Random Instances.

Figure 5.2: Runtime distributions of ALG1 ($\hat{k} = 1$) and ALG3 ($\hat{k} = 200$) across commonly solved planning, scheduling, and random CNF instances.

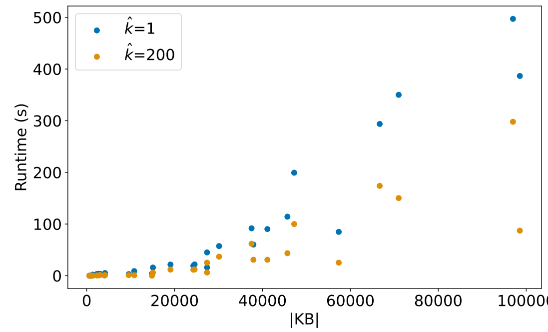
- **Scenario 4:** We randomly removed 40% of the formulae and removed 20% of literals from 40% of the total formulae in the human's model.
- **Scenario 5:** We randomly removed 50% of the formulae and removed 20% of literals from 50% of the total formulae in the human's model.

In general, these experiments aim to answer the following two questions:

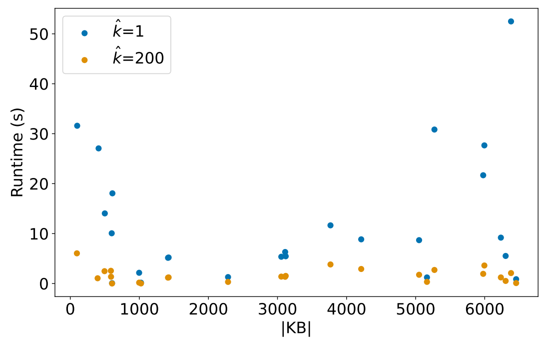
> **Q1**: What is the performance of the algorithms on computing model reconciling explanations across different problem instances?
>
> **Q2**: What is the performance of the algorithms in scenarios with varying degrees of knowledge asymmetry between the agent and the human model?



(a) Planning Instances.



(b) Scheduling Instances.



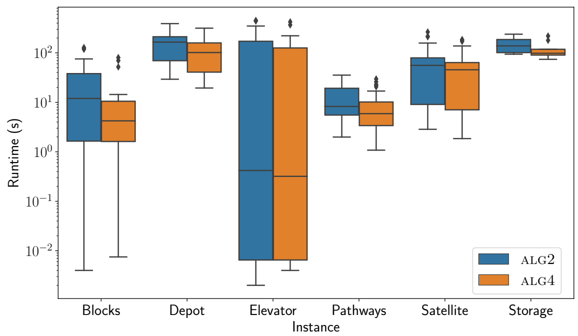
(c) Random Instances.

Figure 5.3: Average runtime of ALG1 ($\hat{k} = 1$) and ALG3 ($\hat{k} = 200$) to compute an explanation across different knowledge base sizes for the planning, scheduling, and random CNF instances.

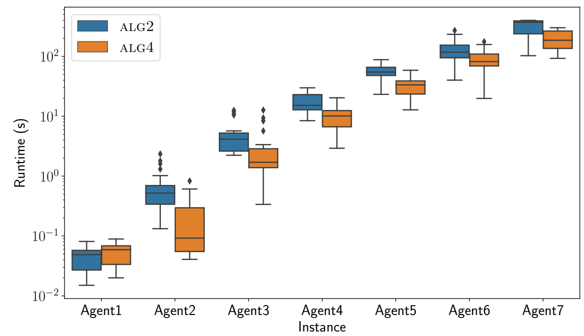
Scenario	Planning						Scheduling						Random CNF					
	ALG2			ALG4			ALG2			ALG4			ALG2			ALG4		
	S	T/O	Runtime	S	T/O	Runtime	S	T/O	Runtime	S	T/O	Runtime	S	T/O	Runtime	S	T/O	Runtime
1	25	12	67.0s	28	9	59.7s	33	2	51.4s	33	2	33.1s	27	5	30.4s	21	11	12.3s
2	25	13	69.2s	27	10	71.8s	31	4	40.9s	31	4	28.0s	26	6	18.7s	20	12	0.5s
3	24	14	67.9s	26	12	68.8s	32	3	60.3s	32	3	37.1s	29	3	20.4s	21	11	2.5s
4	25	13	82.6s	27	11	84.0s	30	4	35.7s	30	4	22.9s	23	9	5.2s	20	12	0.5s
5	22	15	84.3s	24	13	89.9s	30	4	34.4s	30	4	21.5s	24	8	3.8s	20	11	0.5s

Table 5.2: Instances Solved (S) vs. Timed Out (T/O) for the Planning, Scheduling, and Random CNF Benchmarks for ALG2 and ALG4 at $\hat{k} = 200$.

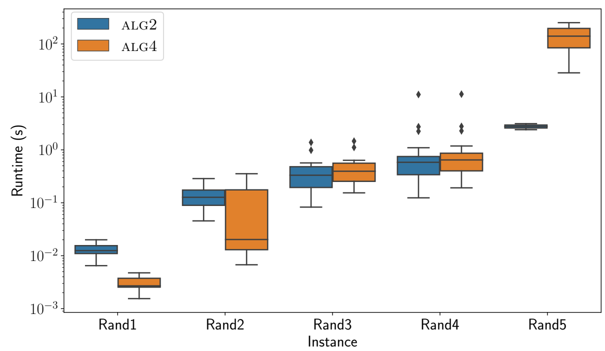
Table 5.2 tabulates the instances solved and timed out by ALG2 and ALG4 at $\hat{k} = 200$ across the five scenarios, where we observe the following trends. For the planning instances, the runtime of both algorithms increases as the difference between the models of the agent and human increases (Scenarios 1 to 5), since both algorithms search over the explanation search space, which increases as the number of differences between the two models increases. As in the previous experiments, ALG4 at $\hat{k} = 200$ yields faster runtimes than ALG2. For the



(a) Planning Instances.



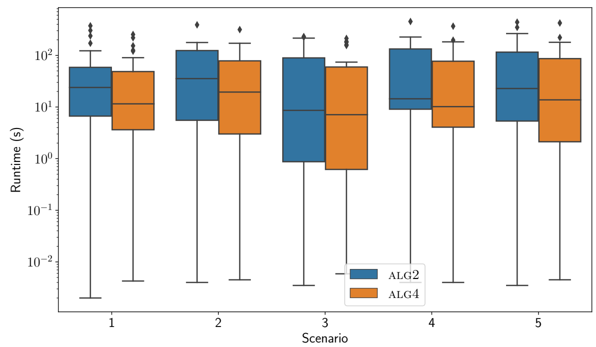
(b) Scheduling Instances.



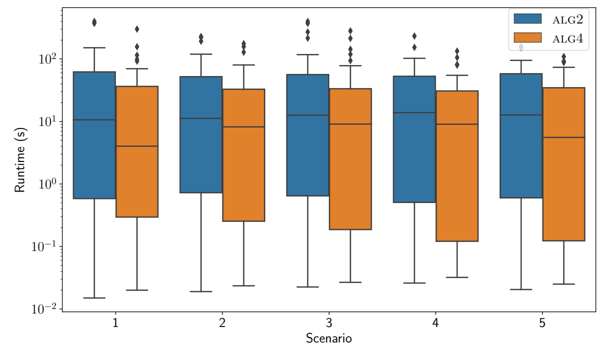
(c) Random Instances.

Figure 5.4: Runtime distributions of ALG2 and ALG4 at $\hat{k} = 200$ to compute an explanation across all commonly solved instances.

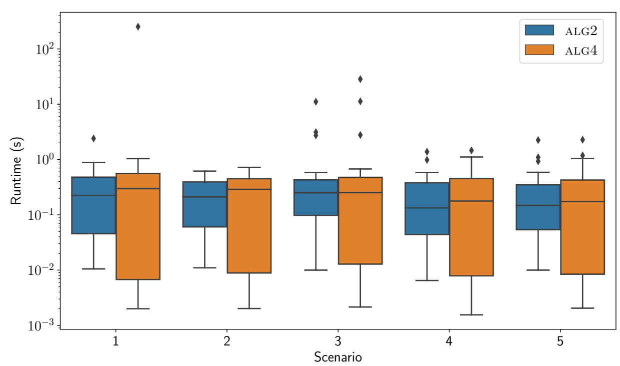
scheduling instances, we observe that the runtimes increase from Scenario 1 to 3, but decrease from Scenarios 4 to 5. Upon closer inspection, this is mainly because the instances solved in these scenarios were easier (i.e., smaller knowledge base sizes) than those solved in the other three scenarios, thus resulting in smaller average runtimes. A similar trend is observed for the random CNF instances. However, in the random CNF instances, ALG2 managed to solve more instances than ALG4. After examining them more closely, we found that the main bottleneck of ALG4 in those instances was computing the most-probable worlds of the explanandum (i.e., the WMaxSAT solver). Even for smaller values of $\hat{k}$, the solver failed to compute all the worlds under the specified time limit – the increase in search space (e.g., because of considering $\mathcal{B}_\alpha$ and $\mathcal{B}_h$) increased the complexity of these instances. We expect that an optimized and more dedicated solver may be able to overcome this limitation. The runtime distributions for ALG2 and ALG4 across all commonly solved instances and across commonly solved instances in each scenario can be seen in Figures 5.4 and 5.5, respectively.



(a) Planning Instances.



(b) Scheduling Instances.

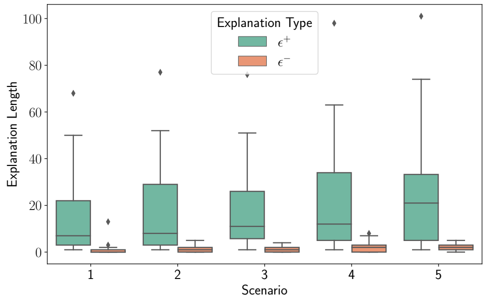


(c) Random Instances.

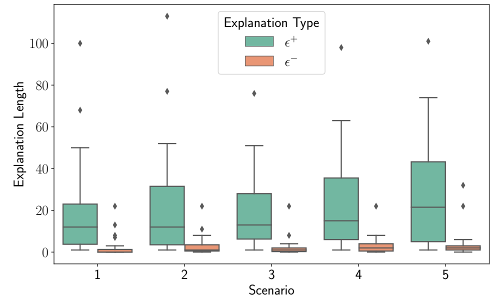
Figure 5.5: Runtime distributions of ALG2 and ALG4 at $\hat{k} = 200$ to compute an explanation across commonly solved instances in each of the five scenarios.

For these instances, we observe, like in the previous experiments, that ALG4 has faster runtimes than ALG2.

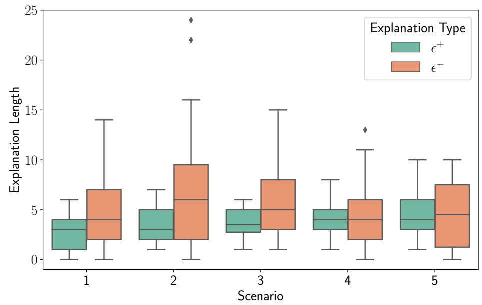
Moreover, in Figure 5.6 we see the distributions of the model reconciling explanation lengths computed by both algorithms. As expected, the general trend is that the size of the explanation ϵ^+ (i.e., formulae from KB_α for KB_h (or $\mathcal{B}_h$)) increases with each scenario, as the difference between the agent and human models increase. The same trend can be seen for ϵ^- – each scenario from 1 to 5 has an increasing amount of inconsistencies between the two models. Interestingly, ϵ^- was largest in the random CNF instances. This indicates that the inconsistencies between the human model and the corresponding ϵ^+ were high. That can also be used to explain why ALG4 failed to solve a subset of random CNF instances – highly inconsistent knowledge bases are considered as the most difficult instances for MaxSAT solvers.



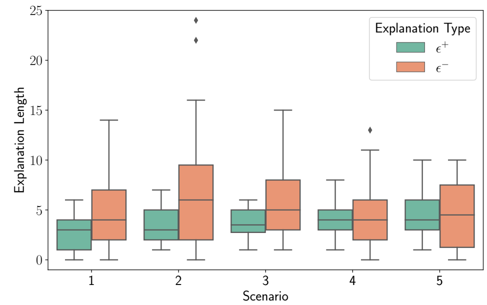
(a) ALG2 on Planning Instances.



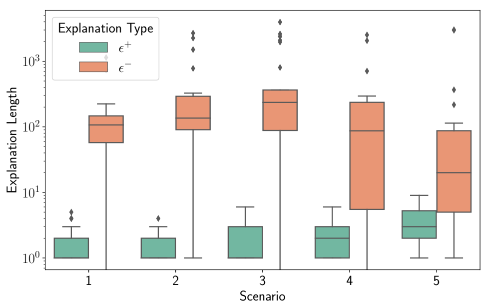
(b) ALG4 on Planning Instances.



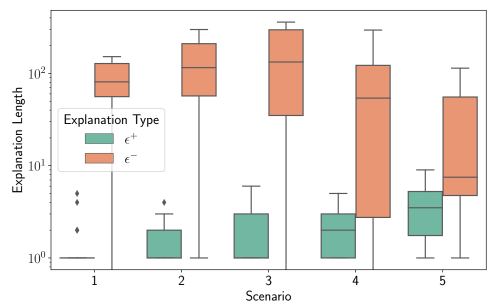
(c) ALG2 on Scheduling Instances.



(d) ALG4 on Scheduling Instances.



(e) ALG2 on Random CNF Instances.



(f) ALG4 on Random CNF Instances.

Figure 5.6: Distributions of the lengths of explanations ϵ^+ and ϵ^- computed by ALG2 and ALG4 at $k = 200$ across all planning, scheduling, and random CNF instances.

In conclusion, the comparative analysis of ALG2 and ALG4 at $\hat{k} = 200$ across varied problem instances shows some trends in performance and computational complexity. The observed increase in runtime with the increase of differences between agent and human models underscores the direct relationship between model disparity and the explanation search space size. Notably, ALG4 consistently outperforms ALG2 in terms of runtime across most scenarios, except in certain random CNF instances where the computation of most-probable worlds becomes a bottleneck due to the limitations of the WMaxSAT solver. This highlights a potential area for further optimization and development of more efficient solvers. Furthermore, the analysis of model reconciling explanation lengths reveals an expected increase in inconsistency measures as the model differences widen, particularly highlighted in random CNF instances.

5.5 Related Work

The algorithms presented in this chapter are inspired by a procedure for computing an SMUS of an inconsistent formula, originally presented by Ignative *et al.* [117]. The method is also related to other similar approaches for enumerating MUSes and MCSes. Moreover, our approach is similar in spirit to the HS-tree presented by Reiter [189]. Although the original purpose was to enumerate diagnoses, Reiter's procedure can be easily adapted to enumerate MUSes (called conflicts in that paper) as already noted by Previti *et al.* [186]. However, the computation of an SMUS might require more substantial modifications. Procedures like the one presented by Reiter, which target MCSes (diagnoses) instead of MUSes (conflicts), can be seen as the dual version of our algorithm. In particular, the algorithm MaxHS [51] applies the same idea of iteratively computing and testing a minimum hitting set for the computation of a MaxSAT solution (the complement of the smallest MCSes). Finally, there are other approaches that exploit the duality between MUSes and MCSes, but instead of iteratively checking if the current hitting set is an MUS, they first compute the set of all MCSes [156]. This has the potential advantage that once all the MCSes are known, every minimal hitting set on the collection of all MCSes is guaranteed to be an MUS (Proposition 3). However, as the number of MCSes is, in the worst case, exponential in the size of the formula, this approach might fail even before reporting the first MUS. This is particularly unnecessary when the target is to return a single explanation.

5.6 Concluding Remarks

This chapter presented efficient algorithms for our logical explainability layer that bridge the gap between theory and practice. By leveraging the duality between minimal correction sets (MCSes) and minimal unsatisfiable subsets (MUSes), we developed methods for generating explanations in both deterministic and probabilistic settings. Our experimental evaluations have demonstrated how these algorithms enable our approach to scale across various domains, extending beyond our initial focus on planning problems.

The key contributions of this chapter are:

1. Algorithms for generating explanations using MCS-MUS duality in deterministic settings.

2. Extensions of these algorithms to probabilistic settings.

3. Comprehensive experimental evaluation demonstrating the scalability of our approach across diverse benchmarks.

It is important to note that while our framework generates explanations in a logical format, these can be translated into more user-friendly formats, such as visualizations and natural language, before being communicated to human users. Indeed, in [146] we considered visualization techniques for effectively communicating explanations to users in a visual format. Moreover, in Chapter 10, we will show how our algorithms can be integrated with large language models to generate natural language explanations, demonstrating the flexibility of our approach in supporting different communication modalities.

Looking ahead, several promising directions could further enhance the practical impact of our algorithms. First, we could explore approximation algorithms for weighted MaxSAT to improve efficiency when computing most-probable worlds, particularly for large knowledge bases. Similarly, heuristic-based approaches for MCS and MUS computation could provide faster explanations for time-sensitive applications while maintaining the logical foundation of our approach. Implementing our algorithms in parallel and distributed computing environments could also enhance the performance for large-scale problems, i.e., by distributing the computation of MCSes and MUSes across multiple processors. Lastly, while we focused on propositional logic, future work could investigate how to adapt our algorithms to other logical systems, such as first-order logic, temporal logic, or description logics. This expansion could broaden the applicability of our approach to a wider range of AI systems and problem domains.

In the next chapter, we will build upon these computational foundations to develop methods for generating personalized explanations. By incorporating abstractions based on user-specified vocabularies, we will show how our logical explainability layer can adapt its explanations to match different users' levels of expertise and understanding.

Chapter 6

Generating Personalized Explanations via Knowledge Forgetting

> *"The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise."*
>
> *— Edsger Dijkstra*

6.1 Introduction & Contribution

Thus far, we have established logic as an effective explainability layer for AI systems, and developed frameworks for generating explanations in both deterministic and probabilistic settings, and providing efficient algorithms for their computation. However, a key challenge remains: how can we ensure our logical explainability layer adapts to different users' levels of understanding? While existing Model Reconciliation Problem (MRP) approaches, including our L-MRP, assume the AI agent has access to a version of the human's model at the same granularity level as the agent's model [37, 205, 36, 231], this assumption potentially limits their practical effectiveness.

The challenge lies in the potential divergence between the agent's version of the human model and the actual human's model, particularly in terms of abstraction level. This discrepancy can lead to the generation of incoherent or unintelligible explanations. Moreover, if the agent lacks confidence in its estimate of the human model, a conservative approach would be to assume the human model is almost empty, resulting in unnecessarily long explanations.

To address these limitations, this chapter proposes a novel approach:

- We assume the agent has access to a vocabulary of task-specific terms known to the human user.
- The agent generates explanations with respect to that vocabulary.

This assumption represents a reasonable compromise between the overly pessimistic view that the human model is almost empty and the overly optimistic view that it is mostly specified. Furthermore, this approach can complement our previous work by integrating with human models that capture information the agent is confident about, leveraging the efficient algorithms developed in the previous chapter.

To illustrate this concept, consider the classical LOGISTICS domain [167]. The vocabulary of the human may include different trucks (e.g., `truck1`, `truck2`) and locations (e.g., `loc1`, `loc2`), and their partial model includes the action dynamics of the `move` operator for trucks. One advantage of this approach is that the vocabulary implicitly encodes the human's knowledge or expertise level of the given task. For instance, the more (or less) terms included in the vocabulary, the higher (or lower) the human's level of expertise, to the extent that a human expert probably knows more task-specific terms than a novice one, all else being equal. Continuing with the example above, the human user is knowledgeable about trucks, but is unaware of the existence of airplanes. The agent could then exploit the vocabulary and construct explanations tailored to the human's level.

Our framework builds on the logic-based variant of MRP (L-MRPs) introduced in Chapter 3. Given an agent knowledge base KB_a encoding a task, an explanandum φ entailed by KB_a, a (possibly partial or empty) human knowledge base KB_h, and a human vocabulary $\mathcal{V}_h$ consisting of task-specific terms, our goal is to find an explanation at an appropriate abstraction level with respect to KB_h and $\mathcal{V}_h$. To achieve this, we employ *knowledge forgetting* [22, 222], a fundamental logic-based operation, to generate abstractions. We then formally define the notion of *personalized explanations* and present an algorithm that can be combined with any off-the-shelf L-MRP approaches for computing them.

The main contributions of this chapter are:

1. A logic-based framework for generating explanations at appropriate abstraction levels with respect to a human vocabulary.

2. Formalization of various settings under which personalized explanations can be generated.

3. An algorithmic approach for computing personalized explanations.

4. Empirical evaluation of our framework on representative benchmarks and through a human-subject study.

This work shows how our logical explainability layer can adapt to different users' needs while maintaining its formal foundations, making it more practical for real-world applications where users have varying levels of expertise.

6.2 Knowledge Forgetting

Knowledge forgetting, henceforth forgetting, has an ordering function in the human mind – it can be seen as a process of omitting information or knowledge from one's memory in such a way that it is no longer present or reproducible. From a cognitive point of view, forgetting is a gradual process in which information that is less used is moved to the "background," from which it either dissipates or recovered through remembering to the foreground [67]. This basic mechanism helps people deal with information overload by suppressing irrelevant information, which allows them to focus on the relevant aspects of a given task, thus improving their cognitive capabilities. For example, when people are trying to focus on a specific task, they tend to "forget" irrelevant aspects around it, or when trying to find a solution under restricted conditions, they have to intentionally "forget" ways of solving the task in more granular environments [121]. This point to the fact that intentional forgetting is a fundamental cognitive process involving many aspects of knowledge and reasoning.

Interestingly, the operation of forgetting aligns with a pragmatic framework in cognitive linguistics called *relevance theory* [237]. Relevance theory suggests that the relevance of a statement transmitted to an individual should minimize their cognitive effort (i.e., effort in processing the statement) and maximize their positive cognitive effect (i.e., the statement leads to a true conclusion). In other words, the more positive the cognitive effects and the less the cognitive effort, the greater the relevance of the statement to the individual. The connection we ought to draw here is that forgetting can be seen as an operation for achieving the objectives suggested by relevance theory, so far as forgetting irrelevant information from a statement can decrease the individual's effort, and by focusing only on what is relevant, yield a positive effect. In the sequel, we look at forgetting from the lens of logic and show how it can be used for that purpose.

6.2.1 Logic-based View of Knowledge Forgetting

Analogous to the cognitive operation of forgetting, which aims at suppressing information from an agent's memory, the logic-based operation of forgetting aims at removing information from an agent's knowledge base. Forgetting has received many logical definitions and interpretations, starting in the mid 1800s with Boole's variable elimination method [22]. For a historical overview of forgetting in logic and AI, we refer the reader to the work by Van Ditmarsch *et al.* [222].

Generally, forgetting is defined through an operation that decreases the language of an agent, insofar as the vocabulary of the agent's language is reduced. Intuitively, forgetting information from an agent's knowledge base that encodes a specific domain affects the agent's ability to express or represent information about that domain, rather than losing information about the domain per se.

Delgrande [57] presents a resolution-based mechanism for computing forgetting, where given a knowledge base KB defined over vocabulary $\mathcal{V}_{\mathrm{KB}}$, the operation of forgetting $\lambda \subseteq \mathcal{V}_{\mathrm{KB}}$ from KB is the logical consequences of KB expressible over $\mathcal{V}_{\mathrm{KB}} \setminus \lambda$.

Given a knowledge base KB and a letter $\lambda \in \mathcal{V}_{\mathrm{KB}}$ in its vocabulary, let $\mathrm{KB}^{\downarrow\lambda}$ and $\mathrm{KB}^{\uparrow\lambda}$ denote the sets of formulae of KB that do not mention λ and do mention λ, respectively:

$$\mathrm{KB}^{\downarrow\lambda} = \{\varphi \in \mathrm{KB} \mid \lambda \notin \mathcal{V}_\varphi\} \tag{6.1}$$

$$\mathrm{KB}^{\uparrow\lambda} = \{\varphi \in \mathrm{KB} \mid \lambda \in \mathcal{V}_\varphi\} \tag{6.2}$$

Additionally, let $Res(\mathrm{KB}^{\uparrow\lambda}, \lambda)$ denote the set of formulae obtained from $\mathrm{KB}^{\uparrow\lambda}$ by carrying out all possible resolutions with respect to letter λ:

$$\begin{gathered} Res(\mathrm{KB}^{\uparrow\lambda}, \lambda) = \{\varphi \mid \exists \varphi_1, \varphi_2 \in \mathrm{KB}^{\uparrow\lambda} \text{ s.t.} \\ \lambda \in \varphi_1, \neg\lambda \in \varphi_2, \varphi = (\varphi_1 \setminus \{\lambda\}) \cup (\varphi_2 \setminus \{\neg\lambda\})\} \end{gathered} \tag{6.3}$$

Combining those definitions, we get the definition of forgetting:

Definition 37 (Forgetting)**.** *Given a knowledge base* KB *and a letter* $\lambda \in \mathcal{V}_{\mathrm{KB}}$ *in its vocabulary,* forgetting λ *from* KB *is defined as* $\mathcal{F}(\mathrm{KB}, \lambda) = \mathrm{KB}^{\downarrow\lambda} \cup \mathit{Res}(\mathrm{KB}^{\uparrow\lambda}, \lambda)$.[43]

Definition 37 can be interpreted as follows: Perform all possible resolutions with respect to the letter to be forgotten, and add these resolvents to those formulae in KB that do not mention that letter. While the resulting KB is weaker than before as it loses its expressivity with respect to what is forgotten, one key advantage is that it still entails the same set of formulae that are irrelevant to what was forgotten:

Property 1. *If* $\mathrm{KB} \models \varphi$*, then* $\forall \lambda \subseteq \mathcal{V}_{\mathrm{KB}} \setminus \mathcal{V}_{\varphi}$*,* $\mathcal{F}(\mathrm{KB}, \lambda) \models \varphi$*.*

Example 21. *Let* $\mathrm{KB} = \{a, b, \neg a \vee c, \neg b \vee \neg c \vee d\}$ *with* $\mathcal{V}_{\mathrm{KB}} = \{a, b, c, d\}$*. Notice that* $\mathrm{KB} \models d$*. For* $\lambda = \{a\}$*, we get* $\mathrm{KB}^{\downarrow a} = \{b, \neg b \vee \neg c \vee d\}$ *and* $\mathrm{KB}^{\uparrow a} = \{a, \neg a \vee c\}$*, and* $\mathit{Res}(\mathrm{KB}^{\uparrow a}, a) = \{c\}$*. Then,* $\mathcal{F}(\mathrm{KB}, \{a\}) = \{b, c, \neg b \vee \neg c \vee d\}$*, where* $\mathcal{F}(\mathrm{KB}, \{a\}) \models d$*.*

Abstractions via Forgetting: As seen from the example above, the forgetting operation can be viewed as a method for simplifying formulae by "forgetting" a set of letters. In essence, if we define an abstraction of a knowledge base as simplifying it, then forgetting is a succinct operation for computing various abstraction levels:

Definition 38 (Abstraction)**.** *Given a knowledge base* KB *and a set of letters* $\lambda \subseteq \mathcal{V}_{\mathrm{KB}}$ *in its vocabulary, a level-*$|\lambda|$ abstraction *of* KB *is* $\mathcal{F}(\mathrm{KB}, \lambda)$*.*

We can now create an *abstraction lattice* defining the abstraction levels that can be achieved on a knowledge base given a set of letters. Figure 6.1 shows a level-3 abstraction lattice based on Example 21. As we will see in the next section, generating personalized explanations boils down to finding the appropriate abstraction level with respect to a set of letters (i.e., the human-specified vocabulary).

[43] Note that computing forgetting for a set of letters can be done iteratively (i.e., $\mathcal{F}(\mathrm{KB}, \lambda_1 \cup \lambda_2) = \mathcal{F}(\mathcal{F}(\mathrm{KB}, \lambda_1), \lambda_2)$).

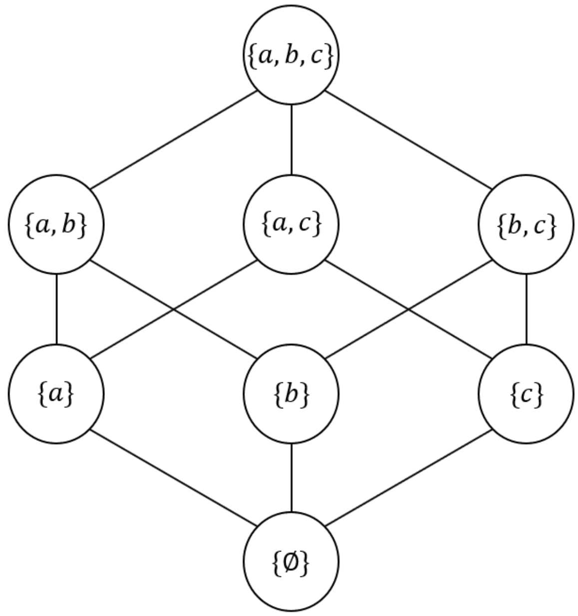


Figure 6.1: Abstraction lattice for $\text{KB} = \{a, b, \neg a \vee c, \neg b \vee \neg c \vee d\}$. At the root is level-0 of the lattice, i.e., the initial $\mathcal{F}(\text{KB}, \{\emptyset\}) = \text{KB}$. The child nodes of the root form level-1 of the lattice and represent (from left to right): $\mathcal{F}(\text{KB}, \{a\}) = \{b, c, \neg b \vee \neg c \vee d\}$, $\mathcal{F}(\text{KB}, \{b\}) = \{a, \neg a \vee c, \neg c \vee d\}$, and $\mathcal{F}(\text{KB}, \{c\}) = \{a, b, \neg a \vee \neg b \vee d\}$. Similarly, the subsequent nodes form level-2, and so on.

6.3 Personalized Explanation Generation

Our framework builds upon the *logic-based model reconciliation problem* (L-MRP) we presented in Chapter 3, where we make the following assumptions:

- The agent has a knowledge base KB_a encoding its knowledge of a task in a logical language. The agent's knowledge base KB_a is logically closed, insofar as the agent is "logically omniscient" about the task.
- The agent has a knowledge base KB_h encoding, possibly incompletely or erroneously, the human user's knowledge of the same task in the same logical language. It is possible for $\text{KB}_h = \emptyset$.
- The human user provides to the agent: (i) An explanandum φ, where $\text{KB}_a \models \varphi$ and $\text{KB}_h \not\models \varphi$, and (ii) a vocabulary $\mathcal{V}_h$. Naturally, $\mathcal{V}_{\text{KB}_h} \subseteq \mathcal{V}_h$ as all the terms in the human model must be in their vocabulary. However, note that *it is possible for the vocabulary to*

include terms that are not in the human model. This is akin to knowing a particular term, but not knowing how it relates to the task or what it really means.

Thus, given the knowledge bases KB_a and KB_h, the corresponding human vocabulary $\mathcal{V}_h$, and an explanandum φ such that $\mathrm{KB}_a \models \varphi$ and $\mathrm{KB}_h \not\models \varphi$, the goal is to find an L-MRP explanation from KB_a to KB_h for φ that is *at an appropriate abstraction level with respect to* $\mathcal{V}_h$. As already mentioned, we will call such an explanation a *personalized explanation.*

The central question behind this setting is what is an appropriate abstraction level. Clearly, an appropriate abstraction level should not contain any irrelevant information with respect to the explanandum:

Definition 39 (Irrelevance). *Given a knowledge base* $\mathrm{KB}_a \models \varphi$*, a set of letters* $\lambda \subseteq \mathcal{V}_{\mathrm{KB}_a}$ *from its vocabulary is* irrelevant *for* KB_a *with respect to* φ *iff* $\mathcal{F}(\mathrm{KB}_a, \lambda) \models \varphi$.

We say that a set of letters λ is irrelevant for KB_a with respect to a formula φ if forgetting λ from KB_a does not affect the entailment of φ in the resulting KB_a. In our context, this definition is easily satisfied by assuming that λ does not contain any letters from the explanandum φ (see Property 1). We enforce this property in our proposed algorithm, which we describe later.

Further, a personalized explanation is not really "personalized" unless it uses at least some letters familiar to the human (i.e., letters from the vocabulary $\mathcal{V}_h$). Naively, one could forget all letters from $\mathcal{V}_{\mathrm{KB}_a}$ except for those in $\mathcal{V}_h$ (and $\mathcal{V}_\varphi$). However, this may result in overly short and trivial explanations of the form "why φ? Because φ", which is the case when $\mathrm{KB}_h = \emptyset$ and $\mathcal{V}_h \cap \mathcal{V}_{\mathrm{KB}_a} = \emptyset$.

Therefore, to avoid forgetting too many letters and oversimplifying explanations to the point that they are trivial, we propose that the goal of forgetting as many letters as needed to get an (L-MRP) explanation is of reasonable complexity. The complexity of explanations can be measured in a variety of ways, including with all the various cost functions (e.g., subset-minimality, cardinality, etc.) previously discussed in Section 3.4.1 of Chapter 3.

Without loss of generality, we will assume that the choice of complexity measure is the cardinality of the explanation. While we continue our description, provide theoretical properties, and propose an algorithm based on this assumption, it is fairly straightforward to see how they can be generalized to other complexity measures as well.

When the choice of complexity measure is explanation cardinality, the constraint that needs to be satisfied is:

$$|\mathcal{F}(\epsilon^+, \lambda)| + |\epsilon^-| \leq \mathcal{UB} \tag{6.4}$$

where $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_h \cup \mathcal{V}_\varphi)$ is the set of letters to forget and $\mathcal{UB}$ is a user-specific maximum cardinality of an explanation. Note that the λ does not include any letters in the vocabulary $\mathcal{V}_h$ because the goal is to personalize explanations by using terms known to the human user. Additionally, λ does not include any letters in $\mathcal{V}_\varphi$ because they are needed to ensure that the updated KB_h of the user entails the explanandum φ (Property 1). Finally, no letters are forgotten from $\epsilon^- \subseteq \mathrm{KB}_h$ because they are all in the vocabulary $\mathcal{V}_h$ of the human user by definition. More formally, extending Definition 19:

Definition 40 (Personalized L-MRP Explanation)**.** *Given knowledge bases* $\mathrm{KB}_a \models \varphi$ *and* $\mathrm{KB}_h \not\models \varphi$*, vocabulary* $\mathcal{V}_h$*, and upper bound* $\mathcal{UB}$*,* $\epsilon = \langle \epsilon^+, \epsilon^- \rangle$ *is a* personalized L-MRP explanation *for* φ *from* KB_a *to* KB_h *with respect to* $\mathcal{V}_h$ *iff* $\epsilon^+ \subseteq \mathrm{KB}_a$*,* $\epsilon^- \subseteq \mathrm{KB}_h$*,* $\lambda \in \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h)$*,* $|\mathcal{F}(\epsilon^+, \lambda)| + |\epsilon^-| \leq \mathcal{UB}$*, and* $(\mathrm{KB}_h \cup \mathcal{F}(\epsilon^+, \lambda)) \setminus \epsilon^- \models \varphi$*.*

Given Definitions 19 and 40 together with Property 1, we can then show that if $\langle \epsilon^+, \epsilon^- \rangle$ is an L-MRP explanation for φ from KB_a to KB_h, then $\langle \mathcal{F}(\epsilon^+, \lambda), \epsilon^- \rangle$ is a personalized L-MRP explanation for φ from KB_a to KB_h for any $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h)$ if its cardinality is no larger than a given upper bound $\mathcal{UB}$. More formally:

Theorem 10. *Given two knowledge bases* $\mathrm{KB}_a \models \varphi$ *and* $\mathrm{KB}_h \not\models \varphi$ *with a corresponding L-MRP explanation* $\langle \epsilon^+, \epsilon^- \rangle$ *for explanandum* φ*, for any set of letters* $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h)$ *and an upper bound* $\mathcal{UB}$*,* $\langle \mathcal{F}(\epsilon^+, \lambda), \epsilon^- \rangle$ *is a personalized L-MRP explanation for the same explanandum* φ *if its cardinality* $|\mathcal{F}(\epsilon^+, \lambda)| + |\epsilon^-| \leq \mathcal{UB}$ *is no larger than* $\mathcal{UB}$*.*

Proof. Assume $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h)$, which is the premise of the theorem. Then:

$$\begin{aligned}
\mathcal{F}((\mathrm{KB}_h \cup \epsilon^+) \setminus \epsilon^-, \lambda) &= \mathcal{F}((\mathrm{KB}_h \setminus \epsilon^-) \cup (\epsilon^+ \setminus \epsilon^-), \lambda) && (6.5)\\
&= \mathcal{F}((\mathrm{KB}_h \setminus \epsilon^-) \cup \epsilon^+, \lambda) && (6.6)\\
&= \mathcal{F}(\epsilon^+ \cup (\mathrm{KB}_h \setminus \epsilon^-), \lambda) && (6.7)\\
&= \mathcal{F}(\epsilon^+, \lambda) \cup (\mathrm{KB}_h \setminus \epsilon^-) && (6.8)\\
&= (\mathcal{F}(\epsilon^+, \lambda) \cup \mathrm{KB}_h) \setminus \epsilon^- && (6.9)
\end{aligned}$$

$$= (\mathrm{KB}_h \cup \mathcal{F}(\epsilon^+, \lambda)) \setminus \epsilon^- \models \varphi \tag{6.10}$$

The simplification from Equations 6.5 to 6.6 is due to the properties of L-MRP explanations that $\epsilon^- \subseteq \mathrm{KB}_h$, $\epsilon^+ \subseteq \mathrm{KB}_a$, and that the intersection $\mathrm{KB}_h \cap \mathrm{KB}_a$ will never be part of the explanation since that information is already common to both knowledge bases (Definition 19). The simplification from Equations 6.7 to 6.8 is because $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h)$ (premise of the theorem) does not contain any letters in KB_h or its subset $\epsilon^- \subseteq \mathrm{KB}_h$. For the same reason, Equation 6.8 can be rewritten as Equation 6.9. Finally, the entailment in Equation 6.10 is because $\mathcal{F}((\mathrm{KB}_h \cup \epsilon^+) \setminus \epsilon^-, \lambda)$ entails φ since $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h) \subseteq \mathcal{V}_{(\mathrm{KB}_h \cup \epsilon^+)\setminus\epsilon^-} \setminus \mathcal{V}_\varphi$ (Property 1). Combining this entailment and the premise that $|\mathcal{F}(\epsilon^+, \lambda)| + |\epsilon^-| \leq \mathcal{UB}$, we can conclude that $\langle \mathcal{F}(\epsilon^+, \lambda), \epsilon^- \rangle$ is a personalized L-MRP explanation (Definition 40). □

6.3.1 Computing Personalized Explanations

Our algorithm, called *Personalized Logical Explanation Algorithm for Symbolic Environments* (PLEASE), exploits Theorem 10 to find personalized L-MRP explanations. Algorithm 6.1 describes its pseudocode. At a high level, PLEASE is composed of the following steps:

(1) Use any off-the-shelf L-MRP solver to find a sequence of L-MRP explanations.
(2) For each L-MRP explanation $\langle \epsilon^+, \epsilon^- \rangle$, iterate through all possible subsets of letters $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h)$.
(3) For each subset of letters λ to forget, check if the cardinality of the explanation with forgotten letters $|\mathcal{F}(\epsilon^+, \lambda)| + |\epsilon^-| \leq \mathcal{UB}$ is within the upper bound $\mathcal{UB}$.
(4) If it is, then return the personalized explanation. If not, repeat with the next L-MRP explanation from the L-MRP solver.

Example 22. *Let* $\mathrm{KB}_a = \{a, d, \neg d \vee b, \neg a \vee \neg b \vee c\}$, $\mathrm{KB}_h = \emptyset$, $\varphi = \{c\}$, *and* $\mathcal{V}_h = \{a, d\}$, *and suppose that we are searching for a personalized explanation whose cardinality is within an upper bound* $\mathcal{UB}$ *of 3. First, notice that since* $\mathrm{KB}_h = \emptyset$, *the only L-MRP explanation is* $\epsilon^+ = \mathrm{KB}_a$ *and* $\epsilon^- = \emptyset$. *As* $|\epsilon^+| + |\epsilon^-| = 4 + 0 = 4$ *is larger than the upper bound, we will iterate through all possible subsets* $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_\varphi \cup \mathcal{V}_h) = \{a, b, c, d\} \setminus (\{c\} \cup \{a, d\}) = \{b\}$, *which in this case is only the letter* b. PLEASE *then checks if forgetting this letter is sufficient to reduce the cardinality of the explanation to within the upper bound:* $|\mathcal{F}(\epsilon^+, \lambda)| + |\epsilon^-| =$

Algorithm 6.1: Personalized Logical Explanation Algorithm for Symbolic Environments (PLEASE)

Input: Agent knowledge base KB_a, human knowledge base KB_h, explanandum φ, human vocabulary $\mathcal{V}_h$, upper bound $\mathcal{UB}$

Result: A *personalized explanation* $\langle \epsilon^+, \epsilon^- \rangle$

while *true* **do**
    $\langle \epsilon^+, \epsilon^- \rangle \leftarrow$ next-L-MRP-exp$(\text{KB}_a, \text{KB}_h, \varphi)$
    if $\langle \epsilon^+, \epsilon^- \rangle = \textit{null}$ **then**
        return null
    else
        foreach $\lambda \subseteq \mathcal{V}_{\epsilon^+} \setminus (\mathcal{V}_h \cup \mathcal{V}_\varphi)$ **do**
            $\epsilon^+ \leftarrow \mathcal{F}(\epsilon^+, \lambda)$
            if $|\epsilon^+| + |\epsilon^-| \leq \mathcal{UB}$ **then**
                return $\langle \epsilon^+, \epsilon^- \rangle$

$|\mathcal{F}(\epsilon^+, b)| = |\{a, d, \neg a \vee \neg d \vee c\}| = 3$. *Since it is,* PLEASE *will return the personalized explanation* $\langle \{a, d, \neg a \vee \neg d \vee c\}, \emptyset \rangle$.

It is fairly straightforward to see that the algorithm is correct and complete, under the assumption that the underlying off-the-shelf L-MRP solver is also correct and complete.

6.4 Empirical Evaluations

We now empirically evaluate our approach both in computational experiments as well as in a human user study.

6.4.1 Computational Experiments

We ran the experiments on a MacBook Pro machine comprising an M1 Max processor with 32GB of memory. The time limit was set to 300s. PLEASE was implemented in Python, where we use the algorithm described in Chapter 5 (Section 9) as the off-the-shelf solver to

Prob.		π^*	$p = 0.2$				$p = 0.4$				$p = 0.6$				$p = 0.8$			
			$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t
BLOCK WORLD	1	4	3	2	1	0.1s	3	2	1	0.1s	54	43	12	1.5s	54	43	12	1.0s
	2	10	3	2	1	0.4s	4	3	1	0.5s	17	14	3	2.5s	17	13	3	2.5s
	3	18	10	8	1	0.5s	20	15	3	1.5s	59	44	19	33.0s	59	46	25	37.0s
LOGIS-TICS	1	12	5	4	1	1.0s	5	4	1	1.0s	10	8	2	2.0s	12	10	4	3.0s
	2	14	4	3	1	2.0s	6	5	1	3.5s	8	6	2	4.0s	11	8	2	5.5s
	3	20	6	5	1	24.5s	6	5	1	25.0s	10	8	2	25.0s	13	10	3	26.0s
TPP	1	5	16	13	3	1.5s	16	13	3	2.0s	16	12	4	1.5s	16	13	3	0.1s
	2	18	43	34	9	1.5s	43	34	9	1.5s	43	34	9	1.5s	43	34	9	1.4s
	3	27	85	68	17	144.5s	85	68	17	144.0s	85	68	17	145.0s	85	68	17	145.0s
DEPOT	1	2	5	3	31	0.5s	6	4	1	1.0s	14	11	3	2.0s	14	11	3	1.2s
	2	6	7	5	1	1.0s	7	5	1	1.5s	14	11	3	32.5s	14	11	3	37.5s
	3	10	11	8	2	2.5s	12	7	3	3.0s	26	21	5	185.0s	26	21	5	184.0s

Table 6.1: Evaluation of PLEASE with different completeness of knowledge bases $|\mathrm{KB}_h|$.

find L-MRP explanations. We used our own implementation for the knowledge forgetting operation.[44]

We encoded some classical planning problems from the International Planning Competition (IPC) in the style of Kautz *et al.* [136], and used them to form the agent's knowledge base KB_a. The explanandum φ for each problem was the plan optimality query, which we constructed as described in Chapter 3 (Section 3.4.2). We varied three parameters – the assumed knowledge base of the human KB_h, the vocabulary of the human $\mathcal{V}_h$, and the upper bound $\mathcal{UB}$. To construct the knowledge base KB_h, we follow the literature by modifying KB_a, specifically, by removing p fraction of actions as well as p fraction of preconditions and effects of each remaining action. To construct the vocabulary $\mathcal{V}_h$, we first extract all the letters that are used in KB_h, then supplement it with letters from KB_a if $|\mathcal{V}_h| \leq q$ fraction of $|\mathcal{V}_{\mathrm{KB}_a}|$. Finally, we parameterize the upper bound $\mathcal{UB}$ as a fraction r of the cardinality of the shortest L-MRP explanation $|\epsilon^*|$. The default values for our three parameters are as follows: $p = 0.8$ for KB_h, $q = 0.4$ for $\mathcal{V}_h$, and $r = 0.8$ for $\mathcal{UB}$.

In our first experiment, we vary the completeness of KB_h by varying $p \in \{0.2, 0.4, 0.6, 0.8\}$. Table 6.1 tabulates the results, where we report the length of an optimal plan $|\pi^*|$, the cardinality of the shortest L-MRP explanations $|\epsilon^*|$ returned by the off-the-shelf L-MRP solver, the cardinality of the personalized L-MRP explanations $|\epsilon|$ returned by PLEASE, the

[44] The code repository is available at https://github.com/YODA-Lab/PLEASE.

Prob.		π^*	$q = 0.2$				$q = 0.4$				$q = 0.6$				$q = 0.8$			
			$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t
BLOCK WORLD	1	4	54	43	12	1.2s	54	43	12	1.0s	54	43	12	1.1s	54	43	11	1.0s
	2	10	17	12	5	3.0s	17	13	3	2.5s	17	13	4	3.0s	17	14	3	2.5s
	3	18	59	47	12	41.0s	59	46	25	37.0s	59	44	13	42.0s	59	44	15	43.0s
LOGIS-TICS	1	12	12	8	5	4.0s	12	10	4	3.0s	12	10	2	3.0s	12	10	3	3.0s
	2	14	11	8	2	5.0s	11	8	2	5.5s	10	8	2	5.2s	10	8	2	5.0s
	3	20	13	9	3	27.5s	13	10	3	26.0s	13	9	4	25.5s	13	10	3	26.0s
TPP	1	5	16	13	3	0.1s	16	13	3	0.1s	16	13	3	0.1s	16	13	3	0.1s
	2	18	43	34	9	1.5s	43	34	9	1.4s	43	34	11	2.0s	43	34	9	1.6s
	3	27	85	68	17	143.0s	85	68	17	145.0s	75	68	17	145.0s	85	67	17	146.0s
DEPOT	1	2	14	11	3	1.2s	14	11	3	1.2s	14	11	3	1.2s	14	11	3	1.0s
	2	6	14	11	3	38.0s	14	11	3	37.5s	14	11	3	37.5s	14	11	3	38.0s
	3	10	26	21	5	186.0s	26	21	5	184.0s	26	21	5	187.0s	26	21	5	185.5s

Table 6.2: Evaluation of PLEASE with different sizes of vocabulary $|V_h|$.

Prob.		π^*	$r = 0.6$				$r = 0.7$				$r = 0.8$				$r = 0.9$			
			$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t	$\lvert\epsilon^*\rvert$	$\lvert\epsilon\rvert$	$\lvert\lambda\rvert$	t
BLOCK WORLD	1	4	54	32	22	1.3s	54	38	16	1.3s	54	43	12	1.0s	54	49	5	1.2s
	2	10	17	10	7	2.3s	17	12	5	2.5s	17	13	3	2.5s	17	15	2	2.5s
	3	18	59	22	33	37.0s	59	22	33	37.5s	59	46	25	37.0s	59	53	6	37.0s
LOGIS-TICS	1	12	12	7	5	3.5s	12	8	4	3.5s	12	10	4	3.0s	12	11	1	3.0s
	2	14	11	7	4	6.0s	11	8	3	5.0s	11	8	2	5.0s	11	9	1	4.5s
	3	20	13	8	5	29.0s	13	9	4	29.0s	13	10	3	26.0s	13	12	1	29.0s
TPP	1	5	16	10	6	0.04s	16	11	5	0.1s	16	13	3	0.1s	16	14	2	0.1s
	2	18	43	26	18	1.5s	43	30	13	1.4s	43	34	9	1.4s	43	30	4	1.5s
	3	27	85	51	34	133.0s	85	59	26	135.0s	85	68	17	145.0s	85	76	9	136.5s
DEPOT	1	2	14	8	6	1.2s	14	10	4	1.5s	14	11	3	1.2s	14	13	1	1.0s
	2	6	14	8	6	35.0s	14	10	4	34.0s	14	11	3	37.0s	14	13	1	35.0s
	3	10	26	16	9	179.0s	26	18	7	180.0s	26	21	5	184.0s	26	23	2	186.5s

Table 6.3: Evaluation of PLEASE with different upper bounds $\mathcal{UB}$.

number of letters forgotten $|\lambda|$, and the runtimes t of PLEASE. We make the following observations: Unsurprisingly, $|\epsilon^*|$ increases as p increases. The reason is that KB_h decreases as p increases. Therefore, more information needs to be provided in the explanation in order for the updated KB_h to entail the explanandum. Additionally, $|\lambda|$ increases as p increases as well because more needs to be forgotten from longer explanations in order to get their cardinality to within $\mathcal{UB}$. Finally, as $|\pi^*|$ increases, the cardinality of both explanations $|\epsilon^*|$ and $|\epsilon|$ increases. Consequently, the runtime t increases as well.

In our second experiment, we vary the size of the vocabulary $|\mathcal{V}_h|$ by varying $q \in \{0.2, 0.4, 0.6, 0.8\}$. Table 6.2 tabulates the results. As q (and, equivalently, the vocabulary size $|\mathcal{V}_h|$) increases, $|\lambda|$ decreases and $|\epsilon|$ increases since fewer letters need to be forgotten before the updated KB_h entails the explanandum. Additionally, the cardinality of the L-MRP explanation $|\epsilon^*|$ and runtimes t remain relatively unchanged for all values of q. This implies that the runtime of PLEASE is dominated by the time needed to find the L-MRP explanation by the off-the-shelf solver, and the time needed to personalize the explanations is relatively small.

In our third experiment, we vary the upper bound $\mathcal{UB}$ by varying $r \in \{0.6, 0.7, 0.8, 0.9\}$. Table 6.3 tabulates the results. As r (and, equivalently, the upper bound $\mathcal{UB}$) increases, $|\lambda|$ decreases and $|\epsilon|$ increases since fewer letters need to be forgotten to get a personalized explanation with a cardinality that is within the upper bound $\mathcal{UB}$. Similar to the second experiment, the cardinality of the L-MRP explanation $|\epsilon^*|$ and runtimes t remain relatively unchanged for all values of r, making the same implication that the runtime of PLEASE is dominated by the off-the-shelf solver.

6.4.2 Human-Subject Experiments

We now evaluate one of the assumptions made in this work, namely, that personalized explanations with respect to a human vocabulary increase the overall comprehension and satisfaction of human users. It is important to note that while we could not control for the knowledge of the human user (i.e., KB_h), it is reasonable to assume that their knowledge grows as their vocabulary $\mathcal{V}_h$ grows. For example, in the LOGISTICS domain example, if a user's vocabulary includes the `move` operator, even if they do not know specifically the preconditions and effects of the operator, they still have an intuitive sense of what it does. As such, they would have a larger KB_h compared to another user who does not know about the operator, everything else being equal. With this in mind, our hypothesis is that:

> *Human users with access to a known vocabulary of task-specific terms (and their background knowledge associated to those terms) have an increased understanding and satisfaction with personalized explanations compared to human users with generic explanations.*

Question	Vocab. $\mathcal{V}_{h1}$ Treat.	Vocab. $\mathcal{V}_{h1}$ Ctrl.	Vocab. $\mathcal{V}_{h1}$ Sig?	Vocab. $\mathcal{V}_{h2}$ Treat.	Vocab. $\mathcal{V}_{h2}$ Ctrl.	Vocab. $\mathcal{V}_{h2}$ Sig?	Vocab. $\mathcal{V}_{h3}$ Treat.	Vocab. $\mathcal{V}_{h3}$ Ctrl.	Vocab. $\mathcal{V}_{h3}$ Sig?
Q1: The explanation helped me understand the robot's decision.	**3.90**	2.65	**Y**	**4.45**	2.60	**Y**	**3.60**	2.70	**Y**
Q2: I am satisfied with the robot's explanation.	**3.90**	2.70	**Y**	**4.40**	2.65	**Y**	**3.65**	2.90	**Y**
Q3: The explanation has sufficient detail.	**3.75**	2.60	**Y**	**3.90**	2.90	**Y**	**3.65**	2.80	**Y**
Q4: The explanation is complete.	**3.75**	2.60	**Y**	**3.95**	2.80	**Y**	**3.15**	2.80	N
Q5: The explanation was useful for understanding.	**3.70**	2.40	**Y**	**4.10**	2.65	**Y**	**3.50**	2.80	N
Q6: I am confident in my understanding.	**4.15**	2.30	**Y**	**4.30**	2.65	**Y**	**3.45**	2.75	N
Q7: I can explain the robot's behavior to others.	**3.95**	2.10	**Y**	**4.05**	2.30	**Y**	**3.35**	2.75	N

Table 6.4: Average scores (max. 5) and statistical significance (t-test, $p = 0.05$) for treatment and control groups.

Study Design: We designed a between-subject user study, wherein the users were divided into three vocabulary group pairs ($\mathcal{V}_{h1}$, $\mathcal{V}_{h2}$, and $\mathcal{V}_{h3}$), each of which consists of a treatment group and a control group. The study comprised a simple imaginary scenario that involved a robot exploring an environment, and a supervisor (i.e., the users) observing its behavior from a station. For simplicity, we simulated the environment as a 5x4 grid and informed the users about the robot's capabilities, such as moving to adjacent locations among other actions. After the users understood the necessary information, they instructed the robot to move to a certain location in the grid. To generate explanations, we told the users that on top of moving to the particular location, the robot also communicated some data to their station and, as supervisors, they requested an explanation so as to understand its behavior. The explanations were in natural language; however, some of the terms in the explanation were changed to random Greek letters. These letters then formed the three vocabulary groups (i.e., $\mathcal{V}_{h1}$, $\mathcal{V}_{h2}$, and $\mathcal{V}_{h3}$, which has one, two, and three letters with meanings described, respectively). Within each vocabulary group, the treatment group received a personalized explanation, where explanations were provided using only the vocabulary known to the group, whereas the control group received the default explanation without any personalization. For additional details about the study design and how the explanations were generated in accordance to our framework see Appendix B.1.

The main task of the users was to evaluate the robot's explanation. To do this, we asked the users seven Likert-type questions pertaining to the understandability and satisfaction of the explanation.

Results: In total, we recruited 120 users (40 for each vocabulary group pair, 20 in the treatment and 20 in the control group) from the online crowdsourcing platform Prolific [175], with the only filter being that the users are fluent in English. Table 6.4 tabulates the average scores for each Likert-type question and whether the scores are statistically significant with respect to a t-test based on a p-value of 0.05. The distributions of all questions can be found in Appendix B.1.

The results presented in Table 6.4 show a clear trend in favor of personalized explanations. When comparing the treatment and control groups for each vocabulary, we observe that the treatment group consistently scores higher on average across the seven Likert-type questions. This indicates that personalized explanations tailored to users' known vocabulary can lead to an increased understanding and satisfaction compared to generic explanations.

In the case of vocabulary groups $\mathcal{V}_{h1}$ and $\mathcal{V}_{h2}$, the treatment group outperforms the control group in all questions, with statistical significance observed at a $p = 0.05$ level. This suggests that personalizing explanations based on a smaller, more focused vocabulary (i.e., one or two terms) has a considerable impact on users' understanding and satisfaction. These results support our hypothesis that personalized explanations can be more effective than generic ones when users have access to a known vocabulary of task-specific terms.

However, when we examine the results for vocabulary group $\mathcal{V}_{h3}$, we notice that the treatment group only shows statistically significant improvements in Q1, Q2, and Q3. This finding may suggest that as the vocabulary size increases, the benefits of personalized explanations become less pronounced. Further investigation is needed to better understand this relationship and its implications on the design of personalized explanations.

In summary, the results of our study demonstrate the value of personalized explanations for enhancing user understanding and satisfaction, especially when a smaller, focused vocabulary is used. While the effectiveness of personalization appears to decrease with larger vocabularies, the overall trend suggests that tailoring explanations to users' known vocabulary can lead to better outcomes than providing generic explanations. Future research could explore the potential trade-offs between vocabulary size and personalization to better understand the optimal conditions for delivering effective explanations.

6.5 Related Work

In Section 6.1, we described a key limitation of existing MRP approaches, namely that it assumes that the human model is accurate. As such, on one hand, it can generate overly long explanations by pessimistically assuming that the human model is almost empty when it captures only information that it is confident about. On the other hand, it can generate incoherent explanations by optimistically (and wrongly) assuming that the human model is mostly specified when it also captures information that it is not confident about.

Nevertheless, our work in this chapter is closely connected to the logic-based variant of model reconciliation (L-MRP) (see Chapter 3), where the underlying optimization and explanation generation problems can be encoded in a logical language. A limitation of existing L-MRP approaches, but not necessarily non-logic-based MRP approaches, is that they assume that $\epsilon^+ \subseteq \mathrm{KB}_a$ must be subsets of *exact clauses* from KB_a (See Definition 19). Therefore, the human user may have to learn and understand a very complicated ϵ^+ with many new terms and concepts when a simpler version with fewer new terms could have sufficed. For example, in classical planning, only one action is allowed to be executed at each time step. A logic-based encoding of this restriction is through the following rule:

$$\bigwedge_{a \in A} \bigwedge_{a' \in A | a \neq a'} (\neg a_t \vee \neg a'_t) \tag{6.11}$$

where A is the set of actions in the problem and a_t represents action a in timestep t. Note that A is the set of *all actions* in the problem. As such, should the human user be unaware of this rule, ϵ^+ would include it and, thus, they need to learn about all possible actions.

Continuing with the LOGISTICS example in the introduction, imagine that the agent is trying to explain that it is not possible to execute, at the same time, both actions `move(truck1, loc1, loc2)` and `move(truck1, loc1, loc3)`, which correspond to moving the truck to both locations `loc2` and `loc3` concurrently. An explanation generated through existing L-MRP methods will include Rule 6.11 with the set of *all actions* when a much simpler and shorter version where A is composed of *exactly the two* `move` *actions* above only would have sufficed. Our goal in this work is to enrich the L-MRP formulation by proposing a new algorithmic framework that can find such simpler explanations through knowledge forgetting.

In the context of classical planning problems, Sreedharan *et al.* [211] considered a related issue. They assumed the user's model is part of an abstraction lattice held by the agent, with each node representing an abstracted planning task model produced by projecting out a set of state fluents. The agent estimates the appropriate level based on user interactions and provides consistent explanations. This is achieved through a foil set (a set of plans) provided by the user, which the agent uses to find a minimal set of models consistent with the foil. This method can be seen as a special case of MRP, with the user model belonging to a set of possible models representing various abstractions of the agent's model. In contrast, our approach follows the standard MRP assumption with a single estimate of the user's model and a human-specified vocabulary that may include terms not in the user's model. This allows us to capture scenarios where users have relevant vocabulary terms without knowledge of their relationship to the problem. This can often be the case when the problem includes terms, such as `move` in LOGISTICS, that are in everyday conversation. Additionally, our approach has the merit of generality, as it can be applied to problems beyond planning, as long as they can be encoded in a logical formalism for which satisfiability of sets is feasible.

Finally, with recent progress, large language models (LLMs) [20] may be able to tackle explainability and reconciliation challenges. As few-shot learners, LLMs excel at producing well-formed sentences [24, 160]. Nevertheless, their primary shortcomings in establishing a robust basis for logical reasoning, mainly due to their dependence on statistical features for inference, has been exploited [187, 45]. Conversely, our framework's symbolic nature offers important theoretical guarantees, such as logically consistent and accurate explanations. The ability to perform consistent logical reasoning is vital for building trust between human users and AI systems.

6.6 Concluding Remarks

This chapter has enhanced our logical explainability layer by addressing a crucial limitation in existing Model Reconciliation Problem (MRP) approaches: the need to adapt explanations to different users' levels of understanding. Through knowledge forgetting, we have shown how we can generate personalized explanations that match users' vocabularies while maintaining formal guarantees.

The key contributions and findings of this chapter are:

1. Development of a logic-based framework that generates explanations tailored to the human user's vocabulary, striking a balance between assuming an almost empty human model and a fully specified one. The use of a human vocabulary allows for implicit encoding of the user's expertise level.

2. Formalization of various settings for generating personalized explanations, providing a flexible approach adaptable to different scenarios.

3. Presentation of an algorithmic approach for computing personalized explanations, which can be integrated with existing L-MRP methods.

4. Empirical evaluation, including both benchmark tests and a human-subject study, demonstrating the efficacy of our framework in practical applications.

While the operation of knowledge forgetting has been extensively studied in various logical settings [244, 162, 235, 69], its application to model reconciliation and explanation generation represents a novel contribution. This work contributes to the growing body of research on human-aware planning, enhancing the potential for effective communication and collaboration between humans and AI agents.

Future work could explore the dynamic adaptation of the human vocabulary based on ongoing interactions, further refining the personalization of explanations. Additionally, investigating the integration of this approach with other explainable AI techniques could lead to more comprehensive and versatile explanation-based systems.

In the next chapter, we will shift our focus and explore how our logical explainability layer can support dynamic, dialogue-based interactions. Through structured argumentative dialogues, we will show how explanations can evolve through continuous interaction between AI systems and human users.

Chapter 7

From Model Reconciliation to Dialectical Reconciliation

"A prediction, or any assertion, that cannot be defended might still be true, but an explanation that cannot be defended is not an explanation."
— David Deutsch (The Fabric of Reality, p. 76)

7.1 Introduction & Contribution

The frameworks we have developed up until now have primarily relied on one-shot explanations and known, a-priori human user models. While these assumptions helped establish our theoretical foundations, they can limit effectiveness in real-world applications where understanding develops through iterative interaction and the AI system's model of user knowledge may be incomplete or inaccurate.

The need for a more interactive framework arises from several observations:

- Human users often have follow-up questions or require clarification on specific aspects of an explanation.
- The initial explanation may not fully address the user's concerns or may reveal misconceptions that need to be addressed.
- A dialogue-based approach allows for a more natural and interactive form of explanation, mirroring human-to-human communication.

- Iterative explanation can lead to a more nuanced understanding of the AI system's reasoning and the problem domain.

These observations lead to a pressing question: "*How can we effectively help the human user understand the AI agent's decisions in a natural manner?*" Cognitive science and psychology literature provides inspiration – people learn and understand better when they engage in argumentation-based dialogues. Such dialogues engage participants' cognitive abilities, enhancing learning and understanding through active engagement, reconstruction, and assimilation of information [168]. In other words, reconciliation is *dialectical*.

Building on this insight, this chapter introduces Dialectical Reconciliation via Structured Argumentative Dialogues (DR-Arg), extending our logical explainability layer to support dynamic interaction between AI systems and human users. DR-Arg builds upon our previous L-MRP framework and extends it to support back-and-forth exchanges between the AI system and the user. It also extends previous efforts in argumentation-based dialogues [17] and is formalized using a game-theoretic approach to dialogues [100, 101].

The key features and contributions of DR-Arg include:

1. **Dialogical Interaction:** DR-Arg does not rely on predefined human user models but allows for a more nuanced exchange of information through dialogue.
2. **Focus on Understanding:** The primary goal is to enhance the explainee's understanding of the explainer's decisions, even if disagreement persists. This distinguishes DR-Arg from traditional argumentation frameworks that aim for mutual agreement through persuasion [97, 184, 178].
3. **Formal Framework:** We provide a formal definition of dialectical reconciliation dialogues, describe their operational semantics using structured (deductive) argumentation [12], and offer theoretical guarantees for termination and success.
4. **Evaluation Metrics:** We introduce the concept of explainee understanding in the context of these interactions and present a method for approximating it.
5. **Empirical Validation:** We evaluate DR-Arg through both computational experiments and human-subject studies, demonstrating its effectiveness in enhancing human-AI interactions.

By introducing this dialectical dimension to our logical explainability layer, we bridge the gap between static explanation generation and dynamic human inquiry. This work shows how logic can serve as an explainability layer even in interactive settings, adapting to the natural flow of human understanding while maintaining formal rigor.

7.2 Essential Background

We provide a partial review of deductive, logic-based argumentation [13], which serves as the underlying machinery of our proposed framework.

We consider a (propositional) language $\mathcal{L}$ that utilizes the classical entailment relation, represented by $\models$. We use $\perp$ to denote falsity and assume that knowledge bases (finite sets of formulae) are consistent unless specified otherwise.

Our approach relies on an intuitive concept of a logical *argument*, which can be thought of as a set of formulae employed to (classically) prove a particular claim, represented by a formula:

Definition 41 (Argument)**.** *Let* KB *be a knowledge base and* ϕ *a formula. An* argument *for* ϕ *from* KB *is defined as* $A = \langle \Gamma, \phi \rangle$ *such that: (i)* $\Gamma \subseteq \mathrm{KB}$*; (ii)* $\Gamma \models \phi$*; (iii)* $\Gamma \not\models \perp$*; and (iv)* $\nexists \Gamma' \subset \Gamma$ *s.t.* $\Gamma' \models \phi$*.*[45]

We refer to ϕ as the *claim* of the argument, denoted as $\mathrm{CL}(A)$, and Γ as the *premise* of the argument, denoted as $\mathrm{PR}(A)$. The set of all arguments for a claim ϕ from KB is represented by $\mathcal{A}(\mathrm{KB}, \phi)$.

Example 23. *Assume* $KB = \{a, b, a \wedge b \rightarrow c, g, g \rightarrow a\}$*. Then, an argument for c from* KB *is* $A_1 = \langle \{a, b, a \wedge b \rightarrow c\}, c \rangle$*. Another argument for c from* KB *is* $A_2 = \langle \{b, g, g \rightarrow a, a \wedge b \rightarrow c\}, c \rangle$*.*

To account for conflicting knowledge between agents, we will make use of a general definition of a *counterargument*, that is, an argument opposing another argument by emphasizing points of conflict on the premises or claim of the argument. With a slight abuse of notation:

[45] The minimality constraint maintains argument relevance by eliminating excess premises and pinpointing specific reasons for inferring a claim, while also preventing negative impacts from superfluous premises.

Definition 42 (Counterargument). *Let* KB_i *and* KB_j *be two knowledge bases,* $A_i = \langle \Gamma_i, \phi_i \rangle$*, and* $A_j = \langle \Gamma_j, \phi_j \rangle$ *be two arguments for* ϕ_i *from* KB_i *and for* ϕ_j *from* KB_j*, respectively. We say that* A_i *(or* A_j*) is a* counterargument *for* A_j *(or* A_i*) iff* $\Gamma_i \cup \Gamma_j \models \bot$.

Example 24. *Assume* $KB_i = \{a, b, a \wedge b \rightarrow c\}$ *and* $KB_j = \{l, d, l \wedge d \rightarrow \neg b, e, e \rightarrow \neg c\}$*, and let* $A_i = \langle \{a, b, a \wedge b \rightarrow c\}, c \rangle$ *be an argument for c from* KB_i*. Then,* $A_{j1} = \langle \{l, d, l \wedge d \rightarrow \neg b, \}, \neg b \rangle$ *and* $A_{j2} = \langle \{e, e \rightarrow \neg c\}, \neg c \rangle$ *are two* counterarguments *for* A_i *from* KB_j.

We denote the set of all counterarguments for an argument A from KB with $\mathcal{C}(\mathrm{KB}, A)$.

7.3 Towards Dialectial Reconciliation

7.3.1 Motivating Example

To illustrate the potential of our approach, consider a scenario where a human user, Alice, is tasked with troubleshooting an AI home assistant robot, named "Roomie", that appears to be disconnected from the internet. Alice is provided with a set of prompts to help diagnose the problem, such as checking the associated mobile app and verifying Roomie's connection to the internet via a wired connector.

Initially, Alice attempts to resolve the issue by following the provided prompts. However, she encounters several complications that hinder her ability to resolve the problem, including an outdated mobile app, and an expired license for the wired connection. Frustrated with the lack of progress, Alice requests an explanation from Roomie.

Roomie provides a brief explanation, stating that the outdated mobile app and expired license are preventing it from establishing a stable internet connection. However, this single-shot explanation does not fully satisfy Alice, as she feels she needs a better understanding of how these factors are interconnected and impact Roomie's performance.

To gain a deeper understanding, Alice engages in an argumentation-based dialogue with Roomie. She presents arguments about the importance of regularly updating the mobile app and renewing the license, citing the need for optimal performance and security. Roomie counters by explaining that while updates and renewals are important, other factors such as network stability and hardware compatibility also play roles in its ability to function properly.

Through the dialogue, Alice and Roomie explore various aspects of the problem, including the potential risks of using outdated software, the benefits of maintaining a stable power supply, and the importance of regular maintenance. This *dialectical interaction* allows Alice to better understand Roomie's reasoning and the evidence behind its explanations. While she may still have reservations about Roomie's arguments, she now has a more comprehensive grasp of the factors contributing to Roomie's disconnection and can make more informed decisions on how to proceed with troubleshooting.

This example demonstrates how a single-shot reconciliation explanation may not always be sufficient in scenarios requiring deeper understanding. In contrast, an argumentation-based dialogue, such as the one enabled by our proposed framework, allows for a more thorough exploration of the reasoning behind the AI system's behavior, enabling users to gain a more nuanced understanding. We also ran a human user study with this motivating example (see Section 7.5.2) highlighting the strengths of our framework.

7.3.2 The DR-Arg Framework

We now introduce the *Dialectical Reconciliation via Structured Argumentative Dialogues* (DR-Arg) framework. We begin by discussing the key assumptions and components of the framework.

Key Assumptions: The DR-Arg framework involves two agents engaging in a dialogue, with one agent taking on the role of an *explainer* (denoted by index R) and the other an *explainee* (denoted by index E). The goal of the dialogue is to *help the explainee understand the decisions made by the explainer from the explainer's perspective*. We use ϕ to represent an explainer's decision and Φ to represent the set of all decisions the explainee seeks to understand.

Three critical assumptions underlie our framework:

1. **Agent Knowledge Bases:** The explainer is associated with a knowledge base KB_R that encodes its own knowledge of the underlying task. The explainee is associated with

knowledge base KB_E that encodes *their approximation of the explainer's knowledge*, which can be $\emptyset$. No agent has explicit access to the other agent's knowledge base.

2. **Explainee Queries:** Initiated by the explainee, the dialogue starts with a `query` $\phi \in \Phi$, where $\mathrm{KB}_E \not\models \phi$ (or $\mathrm{KB}_E \models \neg\phi$) and $\mathrm{KB}_R \models \phi$. The explainee has the flexibility to generate subsequent queries dynamically as the dialogue progresses, reflecting their evolving understanding and the need for additional clarification.

3. **Public Commitment Stores:** Both agents contribute to public *commitment stores* that store their utterances throughout the dialogue, akin to a "chat log". A commitment store for agent $x \in \{R, E\}$ is defined as $CS_x = (CS_x^1, \ldots, CS_x^t)$, where $CS_x^t = \langle l(\gamma), A\rangle$ and $l(\gamma)$ is an instantiated locution (see next section) and A the respective argument (can be empty) accompanying the locution. This feature allows to build more complex and contextually aware arguments.

The main goal of the DR-Arg is formulated as follows:

> *Given an explainer agent with* KB_R*, an explainee agent with* KB_E*, and a set of queries* Φ *such that, for all* $\phi \in \Phi$*,* $\mathrm{KB}_E \not\models \phi$ *(or* $\mathrm{KB}_E \models \neg\phi$*) and* $\mathrm{KB}_R \models \phi$*, the goal of DR-Arg is to* enable $\mathrm{KB}_E \models \phi$ *through dialectical reconciliation.*

A critical aspect of this formulation is successfully *enabling* $\mathrm{KB}_E \models \phi$ during the dialogue between explainee and explainer. At a high level, we aim to find a way to help the explainee transition from a state of *not understanding* a decision ϕ (i.e., $\mathrm{KB}_E \not\models \phi$ or $\mathrm{KB}_E \models \neg\phi$) to a state of *understanding* the decision (i.e., $\mathrm{KB}_E \models \phi$). Our thesis is that a natural way of achieving this transition is through an argumentation-based dialogue that facilitates dialectical reconciliation, i.e., a *dialectical reconciliation dialogue*.

At a high level, a dialectical reconciliation dialogue is a process resolving inconsistencies, misunderstandings, and knowledge gaps between the explainer and the explainee. This is achieved through argument exchange and dialogue moves that collaboratively construct a shared understanding of the explainer's decisions. To successfully achieve a dialectical reconciliation dialogue, the agents should follow certain (dialogue) protocols that guide their interaction:

Locution	Agent Type	Preconditions	Effects
$\texttt{query}(\gamma)$	E	(1) $\exists A \in CS_R^T$ s.t. $\gamma \subseteq \text{PR}(A)$ **and** (2) $\texttt{query}(\gamma) \notin CS_E^T$ **and** (3) $\text{KB}_E \not\models \gamma$ or $\text{KB}_E \models \neg\gamma$	$CS_E^t \leftarrow \langle \texttt{query}(\gamma), \emptyset \rangle$
$\texttt{support}(\gamma)$	R	(1) $\texttt{query}(\gamma) \in CS_E^{t-1}$ **and** (2) $\exists A \in \mathcal{A}(\text{KB}_R, \gamma)$ s.t. $A \notin CS_R^T$	$CS_R^t \leftarrow \langle \texttt{support}(\gamma), A \rangle$
$\texttt{refute}(\gamma)$	E	(1) $\exists A \in CS_R^T$ s.t. $\gamma \subseteq \text{PR}(A) \cup \text{CL}(A)$ **and** (2) $\exists A \in \mathcal{C}(\text{KB}_E \cup CS_R^T, \gamma)$ s.t. $A \notin CS_E^T$	$CS_E^t \leftarrow \langle \texttt{refute}(\gamma), A \rangle$
	R	(1) $\exists A \in CS_E^T$ s.t. $\gamma \subseteq \text{PR}(A) \cup \text{CL}(A)$ **and** (2) $\exists A \in \mathcal{C}(\text{KB}_R \cup CS_E^T, \gamma)$ s.t. $A \notin CS_R^T$	$CS_R^t \leftarrow \langle \texttt{refute}(\gamma), A \rangle$
$\texttt{understand}$	E	(1) $\texttt{query}(\gamma)$ preconditions do not hold **and** (2) $\texttt{refute}(\gamma)$ preconditions do not hold	$CS_E^t \leftarrow \langle \texttt{understand}, \emptyset \rangle$
	R	(1) $\texttt{support}(\gamma)$ preconditions do not hold **and** (2) $\texttt{refute}(\gamma)$ preconditions do not hold	$CS_R^t \leftarrow \langle \texttt{understand}, \emptyset \rangle$

Table 7.1: The DR dialogue protocol. Note that, with a slight abuse of notation, the condition $A \in CS_x^T$ ($x \in \{R, E\}$) is true if there exists an argument A that has been uttered by agent x at any step during the dialogue, i.e., $1 \leq T \leq t-1$.

- Establish a clear dialogue structure, including the use of *locutions* that define permissible speech acts and turn-taking mechanisms.
- Engage in a cooperative and collaborative manner, with both agents focusing on the shared goal of improving the explainee's understanding.
- Employing argumentation techniques, such as offering counterexamples or pointing out logical inconsistencies, to constructively challenge each other's positions.

Following these protocols, the explainer helps the explainee iteratively refine their knowledge base, ultimately converging on a shared understanding that enables $\text{KB}_E \models \phi$ for all decisions $\phi \in \Phi$.

Dialectical Reconciliation Dialogue Type

We now formalize the dialectical reconciliation dialogue type, inspired by Hamblin's dialectical games framework [100, 101]. Here, a dialogue is viewed as a game-theoretic interaction, where utterances are treated as moves governed by rules that define their applicability. In this context, moves consist of a set of *locutions*, which determine the types of permissible utterances agents can make. To align with the goal of DR-Arg, we define the following set of locutions:

$$L = \{\texttt{query}, \texttt{support}, \texttt{refute}, \texttt{understand}\} \tag{7.1}$$

The `query` locution enables the explainee to ask the explainer for an argument supporting the explainee's `query`. The `support` locution allows the explainer to provide a supporting argument for the explainee's `query`. The `refute` locution permits both agents to provide counterarguments, and the `understand` locution allows both agents to acknowledge each other's utterances when no further queries or counterarguments are possible. We impose two restrictions: (1) the `query` locution is only available to the explainee, and (2) the `support` locution is only available to the explainer. These restrictions are reasonable given the goal of DR-Arg; future work will explore relaxing them.

Note that we opted for an `understand` locution instead of a simple `agree` (or `accept`) locution as the goal of DR-Arg is not to convince the explainee about Φ but to help them understand Φ. An `understand` locution reflects this flexibility, where agents do not have to agree with each other; they only have to acknowledge each other's utterances and understand each other's perspectives.

Locutions are typically instantiated with specific formulae that make up the range of possible *dialogue moves* m_t:

$$m_t = \langle x, l(\gamma) \rangle, \tag{7.2}$$

where t is an index indicating the dialogue timestep, $x \in \{R, E\}$ denotes the agent making the move, $l \in L$ is a locution, and $\gamma \in \mathcal{L}$ is a formula that instantiates the locution (e.g., the content of the move).

We now formally define a dialectical reconciliation (DR) dialogue. A DR dialogue requires that the first move must always be a `query` locution from the explainee, and the agents take turns making and receiving moves:

Definition 43 (DR Dialogue)**.** *A* DR *dialogue D is a sequence of moves $[m_1, \ldots, m_{|D|}]$ involving an explainee agent E and an explainer agent R, where the following conditions hold:*

1. *$m_1 = \langle E, \texttt{query}(\phi) \rangle$ is the opening move of the dialogue made by the explainee.*
2. *Each agent can make only one move m_t per (alternating) timestep t.*

We refer to the initial `query` ϕ as the *starting topic* of the dialogue, and to all explainee queries Φ made in the dialogue as the *overall topic* of the dialogue.

A DR dialogue is *terminated* at timestep t if and only if the explainee cannot generate subsequent queries or counterarguments, that is, when the explainee utters the `understand` locution. More formally,

Definition 44 (Terminated DR Dialogue). *A DR dialogue D is* terminated *at timestep t iff $m_t = \langle E, \texttt{understand} \rangle$ and $\nexists t' < t$ s.t. D is terminated at timestep t'.*

Agent Strategy: During the dialogue, the agents essentially determine their moves based on objectives like adhering to rationality or influencing dialogue length. In other words, each agent follows a *strategy* when selecting their next move. For an agent x, a strategy, denoted S_x, is a function taking in its current dialogue D, knowledge base KB_x, and next timestep t to output the next move.

While strategies can take several forms (e.g., preference-based, probabilistic), for simplicity, we assume two ordered strategies: $S_E(D, \mathrm{KB}_E, t) = [\texttt{refute}, \texttt{query}, \texttt{understand}]$ and $S_R(D, \mathrm{KB}_R, t) = [\texttt{support}, \texttt{refute}, \texttt{understand}]$, where the ordered lists show the priorities of dialogue moves for the explainee and explainer, respectively, at $t > 1$.

Now, if the agents follow their respective strategies during the DR dialogue, and the dialogue does not continue after it has terminated, then we say that the dialogue is *well-formed*.

Definition 45 (Well-Formed DR Dialogue). *A DR dialogue D is* well-formed *iff it is terminated at timestep t and, for all timesteps $1 < t' < t$, $m_{t'} \in S_x(D', \mathrm{KB}_x, t')$ for each move $m_{t'}$ from agent x, where $D' \subseteq D$ consists of the first $|D'| = t' - 1$ moves from D.*

Operational Semantics of DR Dialogues

In argumentation-based dialogues, the combination of locutions and formulae by agents is not arbitrary; rather, it is governed by specific rules. This restriction is encapsulated in the concept of a *dialogue protocol*. A dialogue protocol delineates the *operational semantics* of a dialogue, explicating the preconditions and effects for each locution [180]. That is, locutions exhibit action-like properties, influencing and modifying the state of the dialogue.

As described in Definition 43, the dialogue is initiated with a `query` move from the explainee (m_1). Recall also that the `query` and `support` locutions are restricted to the explainee and explainer, respectively. Table 7.1 describes the generation of valid dialogue moves m_t ($t > 1$) during a DR dialogue.

A `query` locution with formula γ is valid if it satisfies three preconditions: (1) γ is part of the premise in an argument previously made by the explainer, (2) γ has not been queried before, and (3) γ is neither entailed by KB_E nor is its negation entailed. The `support` locution, instantiated with formula γ, is permissible when: (1) γ was queried by the explainee in the preceding timestep, and (2) a new argument for γ exists in KB_R. The `refute` locution is instantiated with γ if: (1) γ is in the premises or claim of any argument made by either the explainer (resp. explainee), and (2) an unasserted counterargument refuting γ exists in KB_E (resp. KB_R). The `understand` locution is a valid option if `query` (resp. `support`) and `refute` cannot be uttered by the explainee (resp. explainer). After each move, the respective agents' commitment stores are updated.

Note that our framework remains neutral regarding to which argument (`support` move) or counterargument (`refute` move) is computed first. This can be done in a preference-based fashion by incorporating and minimizing a cost function that measures the complexity of the arguments. For simplicity again, we employ a cost function based on argument length, i.e., $cost(A) = |\mathrm{PR}(A)|$.

Importantly, our framework permits agents to utilize each other's commitment stores when formulating arguments, specifically for the `refute` locution (see precondition (2)). This inter-use of commitment stores enables the agents to draw upon shared information to construct arguments, thereby creating a more realistic representation of dialectical reconciliation.

Illustrative Example

Consider the following explainer and explainee knowledge bases, where all formulae are equally preferred:

$$\mathrm{KB}_R = \{a, b, a \wedge b \rightarrow c, d, d \rightarrow \neg e, f, f \rightarrow d\}$$
$$\mathrm{KB}_E = \{e, e \rightarrow \neg c, g, g \wedge a \rightarrow \neg f\}$$

Dialogue Move	Commitment Store
$m_1 = \langle E, \texttt{query}(\{c\})\rangle$	$CS_E^1 = \langle \texttt{query}(\{c\}), \emptyset\rangle$
$m_2 = \langle R, \texttt{support}, \{c\}\rangle$	$CS_R^2 = \langle \texttt{support}(c), \langle\{a, b, a \wedge b \rightarrow c\}, c\rangle\rangle$
$m_3 = \langle E, \texttt{refute}(\{c\})\rangle$	$CS_E^3 = \langle \texttt{refute}(\{c\}), \langle\{e, e \rightarrow \neg c\}, \neg c\rangle\rangle$
$m_4 = \langle R, \texttt{refute}(\{e\})\rangle$	$CS_R^4 = \langle \texttt{refute}(\{e\}), \langle\{d, d \rightarrow \neg e\}, \neg e\rangle\rangle$
$m_5 = \langle E, \texttt{query}(\{d\})\rangle$	$CS_E^5 = \langle \texttt{query}(\{d\}), \emptyset\rangle$
$m_6 = \langle R, \texttt{support}(\{d\})\rangle$	$CS_R^6 = \langle \texttt{support}(\{d\}), \langle\{f, f \rightarrow d\}, d\rangle\rangle$
$m_7 = \langle E, \texttt{refute}(\{f\})\rangle$	$CS_E^7 = \langle \texttt{refute}(\{f\}), \langle\{g, a, g \wedge a \rightarrow \neg f\}, \neg f\rangle\rangle$
$m_8 = \langle R, \texttt{understand}\rangle$	$CS_R^8 = \langle \texttt{understand}, \emptyset\rangle$
$m_9 = \langle E, \texttt{understand}\rangle$	$CS_E^9 = \langle \texttt{understand}, \emptyset\rangle$

Table 7.2: Example of DR dialogue.

The starting topic is c, where $\text{KB}_R \models c$ and $\text{KB}_E \models \neg c$.

A generated DR dialogue is shown in Table 7.2. The dialogue begins with the explainee asking the explainer about c (m_1), and the explainer provides an argument `support`ing it (m_2). The explainee counters by refuting c with e (m_3), which the explainer then `refute`s with d (m_4). Next, the explainee poses a new `query` about d (m_5), and the explainer `support`s it with f (m_6). The explainee subsequently `refute`s f with g and a (from the explainer's commitment store) (m_7). Finally, both agents utter `understand` (m_8 and m_9), leading to the termination of the dialogue.

It is important to note that the goal we pursue in this work (dialectical reconciliation) sets our framework apart from traditional argumentation frameworks that aim to achieve mutual agreement through persuasion [97, 184, 178] or obtain information through information-seeking [178, 73]. To better highlight the differences, let us consider the logic-based persuasion and information-seeking frameworks presented in [177, 178]. In these frameworks, agents are assumed to have "dialogical attitudes" (akin to agent strategies) when choosing their assert and accept moves. The attitudes relevant to our setting are the confident agent, who asserts any proposition for which an argument can be constructed, and the cautious agent, who accepts a proposition only if they cannot construct a counterargument against it. In our example, given that the starting dialogue topic is c, the goal in persuasion is for the explainer to persuade the explainee to accept c, while in information-seeking, the explainee aims to gather information about c. The corresponding dialogues are shown in Table 7.3.

Persuasion	Information-seeking
$m_1 = \langle R, \texttt{assert}(c)\rangle$	$m_1 = \langle E, \texttt{question}(c)\rangle$
$m_2 = \langle E, \texttt{assert}(\neg c)\rangle$	$m_2 = \langle R, \texttt{assert}(c)\rangle$
$m_3 = \langle R, \texttt{challenge}(\neg c)\rangle$	$m_3 = \langle E, \texttt{challenge}(c)\rangle$
$m_4 = \langle E, \texttt{assert}(\{e, e \rightarrow \neg c\})\rangle$	$m_4 = \langle R, \texttt{assert}(\{a, b, a \wedge b \rightarrow c\})\rangle$
$m_5 = \langle R, \texttt{assert}(\neg e)\rangle$	$m_5 = \langle E, \texttt{accept}(\{a\}\rangle$
$m_6 = \langle E, \texttt{assert}(e)\rangle$	$m_6 = \langle E, \texttt{accept}(\{b\}\rangle$
$m_7 = \langle R, \texttt{challenge}(e)\rangle$	$m_7 = \langle E, \texttt{accept}(\{a \wedge b \rightarrow c\}\rangle$
$m_8 = \langle E, \texttt{assert}(\{e\})\rangle$	

Table 7.3: Example of persuasion and information-seeking dialogues.

The differences between a DR dialogue and persuasion and information-seeking dialogues are evident in this example. Compared to persuasion, the primary difference lies in the goal. A DR dialogue aims for understanding, while persuasion seeks to change the explainee's beliefs. This is evident in the dialogue moves, where dialectical reconciliation allows for a back-and-forth exchange of arguments and counterarguments (m_3 to m_8) until a point of understanding is reached (m_9). In persuasion, the dialogue ends when the explainer concedes (m_8), failing to persuade E about c.

Compared to information-seeking, the main difference is the level of interaction. A DR dialogue enables the explainee to provide counterarguments (e.g., refuting c in m_3) and the explainer to offer additional information (e.g., m_4 onwards). This kind of exchange is not possible in the information-seeking protocol, where the explainee simply accepts the explainer's assertions (m_4) without the opportunity to challenge or seek further clarification.

This simple example shows that a DR dialogue provides a more interactive and collaborative framework for understanding, compared to the one-sided nature of persuasion and the limited interaction in information-seeking.

7.3.3 Properties of DR Dialogues

We now describe two properties for assessing the efficacy of a DR dialogue: *termination* and *success*.

Termination: This property ensures that the dialogue concludes within a finite number of steps and is devoid of any deadlocks, guaranteeing that at every stage, each agent has at least one viable move.

Theorem 11. *Every DR dialogue is guaranteed to terminate.*

PROOF. First, the operational semantics (see Table 7.1) outline the constraints and conditions under which each dialogue move can be executed. Second, the agents' knowledge bases are finite, meaning that there are only a limited number of different moves that can be generated, and the agents cannot repeat these moves. As such, the dialogue will not continue indefinitely.

We now prove through contradiction that a deadlock cannot happen. Assume that a deadlock happened, where an agent x does not have any available moves to make and the dialogue has not terminated. There are two cases:

- Agent x is an explainee. When the explainee cannot make any `query` or `refute` moves, it can always make the `understand` move since its preconditions are that the preconditions of the `query` and `refute` moves do not hold.
- Agent x is an explainer. Similar to the previous case, when the explainer cannot make any `support` or `refute` moves, it can always make the `understand` move.

This contradicts our assumption and the dialogue is thus deadlock-free. Therefore, a DR dialogue is guaranteed to terminate. □

Success: The success of a terminated dialogue is contingent upon the achievement of its primary goal. For DR-Arg, this entails the explainee comprehending the overall topic Φ, from the explainer agent's perspective. This is formally denoted as $\mathrm{KB}_E \models \phi$ for each $\phi \in \Phi$, or more succinctly, $\mathrm{KB}_E \models \Phi$. Achieving this involves a *knowledge update* in KB_E, incorporating the explainer's arguments from the dialogue. We adopt the following general knowledge base update from the Section 3.4 of Chapter 3:

Definition 46 (Updated Knowledge Base)**.** *The* updated knowledge base KB_E *upon integrating argument A is defined as* $\widehat{\mathrm{KB}}_E^A = (\mathrm{KB}_E \cup \mathrm{PR}(A)) \setminus M$*, where* $M \subseteq \mathrm{KB}_E \setminus \mathrm{PR}(A)$ *is a* $\subseteq$*-minimal subset whose (potential) removal ensures that* $(\mathrm{KB}_E \cup \mathrm{PR}(A))$ *remains consistent.*

For simplicity, we assume that the knowledge base update transpires post-dialogue. Performing this update during the dialogue is equally feasible, given that the explainee has access to the explainer's commitment store, which aids in formulating new arguments. This means that the timing of the update does not affect the argumentation dynamics.

Now, a crucial observation is that not all arguments presented by the explainer are necessary to update KB_E for it to entail Φ. An incremental update strategy can be employed, beginning with the most recent argument and proceeding until $\mathrm{KB}_E \models \Phi$ is fulfilled. Should retraction be needed for consistency, it is confined to the original contents of KB_E, preserving the integrity of the added arguments. This approach assures that $\mathrm{KB}_E \models \Phi$ is enabled. Hence, a DR dialogue that attains its objective is deemed *successful*.

Definition 47 (Successful DR Dialogue)**.** *A terminated DR dialogue D regarding topic Φ is* successful *iff* $\widehat{\mathrm{KB}}_E^A \models \Phi$ *for some* $A \subseteq CS_R$.

Integrating Definition 47 with the underlying principles of the DR-Arg framework leads to an important conclusion:

Theorem 12. *A terminated DR dialogue D on topic Φ is always successful.*

PROOF. First, recall that the topic of the dialogue φ must be entailed by the explainer (i.e., $KB_R \models \varphi$), which means that an argument for φ from KB_R always exists (Definition 41).

Now, notice that for a terminated dialogue D, the explainer's commitment store CS_R contains the explainer's set of arguments that have been presented during the dialogue. Since $KB_R \models \varphi$, and the arguments in CS_R are derived from KB_R, it follows that using the arguments in CS_R to update the explainee's knowledge base KB_E (w.r.t. Definition 46) will enable $KB_E \models \varphi$, as in the worst case, the entire CS_R will be used to update KB_E.

Therefore, the explainee's knowledge base will eventually entail φ (i.e., $KB_E \models \varphi$) and, as such, a terminated DR dialogue on topic φ is always successful. □

7.4 Approximating Explainee Understanding

Understanding, a multifaceted and abstract concept, is challenging to quantify and often involves the explainee's cognitive process of forming a functional mental model of the subject matter, which includes its causes, consequences, and interconnections. This process resembles constructing a complex "blueprint" through the narrative provided by the explainer, effectively facilitated by argumentation-based dialogue. Such dialogues engage the explainee's cognitive abilities, enhancing learning and understanding through active engagement, reconstruction, and assimilation of information, as evidenced in cognitive psychology studies [122, 168]. Our framework is motivated by these insights, employing argumentation to guide the explainee in developing a comprehensive understanding of the phenomenon under discussion.

As stated, our main objective is to enhance the explainee's understanding of the explainer's decisions. To quantify and approximate this understanding, we propose a simple metric that measures the similarity between the explainee's knowledge base (KB_E) and the explainer's knowledge base (KB_R). We postulate that *the explainee's understanding is likely to improve as the similarity between* KB_E *and* KB_R *increases.*

We define the similarity between KB_E and KB_R using syntactic and semantic measures. Syntactic similarity assesses structural likeness (e.g., similarity of formulae), while semantic similarity examines the logical consequences of the knowledge bases. We employ a weighted Sørensen-Dice similarity index [61, 203] as follows:

$$\Sigma = a \cdot \frac{2 \cdot |\mathrm{KB}_E \cap \mathrm{KB}_R|}{|\mathrm{KB}_E| + |\mathrm{KB}_R|} + (1-a) \cdot \frac{2 \cdot |\mathcal{B}_E \cap \mathcal{B}_R|}{|\mathcal{B}_E| + |\mathcal{B}_R|} \tag{7.3}$$

where $a \in [0, 1]$ is a parameter indicating the weight of each metric component. Here, $\mathcal{B}_E$ and $\mathcal{B}_R$ represent the backbone literals of KB_E and KB_R, respectively, which are the literals entailed by each knowledge base [176].[46] This formula approximates the explainee's level of understanding as the similarity between KB_E and KB_R.

Note that we assume that the explainee's knowledge base is dynamic, capable of assimilating new information from the explainer. We also assume that the explainee, as a rational agent,

[46]Note that instead of the backbone literals of the knowledge bases, we could alternatively consider their prime implicates, which are their strongest consequences [118].

actively seeks to understand the explainer's perspective and integrates this information into KB_E (Definition 46).[47]

Example 25. *Consider the DR dialogue from the illustrative example. Upon dialogue termination, the explainee sequentially updates* KB_E *with the explainer's arguments until the dialogue topic* $\Phi = \{c, d\}$ *is entailed by* KB_E *(i.e.,* $\mathrm{KB}_E \models c$ *and* $\mathrm{KB}_E \models d$*). Table 7.4 illustrates the evolution of the knowledge base similarity with each update.*

#	**Premise to Add**	**Updated** KB_E	**Similarity Metric**
1	$\{f, f \rightarrow d\}$	$\{e, e \rightarrow \neg c, g, f, f \rightarrow d\}$	$\Sigma = 0.5 \cdot \frac{2 \cdot 2}{12} + 0.5 \cdot \frac{2 \cdot 2}{11} = 0.35$
2	$\{d, d \rightarrow \neg e\}$	$\{e \rightarrow \neg c, g, f, f \rightarrow d, d, d \rightarrow \neg e\}$	$\Sigma = 0.5 \cdot \frac{2 \cdot 4}{13} + 0.5 \cdot \frac{2 \cdot 3}{11} = 0.58$
3	$\{a, b, a \wedge b \rightarrow c\}$	$\{e \rightarrow \neg c, g, f, f \rightarrow d, d, d \rightarrow \neg e, a, b, a \wedge b \rightarrow c\}$	$\Sigma = 0.5 \cdot \frac{2 \cdot 7}{16} + 0.5 \cdot \frac{2 \cdot 6}{13} = 0.90$

Table 7.4: Example of knowledge base update and similarity metric.

It is interesting to see how this example underscores the potential advantage of dialectical reconciliation over a single-shot reconciliation approach. For instance, using our single-shot reconciliation approach, we get the explanation tuple $\mathcal{E} = \langle \mathcal{E}^+, \mathcal{E}^- \rangle = \langle \{a, b, a \wedge b \rightarrow c\}, \{e\} \rangle$*, where* $\mathcal{E}^+$ *and* $\mathcal{E}^-$ *denote the formulae to be added and retracted from* KB_E*, respectively. Updating* KB_E *with* $\mathcal{E}$ *(using Definition 46) results in* $KB_E = (\mathrm{KB}_E \cup \mathcal{E}^+) \setminus \mathcal{E}^- = \{e \rightarrow \neg c, g, g \wedge a \rightarrow \neg f, a, b, a \wedge b \rightarrow c\}$*. Calculating the similarity score between this updated* KB_E *and* KB_R*, we get* $\Sigma = 0.50$*. Unsurprisingly, the single-shot reconciliation approach yields a lower similarity score than dialectical reconciliation.*

7.5 Empirical Evaluations

We present two forms of empirical evaluations – a computational experiment and a human-subject experiment.

[47] Recall that KB_E is what the explainee thinks the agent's knowledge is, which means that they have no qualms adopting information from KB_R.

7.5.1 Computational Experiments

For our computational evaluation of DR-Arg, we utilize the following metrics to assess its performance:

- **Dialogue Length L:** The total number of dialogue moves exchanged between the explainer and explainee agents.
- **Dialogue Time T:** The duration of the dialogue, defined as the computational efforts required to generate arguments, assuming that communication cost is 0.
- **Number of Updates N:** The total count of updates to the explainee's knowledge base after the dialogue, reflecting the volume of new information incorporated.
- **Change in Similarity $\Delta\Sigma$:** The change in the similarity between KB_E and KB_R (for $a = 0.5$), comparing their initial (pre-interaction) and final (post-interaction) levels.

Setup: We created 16 unique pairs of KB_R and KB_E with sizes of $10^2 - 10^5$ by doing the following. (1) We generated random inconsistent propositional KBs of varying sizes of $10^2 - 10^5$. (2) We constructed KB_R by removing a minimal correction set (MCS) from the inconsistent KB to make them consistent.[48] (3) To create KB_E, we controlled the fraction of conflicts between the explainer and explainee with $c = |\text{KB}_E|/|\text{KB}_R|$. Specifically, starting with an empty KB_E, we added formulae from MCS and, if needed, negations of random formulae from KB_R to meet the desired ratio. This process generated distinct KBs with conflict levels determined by c. (4) Lastly, to have KBs of approximately the same size and with some similarity between them, we added a $1-c$ fraction of formulae from KB_R to KB_E, as long as KB_E remained satisfiable.

For generating arguments and counterarguments, we used a standard method from the literature [11]. The dialogue topic comprised a single `query` ϕ, created by finding a formula entailed by KB_R but not by KB_E. We identified this formula by examining the logical consequences of both knowledge bases. This process ensured the `query` addressed the knowledge discrepancy between the explainer and explainee, allowing to simulate a dialectical reconciliation dialogue.

[48] A MCS is a $\subseteq$-minimal set of formulae whose removal renders an inconsistent KB consistent [165].

We implemented a prototype of DR-Arg in Python using PySAT [115], and ran experiments with a time limit of 900s on a MacBook Pro machine with an M1 Max processor and 32GB of memory.[49]

Results: Table 7.5 presents the evaluation results of DR-Arg on various knowledge base sizes |KB| and fractions of conflicts c, allowing us to observe how they influence the dialogue time T, dialogue length L, number of updates N, the change in similarity with DR-Arg $\Delta\Sigma_{DR}$, and the change in similarity with a state-of-the-art single-shot reconciliation approach $\Delta\Sigma_{SSR}$ [225]. The results reveal several trends and insights:

- Increasing |KB| led to longer dialogue times (T), reflecting the higher computational demand for larger knowledge bases.
- Both the dialogue length (L) and the number of knowledge base updates (N) generally increased with larger |KB| and higher conflict ratios (c), indicating more extensive interactions required to resolve greater inconsistencies.
- A noticeable increase in $\Delta\Sigma_{DR}$ was observed with the rise in N, suggesting that more updates correlate with a greater improvement in the explainee's understanding. Notably, $\Delta\Sigma_{DR}$ consistently outperformed $\Delta\Sigma_{SSR}$, underscoring the advantage of DR-Arg's iterative, multi-move approach over single-shot reconciliation methods.

7.5.2 Human-Subject Experiments

We conducted a study involving the simulated scenario described in our motivating example (see Section 7.3.1). As a brief recap, a human user is presented with the task of troubleshooting an AI home assistant robot named "Roomie" that appears to be disconnected from the internet. The user is given a set of prompts to help them diagnose the problem, such as checking the associated mobile app, confirming Roomie's connection to the charging base, verifying Roomie's connection to the internet via a wired connector, and noting a flashing light next to the LAN port. However, the user is faced with several complications that hinder their ability to resolve the issue. These include an outdated mobile app, an expired license for the wired connection, and a low battery indicated by the flashing light. These obstacles

[49] Code repository: https://github.com/YODA-Lab/Dialectical-Reconciliation-with-Structured-Argumentation.

$\lvert KB \rvert$	$c = 0.2$					$c = 0.4$					$c = 0.6$					$c = 0.8$				
	T	L	N	$\Delta\Sigma_{DR}$	$\Delta\Sigma_{SSR}$	T	L	N	$\Delta\Sigma_{DR}$	$\Delta\Sigma_{SSR}$	T	L	N	$\Delta\Sigma_{DR}$	$\Delta\Sigma_{SSR}$	T	L	N	$\Delta\Sigma_{DR}$	$\Delta\Sigma_{SSR}$
2×10^2	0.05s	21	5	11.50%	9.00%	0.04s	11	1	10.10%	9.20%	0.02s	9	2	9.90%	9.20%	0.05s	9	2	9.95%	9.10%
4×10^2	0.07s	15	6	4.50%	2.50%	0.07s	15	6	5.20%	4.76%	0.05s	11	5	5.63%	4.19%	0.06s	11	5	5.60%	5.30%
6×10^2	0.10s	11	5	2.83%	1.37%	0.10s	11	5	2.15%	1.43%	0.20s	23	11	4.27%	1.58%	0.40s	59	29	11.57%	1.92%
8×10^2	0.30s	41	16	5.09%	0.80%	0.40s	43	20	6.45%	0.74%	0.40s	43	9	3.47%	0.73%	0.50s	43	8	3.50%	0.72%
2×10^3	0.50s	5	2	0.53%	0.83%	1.00s	23	9	2.50%	0.50%	2.40s	69	31	5.48%	0.45%	1.10s	25	10	3.57%	0.72%
4×10^3	4.30s	61	29	4.88%	0.37%	5.50s	71	34	6.05%	1.43%	10.20s	109	54	6.72%	0.59%	8.50s	85	42	6.37%	1.73%
6×10^3	3.50s	13	6	0.89%	0.20%	113.00s	87	40	4.65%	0.18%	3.70s	13	6	3.03%	0.24%	8.30s	57	28	4.93%	0.23%
8×10^3	7.60s	43	21	3.30%	1.53%	5.70s	19	9	4.03%	2.86%	37.90s	43	21	5.13%	4.18%	5.60s	19	9	4.45%	4.19%
2×10^4	21.20s	9	4	0.88%	0.15%	21.70s	9	4	0.10%	0.75%	21.60s	9	4	2.25%	0.68%	21.70s	9	4	2.49%	0.07%
4×10^4	38.40s	44	17	3.20%	1.95%	45.50s	66	18	4.30%	2.13%	50.20s	61	16	5.40%	4.19%	55.80s	68	23	6.20%	3.32%
6×10^4	125.30s	90	33	9.40%	7.31%	133.00s	111	52	29.40%	5.15%	129.60s	101	48	33.20%	17.20%	141.50s	120	61	44.90%	21.32%
8×10^4	149.00s	95	32	15.60%	4.79%	155.00s	129	59	25.40%	13.41%	161.50s	121	42	30.10%	21.29%	172.50s	155	72	39.30%	19.47%
2×10^5	220.20s	159	63	20.30%	13.14%	232.50s	191	82	32.00%	22.08%	242.00s	202	95	39.00%	25.50%	254.80s	233	108	50.20%	25.59%
4×10^5	386.60s	245	111	28.10%	15.48%	411.30s	287	135	37.90%	29.76%	430.00s	306	151	45.10%	32.70%	456.60s	340	168	57.40%	31.00%
6×10^5	561.20s	322	151	33.80%	19.29%	594.40s	378	178	41.80%	34.15%	622.60s	405	206	49.70%	37.80%	656.70s	446	227	63.10%	34.94%
8×10^5	739.20s	402	192	38.00%	21.92%	781.90s	473	229	45.20%	37.31%	816.30s	508	262	53.30%	41.30%	862.70s	556	287	67.60%	37.76%

Table 7.5: Evaluation of DR-Arg on various knowledge base sizes |KB| and fractions of conflicts c. The results represent averages from five runs per scenario.

create a realistic scenario for the user to navigate, as they must interact with Roomie to understand the underlying issues in order to get it up and running again.

Overall, this study provides a valuable opportunity to explore how humans interact with AI systems in real-world situations, and how they approach troubleshooting and problem-solving when faced with unexpected obstacles. From a technical standpoint, this narrative allowed us to approximate a human model, facilitating the use of a single-shot model reconciliation-based method as a baseline. A detailed setup of the study can be found in Appendix C.

Study Design: Participants were introduced to the problem through a narrative dialogue that explained the scenario's premise and known information. After posing the initial `query` "Why are you disconnected?", participants were divided into two groups:

- **Single-Shot (SSR)**: Group 1 received a single-shot model reconciliation explanation, where the human model was assumed to include the information provided during the scenario's introduction. The explanation was computed using the solver in [225].
- **DR-Arg**: Group 2 interacted with DR-Arg's explanations, choosing from four unique questions (i.e., counterarguments) in a game-like format. They could continue asking questions or decide to end the interaction.

Upon completing their interaction with Roomie, participants were asked four multiple-choice questions to evaluate their understanding of the issues, generating a comprehension score. They also responded to three Likert-scale questions (1: strongly disagree, 5: strongly agree) to gauge their satisfaction with the interaction and explanations, resulting in a satisfaction score. Our hypothesis was:

H: DR-Arg will achieve higher comprehension and satisfaction compared to SSR.

Study Results and Discussion

We recruited 100 participants through Prolific [175], of whom 97 completed the study. The participants were diverse in terms of age, gender, and educational background, with all of them being proficient in English and having at least an undergraduate degree. They were compensated with a base payment of $2.50 and had the opportunity to earn an additional $2.00 bonus for correctly answering the comprehension questions.[50]

In the DR-Arg group, participant engagement varied, leading us to further classify this group for analysis. Specifically, we divided the DR-Arg participants into two subgroups based on their interaction depth:

- **DR-Arg$_{\textbf{Single}}$**: This subgroup is comprised of participants who chose to end the interaction after only one question.
- **DR-Arg$_{\textbf{Multi}}$**: This subgroup is comprised of participants who engaged with more than one question.

This classification allowed us to evaluate the impact of deeper interaction on comprehension and satisfaction.

The study results, presented in Table 10.1, display the average scores for comprehension and satisfaction, alongside the statistical significance of differences between the SSR and DR-Arg groups.

[50] The study was approved by our institution's ethics board and adhered to the guidelines for responsible research practices.

	SSR	**DR-Arg**	**DR-Arg$_{\mathbf{Single}}$**	**DR-Arg$_{\mathbf{Multi}}$**
Number of Participants	49	48	11	37
Comprehension Score (out of 4)	0.30	2.60	1.18	**3.02**
Satisfaction Score (out of 5)	2.94	3.57	3.09	**3.74**

Table 7.6: Results of the user study.

The results of the user study are presented in Table 10.1. As hypothesized, the DR-Arg participants outperformed the SSR group in terms of both comprehension and satisfaction scores. The differences between the two groups were statistically significant according to independent samples t-tests, with p-values below 0.05.

The DR-Arg$_{\text{Single}}$ subgroup achieved better comprehension scores than the SSR group, suggesting that even a single interaction with DR-Arg can lead to improved understanding compared to a single-shot explanation. However, the most notable results were observed in the DR-Arg$_{\text{Multi}}$ subgroup, which obtained the highest comprehension and satisfaction scores among all groups. This finding highlights the effectiveness of deeper, multi-query interactions in dialectical reconciliation for enhancing user understanding and satisfaction.

As anticipated, the SSR participants scored lower on comprehension questions, possibly due to their inability to ask follow-up questions and only receiving information based on Roomie's assumed model of them. In contrast, the DR-Arg participants outperformed the SSR group, with the results being statistically significant according to a t-test with a p-value of 0.05. The DR-Arg$_{\text{Single}}$ subgroup showed improved comprehension over SSR, indicating that even minimal interaction with DR-Arg is more informative than a single-shot explanation. However, the most notable results were observed in the DR-Arg$_{\text{Multi}}$ subgroup, which achieved the highest comprehension and satisfaction scores. This underscores the efficacy of deeper, multi-query interactions in dialectical reconciliation for enhancing understanding and user satisfaction.

The study confirms our hypothesis **H**, illustrating that dialectical reconciliation is more effective in fostering understanding and addressing human user concerns than a single-shot approach.

7.6 Related Work

The influential work by Walton [234] provides a valuable framework for categorizing dialogues based on participants' knowledge, objectives, and governing rules. This categorization is essential for understanding the distinct characteristics and purposes of different dialogue types. Each dialogue type revolves around a central topic, typically a proposition, that serves as the subject matter of discussion.

Related dialogue types include: persuasion [97, 184], where an agent attempts to convince another agent to accept a proposition they initially do not hold; information-seeking [178, 73], where an agent seeks to obtain information from another agent believed to possess it; and inquiry [111, 16], where two agents collaborate to find a joint proof for a query that neither could prove individually.

While many dialogue systems have been proposed for these dialogue types [17], to the best of our knowledge, no existing dialogue frameworks have been developed exclusively for model reconciliation processes [37]. This is a crucial aspect of communication that sets our framework apart from related dialogue types, such as persuasion and information-seeking. To better illustrate this, in Section 7.3.2, we provide an example that clarifies the distinctions between our proposed dialogue type and persuasion and information-seeking.

On a similar thread, our work fits well within the literature on argumentation-based explainable AI [49]. However, a big difference with most existing approaches within that space [74, 41, 174, 25, 188] is that they are based on forms of *abstract* argumentation, which in our specific setting offers limited expressivity as the internal structure of arguments is ignored. In a practical explanatory dialogue setting with implementations for user studies (such as in our case), one must know and express the contents of the arguments conveyed, and how they can be used to generate new arguments and counterarguments.[51]

In similar spirit, Dennis *et al.* [59] proposed a framework for explaining the behavior of BDI systems. However, the differences lie in the underlying formalisms (BDI vs structure deductive argumentation), and importantly, their methodology lacks an experimental evaluation. In contrast, we include both computational experiments and a human-user study, providing

[51]That is why we opted to using deductive argumentation, a form of *structured* argumentation, whose key feature is the clarification of the nature of arguments and counterarguments.

a more robust and empirically grounded understanding of the framework's effectiveness. In an orthogonal direction, Teze *et al.* [215] proposed an argumentation-based approach for epistemic planning that allows for handling contextual preferences of users during plan construction, but without explainability considerations. In contrast, our framework can be used to explain planning problems to users via argumentation-based dialogues.

Finally, our work is motivated by the model reconciliation process (MRP) [37, 211, 209], and specifically the logic-based variant [202, 225, 231, 227]. Our framework addresses two MRP limitations: (1) the explainer agent's assumed knowledge of the human model (we relax this assumption) and (2) single-shot interactions (we focus on dialogue-based interactions). Notably, Dung *et al.* [65] tackle these limitations using answer set programming, but their approach is tied to planning problems while ours can be used to express general problems. Specifically, our framework relies on the general notion of argument/counterargument, while theirs discuss only arguments related to optimal planning, and it is not clear how to extend it to our general context. Moreover, their framework is purely theoretical and lacks experimental evaluation.

7.7 Concluding Remarks

While argumentation is often advocated as suitable for explanation, its effectiveness for human users remains understudied. This chapter has enhanced our logical explainability layer by introducing Dialectical Reconciliation via Structured Argumentative Dialogues (DR-Arg), showing how structured argumentation can enable dynamic interaction between AI systems and human users. Our main contributions and findings are:

1. We provided a formal framework for dialectical reconciliation dialogues, including operational semantics and theoretical guarantees for termination and success.
2. We introduced a metric for approximating explainee understanding in the context of these interactions.
3. Our empirical evaluations, through both computational experiments and human-subject studies, attested to the efficacy of dialectical interactions in enhancing human-AI interactions.

These findings highlight the potential of argumentation-based approaches in enhancing the human-AI interaction of AI systems, particularly in domains where explainability is crucial.

Despite the promising aspects of our framework, it is important to acknowledge its limitations and potential areas for improvement. DR-Arg follows a fixed structure in presenting arguments and does not consider the effectiveness of personalizing the interactions according to the user's beliefs and preferences. DR-Arg also assumes seamless communication through well-defined dialogue moves, which may not reflect real-world complexities such as miscommunication or uncertainty. Finally, the current framework is limited to deductive argumentation and propositional logic, which may not be sufficient to express complex relationships and dependencies in real-world domains.

To address these limitations, we suggest the following future directions: (1) Develop an adaptive approach that tailors arguments to individual users' needs and preferences based on user feedback and prior interactions [211, 227]. In Tang *et al.* [214], we have taken a step towards this end by proposing a probabilistic framework to approximate human user models from argumentation-based dialogues; (2) Integrate DR-Arg with large language models [24] to translate formal arguments and logical structures into intuitive, natural language expressions, enhancing accessibility and user-friendliness while maintaining logical coherence; and (3) Consider alternative structured argumentation frameworks, such as ABA [21, 50] or probabilistic argumentation frameworks [144, 114], to enable more complex reasoning and argument generation for a wider range of real-world problems.

In the next chapter, we will switch gears and examine how our logical explainability layer can address scheduling problems, demonstrating its practical applicability while handling additional challenges like privacy concerns.

Chapter 8

Explanation-guided Belief Revision

> *"The growth of knowledge depends entirely on disagreement."*
>
> *— Karl Popper*

8.1 Introduction & Contribution

Throughout this thesis, we have developed a logical explainability layer for AI systems, from the foundational Logic-based Model Reconciliation Problem (L-MRP) to its extensions for probabilistic settings, efficient computation, personalization, and dialectical interaction. While these approaches have demonstrated the usefulness and power of logic, they share an underlying assumption: that effective explanation involves minimal updates to the human's model. However, as we move deeper into human cognition and real-world interactions, we find that this assumption of minimality may not align with how humans actually process and integrate new information.

When faced with new information that challenges their existing beliefs, people do not simply make minimal adjustments. Instead, they engage in a more complex process of understanding and reconciliation. This discrepancy between our previous models and actual human cognition prompts us to reconsider our approach to revising human models in the context of explainable AI.

One perspective on rational change is the principle of *minimalism* (or information economy), which emphasizes a minimal change to one's beliefs. James [119] eloquently expressed this notion, suggesting that new information should be integrated in a way that slightly stretches people's existing beliefs just enough to incorporate the new information. While minimalism has been championed by numerous cognitive scientists in the context of both scientific and everyday decision-making [87, 105], it can be limiting in complex real-world situations where multiple interrelated beliefs may need to be updated simultaneously.

Cognitive studies on human belief revision indicate that people may not always aim for minimality when revising their beliefs. Instead, they first seek to understand the nature of the inconsistency [216, 123]. When encountering conflicting information, individuals often generate *explanations* to reconcile these inconsistencies, as explanations offer a clearer guide for future actions than mere belief adjustments [43, 137] and play a vital role in communicating one's understanding of the world [39, 159]. This approach suggests that people construct explanations to resolve inconsistencies, leading to belief revisions that are often not minimal [123, 139].

Drawing inspiration from these cognitive studies, we argue that minimalism should not be the primary principle in belief revision frameworks aimed at human users. We argue that a drive for *explanatory understanding*, rather than mere consistency of beliefs, is a key feature of human reasoning that belief revision frameworks should account for. Building on this foundation, we introduce the *human-aware belief revision* framework, which formalizes the process of revising human beliefs in light of explanations.

In summary, the contributions of this chapter are:

1. We introduce a framework for human-aware belief revision inspired by human cognition, focusing on explanatory understanding over minimal changes, which better reflects real-world cognitive processes.

2. We present the *explanation-guided* revision operator that revises the human model in a (possibly) non-minimal way while preserving new information, offering a more flexible approach to belief revision.

3. We conduct empirical evaluations through two human-subject studies that provide robust evidence for the applicability of our proposed framework, validating its effectiveness in real-world scenarios.

Our key insight is that *an explanation-guided belief revision operator not only aligns more closely with natural human reasoning patterns but also offers a more effective and human-aware approach to belief revision that extends beyond the minimal update approach of L-MRP.*

8.2 Background

8.2.1 Belief Revision Theory

Belief revision is generally divided into two perspectives [88]: the *coherence model*, which deals with sets of formulae that are closed under a consequence relation (belief sets) [2], and the *foundational model*, which focuses on non-closed sets (belief bases) [103, 104]. In this work, we adopt the foundational model approach. There, revision operators are typically constructed via *kernel functions* [103] that select among (minimal) subsets of a belief base that contribute to making it imply a formula. This is a general approach that relies on incision functions to determine the beliefs to be removed from each kernel. For a complete overview of different approaches to belief revision, please refer to [75].

Minimalism

The *principle of minimalism* (also called the principle of informational economy or minimal mutilation) is one of the basic conceptual principles underlying belief revision frameworks:

> **Principle of Minimalism:** When an agent with a prior belief base is presented with a new belief that is inconsistent with it, they should revise it with respect to the new belief to get a posterior belief base that is *the closest belief base to their prior belief base.*

In essence, the principle of minimalism states that the agent's primary goal when resolving inconsistencies is to make minimal changes to their (existing) beliefs. This basic principle is encapsulated by certain postulates, such as recovery [84, 2], core-retainment, and relevance [102, 72], or by constructing revision operators that restrict revisions to minimal subsets of the original belief base [72].

Explanatory Understanding However intuitive and plausible minimalism appears to be, it does not always hold true in human belief revision dynamics. Studies in cognitive science show that when people are faced with inconsistencies, they first construct (or seek) explanations to resolve the inconsistencies, which consequently lead them to revise their beliefs in a non-minimal fashion [71, 123, 139, 233]. We will refer to this as *explanatory understanding*:

> **Explanatory Understanding:** When an agent with a prior belief base is presented with a new belief that is inconsistent with it, they first seek for an explanation to explain the origin of the inconsistency, which can then lead them to make greater than minimal changes to their belief base.

The explanatory understanding poses a challenge to the principle of minimalism, and thus to most belief revision frameworks that adhere to it. Crucially, this means that existing belief revision frameworks may fail to be successfully applied to human-AI settings. In this work, we argue against minimalism and for explanatory understanding in human-centric belief revision frameworks.

8.2.2 Logical Preliminaries

We will adopt an equality-free first-order language $\mathcal{L}$ comprising (finite) sets of constants $\mathcal{C}$, variables $\mathcal{V}$, predicates $\mathcal{P}$, and no explicit existential quantifiers. An atom takes the form $p(t_1, \ldots, t_n)$, where p is a predicate and $t_i \in \mathcal{C} \cup \mathcal{V}$ are terms. A *ground* (or instantiated) atom is an atom without variables, otherwise it is *lifted*. A formula is built out of atoms using quantifier $\forall$ and the usual logical connectives $\neg$, $\vee$, and $\wedge$. An *interpretation* is an assignment of $true$ or $false$ to each ground atom in a set of formulae. If an interpretation satisfies a set of formulae then it is called a *model* of that set. A set of formulae is *consistent* if it has at least one model, otherwise it is *inconsistent*.

To align with the cognitive flexibility observed in human reasoning, we make the fundamental assumption that beliefs are *defeasible*—that is, they are open to change and retraction. To capture a more nuanced structure of human beliefs, we represent a human model as a *belief base* consisting of *facts* and *general rules* from language $\mathcal{L}$. This approach allows us to model both specific pieces of information and the broader principles that guide human reasoning. Formally,

Definition 48 (Belief Base)**.** *A belief base is a tuple* $\mathcal{B} = \langle \mathcal{F}, \mathcal{R} \rangle$*, where:*

- *$\mathcal{F} \subseteq \mathcal{L}$ is a set of facts, each of which is a ground atom or its negation.*
- *$\mathcal{R} \subseteq \mathcal{L}$ is a set of general rules, each having the form of a conditional $P(\vec{x}) \rightarrow Q(\vec{y})$, where $\vec{x}$ and $\vec{y}$ are variable tuples, and $P(\vec{x})$ and $Q(\vec{y})$ are formulae in $\mathcal{L}$.*

We denote the ground version of $\mathcal{B}$ as $\mathcal{B}^\gamma$.

A belief base $\mathcal{B}$ entails a formula φ, denoted by $\mathcal{B} \models \varphi$, iff $\mathcal{B}^\gamma \cup \neg\{\varphi\} \models \bot$, where $\bot$ denotes falsum. We will also use $\Gamma(\mathcal{B})$ to denote the (primitive) *consequences* (e.g., set of all ground atoms) of $\mathcal{B}$, i.e., $\Gamma(\mathcal{B}) = \{\phi \mid \phi \in \mathcal{L}, \mathcal{B} \models \phi\}$.[52]

For convenience, and unless specified otherwise, we will write a belief base as the set $\mathcal{B} = \mathcal{F} \cup \mathcal{R}$, implicitly assumed to be a conjunction of the facts and rules.

8.3 Towards Human-Aware Belief Revision

Consider an illustrative scenario involving two agents, Alice and Bob. Suppose that Alice believes that (a) *If people are worried, then they have insomnia*, (b) *Diana is worried*, and (c) *Charlie is worried.* As such, she believes that (d) *Charlie and Diana have insomnia.* Now, suppose that Bob tells Alice that (e) *Charlie does not have insomnia.* Before incorporating this new information that contradicts her beliefs, Alice will naturally seek an explanation from Bob. For instance, Bob would then explain to Alice that (f) *Charlie has a coping strategy* and that (g) *people with coping strategies may not have insomnia despite being worried.* If the explanation stands up to scrutiny, Alice will then integrate this explanation and into her beliefs. In other words, it is the explanation that will drive Alice's revision process.

An explanation aims to explain (or, "rationalize") a particular phenomenon, referred to as the *explanandum*, to someone. Essentially, an explanation is reasoning in "reverse", e.g., a set of beliefs adduced as explanations of the explanandum. More formally,

Definition 49 (Explanation)**.** *We say that $\mathcal{E} \subseteq \mathcal{L}$ is an* explanation *for an explanandum $\varphi \subseteq \mathcal{L}$ iff: (1) $\mathcal{E} \models \varphi$; (2) $\mathcal{E} \not\models \bot$; and (3) $\forall \mathcal{E}' \subset \mathcal{E}$, $\mathcal{E}' \not\models \varphi$.*

[52] This is also called the backbone of the belief base [176].

The first condition determines that the explanandum is derived by the set of beliefs $\mathcal{E}$, while the second condition averts the possibility that an explanandum is derived from an inconsistent set. The final condition ensures that there are no irrelevant beliefs in the explanation. Note that we assume the explanation is also a belief base (Definition 48).

Example 26. *Building on the illustrative scenario, Alice's belief base is* $\mathcal{B}_A = \{Wor(Charlie), Wor(Diana), \forall x. Wor(x) \rightarrow Ins(x)\}$*. It is easy to see that* $\mathcal{B}_A \models Ins(Charlie)$*. Upon encountering the contradictory explanandum* $\neg Ins(Charlie)$*, the explanation for it from Bob is* $\mathcal{E} = \{Wor(Charlie), Cop(Charlie), \forall x. Wor(x) \wedge Cop(x) \rightarrow \neg Ins(x)\}$*.*

What is important to note here is that the explanation should aim to enable Alice's *explanatory understanding* of the explanandum, that is, Alice understands why Charlie does not have insomnia if she can produce an explanation for it. Therefore, a good explanation should drive the receiving agent's belief revision in such a way that explanatory understanding of the given explanandum is satisfied.

Definition 50 (Explanatory Understanding)**.** *Let* $\mathcal{B}$ *be the belief base of an agent,* $\mathcal{E}$ *an explanation for explanandum* φ*, and* $\odot$ *a revision operator such that* $\mathcal{B} \odot \mathcal{E}$ *denotes the revision of* $\mathcal{B}$ *with* $\mathcal{E}$*. We say that the agent has explanatory understanding of* φ *after receiving* $\mathcal{E}$ *if* $\mathcal{B} \odot \mathcal{E} \models \varphi$*.*

Example 27. *Continuing from Example 26, notice that the Alice's belief base* $\mathcal{B}_A$ *and the explanation* $\mathcal{E}$ *are inconsistent, i.e.,* $\mathcal{B}_A \cup \mathcal{E} \models \perp$*. Thus, a revision operator* $\odot$ *must re-establish consistency within Alice's belief base while enabling explanatory understanding of the explanandum* $\neg In(Charlie)$*. But how this revision should happen? Should Alice, in light of* $\mathcal{E}$*, still maintain the general belief that people have insomnia if they are worried?*

When resolving inconsistencies with explanations, we would like to have a revision operator that, on the one hand, yields a consistent revised belief base, but on the other hand guarantees explanatory understanding of the explanandum. But how should such a revision operator behave? In what follows, we argue it should not be constrained by the principle of minimalism, e.g., minimal changes to the belief base, but rather be guided by the explanation, which might lead to larger than minimal revisions.

8.3.1 The Explanation-Guided Revision Operator

We build upon the concept of *kernel revision* [103]. Particularly, we assume that the belief base $\mathcal{B}$ we are dealing with is fully grounded,[53] and start by defining the notion of *correction kernel*:

Definition 51 (Correction Kernel)**.** *Let* $\mathcal{B}$ *be a belief base and* $\mathcal{E}$ *an explanation for explanandum* φ*. If* $\mathcal{B} \cup \mathcal{E} \models \bot$*, then the correction kernel of* $\mathcal{B} \cup \mathcal{E}$ *is defined as* $(\mathcal{B} \cup \mathcal{E})^{\perp} = \{\mathcal{B}' \,|\, \mathcal{B}' \subseteq \mathcal{B} \cup \mathcal{E}, (\mathcal{B} \cup \mathcal{E}) \setminus \mathcal{B}' \not\models \bot,$ *and* $(\mathcal{B} \cup \mathcal{E}) \setminus \mathcal{B}' \neq \emptyset\}$*. If* $\mathcal{B} \cup \mathcal{E} \not\models \bot$*, then* $(\mathcal{B} \cup \mathcal{E})^{\perp} = \mathcal{B} \cup \mathcal{E}$*.*

In other words, the correction kernel of an inconsistent belief base is the set of all subsets of the belief base whose removal render the belief base consistent. The elements of the correction kernel are called *correction sets*. Note how we do not impose any minimality constraints on the resulting correction sets.

Example 28. *Consider the belief base* $\mathcal{B} = \{Wor(Charlie), Wor(Charlie) \rightarrow Ins(Charlie)\}$ *and explanation* $\mathcal{E} = \{\neg Ins(Charlie)\}$ *for explanandum* $\neg Ins(Charlier)$*. The correction kernel of* $\mathcal{B} \cup \mathcal{E}$ *is* $(\mathcal{B} \cup \mathcal{E})^{\perp} = \{\{Wor(Charlie)\}, \{Wor(Charlie) \rightarrow Ins(Charlie)\}, \{\neg Ins(Charlie)\}, \{Wor(Charlie), Wor(Charlie) \rightarrow Ins(Charlie)\}, \{Wor(Charlie), \neg Ins(Charlie)\}, \{\neg Ins(Charlie), Wor(Charlie) \rightarrow Ins(Charlie)\}\}$*.*

As mentioned earlier, we want our operator to not only restore consistency (with no minimality guarantees), but also to enable explanatory understanding of the explanandum φ (Definition 50). This leads us to define the notion of the *φ-preserving (explanandum-preserving) selection function*:

Definition 52 (φ-Preserving Selection Function)**.** *Let* $\mathcal{B}$ *be a belief base and* $\mathcal{E}$ *an explanation for explanandum* φ*. A* φ*-preserving selection function for* $\mathcal{B}$ *with respect to* $\mathcal{E}$ *is defined as* $\Sigma : 2^{2^{\mathcal{L}}} \mapsto 2^{\mathcal{L}}$ *such that: (1)* $\Sigma((\mathcal{B} \cup \mathcal{E})^{\perp}) \in (\mathcal{B} \cup \mathcal{E})^{\perp}$*; and (2)* $(\mathcal{B} \cup \mathcal{E}) \setminus \Sigma((\mathcal{B} \cup \mathcal{E})^{\perp}) \models \varphi$*.*

A φ-preserving selection function selects a correction set from the correction kernel $(\mathcal{B} \cup \mathcal{E})^{\perp}$ (condition (1)) whose removal does not affect the entailment of the explanandum φ (condition (2)). Note that in the case where $\mathcal{B} \cup \mathcal{E} \not\models \bot$, then $\Sigma((\mathcal{B} \cup \mathcal{E})^{\perp}) = \emptyset$.

[53] This restriction is only made for convenience and illustrative purposes. The results continue to hold for lifted belief bases.

Example 29. *Consider the belief base and explanation from Example 8.2. Possible results of the* $\neg In(Charlie)$*-preserving selection function* $\Sigma((\mathcal{B} \cup \mathcal{E})^{\perp})$ *are* $\{Wor(Charlie)\}$, $\{Wor(x) \rightarrow Ins(x)\}$ *and* $\{Wor(Charlie), Wor(x) \rightarrow Ins(x)\}$.

Having defined the correction kernel and the φ-preserving selection function, we now formally define the *explanation-guided belief revision operator*, which is a function mapping a belief base and an explanation to a revised belief base:

Definition 53 (Explanation-Guided Belief Revision Operator)**.** *Let* $\mathcal{B}$ *be a belief base,* $\mathcal{E}$ *an explanation for explanandum* φ*, and* Σ *a* φ*-preserving selection function. The operator of explanation-based belief revision on* $\mathcal{B}$ *with* $\mathcal{E}$ *is defined as* $\odot : 2^{\mathcal{L}} \times 2^{\mathcal{L}} \mapsto 2^{\mathcal{L}}$, $\mathcal{B} \odot \mathcal{E} = (\mathcal{B} \cup \mathcal{E}) \setminus \Sigma((\mathcal{B} \cup \mathcal{E})^{\perp})$.

The mechanism of the explanation-guided revision operator is to first add the explanation $\mathcal{E}$ to the belief base $\mathcal{B}$, and then retract from the result a correction set by means of a selection function that makes a choice among possible sets in the correction kernel of $\mathcal{B} \cup \mathcal{E}$, while also ensuring that the resulting belief base entails the explanandum.

Example 30. *Consider the belief base* $\mathcal{B}$ *and explanation* $\mathcal{E}$ *from Example 29, and assume that we select* $\Sigma((\mathcal{B} \cup \mathcal{E})^{\perp}) = \{Wor(Charlie) \rightarrow Ins(Charlie)\}$*. Then the explanation-guided revision on* $\mathcal{B}$ *with* $\mathcal{E}$ *is* $\mathcal{B} \odot \mathcal{E} = \{Wor(Charlie), Wor(Charlie) \rightarrow Ins(Charlie), \neg Ins(Charlie)\} \setminus \{Wor(Charlie) \rightarrow Ins(Charlie)\} = \{Wor(Charlie), \neg Ins(Charlie)\}$*. Another possible revision is* $\mathcal{B} \odot \mathcal{E} = \{\neg Ins(Charlie)\}$*. In fact, no matter the choice of the selection function, it is guaranteed that* $\mathcal{B} \odot \mathcal{E} \models \neg Ins$*.*

It is important to mention that the operator's result–specifically, the determination of which beliefs to retract–is influenced by the epistemic attitude of the agent, insofar as the agent's preexisting beliefs, level of skepticism, and openness to new information shape the outcome of belief revision. As we focus on the importance of explanations and non-minimal revisions in this work, we leave this nuanced aspect of belief dynamics for future work.

A Simple Measure of Belief Change: Measures for calculating the amount of belief change typically depend on counting all the beliefs that change their values [71, 105]. As such, to have an effective theoretical measure for quantifying the amount of change in a belief base, we propose the following definition:

Definition 54 (Measure of Belief Change)**.** *Let $\mathcal{B}$ be a prior belief base and $\mathcal{B}'$ a posterior belief base. We define the measure of belief change between $\mathcal{B}$ and $\mathcal{B}'$ as:*

$$\mathcal{D}(\mathcal{B}, \mathcal{B}') = \frac{|\Gamma(\mathcal{B}) \vartriangle \Gamma(\mathcal{B}')|}{|\Gamma(\mathcal{B}) \cup \Gamma(\mathcal{B}')|},$$

where $\Gamma(\mathcal{B}) \vartriangle \Gamma(\mathcal{B}') = (\Gamma(\mathcal{B}) \setminus \Gamma(\mathcal{B}') \cup (\Gamma(\mathcal{B}') \setminus \Gamma(\mathcal{B}))$, and $\Gamma(\mathcal{B})$ is the consequences of $\mathcal{B}$.

Example 31. *Consider the prior belief base $\mathcal{B} = \{Wor(Charlie),\ Wor(Diana),\ Wor(Charlie) \rightarrow Ins(Charlie),\ Wor(Diana) \rightarrow Ins(Diana)\}$, with consequences $\Gamma(\mathcal{B}) = \{Wor(Charlie), Wor(Diana), Ins(Charlie), Ins(Diana)\}$.*

Now, assume the explanation $\mathcal{E} = \{Wor(Charlie),\ Cop(Charlie),\ Wor(Charlie) \wedge Cop(Charlie) \rightarrow \neg Ins(Charlie)\}$ for explanandum $\neg Ins(Charlie)$, and consider the following revisions of $\mathcal{B}$ with $\mathcal{E}$:

- ***Minimal:*** *$\mathcal{B}' = \mathcal{B} \odot \mathcal{E} = \{\ Wor(Charlie),\ Wor(Diana),\ Wor(Diana) \rightarrow Ins(Diana),\ Cop(Charlie),\ Wor(Charlie) \wedge Cop(Charlie) \rightarrow \neg Ins(Charlie)\}$. The consequences of $\mathcal{B}'$ are $\Gamma(\mathcal{B}') = \{Wor(Charlie), Wor(Diana), Ins(Diana),\ Cop(Charlie), \neg Ins(Charlie)\}$. Computing the measure of belief change between $\mathcal{B}$ and $\mathcal{B}'$ we get $\mathcal{D}(\mathcal{B}, \mathcal{B}') = \frac{3}{6} = 0.5$.*
- ***Non-minimal:*** *$\mathcal{B}'' = \mathcal{B} \odot \mathcal{E} = \{Wor(Charlie),\ Wor(Diana),\ Cop(Charlie),\ Wor(Charlie) \wedge Cop(Charlie) \rightarrow \neg Ins(Charlie)\}$, with consequences $\Gamma(\mathcal{B}'') = \{Wor(Charlie), Wor(Diana),\ Cop(Charlie), \neg Ins(Charlie)\}$. Then, computing the measure of belief change we get $\mathcal{D}(\mathcal{B}, \mathcal{B}'') = \frac{4}{6} = 0.66$.*

As seen in the above example, a non-minimal revision obviously yields a higher belief change. At a first glance, non-minimal revisions might seem counter-intuitive–why should an agent discard more beliefs than what is minimally necessary for consistency? Are not these beliefs irrelevant to the explanandum anyway? But as we will see in the next section, these are not irrelevant beliefs. In fact, our operator allows us to capture how people revise their beliefs: *non-minimally with explanations as their guide.*

Illustrative Example with Existing Revision Operator

A reader familiar with the belief revision literature may wonder what is the difference between the explanation-guided belief revision operator we presented in this section and existing belief base revision frameworks. First, we want to note that our framework is specifically intended for human-aware AI settings. Now, closest to our operator is the belief revision operator proposed by Falappa *et al.* [72], which considers the revision of belief bases with explanations (or sets of formulae). The following example is aimed at illustrating the primary differences between the two approaches.

The belief revision operator of Falappa *et al.* [72], referred to by the authors as kernel revision by a set of sentences, is defined as follows:

Definition 55 ([72] Revision Operator)**.** *Let $\mathcal{B}$ be a belief base and E an explanation. The kernel revision by a set of sentences $\diamond_F$ is defined as $\mathcal{B} \diamond_F E = (\mathcal{B} \cup E) \setminus \sigma((\mathcal{B} \cup E)^{\perp\!\!\perp}\bot)$, where $(\mathcal{B} \cup E)^{\perp\!\!\perp}\bot$ is a kernel set [103] (e.g., minimal unsatisfiable subsets of $\mathcal{B} \cup E$), and $\sigma((\mathcal{B} \cup E)^{\perp\!\!\perp}\bot)$ an incision function such that $\sigma((\mathcal{B} \cup E)^{\perp\!\!\perp}\bot) \subseteq \bigcup(\mathcal{B} \cup E)^{\perp\!\!\perp}\bot$ and if $X \in (\mathcal{B} \cup E)^{\perp\!\!\perp}\bot$ and $X \neq \emptyset$ then $X \cap \sigma((\mathcal{B} \cup E)^{\perp\!\!\perp}\bot) \neq \emptyset$.*

First, we see from the above definition that the revision by Falappa *et al.* [72] does not satisfy explanatory understanding, i.e., it makes it possible for the revised belief base to not entail the explanandum. Second, it follows the principle of by minimalism. Let us illustrate the differences in the following example.

Let w_c be Wor($Charlie$), w_d be Wor($Diana$), c_c be Cop($Charlie$), i_c be Ins($Charlie$), and i_d be Ins($Diana$).

Consider the belief base:

$$\mathcal{B} = \{w_c, w_d, w_c \rightarrow i_c, w_d \rightarrow i_d\},$$

which has consequences $\Gamma(\mathcal{B}) = \{w_c, w_d, i_c, i_d\}$. Now, assume the following explanation for explanandum $\neg i_c$:

$$\mathcal{E} = \{w_d, c_c, w_c \wedge c_c \rightarrow \neg i_c\}.$$

Using the framework by Falappa *et al.* [72] to revise $\mathcal{B}$ with $\mathcal{E}$, we go through the following steps:

1. **Kernel Set:** $(\mathcal{B} \cup \ \mathcal{E})^{\perp\!\!\perp}\bot = \{w_c, c_c, w_c \rightarrow i_c, w_c \wedge c_c \rightarrow \neg i_c\}\}$.

2. **Incision Function:** Possible results of $\sigma((\mathcal{B} \cup E)^{\perp\!\!\!\perp}\bot)$ are: $\{w_c\}$, $\{w_c \to i_c\}$ $\{w_c \wedge c_c \to \neg i_c\}$, $\{c_c, w_c \wedge c_c \to \neg i_c\}$, and so on.

What is important to highlight here are two things: (1) *The incision function only selects subsets to retract from the minimal unsatisfable set, i.e., only those beliefs deemed "relevant" wrt minimality*; and (2) *It is possible for the revised belief base to reject the explanandum*, i.e., $\mathcal{B} \diamond_F \mathcal{E} \not\models \neg i_c$.

In contrast, our operator works differently. First, it ensures that the revised belief base always entails the explanandum, i.e., $\mathcal{B} \odot E \models \neg i_c$, in all possible results of the operator, while also not constrained to choose beliefs based on a minimality criterion. Importantly, notice how $w_d \to i_d \in \mathcal{B}$ will never be in any of the possible solutions of Falappa *et al.*'s operator, as it is not included in any of the kernels in $(\mathcal{B} \cup E)^{\perp\!\!\!\perp}\bot$. However, in our case $w_d \to i_d \in (\mathcal{B} \cup E)^{\bot}$, and as such, it is possible for it to be in the solution of our operator. Again, while this might seem as an "irrelevant" revision that should be avoided (i.e., violates the minimal change principle), our empirical findings strongly support it, namely that given an explanation of an inconsistency, people, more frequently than not, make non-minimal revisions to their beliefs.

Finally, it is very important to note that we do not wish to replace any of the existing belief revision operators from the literature. Each has its own suitability and application. What we wish to do is present a framework that is tailored to human belief revision processes. That is, our operator is intended for human-aware AI settings.

8.3.2 Rationality Postulates and Axiomatization

Following the tradition in belief revision literature [84, 72, 75], we now present some basic rationality postulates for the explanation-guided belief revision. Rationality postulates are fundamental guidelines dictating the behavior of a revision operator – they specify what the operator's response should be, when provided with certain inputs, but not its internal mechanism.[54]

We consider the following postulates:

[54] The internal mechanism is specified in the construction of the operator we presented in the previous section.

- *Inclusion*: $\mathcal{B} \odot E \subseteq \mathcal{B} \cup E$.
  This postulate establishes that if the agent revises its belief base $\mathcal{B}$ with the explanation E, then the new belief base will stem from the union of $\mathcal{B}$ and E.

- *Vacuity*: If $\mathcal{B} \cup E \not\models \bot$, then $\mathcal{B} \odot E = \mathcal{B} \cup E$.
  This postulate establishes that if the belief base $\mathcal{B}$ is consistent with the explanation E, then the revision simply expands $\mathcal{B}$ with E.

- *Consistency*: If $\mathcal{B} \cup E \models \bot$, then $\mathcal{B} \odot E \not\models \bot$.
  This postulate ensures that the revised belief base is consistent.

- *Reversion:* If $(\mathcal{B} \cup E)^{\perp} = (B \cup E')^{\perp}$, then $(B \cup E) \setminus (\mathcal{B} \odot E) = (B \cup E') \setminus (\mathcal{B} \odot E')$.

  This postulates states that if, if $\mathcal{B} \cup E$ and $\mathcal{B} \cup E'$ have the same correction kernels, then the beliefs retracted in the respective revisions with respect to E and E' are the same.

- *Constrained Acceptance*: If $\mathcal{B} \not\models \neg\varphi$, then $\mathcal{B} \odot E \models \varphi$.
  This postulate establishes that if the agent receives an explanation E for an explanandum φ that is not rejected in its belief base $\mathcal{B}$, then φ will be accepted in its revised belief base.

- *Unconstrained Acceptance*: If $\mathcal{B} \models \neg\varphi$, then $\mathcal{B} \odot E \models \varphi$.
  This postulate establishes that if the agent receives an explanation E for an explanandum φ that is rejected in its belief base $\mathcal{B}$, then φ will be accepted in its revised belief base.

- *Strong Acceptance*: $\mathcal{B} \odot E \models \varphi$.
  This postulate ensures that the explanandum φ can be inferred from the revised belief base.

Two obvious relations between the postulates follow.

Proposition 6. *If the operator $\odot$ satisfies vacuity, then it satisfies consistency and strong acceptance.*

Proof. From vacuity, we have that $\mathcal{B} \cup E \not\models \bot$ and $\mathcal{B} \odot E = \mathcal{B} \cup E$. It follows then that $\mathcal{B} \odot E \not\models \bot$, satisfying consistency. Now, from the definition of an explanation we have that

$E \models \varphi$, and from the monotonicity of classical logic we have that $\mathcal{B} \odot E \models \varphi$. Therefore, strong acceptance is satisfied. $\square$

Proposition 7. *If the operator $\odot$ satisfies strong acceptance, then it satisfies constrained acceptance and unconstrained acceptance.*

Proof. From strong acceptance, $\mathcal{B} \odot E \models \varphi$, thus automatically satisfying constrained and unconstrained acceptance. $\square$

Finally, an axiomatic characterization of the explanation-guided belief revision operator is as follows.

Theorem 13. *Let $\mathcal{B}$ be a belief base and E an explanation for explanandum φ. The operator $\odot$ is an explanation-guided belief revision operator on $\mathcal{B}$ with E if and only if it satisfies inclusion, consistency, reversion, and strong acceptance.*

Proof. We first prove in the direction of construction to postulates. Let $\mathcal{B} \odot E = (\mathcal{B} \cup E) \setminus \Sigma((\mathcal{B} \cup E)^{\perp})$:

- *Inclusion*: Since $\mathcal{B} \odot E = (\mathcal{B} \cup E) \setminus \Sigma((\mathcal{B} \cup E)^{\perp})$, it follows that $\mathcal{B} \odot E \subseteq \mathcal{B} \cup E$.
- *Consistency*: If $\mathcal{B} \cup E \models \perp$, then the operator retracts a correction set $\Sigma((\mathcal{B} \cup E)^{\perp})$ from $\mathcal{B} \cup E$, thus rendering it consistent.
- *Reversion*: Suppose that $(\mathcal{B} \cup E)^{\perp} = (B \cup E')^{\perp}$. Because Σ is well-defined we have that $\Sigma((B_A \cup E)^{\perp}) = \Sigma((B_A \cup E')^{\perp})$. Now, if $\alpha \in (\mathcal{B} \cup E) \setminus \mathcal{B} \odot E$, then $\alpha \in \Sigma((B_A \cup E)$, and consequently, $\alpha \in \mathcal{B} \cup E'$ and $\alpha \notin \mathcal{B} \odot E'$. Hence, $(\mathcal{B} \cup E) \setminus \mathcal{B} \odot E \subseteq (\mathcal{B} \cup E') \setminus \mathcal{B} \odot E'$. Similarly, starting with $\alpha \in (\mathcal{B} \cup E') \setminus \mathcal{B} \odot E'$, we get that $(\mathcal{B} \cup E') \setminus \mathcal{B} \odot E' \subseteq (\mathcal{B} \cup E) \setminus \mathcal{B} \odot E$, thereby satisfying reversion.
- *Strong Acceptance*: Since $E \models \varphi$, and since the φ-preserving selection function $\Sigma((\mathcal{B} \cup E)^{\perp})$ selects sets that do not affect the entailment of φ in $\mathcal{B} \cup E$, it follows that $(\mathcal{B} \cup E) \setminus \Sigma((\mathcal{B} \cup E)^{\perp}) \models \varphi$, thus satisfying strong acceptance.

We now prove in the reverse direction from postulates to construction. We need show that if a belief revision operator satisfies the postulates then it is possible to build the explanation-guided belief revision operator. Let Σ be a function such that for every pair of belief bases $\mathcal{B}$ and E, it holds that: $\Sigma((\mathcal{B} \cup E)^{\perp}) = \{\alpha \,:\, \alpha \in (\mathcal{B} \cup E) \setminus \mathcal{B} \odot E\}$. We must show that:

1. Σ is well-defined.
   That is, if E and E' are belief bases such that $(\mathcal{B} \cup E)^{\perp} = (\mathcal{B} \cup E')^{\perp}$, we will show that $\Sigma((\mathcal{B} \cup E)^{\perp}) = \Sigma((\mathcal{B} \cup E')^{\perp})$. Since E and E' have the same correction kernels, it follows from reversion that $(\mathcal{B} \cup E) \setminus \mathcal{B} \odot E = (\mathcal{B} \cup E') \setminus \mathcal{B} \odot E'$. Therefore, $\Sigma((\mathcal{B} \cup E)^{\perp}) = \{\alpha : \alpha \in (\mathcal{B} \cup E) \setminus \mathcal{B} \odot E\} = \{\alpha : \alpha \in (\mathcal{B} \cup E') \setminus \mathcal{B} \odot E'\} = \Sigma((\mathcal{B} \cup E')^{\perp})$. Hence, Σ is well-defined.

2. $\Sigma((\mathcal{B} \cup E)^{\perp}) \in (\mathcal{B} \cup E)^{\perp}$.
   Let $\alpha \in \Sigma((\mathcal{B} \cup E)^{\perp})$. Then, $\alpha \in (\mathcal{B} \cup E) \setminus \mathcal{B} \odot E$. Since $\alpha \in (\mathcal{B} \cup E)$ and $\alpha \notin \mathcal{B} \odot E$, there is a correction set $\mathcal{B}' \subseteq \mathcal{B} \cup E$ such that $\alpha \in \mathcal{B}'$. Therefore, $\alpha \in (\mathcal{B} \cup E)^{\perp}$.

3. $(\mathcal{B} \cup E) \setminus \Sigma((\mathcal{B} \cup E)^{\perp}) \models \varphi$. Follows from the strong acceptance postulate $\mathcal{B} \odot E \models \varphi$.

From the inclusion postulate and the definition of $\Sigma((\mathcal{B} \cup E)^{\perp})$, we can conclude that the operator is an explanation-guided belief revision operator. $\square$

8.4 Human Belief Revision: Empirical Findings

Do the dynamics of human belief revision align with the principle of minimalism or with explanatory understanding?

This is the main question we are investigating in this section. As described earlier, the explanatory understanding suggests that in resolving inconsistencies, people seek explanations rather than simple minimal edits to their beliefs. The explanations then entail the revision, which may not be minimal. This is in contrast to the principle of minimalism, the most common principle in belief revision theory to date, which presupposes that an agent's primary goal when resolving inconsistencies is to make minimal changes in their beliefs.

To carry out our investigation, we conducted two human-subject study experiments using three types of inconsistent problems commonly referenced in cognitive science literature [71, 181, 26]. These problems were selected for their relevance in testing the depth of belief revision in response to inconsistencies.

The first type (**Type I**) consists of a conditional generalization statement S_1, a (non-conditional) ground categorical statement S_2, and a fact F that is inconsistent with what the statements imply. For example:

- S_1: *If people are worried, then they find it difficult to concentrate.*
- S_2: *Alice was worried.*
- F: *In fact, Alice did not find it difficult to concentrate.*

The second type (**Type II**) of inconsistency consists of two conditional generalization statements S_1 and S_2, a ground categorical statement S_3, and a fact F that is inconsistent with the consequences of one of the conditional statements and the categorical statement. For example:

- S_1: *If people are worried, then they find it difficult to concentrate.*
- S_2: *If people are worried, then they have insomnia.*
- S_3: *Alice was worried.*
- F: *In fact, Alice did not find it difficult to concentrate.*

Finally, the third type (**Type III**) of inconsistency consists of two conditional generalizations S_1 and S_2, a categorical statement S_3, and a fact F that is inconsistent with the consequences of both conditional statements and the categorical statement. For example:

- S_1: *If people are worried, then they find it difficult to concentrate.*
- S_2: *If people are worried, then they have insomnia.*
- S_3: *Alice was worried.*
- F: *In fact, Alice did not find it difficult to concentrate and did not have insomnia.*

In all three types, minimalism posits that a minimal resolution will be an explanation that rejects only the categorical statement (i.e., Alice was *not* worried.). However, almost all conditional generalizations about events are susceptible to what psychologists refer to as *disabling conditions* – conditions describing how the conditional fails [62, 71, 181]. For instance, "*Is it really the case that people find it difficult to concentrate when they are worried?*" One can easily think of a disabling condition for this conditional, for example, "*people with effective coping strategies may still be able to concentrate despite being worried*".

Because of people's propensity to envisage disabling conditions, their explanations are more likely to invoke such conditions than to imply that a categorical statement is wrong. But these explanations do not invoke a minimal change, because, logically speaking, they also remove the support for other consequences apart from the one giving rise to the inconsistency. For example, rejecting S_1 implies rejecting all of its groundings, which means you cannot infer that people find it difficult to concentrate if they are worried, for any instantiation of this rule (see also Example 31) . Surely, this is not a minimal change.

The upcoming experiments aim to elucidate whether people's revisions follow minimalism or are more aligned with the explanatory understanding, providing some insights into human belief revision processes.

8.4.1 Experiment 1

Our first experiment looked at the three problem types described above and was aimed at providing some empirical data on what kinds of explanations do people seek in the face of inconsistencies. In other words, do people seek explanations that resolve categorical or conditional statements?

Participants and Design We recruited 62 participants from the online crowdsourcing platform Prolific [175] across diverse demographics, with the only filter being that they are fluent in English. The participants carried out three different problems of each of the three types (Type I, Type II, and Type III), for a total of nine problems. The statements were taken from common, everyday events including subjects such as economics, intuitive physics, and psychology. The conditional statements in all problems were selected to be highly plausible and interpretable, similar to those in the high-plausibility category used by Politzer *et al.* [181]. All problem sets as well as more details of the study can be found in Appendix D.

The participants' main task was to explain the inconsistencies presented to them, and we examined the revisions implied by their explanations. After providing their explanations for every problem, each participant was asked a question about how they approached explaining what was going on and if they followed any strategies when doing so. They also answered a Likert-type question about whether being provided an explanation will help them understand the inconsistency.

Results All participants came up easily with reasons to explain the inconsistencies they encountered. To analyze the results, we employed a coding scheme similar to that by Byrne *et al.* [26]. Explanations provided by the participants were categorized into two main types: (1) Those implying non-minimal revisions (e.g., revisions to conditional generalizations), and (2) those implying minimal revisions (e.g., revisions to categoricals). Explanations implying non-minimal revisions were either disabling conditions that would prevent the consequences of the generalization, or of the form "It is not the case that if X then Y", "X is not sufficient for Y", and other similar ones. Explanations that implied minimal revisions rejected the categorical statements and were of the form "not X", "perhaps not X", and so on. This coding scheme classified 89% of the responses. The remaining responses either affirmed or denied the new information, or were too vague to classify.

Table 8.1 illustrates the distribution of explanations implying either non-minimal or minimal revisions. The data reveal a compelling trend: an overwhelming majority of explanations across all questions leaned towards non-minimal revisions. A Wilcoxon test performed on the aggregated data yielded a p-value significantly smaller than 0.05 ($p \approx 2.96 \times 10^{-50}$), providing robust evidence that the observed proportions of non-minimal and minimal classifications are far from what would be expected by random chance. Specifically, non-minimal revisions were substantially more frequent than minimal revisions.

To probe the robustness of this finding, we conducted individual statistical tests for each question. Wilcoxon tests for each problem revealed p-values well below the 0.05 threshold, affirming the prevalence of non-minimal revisions over minimal. Moreover, effect size measurements (Cohen's d) were conducted to quantify the magnitude of these differences, where it was consistently high across all instances.

Collectively, these results offer empirical support for the prevalence of non-minimal revisions in participants' explanations. This inclination suggests that individuals engage deeply in

Problem Type	Total Valid Responses (Count)	Non-Minimal Explanation (Count and %)	Minimal Explanation (Count and %)	Wilcoxon test (p-value)	Effect Size (Cohen's d)
Type I	161	132 (81.99%)	29 (18.01%)	4.76×10^{-16}	1.28
Type II	161	140 (86.96%)	21 (13.04%)	6.69×10^{-21}	1.48
Type III	177	144 (81.36%)	33 (18.64%)	7.23×10^{-17}	1.25
Aggregate	499	416 (83.37%)	83 (16.63%)	2.96×10^{-50}	1.33

Table 8.1: Results from Experiment 1, with *Aggregate* representing combined data from all problem types.

resolving inconsistencies, often opting for more comprehensive explanatory frameworks that necessitate altering their existing beliefs to a greater extent than minimalism would predict. These results also provide insights into what kinds of explanations people tend to create.

8.4.2 Experiment 2

In this experiment we look at how people actually revise their beliefs when they are given an explanation for an inconsistency.

Participants and Design We recruited 60 participants from the Prolific platform with the same requirements as before. In this study, rather than having the participants generate their own explanations, they were presented with some of the most plausible explanations (that are disabling conditions) created by participants in Experiment 1, and then asked to describe how they would revise their information in light of the explanation. To ensure that they do not discard the explanations, we added some validity to the explanation by telling the participants that the explanation comes from a trustworthy source. Unlike in Experiment 1, however, the participants were only shown problems of Type II and III.

Results We employed a specific coding scheme to analyze how participants chose to revise their beliefs. In accordance with this scheme, participants indicated whether they would *keep*, *discard*, or *alter* the beliefs.[55]. When choosing to alter a belief, participants were asked to provide details about how they would go about it. Like before, a minimal revision is one that discards or alters the categorical statement, and a non-minimal revision one that discards or alters either a generalization, or a combination of more than two statements. This coding

[55]We adopted a measure of belief change similar to those used in previous studies [71, 105, 233, 139], which typically count the number of beliefs that change their values.

Problem Type	Total Valid Responses (Count)	Non-Minimal Revision (Count and %)	# of Changes (Avg.)	Minimal Revision (Count and %)	Wilcoxon Test (p-value)	Effect Size (Cohen's d)
Type II	159	153 (96.23%)	1.71	6 (3.77%)	2.09×10^{-31}	3.56
Type III	154	136 (88.31%)	2.06	18 (11.69%)	1.93×10^{-21}	1.89
Aggregate	313	289 (92.33%)	1.88	24 (7.67%)	1.01×10^{-50}	2.43

Table 8.2: Results from Experiment 2, with *Aggregate* representing combined data from all problem types.

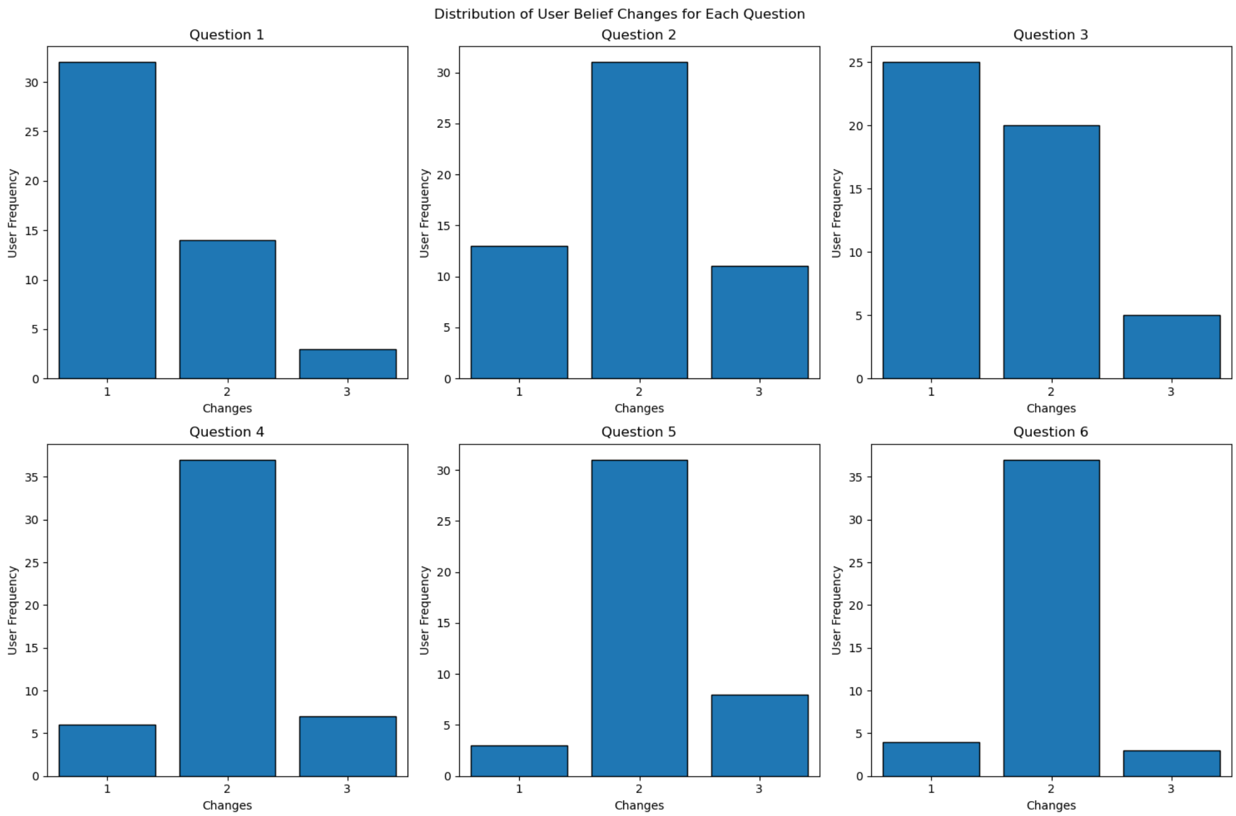


Figure 8.1: Distribution of Belief Changes per Question in Experiment 2.

scheme classified 87% of the responses, while the remaining responses were either yielding inconsistent revisions (e.g., not revising anything) or too vague to classify.

Table 8.2 provides an overview of the results. The data reveal a clear trend: A significant majority of revisions were non-minimal across both problem types. In Type II problems, 96.23% of the responses were non-minimal with an average of 1.65 changes to beliefs, compared to 3.77% that were minimal. A Wilcoxon test produced a p-value of 2.09×10^{-31}, and the effect size d was 3.56. In Type III problems, 88.31% were non-minimal and 11.69% were minimal. The average number of changes were 1.82. The Wilcoxon p-value was 1.93×10^{-21}, and the effect size d was 2.06. Moreover, aggregated data showed 92.33% non-minimal revisions, with an average of 1.88 belief changes, and 7.67% minimal revisions. The Wilcoxon p-value was 1.01×10^{-50} and the effect size 1.73.

These findings not only strongly corroborate those of Experiment 1, but further solidify the evidence that people predominantly opt for non-minimal revisions when presented with explanations. Interestingly, even if we relax the assumption of what constitutes a minimal revision in our coding scheme, i.e., by supposing that it is discarding or altering one statement (conditional or generalization), these results clearly indicate that the average number of belief changes in people tend to be more than one. This can be seen in Figure 8.1, where we plot the average number of belief changes per questions.

In summary, the results from both Experiment 1 and Experiment 2 offer compelling empirical data in the context of human belief revision processes. Far from making minimal changes to their existing beliefs, participants predominantly favored explanations that led to more comprehensive revisions. This demonstrates a natural inclination towards understanding the underlying factors that give rise to inconsistencies rather than merely resolving them in a superficial, yet minimal manner. The findings not only validate the explanatory understanding but also suggest a potential need for reevaluation of belief revision theories aimed at modeling actual humans. Our proposed human-aware belief revision framework is step towards this direction.

8.5 Related Work

The study of belief revision has been a focal point of research in both philosophy and AI, evolving over several decades. Among the most influential frameworks in the this domain is the AGM framework [2, 87, 85]. While the AGM framework has been highly influential [75], it unfortunately does not allow for an account of explanation. Crucially, the hallmark of AGM is the principle of minimalism, a point that we critically reconsidered in this work. When it comes to human belief revision, the explanatory understanding together with our empirical results indicate that humans tend to perform non-minimal revisions to their beliefs that are guided by explanations.

Our work is very closely related with the belief revision operator proposed by Falappa *et al.* [72], where they emphasized the role of explanations in belief revision, particularly when the incoming information is inconsistent with the existing beliefs. However, there are some subtle differences: their revision operator adheres to minimalism, and makes it possible for the explanation to be rejected, consequently rejecting the explanandum.

We are not the first to question adherence to minimal changes in belief revision. Notably, Rott [197] has critically examined minimalism in the context of AGM theory. He argued that, while intuitively appealing, may not adequately capture the complexities of real-world belief revision processes. Kern *et al.* [138] has also addressed the inadequacy of the minimalist approach guided by the AGM postulates in preserving conditional beliefs (if-then statements) during revision, and presented a thorough axiomatization of conditional preservation in belief revision. Nonetheless, in our framework we do not make such restrictions, and importantly, the empirical results we obtained refute the preservation of conditionals, at least in human belief revision.

Cognitive scientists and psychologists have also critiqued the principle of minimal change in belief revision, showing that human reasoners use a different strategy when revising their beliefs with new, conflicting information [71, 181, 70, 139, 123], namely the *explanatory understanding*. As we expressed throughout this chapter, this approach suggests that people create explanations to resolve inconsistencies, leading them to make greater than minimal changes to the information that they have.

8.6 Concluding Remarks

The field of belief revision theory has experienced remarkable progress, primarily influenced by the foundational work of Alchourrón, Makinson, and Gärdenfors. Their studies on revisions in legal codes [3], the introduction of rationality postulates for change operators [84], and the development of the *AGM model* [2] have set the stage for subsequent advancements in the area. It is well known that one of the basic conceptual principles underlying the AGM model, as well as most belief revision frameworks, is the principle of minimalism.

Contrary to the minimalist approach that has dominated belief revision theory, our empirical findings point towards a different paradigm–they suggest that in certain situations, individuals might opt for broader revisions to their belief systems, driven by the desire for a more comprehensive understanding and explanation of the information they encounter. In particular, our user studies revealed a notable propensity among participants to favor non-minimal, explanation-driven revisions when faced with inconsistencies. This preference persisted across different scenarios, suggesting that such an inclination might be a fundamental aspect of human reasoning. This finding aligns with the explanatory understanding, which proposes that humans prioritize generating coherent and plausible explanations over merely maintaining consistency with minimal changes.

The key contributions and findings of this chapter are:

1. We introduce a framework for human-aware belief revision inspired by human cognition, focusing on explanatory understanding over minimal changes, which better reflects human belief revision processes. Particularly, we proposed the *explanation-guided* revision operator that, given an explanation for an explanandum, revises the human model in a (possibly) non-minimal way while preserving the explanandum.

2. We presented findings from two human-subject studies that serve as robust evidence for the claim that people tend to make non-minimal revisions to their beliefs in light of inconsistency.

We believe that these results can have implications that extend to the domains of explainable AI [99] and human-aware AI [127]. As efforts in these fields converge towards fostering transparent, explainable, and synergistic interactions between humans and AI systems, aligning the AI systems' decision-making processes with human cognitive models can not only enhance explainability but also elevate the efficacy of human-AI collaborations.

Our work also provides new insights into the Model Reconciliation Problem (MRP) [37, 207, 202, 225, 231]. While previous approaches, including our earlier work, emphasized minimal changes to human mental models, this research suggests that human-aware AI systems should accommodate more substantial revisions to better match human cognition. By integrating explanation-guided belief revision into our logical explainability layer, we enable AI systems to generate explanations that better reflect the complexity of human understanding. As AI systems progressively permeate society's decision-making structures, the imperative to align AI processes with human cognition becomes ever more pressing.

In conclusion, this work is an attempt to contribute a new perspective to belief revision theory by introducing and empirically grounding a human-aware belief revision framework. Our empirical findings provide support for this framework, highlighting the importance of explanations in human belief revision processes, and the divergence from minimalism. Ultimately, belief revision is not an isolated process, but an integral component of people's broader quest for explanatory understanding.

In the next chapter, we will switch gears and examine how our logical explainability layer can address real-world scheduling problems, demonstrating its practical applicability while handling additional challenges like privacy concerns.

Chapter 9

Towards Explainable Agent Scheduling Problems

"Premature optimization is the root of all evil."

— Donald Knuth

9.1 Introduction & Contribution

Having developed our logical explainability layer through theoretical foundations, efficient algorithms, personalization, dialectical interaction, and explanation-guided belief revision, we now demonstrate its practical application to a challenging real-world domain: *Agent Scheduling Problems* (ASPs).

ASPs involve allocating a finite set of resources to multiple agents over a specific time frame. These problems are pervasive in real-world scheduling systems, ranging from personnel shift assignments [221] to machine job allocation [236], and even scheduling awake and asleep periods for Mars rovers [40]. Apart from generating a schedule that allocates resources to agents, it is crucial to ensure that both the schedule and the underlying decision-making process are explainable. An agent may require an explanation for why certain scheduling decisions were not satisfied or why a schedule could not be generated at all. In such cases, understanding the reasons behind these issues is not only enlightening but also necessary for rectifying the problem. Additionally, privacy plays a significant role due to the sensitive nature of personal information that may be included in ASPs, such as agents' constraints and preferences. Preserving privacy helps protect individual agents from potential discrimination or unauthorized access to their information, fostering trust and willingness to participate in the scheduling process. Therefore, incorporating explanation generation modalities with privacy-preserving considerations into ASP systems is highly desirable.

To address this need, we present a logic-based framework aimed at making ASPs explainable. Our framework accommodates two types of queries: (1) *Reason-seeking queries*, which clarify why a scheduling decision was (or was not) derived, similar to the explanations generated in our L-MRP framework; and (2) *Modification-seeking queries*, which offer guidance on rendering infeasible scheduling decisions feasible, aligning with our work on human-aware belief revision where explanations guide future actions. Recognizing the importance of privacy in multi-agent scheduling, we introduce the concept of agent access rights to distinguish between *public* and *private information.* We define a privacy-loss function to quantify the amount of private information disclosed in explanations. This leads to the notion of *privacy-aware explanations*, a novel concept that ensures explanations maintain the confidentiality of sensitive information while still offering valuable insights. To operationalize this framework, we present the *Query Understanding and Efficient Response with Intelligible Explanations of Schedules* (QUERIES) algorithm for computing privacy-aware explanations. This algorithm builds upon our previous work on efficient explanation generation, adapting it to the specific challenges of ASPs and privacy considerations.

Our main contributions are as follows:

- We introduce a general logic-based explanation generation framework for ASPs that addresses both reason-seeking queries and modification-seeking queries.
- We propose a privacy-loss function to quantify the amount of private information included in an explanation and define the concept of privacy-aware explanations.
- We present the QUERIES algorithm for computing explanations. Empirical evaluations demonstrate the effectiveness and versatility of our approach.

This work demonstrates how our logical explainability layer can be adapted to address domain-specific challenges while preserving its core strengths. By incorporating privacy considerations and supporting different query types, we show how logic can serve as an explainability layer even in complex multi-agent settings where privacy is paramount.

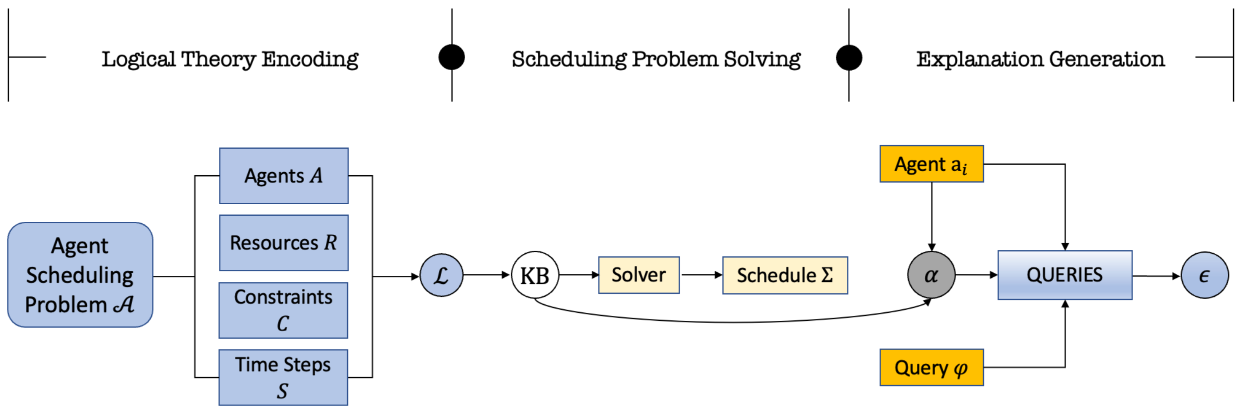


Figure 9.1: Overview of Our Explainable Logic-based Agent Scheduling Problem Pipeline.

9.2 Background

We now provide some background on the satisfiability (SAT) problem, a general agent scheduling problem (ASP) definition, and our logic-based representation of that problem.

9.2.1 Satisfiability

We assume familiarity with propositional logic. A knowledge base KB is a set of constraints, where each constraint is built up recursively from *literals* (i.e., variables or its negations) using the usual logical connectives.

Satisfiability (SAT) [42] is the prototypical NP-complete problem of finding an assignment of truth values to variables in order to make a knowledge base KB true. If there exists a truth value assignment μ that makes KB true, then we say that μ is a *model* of KB and KB is *satisfiable*, otherwise KB is *unsatisfiable*, denoted by KB $\models \perp$. A KB *entails* a constraint φ, denoted KB $\models \varphi$, iff KB $\cup \{\neg\varphi\} \models \perp$.

Partial weighted MaxSAT [154] is an extension of SAT in which constraints are partitioned into *hard* and *soft* constraints, where each soft constraints is given a weight. Hard constraints must always be satisfied in a solution, whereas soft constraints may not. The goal of MaxSAT is to find an assignment that satisfies the hard clauses and maximizes the sum of weights of the satisfied soft clauses.

9.2.2 Agent Scheduling Problem

In general, the goal of an *agent scheduling problem* (ASP) is to distribute a set of resources to a set of agents over a scheduling horizon. Formally, it can be defined as a tuple $\mathcal{A} = \langle A, R, S, C \rangle$, where $A = \{a_i\}_{i=1}^n$ is a set of agents, $R = \{r_j\}_{j=1}^m$ is a set of resources, $S = \{s_t\}_{t=1}^h$ is a set of time steps, and C is a set of constraints that consists of *domain constraints*, which are intrinsic and describe the problem's dynamics, as well as *agent constraints*, which are extrinsic and describe the agents' personal constraints.

A solution to an ASP $\mathcal{A}$ is a *schedule* Σ, that is an $|A| \times |R| \times |S|$ matrix, where each cell $\Sigma[i, j, t] = 1$ if agent a_i is assigned resource r_j at time step s_t and $\Sigma[i, j, t] = 0$ otherwise. A schedule is *feasible* if all the domain constraints, which are treated as hard constraints, are satisfied. A schedule is *optimal* if it is feasible and all the agent constraints, which are treated as soft constraints, are maximized.

9.2.3 Logic-based Agent Scheduling Problems

We will model an ASP $\mathcal{A}$ as a logic-based problem, that is, we encode $\mathcal{A}$ into a set of logical constraints for which satisfiability can be decided. By using an appropriate logical language, the problem's dynamics are encoded into a knowledge base KB that expresses all the scheduling constraints that a desired schedule should satisfy. Specifically, the knowledge base KB consists of domain constraints C_D and agent constraints C_A, where C_D are treated as hard constraints and C_A as weighted soft constraints. As such, the scheduling problem turns into a MaxSAT problem, where the quality of a feasible schedule depends on the degree to which the soft clauses are satisfied. The objective function of a candidate schedule is then defined as the sum of weights of satisfied soft constraints, and an optimal schedule is the solution with the highest possible objective value. A plethora of scheduling problems has been modeled using logic-based approaches [44, 179, 147, 145, 4, 106, 18, 58].

For ease of presentation, in this work we will use propositional logic to encode ASPs. We formally define a *logic-based ASP* (L-ASP) as follows:

Definition 56 (L-ASP). *An* L-ASP *is a tuple* $\mathcal{L} = \langle A, R, S, \mathrm{KB} \rangle$, *where* $\mathrm{KB} = C_D \cup C_A$ *and:*

- *C_D is the set of domain-specific (hard) constraints. These constraints are intrinsic to the problem and must be satisfied by a solution.*
- *$C_A = \bigcup_{i=1}^{n} C_i$ is the set of agent (weighted soft) constraints. Each $C_i = \{(w_k, c_k^i)\}_{k=1}^{l}$, where each c_k^i is a constraint associated with agent a_i and w_k is its corresponding weight.*

A schedule can be derived by using off-the-shelf SAT solvers [14] to search for a model μ of KB that satisfies all of the constraints in C_D and possibly some of the constraints in C_A. If a model μ exists, then a *feasible* schedule Σ_μ is derived by extracting from μ the truth values of the variables corresponding to agents, resources, and time steps. Otherwise, the scheduling problem is *infeasible*, i.e., no feasible schedule exists. Finally, a schedule Σ_μ is deemed *optimal* if a model μ exists and maximizes the cumulative sum of weights of satisfied soft constraints in C_A.

Note that the knowledge base KB $= C_D \cup C_A$ may be unsatisfiable due to inconsistencies in the domain constraints and/or agent constraints. However, if a schedule Σ_μ exists, then that means that Σ_μ logically follows from a satisfiable subset $KB_\mu \subseteq$ KB. In the next section, we use KB to denote the knowledge base from which explanations are derived. Depending on the context, KB could refer to either a satisfiable subset of the original knowledge base (i.e., KB_μ) or the overall unsatisfiable knowledge base.

9.3 Explainable Agent Scheduling Problems

To better understand the challenges faced by agent scheduling problems and the importance of generating effective explanations, let us first engage in a thought experiment inspired by a simplified version of the *employee shift assignment* problem [221].

9.3.1 Motivating Thought Experiment

Consider a scenario based on the *employee shift assignment* problem [221]. In this scenario, an automated scheduling agent named Alice is responsible for assigning shifts to employees at a company. Specifically, there are three shift types – *morning*, *afternoon*, and *evening* –

Employee Name	*Monday*	*Tuesday*	*Wednesday*
Thanos	*morning*	*evening*	–
Irene	*afternoon*	–	*evening*
Vicky	–	*afternoon*	*afternoon*
Rose	–	*morning*	*morning*

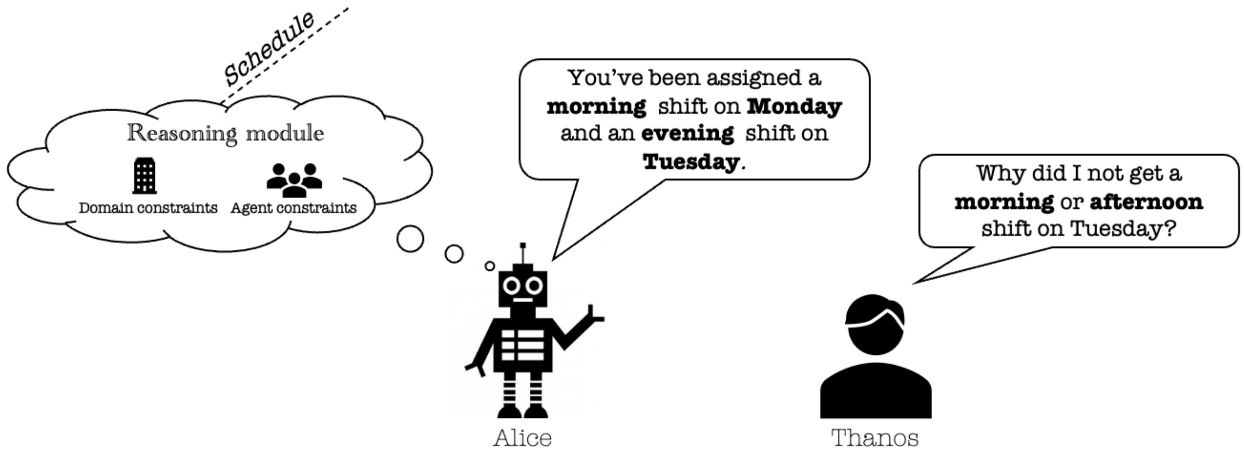


Figure 9.2: Instance of the thought experiment with Alice and Thanos.

and four employees – *Thanos*, *Irene*, *Vicky*, and *Rose* – who need to be assigned shifts over three days from *Monday* to *Wednesday*.

The scheduling problem consists of the following *domain constraints*:

C_1: *All employees must be assigned a total of two shifts.*
C_2: *Employees cannot be assigned multiple shifts per day.*
C_3: *No two employees can be assigned the same shift the same day.*
C_4: *Employees cannot be assigned a morning shift right after an evening shift.*

Moreover, each employee has *personal constraints*:

C_T: *Thanos wants only morning or afternoon shifts.*
C_I: *Irene does not want evening shifts.*
C_V: *Vicky wants the afternoon shift on Tue. and Wed.*
C_R: *Rose wants the morning shift on Tue. and Wed.*

Here, Alice's objective is to find a schedule that satisfies all domain constraints and, as much as possible, accommodates the employee constraints according to their weights, which in this example are based on the employees' seniority levels.

Let us assume that Alice finds a feasible schedule, but it does not meet Thanos' constraint of being assigned morning or afternoon shifts. Thanos, in turn, may inquire about the reason for this assignment. To generate an effective explanation, Alice needs a framework that can generate explanations that are informative and tailored to the specific needs of the explainee, that is, Alice must first recognize the nature of the explainee's query.

In our thought experiment, Thanos' query is a *reason-seeking query*, as he wants to know "why" his constraint was unsatisfied in the schedule. In response, Alice should provide a (reason-seeking) explanation that identifies the reasons behind her (scheduling) decision. For example, Alice might explain that due to the constraints of the problem and the higher priority given to the preferences of Rose and Vicky, it was not possible to assign Thanos morning shifts on Tuesday or Wednesday without affecting the overall quality of the allocation.

However, providing a reason-seeking explanation alone may not be sufficient in all scenarios. Suppose Alice could not create a feasible schedule at all due to conflicting constraints. In this case, a higher-level employee, such as a manager, may want to understand "how" to adjust the scheduling problem to derive a feasible schedule. This type of query is a *modification-seeking query*, which requires an explanation that helps the manager identify issues preventing a feasible schedule and suggest potential modifications.

In addition to addressing these two types of queries, Alice's explanations should respect the *privacy* of the other employees. To achieve this, Alice could only reveal information according to the employees' *access rights*. In doing so, Alice distinguishes between *public information* (information that can be revealed to employees with access rights) and *private information* (information that cannot be revealed to employees without access rights).

This thought experiment demonstrates some of the challenges of generating explanations in the context of agent scheduling problems. Indeed, in the next section we present an explanation generation framework that can handle the complexity of the problem, account for the explainee's needs and access rights, and produce informative explanations.

9.3.2 Explanation Generation Framework

We now present our explanation generation framework for agent scheduling problems. We particularly address the following problem:

> Given a logic-based L-ASP $\mathcal{L} = \langle A, R, S, \text{KB} \rangle$ and a *query* φ with respect to KB, the goal is to find an *explanation* for φ that can be inferred from KB.

As discussed in Section 9.3.1, we are interested in a framework that can generate explanations for agent scheduling problems that are not only informative but also tailored to the specific needs of the explainee. Such a framework should in principle:

- Address two general types of queries: *reason-seeking queries*, which aim to uncover *why* certain scheduling decisions were (or not) made, and *modification-seeking queries*, which focus on identifying potential modifications to the problem.
- Generate informative and concise explanations for the two query types.
- Preserve the privacy of other agents by only revealing information with respect to *access-rights*.

A general pipeline is shown in Figure 9.1. We now describe how to generate explanations for the two query types.

Explaining Reason-Seeking Queries

A reason-seeking query, denoted by φ_r, aims to uncover *why* certain scheduling decisions were made. Recall from Section 9.3.1 that Thanos wants to know why Alice did not assign him only morning shifts. Alternatively, a higher-level employee (e.g., a manager) may want to understand why a feasible schedule cannot be generated.

To explain reason-seeking queries, we assume that $\text{KB} \models \varphi_r$. There are two possible scenarios to consider:

- **Agent Constraints in a Schedule:** If the query φ_r captures an unsatisfied (or satisfied) agent constraint in a schedule Σ_μ, then $\varphi_r \in \neg C_A$ (or $\varphi_r \in C_A$).[56] In this scenario, an explanation should identify the reasons why the constraint holds true with respect to the schedule. Note that the knowledge base KB here is satisfiable (see Section 9.2.3).
- **Infeasible Scheduling Problems:** If the query φ_r is aimed at capturing why a problem is infeasible, i.e., why a feasible schedule cannot be generated, then generally $\varphi_r = \bot$. In this case, the explanation should identify the inconsistencies within the scheduling constraints that lead to infeasible schedules. Note that the knowledge base KB here is unsatisfiable, i.e., there is no model of KB from which a feasible schedule can be extracted.

Formally now, an explanation for a reason-seeking query is defined as follows:

Definition 57 (Reason-seeking Explanation)**.** *Given a knowledge base* KB *that encodes an L-ASP $\mathcal{L}$ and a reason-seeking query φ_r, we consider an explanation $\epsilon_r \subseteq$* KB *to be a* reason-seeking explanation *for φ_r if:*

- *ϵ_r is sufficient: $\epsilon_r \models \varphi_r$, meaning that the explanation ϵ_r entails the query φ_r.*
- *ϵ_r is minimal: For all proper subsets $\epsilon'_r \subset \epsilon_r$, $\epsilon'_r \not\models \varphi_r$, indicating that no smaller subset of ϵ_r are sufficient.*

These conditions ensure that the reason-seeking explanation is both sufficient and minimal in addressing the query.

Explaining Modification-Seeking Queries

Modification-seeking queries, denoted by φ_m, focus on identifying potential modifications to a scheduling problem to address specific issues. For example, Thanos may want to know how to incorporate his unsatisfied constraint in Alice's schedule, or a manager may seek ways to adjust the scheduling problem to generate a feasible schedule.

To explain modification-seeking queries, we assume that KB $\not\models \varphi_m$. Specifically, to explain these query types, we seek to identify a set of constraints from the knowledge base KB that, when retracted, KB $\models \varphi_m$. Like before, there are two possible scenarios to consider:

[56]Note that $\neg C_A$ denotes the logical negation of all the constraints in C_A.

- **Unsatisfied Agent Constraints in a Schedule:** If the query φ_m concerns accommodating an unsatisfied agent constraint in a schedule Σ_μ, then $\varphi_m \in C_A$.
- **Infeasible Scheduling Problems:** If the query φ_m is aimed at explaining how a problem can be modified such that a feasible schedule can be found, then $\varphi_m = \top$.

We now define an explanation for a modification-seeking query as follows:

Definition 58 (Modification-seeking Explanation)**.** *Given a knowledge base* KB *that encodes an L-ASP $\mathcal{L}$ and a modification-seeking query φ_m, we consider an explanation $\epsilon_m \subseteq$* KB *to be a* modification-seeking explanation *for φ_m if:*

- *ϵ_m enables the entailment of φ_m:* $\text{KB} \setminus \epsilon_m \models \varphi_m$*, meaning that the query φ_m is entailed when the constraints in ϵ_m are removed from the knowledge base.*
- *ϵ_m is minimal: For all proper subsets $\epsilon'_m \subset \epsilon_m$,* $\text{KB} \setminus \epsilon'_m \not\models \varphi_m$*, indicating that no smaller subset of ϵ_m can satisfy the query when removed from the knowledge base.*

These conditions ensure that the modification-seeking explanation is both effective and minimal in addressing the query.

Privacy-Aware Explanations

It is reasonable to assume that individuals might prefer explanations for scheduling decisions that only encompass public information, as they could perceive these as more satisfying and equitable compared to explanations that incorporate private information as well. To explore this possibility and incorporate potential privacy preferences into our framework, we propose that agents have *access rights* on the different pieces of information about the scheduling problem. Specifically, we assume an *access-rights* function:

$$\alpha : A \times \text{KB} \to \{0, 1\} \tag{9.1}$$

that determines whether an agent $a_i \in A$ has *access rights* to a constraint $c \in \text{KB}$, returning 1 if a_i has access to c and 0 otherwise.

While we have motivated access rights through the lens of privacy, note that the function can also encode access rights through other means as well (e.g., security clearances and other administrative compartmentalization protocols).

Given an agent a_i and the function α, we define the *privacy loss* ρ_i of an explanation ϵ with regard to the agent as the count of constraints inaccessible to it:

$$\rho_i(\epsilon) = |\epsilon| - \sum_{c \in \epsilon} \alpha(a_i, c) \tag{9.2}$$

Lastly, we define an explanation ϵ_i as being *privacy-aware* in relation to agent a_i and query φ if it incurs the least privacy loss among all possible explanations E for the query φ:

$$\epsilon_i = \underset{\epsilon \in E}{\text{argmin}} \ \rho_i(\epsilon) \tag{9.3}$$

Illustrating Example

Consider the employee shift assignment problem presented in Section 9.3.1. To represent the problem using (propositional) logic, we employ Boolean decision variables $x_{i,j,t}$ for all $a_i \in A$, $r_j \in R$, and $s_t \in S$, where each variable is set to true if and only if agent a_i is assigned shift r_j on day s_t. Otherwise, it is set to false. These variables comprise the domain constraints C_D and agent constraints C_A which make up the knowledge base KB. Note that we assume the following weights for employee constraints C_A: $w(C_R) = w(C_V) > w(C_T) > w(C_I)$. [57]

Recall from Section 9.3.1 that Alice has generated a schedule (see Figure 9.2) that does not satisfy Thanos' constraint, prompting him to ask Alice a reason-seeking query. In our logic-based framework, this translates to the query $\varphi_r = \{\neg x_{1,1,2} \vee \neg x_{1,2,2}\}$. There are two reason-seeking explanations for this query:

[57] For more details on the encoding, please refer to the supplement available at https://github.com/YODA-Lab/QUERIES.

- $\epsilon_{r1} = \{x_{4,1,2}, \neg x_{4,1,2} \vee \neg x_{1,1,2}\}$, stating that only one employee can be assigned a morning shift on the same day (domain constraint) and that Rose's preference was given a higher priority that day.
- $\epsilon_{r2} = \{x_{3,2,2}, \neg x_{3,2,2} \vee \neg x_{1,2,2}\}$, stating that only one employee can be assigned an afternoon shift on the same day (domain constraint) and that Vicky's preference was given a higher priority that day.

Now, assume that the access-rights function α is defined such that Thanos has access-rights to the domain constraints and Rose's constraints, but not to the constraints of other agents. In this case, the privacy loss ρ_1 of both explanations would be calculated as follows:

- $\rho_1(\epsilon_{r1}) = |\epsilon_{r1}| - \sum_{c \in \epsilon_{r1}} \alpha(1, c) = 2 - 2 = 0$, since Thanos has access to Rose's information.
- $\rho_1(\epsilon_{r2}) = |\epsilon_{r2}| - \sum_{c \in \epsilon_{r2}} \alpha(1, c) = 2 - 1 = 1$, since Thanos does not have access to Vicky's information.

As $\rho_1(\epsilon_{r1}) < \rho_1(\epsilon_{r2})$, the privacy-aware explanation in this case would be ϵ_{r1}.

9.4 QUERIES: Computing Explanations

We now present the *Question Understanding and Efficient Response with Intelligible Explanations of Schedules* (QUERIES) algorithm, which generates privacy-aware explanations ϵ_i^* for reason-seeking and modification-seeking queries φ of an agent a_i. The core of QUERIES is based on *reasoning via inconsistency*. In particular, it leverages a set of methods that are directly applicable to logic-based explanation generation problems, namely, *minimal unsatisfiable sets* (MUS) and *minimal correction sets* (MCS) [186, 165], both of which emerge when a set of clauses is unsatisfiable. Particularly, an MUS can be interpreted as explaining why a set of clauses is unsatisfiable by identifying a minimal set of conflicting clauses that cause the unsatisfiability. An MUS can then be used to find a reason-seeking explanation:

Proposition 8. *Given a knowledge base* KB *and a reason-seeking query* φ_r, $\epsilon_r = M \setminus \{\neg\varphi_r\}$ *is a reason-seeking explanation for* φ_r *if* M *is an MUS of* $\mathrm{KB} \cup \{\neg\varphi_r\}$.

PROOF (SKETCH). The existence of a reason-seeking query φ_r implies that $\text{KB} \models \varphi_r$, which in turn implies that $\text{KB} \cup \{\neg\varphi_r\} \models \bot$ according to the definition of entailment. That is, the negation of φ_r is inconsistent with a set of constraints from KB and, as such, an MUS M of $\text{KB} \cup \{\neg\varphi_r\}$ exists. If $\neg\varphi_r \in M$, then $M \setminus \{\neg\varphi_r\}$ is satisfiable and $M \setminus \{\neg\varphi_r\} \models \varphi_r$. Therefore, $M \setminus \{\neg\varphi_r\}$ is a reason-seeking explanation for φ_r. $\square$

Similarly, an MCS explains how to restore consistency in an inconsistent KB by identifying a minimal set of clauses from KB such that when removed, KB becomes satisfiable. A modification-seeking explanation can be then be generated via an MCS:

Proposition 9. *Given a knowledge base* KB *and a modification-seeking query φ_m, C is a modification-seeking explanation for φ_m if C is an MCS of* $\text{KB} \cup \{\varphi_m\}$ *and $\varphi_m \notin C$.*

The proof of Proposition 2 follows from the fact that a modification-seeking explanation for φ_m is indeed an MCS of $\text{KB} \cup \{\varphi_m\}$.

Algorithm 9.1 presents the pseudocode of QUERIES, which generates explanations for an agent a_i. At a high level, it iterates over all constraints in KB and assigns large weights $k >> 1$ to constraints that are public to agent a_i with respect to access-rights function α. Then, the MUS (or MCS) solver prioritizes the constraints with the largest weights, which means that the output of the solver is a set of constraints with the largest cumulative sum of weights (i.e., privacy-aware explanation).

The completeness of QUERIES lies in the assumption we made for the two query types, which is that an explanation for both query types always exists. The correctness of QUERIES lies in the correctness of the MUS and MCS solvers and the assumption that k is sufficiently large such that explanations with the largest cumulative sum of weights are privacy-aware explanations.

9.5 Empirical Evaluations

We now empirically evaluate our approach both in simulated computational experiments as well as in a human user study.

Algorithm 9.1: QUERIES Algorithm

Input: KB, φ, a_i, α, k
Result: *privacy-aware explanation* ϵ for φ for a_i

forall $c \in$ KB **do**
  if $\alpha(a_i, c) = 1$ **then**
    assign weight k to c
if φ *is a reason-seeking query* **then**
  $\epsilon \leftarrow getMUS(\mathrm{KB}, \varphi)$
else if φ *is a modification-seeking query* **then**
  $\epsilon \leftarrow getMCS(\mathrm{KB}, \varphi)$
return ϵ

9.5.1 Computational Experiments

We now present a computational evaluation of QUERIES for the following four queries, two for each query type, where C_a is an agent's clause and Σ an infeasible schedule:[58]

- Reason-seeking query (agent): *Why is C_a unsatisfied?*
- Modification-seeking query (agent): *How to satisfy C_a?*
- Reason-seeking query (schedule): *Why is Σ infeasible?*
- Modification-seeking query (schedule): *How to make Σ feasible?*

We ran our experiments on a MacBook Pro machine comprising an M1 Max processor with 32GB of memory. The time limit was set to $500s$. Our implementation of QUERIES is written in Python and integrates calls to MUS and MCS oracles through the PySAT toolkit [115].[59]

To comprehensively evaluate our approach, we ran three sets of experiments: (1) To demonstrate the scalability of our approach, we evaluated it on our motivating employee shift assignment problem of varying size; (2) To demonstrate the impact of privacy or access rights, we evaluated our algorithm on the same scheduling problem, but agents have varying

[58] C_a was randomly selected from a pool of unsatisfied clauses of agent a and Σ was generated by randomly flipping 20% of the values of a feasible schedule.
[59] The code repository is available at https://github.com/YODA-Lab/QUERIES.

access rights; and (3) To demonstrate the generality of our approach, we evaluated it on an SMT-based encoding of the job-shop scheduling problem.

Experiment 1: Scalability: In this experiment, we vary the scale and complexity of the agent scheduling problem by varying the number of agents $|A|$, resources $|R|$, and time steps $|S|$ in the problem. Specifically, we created 14 random instances, where each instance has $|A| = 10 \cdot i$ agents, $|R| = 10 \cdot i$ resources, and $|S| = 10$ time steps, with i taking the values $1, 1.5, 2, \ldots, 7.5$. For the domain constraints, we extended the ones described in Section 9.3.1 to include more agents, shift types, and time steps, as well as included an additional constraint describing the maximum number of consecutive shifts an employee can undertake without a day off. For the agent constraints, we generated 5 types of constraints to reflect different kinds of preferences similar to those presented in Section 9.3.1, and randomly assigned them to the agents. We set the fraction $p = 0.5$ of agents that each agent has access rights to. If an agent a_i has access rights to agent a_j, then a_i is aware of all of agent a_j's constraints.

Figure 9.3 plot the runtimes of QUERIES as a function of the cardinalities of the knowledge base |KB| and the explanation $|\epsilon|$ found. Unsurprisingly, the runtimes increase as the cardinalities increase. The reason is that the search space grows with |KB|, also reflected in $|\epsilon|$. Also, modification-seeking queries took longer to solve than reason-seeking queries. The reason is that our off-the-shelf MCS solver, used for modification-seeking queries, is less efficient than our off-the-shelf MUS solver, used for reason-seeking queries.

Experiment 2: Access Rights: In this experiment, we use the same employee shift assignment problem, where we set the number of agents $|A| = 40$, resources $|R| = 40$, and time steps $|S| = 5$. We vary the fraction $p = \{0, 0.1, 0.2, \ldots, 1\}$ of other agents that each agent has access rights to.

Figure 9.4 plots, as a function of access rights fraction p, the runtimes of QUERIES, privacy losses $\rho_i(\epsilon)$ of explanations, and cardinality of explanations $|\epsilon|$. Similar to the previous experiment, the runtimes are larger for modification-seeking queries than reason-seeking queries. However, unlike the previous experiment, there is a significant difference in $|\epsilon|$ for the different queries in this experiment. As the modification-seeking queries required longer explanations, they took longer to solve than reason-seeking queries.

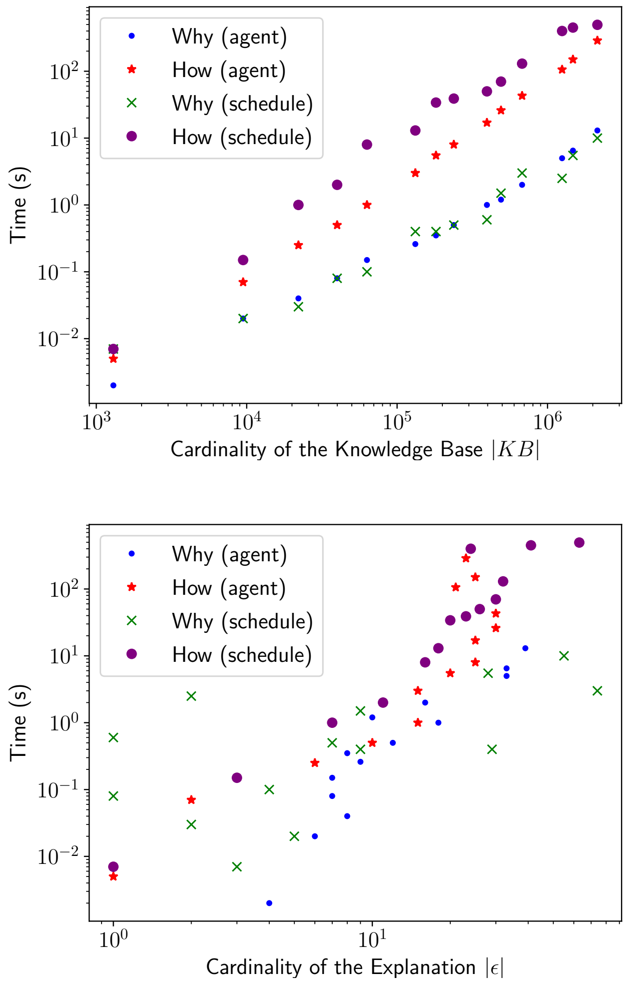


Figure 9.3: Results of Experiment 1 on the Scalability of QUERIES

Additionally, the runtimes stay relatively constant for all values of p, reflecting the fact that the runtimes for the MCS and MUS computations are independent of the weights of the clauses. Also, as expected, the privacy loss decreases as p increases since fewer clauses are private as p increases. Finally, as p increases, $|\epsilon|$ either decreases or remains constant, indicating that the solver can find shorter (i.e., better) explanations when the explanation space expands with larger values of p.

Experiment 3: SMT and Job-Shop Scheduling: Finally, to demonstrate that our explainable scheduling framework and algorithm can be generalized to other scheduling problems as well as other types of logic aside from propositional logic, we evaluate our approach

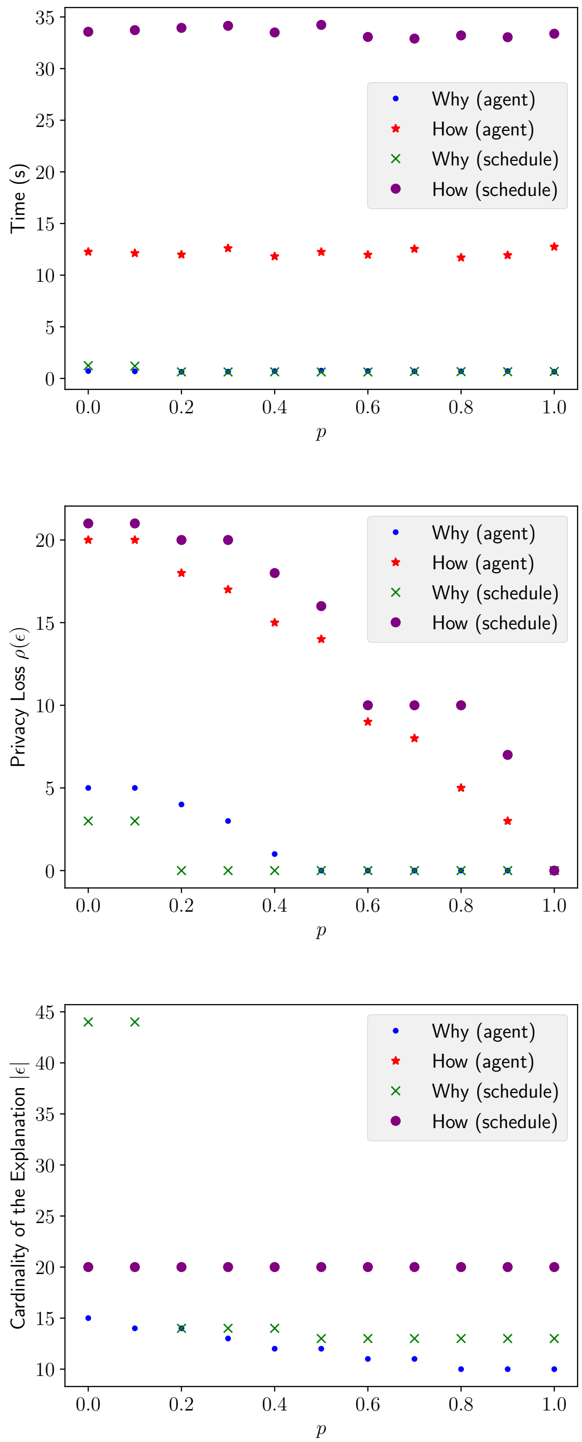
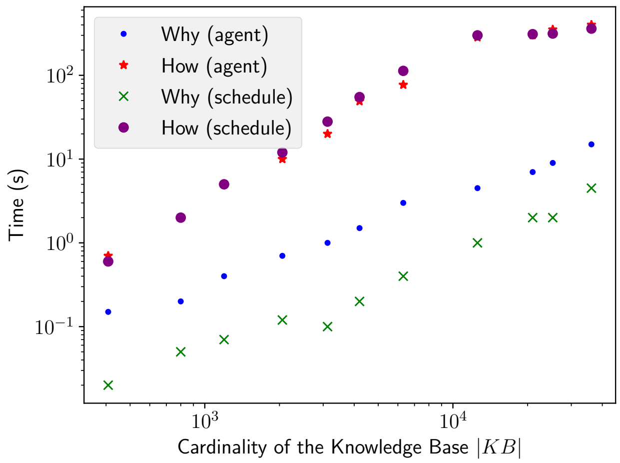


Figure 9.4: Results of Experiment 2 on the Impact of Privacy and Access Rights



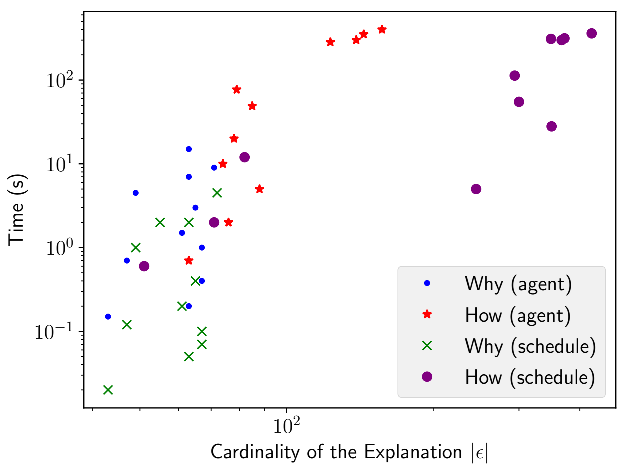


Figure 9.5: Results of Experiment 3 on SMT-based Encoding of Job-Shop Scheduling

on a *Satisfability Modulo Theory* (SMT) encoding of the *job-shop scheduling problem* [194]. SMT is a decision problem that extends Boolean logic and allows for richer representations of real-world problems with logical formulae that are based on a combination of background theories such as integers and reals [55].

The job-shop scheduling problem involves assigning a set of jobs, each with its own processing time, to machines in a way that ensures all jobs are completed. We encoded this problem in Python using the Z3 solver [54], and generated 11 instances by varying the number of jobs, processing times, and machines. For the MUS and MCS solvers, we used off-the-shelf implementations available within Z3. Similar to the previous experiment, we generated queries with an unsatisfied constraint and an infeasible schedule.

Figure 9.5 plots the runtimes of QUERIES as a function of the cardinalities |KB| and $|\epsilon|$. We observed trends similar to those in Experiment 1, attributable to the same reasons described earlier.

9.5.2 Human-Subject Experiments

We now present a user study aimed at examining the assumptions made in our framework. In particular, we hypothesize:

> *Within agent scheduling problems, individuals prefer explanations containing only public information (e.g., publicly acknowledged rules and constraints) over those including private information (e.g., other employees' names and personal constraints), as they perceive them as more satisfactory.*

To evaluate this hypothesis, we conducted a human user study involving 60 English-speaking participants recruited through the online platform Prolific [175]. The study is centered around the employee shift assignment problem introduced earlier, with participants engaging in a thought experiment by assuming the role of an employee in a hypothetical company.

We informed the participants that Alice, an automated scheduling agent, was responsible for creating a schedule under the previously described domain constraints, ensuring that this information was public and known to all users. Participants were asked to choose a personal constraint from four available options, making them aware of only their own personal constraint, while the remaining agent constraints were considered private information. The participants then received their shift assignments, and were notified that their personal constraint was not satisfied in Alice's schedule.

Their primary task was to select an explanation out of two options: a *generic explanation*, which contained another employee's name and private constraint as the reason for their unsatisfied constraint, and a *privacy-aware explanation*, which included only a public domain constraint. Participants then answered questions about their choice of explanation and their satisfaction levels.

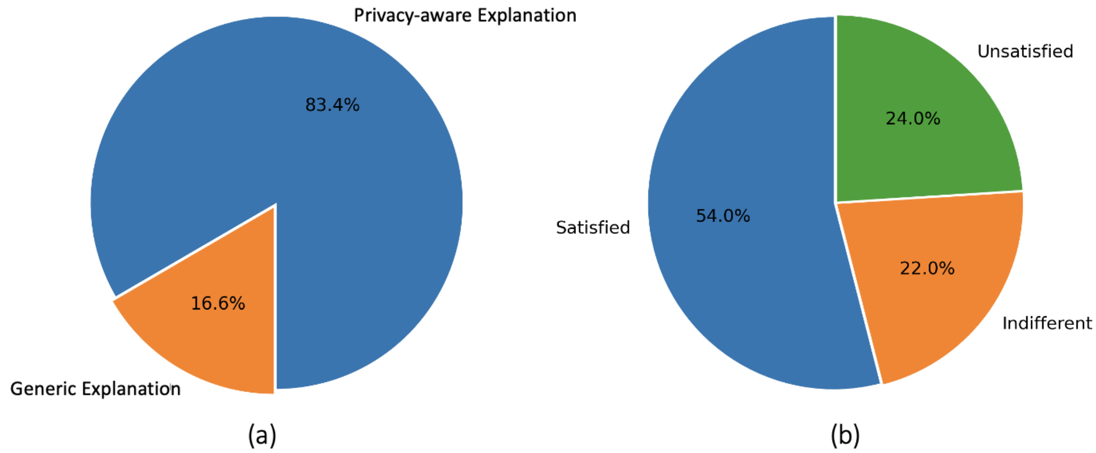


Figure 9.6: Human user study results from 60 users: (a) Percentage of users that selected generic and privacy-aware explanations; and (b) Percentage of users that were satisfied, indifferent, or unsatisfied with the privacy-aware explanation.

Figure 9.6 presents the main results of the study. The majority (83.4%) of participants preferred the privacy-aware explanation (Figure 9.6(a)). Among those who chose the privacy-aware explanation, 54% were satisfied, while the remaining participants were either indifferent (22%) or unsatisfied (24%), as shown in Figure 9.6(b). In the analysis of responses to the justification question, i.e., "why they selected the particular explanation", we observed a common trend: the privacy-aware explanation was considered more "informative" and "equitable" to all employees. Here, informative meant that it contained well-justified rules (i.e., constraints known to them), while "equitable" implied that it was not personal in the sense that it did not disclose other employees' information. Finally, when asked whether an explanation for a scheduling decision should include only public information, only private information, or a combination of both, the vast majority (88%) responded that only public information should be included, while the remaining participants (12%) suggested a combination of both public and private information.

In conclusion, our study supports the hypothesis that individuals prefer explanations containing only public information, which they perceive as not only more satisfactory but also more equitable.. Based on these findings, our explanation generation framework is designed to align with people's expectations for a scheduling decision explanation in this particular context.

9.6 Related Work

There is a small body of literature on *explainable scheduling*, with EXPRES [183] being the most relevant related work. It uses a MILP to find explanations for unsatisfied user preferences. Nevertheless, it is limited to only identifying a set of reasons for unsatisfied user preferences, thus lacking the ability to address and explain other types of queries, such as how (or why) a schedule can be (or is) (in)feasible. With regards to privacy, EXPRES preserves privacy by post-processing explanations to remove identifying reference to agents. In contrast, we give a more thorough treatment on this issue as we found that it is key to users in our user study. On a similar thread, Cyras *et al.* [48] proposed an argumentation-based approach for explaining why a schedule is (or not) feasible and why a preference was unsatisfied in the schedule, as we also tackle in this chapter. The key differences between their approach and ours is that they do not consider any privacy preservation strategies, they are restricted to makespan scheduling problems, and they did not provide any experimental evaluation of their approach. Finally, Agrawal *et al.* [1] and Bertolucci *et al.* [10] also consider the problem of explaining scheduling decisions, however, their scope is limited to specific domain applications – scheduling Mars rovers and operating rooms, respectively.

A related research area is *explainable planning*, which has a larger body of work. Most of the approaches in this area aim at explaining planning-specific queries, such as why a plan is feasible/optimal and why a particular action is (or not) included in a plan [243, 37, 81, 206, 202, 231]. Closely related is the algorithm we presented in Chapter 5, which also uses minimal correction sets (MCS) and minimal unsatisfiable sets (MUS) to find explanations. However, the key difference is that we now consider privacy preservation and take a philosophically different approach of finding explanations, i.e., we do not reconcile the differences between the mental models of the explainer and explainee. Finally, for a further exposition on the relationship between our approach and previous works such as diagnosis and MUS generation, we refer the reader to Section 3.7.1 of Chapter 3.

9.7 Concluding Remarks

In this chapter, we addressed the challenge of generating explanations for agent scheduling problems, extending our previous work to a multi-agent domain. Our logic-based framework for privacy-aware explanations in ASPs represents an advancement in making AI systems more transparent and trustworthy, while also addressing the crucial aspect of privacy in multi-agent settings.

Key contributions and findings include:

1. A general approach to explainable ASPs that handles both reason-seeking and modification-seeking queries, demonstrating the versatility of our logical frameworks developed in earlier chapters.

2. The introduction of privacy considerations in explanation generation, quantified through a privacy-loss function.

3. Empirical evaluation of our framework's efficacy through experiments and a user study, highlighting the importance of privacy, fairness, and informativeness in scheduling explanations.

However, our work also reveals several important considerations and directions for future research.

Privacy: Despite optimizing for privacy, explanations may still contain private constraints with respect to the explainee. As such, privacy leakage can occur when these explanations are relayed to the explainee. To address this issue and preserve the agents' privacy, we can post-process the explanation by abstracting away the remaining private constraints. This process can take different forms, such as *masking* all identifying references to the agents' whose private constraints are included in the explanation or by completely *retracting* the private constraints from the explanation.

As an example, consider that Thanos has no access rights to any of the agent constraints. Then, the reason-seeking explanation $\epsilon_r = \{x_{4,1,2}, \neg x_{4,1,2} \vee \neg x_{1,1,2}\}$ that is generated for him unfortunately includes Rose's identity and private constraint ($= x_{4,1,2}$). Post-processing ϵ_r will allow us to retract $x_{4,1,2}$ from ϵ_r and mask the identity of Rose from the remaining clause $\neg x_{4,1,2} \vee \neg x_{1,1,2}$, for example, by transforming the clause to its generalized form $atmost_1(\{x_{1,j,t}, x_{2,j,t}, x_{3,j,t}, x_{4,j,t}\})\ {\scriptstyle \forall r_j \in R, s_t \in S}$ (domain constraint C_3).

Explanation Delivery: After the (potential) abstraction phase, the (post-processed) explanation needs to be communicated to the agent. Unless the explainee agent is a domain expert, the explanation should not be communicated in a logical representation, but rather in a human-understandable format such as natural language. A trivial direction could be to leverage the expressivity and symbolic nature of logic. That is, we can define natural language templates and use them to map the generated explanations. In particular, notice that each constraint "symbolizes" a specific constraint type and is grounded on (propositional) variables, with each variable denoting a scheduling element such as an agent, a resource, or a time step. For instance, $\epsilon_r = \{x_{4,1,2}, \neg x_{4,1,2} \vee \neg x_{1,1,2}\}$ says that Rose is assigned the morning shift on Tuesday ($x_{4,1,2}$), and that either Rose or Thanos can be assigned a morning shift on Tuesday ($\{\neg x_{4,1,2} \vee \neg x_{1,1,2}\}$). As such, a logic-based explanation can be transformed into a natural language explanation by identifying and mapping the constraints to their respective pre-defined, natural language templates.

Another possibility is to leverage Large Language Models (LLMs) [20] to translate logical explanations into natural language. However, the accuracy of such translations will need to be validated through additional research as LLMs have been shown to have confabulation issues [246]. In the next chapter, we will explore this second approach in depth, presenting a system that combines our logical explainability layer with LLMs to generate natural and effective explanations. This integration shows how our formal logical foundation can be made more accessible while maintaining formal guarantees such as correctness.

Chapter 10

Demonstration: Trustworthy Reasoning for Contrastive Explanations in Course Scheduling

"The best way to explain it is to do it."

— *Lewis Carroll*

10.1 Introduction & Contribution

While we have developed several approaches to logic-based explanation generation that ensure soundness and validity, a key challenge remains: how can we make these formal explanations more accessible to users without sacrificing their rigorous foundations? Course scheduling provides an ideal test case for addressing this challenge, as students need clear, understandable explanations for scheduling decisions while the system must maintain logical consistency.

The emergence of large language models (LLMs) offers a potential solution to this accessibility challenge. While LLMs excel at generating coherent and contextually relevant text [24], their reliance on statistical inference leads to challenges in maintaining logical consistency and accuracy in reasoning and planning tasks [166, 220]. This limitation is particularly apparent when explanations need to be both linguistically coherent and logically sound. In contrast, symbolic, logical methods provide a robust medium for reasoning and planning due to their ability to perform valid and sound inference. This realization offers an opportunity to combine the strengths of both LLMs and symbolic methods, creating synergistic systems that ensure decisions are not only provably correct and robust, but also communicated in a user-friendly manner.

In this chapter, we present **TRACE-cs** (*Trustworthy ReAsoning for Contrastive Explanations in Course Scheduling Problems*), a demonstration system that integrates our explanation generation algorithms with the natural language capabilities of LLMs for generating explanations in course scheduling problems. This approach ensures that explanations are both provably trustworthy and communicated to users in natural language, addressing both the logical soundness and user comprehension aspects of explainable AI. Specifically, TRACE-cs:

1. Generates natural language explanations for contrastive user queries (e.g., "Why course X instead of course Y?").
2. Leverages the algorithm from Chapter 5 for generating logic-based explanations.
3. Utilizes an LLM-powered user interface for natural language interactions.

This demonstration synthesizes key themes from throughout this thesis - from the formal foundations of our logical explainability layer to its practical application in scheduling problems - while addressing the crucial challenge of explanation communication. By showing how logic can serve as an explainability layer that is both rigorous and user-friendly, TRACE-cs points the way toward AI systems that are both trustworthy and accessible in critical decision-making contexts.

10.2 TRACE-cs Overview

We now provide an overview of the TRACE-cs system, illustrated in Figure 10.1.

Symbolic Module. The Symbolic Module forms the core of TRACE-cs, handling the scheduling logic and explanation generation:

- **Encoder:** Encodes specific scheduling constraints into logical formulae, creating a knowledge base KB that represents the scheduling problem. This includes encoding

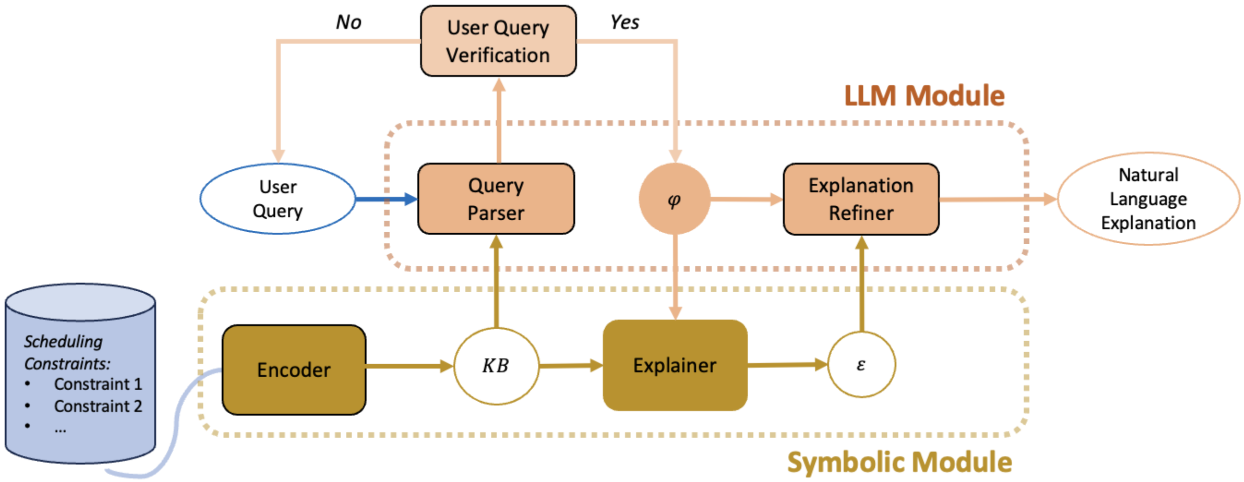


Figure 10.1: The TRACE-CS workflow.

course prerequisites, credit requirements, semester constraints, and so on. Each formula has an associated label attached to it, describing in English the type of scheduling constraint it encodes.

- **Explainer:** Utilizes the explanation generation algorithm we presented in Chapter 9. It takes as input the knowledge base KB from the Encoder and a user contrastive query φ (processed by the LLM module), and generates contrastive explanations. The output is a set of logical formulae along with their corresponding labels.

LLM Module. The LLM Module serves as the interface between the user and the Symbolic Module, handling natural language processing tasks:

- **Query Parser:** Interprets a user's contrastive query in natural language and converts it into a symbolic representation φ compatible with the encoded knowledge base KB. This process employs in-context learning to ensure accurate interpretation. However, recent work by *Karia et al.* [129] highlights the potential limitations of LLMs in formal interpretation tasks, underscoring the importance of human verification in our system. Thus, TRACE-CS includes a step for user verification of the extracted query information before proceeding to explanation generation.

- **Explanation Refiner:** Takes the symbolic explanation ε from the Explainer and translates it into natural language sentences. This translation process also uses in-context learning, utilizing the labels attached to each formula in ε to ensure accurate and coherent explanations.

Figure 10.1 shows the workflow of TRACE-cs:

1. The user submits a contrastive query in natural language;
2. The Query Parser extracts the information from the query and converts it into a symbolic representation φ consistent with the knowledge base KB created by the Encoder;
3. The user verifies if the extracted query information corresponds to the original query, and proceeds to the next step if it is;
4. The Explainer generates a symbolic explanation ϵ for φ with respect to KB;
5. The Explanation Refiner converts ϵ into natural language and outputs it to the user.

This workflow ensures that user queries are accurately interpreted, logically processed, and explained in a user-friendly manner, combining the strengths of symbolic reasoning and natural language processing.

10.3 Proof-of-Concept: Academic Course Schedules

We implemented TRACE-cs in Python as a proof-of-concept for scheduling courses for an undergraduate computer science student across the eight academic semesters at Washington University in St. Louis. To create a comprehensive and realistic scheduling environment, we scraped the computer science course catalog and degree requirements from the university's official website. The Symbolic Module was implemented using PySAT [115] and the LLM Module was implemented using the GPT-4 model [173]. Figure 10.2 shows the user interface of our implementation.[60]

[60] Code repository of the system with full implementation details can be found here: https://github.com/YODA-Lab/TRACE-CS.

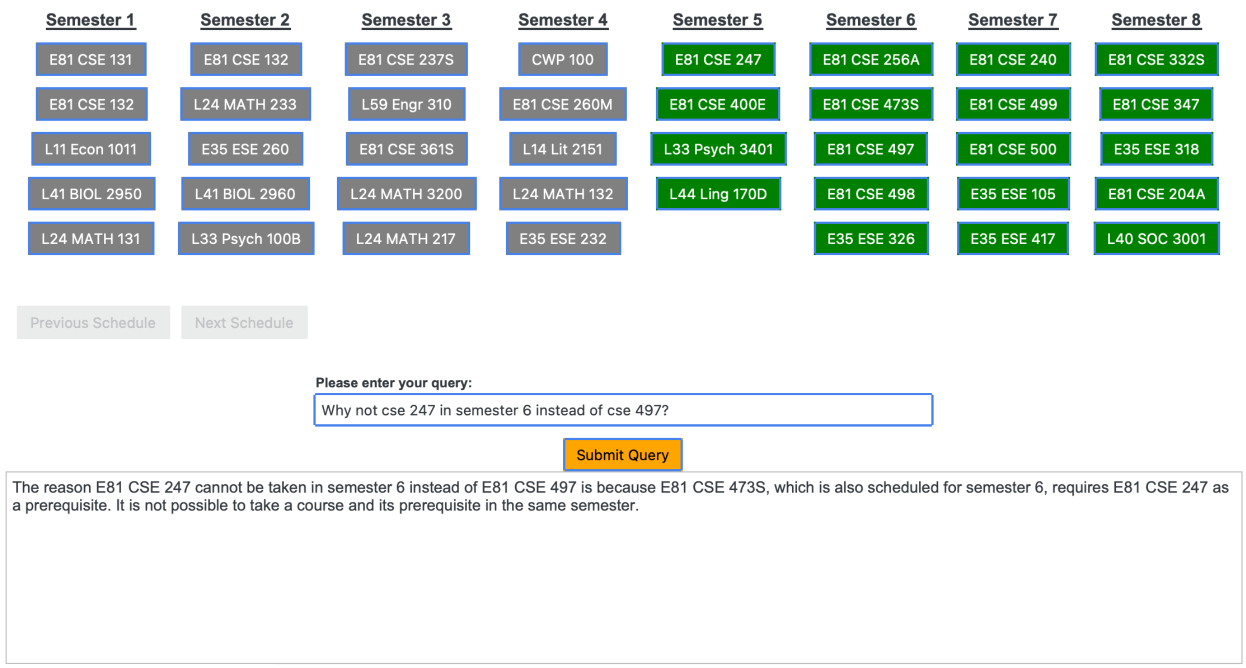


Figure 10.2: The course scheduling user interface.

	TRACE-CS	Zero-shot LLM	Few-shot LLM
Explanation Correctness	100%	44%	49%
Explanation Verbosity	46	113.3	59

Table 10.1: Results from 100 queries comparing TRACE-CS with a zero-shot and a few-shot LLM-only approach.

To evaluate the effectiveness of TRACE-CS, we conducted a comparative experimental study against zero-shot and few-shot LLM-only approaches. Specifically, we generated 10 distinct schedules and created 10 queries for each schedule, totaling 100 schedule-query pairs. Our evaluation metrics were explanation correctness with respect to the degree and course constraints, and explanation verbosity measured by the average number of words per explanation.

Table 10.1 shows the results, where TRACE-CS significantly outperformed both zero-shot and few-shot LLM approaches in terms of explanation correctness, achieving 100% accuracy compared to 44% and 49%, respectively. These results underscore the effectiveness of a hybrid approach in providing trustworthy explanations for course scheduling scenarios.

Moreover, TRACE-CS demonstrated superior performance in terms of explanation verbosity. With an average of 46 words per explanation, TRACE-CS provided more concise explanations compared to both the zero-shot LLM (113.3 words) and few-shot LLM (59 words) approaches. This indicates that TRACE-CS not only generates more accurate explanations but also does so more efficiently, presenting information in a more digestible format for users.

These results collectively demonstrate that TRACE-CS not only provides more accurate explanations but also presents them more concisely. This combination of accuracy and brevity is crucial for effective communication in complex scheduling scenarios, reinforcing the value of our hybrid approach in bridging the gap between symbolic reasoning and natural language explanation generation.

10.4 Related Work

Explainable scheduling research has predominantly relied on logical symbolic methods [48, 1, 10, 183, 182, 231, 226, 241]. While grounded in sound inference procedures, these approaches often produce explanations that are difficult to communicate to users due to their logic-based nature. Attempts to mitigate this limitation have used templates mapping logical explanations to pre-specified natural language sentences [183, 226] or visualization interfaces [47, 146, 182].

Concurrently, LLMs have revolutionized natural language processing and found applications across diverse domains, including planning [128], code generation [198], and medical applications [245]. However, the integration of LLMs with symbolic explainable scheduling systems remains largely unexplored. Our work, TRACE-CS, represents the first attempt to address this gap by presenting a novel hybrid system that synergistically combines a symbolic explainable scheduling module with an LLM module.

10.5 Concluding Remarks

In this chapter, we demonstrated how our logical explainability layer can be enhanced through integration with large language models (LLMs) to provide explanations that are both correct and naturally expressed. TRACE-cs represents the culmination of our work, showing how the theoretical frameworks developed throughout this thesis can be made accessible to users while maintaining their formal guarantees. Our experimental results demonstrate that this enhanced explainability layer significantly outperforms LLM-only approaches in explanation correctness, validating the importance of maintaining logical foundations even when prioritizing accessibility.

The success of TRACE-cs demonstrates a key insight: logic can serve as an explainability layer that is both theoretically sound and practically useful. While LLMs provide natural language capabilities, it is our logical foundation that ensures explanations remain trustworthy and reliable. This combination proves particularly powerful in course scheduling, where explanations must be both precise and easily understood.

Looking ahead, this integration of formal logic and natural language capabilities points toward broader applications of our logical explainability layer. The principles we have developed - from formal foundations to efficient computation, from personalization to privacy awareness, and now to natural language integration - provide a comprehensive framework for explainable AI systems that can: reason logically while communicating naturally, maintain formal guarantees while adapting to user needs, preserve privacy while providing meaningful explanations, support complex decision-making while remaining accessible.

As AI systems continue to evolve and tackle more real-world problems, our work demonstrates how logic can serve as a robust explainability layer that bridges the gap between formal correctness and practical usability.

Chapter 11

Epilegomena

“ Ἓν οἶδα ότι ούδὲν οἶδα. ”
(I know that I know almost nothing, and hardly that)

— *Socrates*

As we conclude this thesis, Socrates' humble assertion resonates deeply with our exploration of explainable AI systems. While we have contributed in developing a general logic-based explainability layer for AI systems, each advance has revealed new depths in the challenge of bridging the decision-making processes of AI systems and human understanding.

Our main thesis posited that *logic-based frameworks can serve as an explanatory representational layer for AI systems, enabling the generation of rigorous, flexible, and human-aware explanations across diverse problem domains by capturing the system's decisions in a formal logical language that supports inference and reasoning*. Our contributions, through systematic development and empirical validation, have provided some evidence supporting this thesis across multiple dimensions.

First, we established the theoretical foundations through the *Logic-based Model Reconciliation Problem* (L-MRP). This framework showed how logic could serve as an explainability layer for planning systems by expressing knowledge and explanations through logical knowledge bases. While our initial focus was on planning problems, L-MRP's applicability extends to various problem types that can be encoded in logical formalisms where satisfiability of sets can be checked. This broad applicability highlights the flexibility of our approach and its potential for more diverse problem domains. However, we also recognized the limitations of deterministic models in representing real-world scenarios.

To address these limitations, we extended our framework to handle uncertainty by developing a probabilistic framework for explanation generation. We introduced concepts such as explanatory gain and explanatory power to quantify the effectiveness of explanations in uncertain environments, including uncertain human user models. This extension allowed us to address more complex scenarios where decision-making often involves incomplete or probabilistic information.

Recognizing the need for efficiently computing explanations, we developed algorithms leveraging the duality between minimal correction sets (MCSes) and minimal unsatisfiable sets (MUSes). These algorithms, applicable to both deterministic and probabilistic settings, enhanced our ability to generate explanations across various problem types and knowledge representations, marking progress towards a practical implementation of our theoretical frameworks.

We then addressed the human aspect of explainability through several advances:

- Personalized explanation generation through knowledge forgetting. This approach allowed us to generate explanations at appropriate abstraction levels based on the human user's vocabulary, addressing the challenge of tailoring explanations to individual users' levels of expertise.
- Dialectical reconciliation via structured argumentative dialogues. This approach addressed the limitations of single-shot explanation processes and facilitated dynamic, argumentation-based interactions between AI systems and human users, enhancing understanding through active engagement and iterative explanation.
- Human-aware belief revision. We introduced a framework for revising human user models that focuses on explanatory understanding rather than mere consistency of beliefs. Through two human-subject studies, we demonstrated that, when faced with inconsistencies, people often opt for non-minimal, explanation-guided revisions to their beliefs. These results challenge the principle of minimal change that is implicit across most belief revision theories in the literature.
- Privacy-aware explanations in multi-agent scheduling problems. We presented a framework for addressing two general queries: (i) why certain scheduling decisions have been made; and (ii) how to alter certain scheduling decisions. This approach demonstrated

a commitment to ethical AI systems that respect individual privacy while maintaining transparency of decisions.

Finally, we presented TRACE-cs, a system for explainable course scheduling that applied our theoretical frameworks to a real-world problem. By combining our explanation generation framework with large language models, TRACE-cs demonstrated the potential for generating explanations that are both logically sound and linguistically accessible to human users.

Framework Limitations and Scope

It is important to highlight the general scope and limitations of our frameworks, particularly L-MRP. While our framework has demonstrated effectiveness across various domains, several key assumptions and limitations warrant careful consideration.

First, L-MRP makes fundamental assumptions about human inferential capabilities. We presume that human users possess the reasoning capacity to process and understand the reconciling explanations provided by the AI system. Note that this does not imply that humans can reason with the same efficiency as AI systems – much like how humans can perform arithmetic operations correctly but are significantly slower than calculators. Rather, we assume that given sufficient time and a properly formulated reconciling explanation, humans can validate the logical correctness of the AI system's decision-making process.

Further, an explanation can fail in several critical ways. Cognitive failures occur when the logical steps exceed human working memory capacity or when the explanation requires domain knowledge the human user lacks. Social failures arise when the explanation, though logically sound, does not align with how humans naturally exchange information and reasoning. Perhaps most importantly, explanatory failure can occur through rejection, i.e., when a human user chooses not to accept the explanation. This rejection often stems from misalignment between the explanation and the human user's (mental) models, beliefs, or expectations about what constitutes a satisfying explanation. For instance, a human user might reject an explanation that fails to address their specific concerns or one that does not match their preferred reasoning patterns, even if the explanation is technically complete and correct.

At the other end of the spectrum, the interplay between understanding, acceptance, and trust creates a complex landscape for explanation generation. While L-MRP provides guarantees of correctness through its formal representational layer, it cannot guarantee user acceptance or trust. This limitation reflects a fundamental challenge in explainable AI: the gap between technical correctness and human satisfaction cannot be bridged through logic alone. This connects to the philosophical perspectives discussed in our introduction – while logical frameworks like the Deductive-Nomological model provide formal rigor, effective explanations must also satisfy the pragmatic and social aspects emphasized by van Fraassen and others. Our framework attempts to balance these competing demands by using logic as a representational foundation while remaining sensitive to the psychological and social dimensions of explanation.

The class of problems where L-MRP can be effectively applied is thus constrained by these considerations. The framework is most suitable for domains where:

- The decision-making process can be expressed in logical terms that align with human reasoning patterns.
- The complexity of the reconciliation process remains within human cognitive capacity.
- The domain knowledge required for understanding the explanation is accessible to the intended human users.
- The explanation structure resonates with the human users' (mental) models.
- The context allows for iterative refinement when initial explanations prove unsatisfactory.

These constraints define the framework's optimal application space rather than limiting its utility. As demonstrated through our empirical studies and implementations like TRACE-cs, many real-world decision-making scenarios fall within these boundaries. Throughout this thesis, we have developed various techniques to address these limitations: our dialectical reconciliation framework facilitates interactive explanation refinement (Chapter 7, our personalized explanation generation approach manages cognitive complexity through abstraction (Chapter 6), and our probabilistic framework handles uncertainty in human user models (Chapter 4. While these advances represent progress, they also highlight an essential insight: the challenge of bridging formal logical systems with human cognitive processes is not merely a technical problem to be solved, but rather a fundamental aspect of human-AI interaction that requires continuous adaptation and refinement.

Afterthoughts and Future Vision

As we reflect on this body of work, several key themes emerge:

- **Logical Foundations Enable Trust:** Logic provides the backbone for generating provably correct explanations, offering formal guarantees about completeness and soundness that are essential for trustworthy AI systems.
- **Efficiency Bridges Theory and Practice:** While theoretical frameworks provide foundations, efficient algorithms for computing explanations are essential for real-world applications, as demonstrated through our MCS-MUS based techniques.
- **Flexibility Across Contexts:** An explainability layer must handle diverse scenarios - from deterministic to probabilistic reasoning, from one-shot explanations to interactive dialogues, from single-agent to multi-agent settings.
- **Human-Awareness Shapes Understanding:** Effective explanations must align with human cognitive processes, from personalized explanations to explanation-guided belief revision, ensuring explanations resonate with human reasoning patterns.
- **Communication Methods Matter:** The medium of explanation significantly impacts its effectiveness, whether through structured dialogues, visual representations, or natural language, highlighting the importance of choosing appropriate communication channels.
- **Privacy Preserves Trust:** In multi-agent settings, explanations must balance transparency with privacy, using techniques like our privacy-loss quantification to protect sensitive information while maintaining meaningful explanations.
- **Hybrid Systems Bridge Theory and Practice:** Combining the formal guarantees of logic with complementary capabilities like natural language generation (through LLMs) creates systems that are both rigorous and accessible. This synthesis, as demonstrated in TRACE-cs, shows how we can maintain logical soundness while making explanations more natural and user-friendly.

These themes not only summarize the contributions of this thesis but also suggest future research directions in explainability. As AI systems become more integrated into daily life, the need for explanations that are accurate, understandable, trustworthy, and respectful of privacy will continue to grow. While this thesis attempted to contribute to the field of explainable decision-making in AI systems, we remain acutely aware of the vast territory yet to be explored. The frameworks, methodologies, and insights developed here could serve as steps towards this vision, paving the way for future innovations in human-aware and trustworthy AI systems.

Our research journey has also led us to develop a new understanding of explanation. The ability to explain is arguably the most significant cognitive functionality of the human species. It underpins our capacity to share insights, justify decisions, and convey thoughts and intentions. This intrinsic need to understand and articulate ideas permeates everyday conversations and extends to the realms of science and research, where we formulate theories to explain certain phenomena.

This *explanatory creativity* is a defining characteristic of human cognition and a key driver of our intellectual and societal progress. Indeed, what we are, fundamentally, is *Universal Explainers* – entities with the unique ability to generate, refine, and communicate explanations across a vast range of domains and complexities. Our attempt to articulate this very idea is itself an explanation; this whole document has served as a, rather long, explanation of my thesis. Our logical explainability layer aims not to replace this capacity but to enhance it, creating a foundation for a long-term vision: *Human-AI Collaborative Decision-Making.*

In this ambitious vision, AI systems will not only match but also enhance our explanatory abilities, engaging in dynamic, bi-directional exchanges of ideas with humans. This "universal explainer" concept will bridge the gap between human intuition and AI's computational power, enabling us to tackle problems that neither humans nor AI could solve independently. By combining human creativity and context-awareness with the AI system's ability to process vast amounts of data and identify patterns, we can unlock new realms of explanatory creativity. This symbiosis will lead to breakthroughs in fields ranging from fundamental scientific research to complex societal challenges. Imagine AI systems that can propose novel scientific theories, collaboratively refine them with human scientists, and then design and interpret experiments to test these theories – all while maintaining a clear chain of reasoning that humans can scrutinize and build upon.

Furthermore, this collaborative intelligence will democratize complex problem-solving, making advanced scientific and analytical tools accessible to a broader audience. It will empower individuals and organizations to make more informed decisions, fostering a society that is better equipped to address global challenges such as climate change, healthcare crises, and sustainable development.

In the grand tapestry of AI research, our work may represent a small contribution. Yet, it is through the collective efforts of researchers worldwide, each contributing their unique perspectives and insights, that we will ultimately weave a future where AI systems are not just powerful tools, but trusted partners in human endeavors. As we look to this future, we do so with excitement, humility, and an unwavering commitment to the pursuit of knowledge that benefits humanity.

References

[1] J. Agrawal, A. Yelamanchili, and S. Chien. Using explainable scheduling for the Mars 2020 rover mission. *arXiv preprint arXiv:2011.08733*, 2020.

[2] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, 1985.

[3] C. E. Alchourrón and D. Makinson. Hierarchies of regulations and their logic. In *New Studies in Deontic Logic: Norms, Actions, and the Foundations of Ethics*, pages 125–148. 1981.

[4] C. Ansótegui, M. Bofill, M. Palahí, J. Suy, and M. Villaret. Satisfiability modulo theories: An efficient approach for the resource-constrained project scheduling problem. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation (SARA)*, pages 2–9, 2011.

[5] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[6] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge - A Logical Approach to Probabilities.* MIT Press, 1990.

[7] C. Barrett and C. Tinelli. Satisfiability modulo theories. In *Handbook of Model Checking*, pages 305–343. 2018.

[8] V. Belle. Logic meets probability: Towards explainable AI systems for uncertain worlds. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5116–5120, 2017.

[9] V. Belle and I. Papantonis. Principles and practice of explainable machine learning. *Frontiers in Big Data*, 4(688969), 2021.

[10] R. Bertolucci, C. Dodaro, G. Galatà, M. Maratea, I. Porro, and F. Ricca. Explaining ASP-based operating room schedules. In *Proceedings of the Workshop on Explainable Logic-Based Knowledge Representation*, 2021.

[11] P. Besnard, É. Grégoire, C. Piette, and B. Raddaoui. MUS-based generation of arguments and counter-arguments. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI)*, pages 239–244, 2010.

[12] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, pages 203–235, 2001.

[13] P. Besnard and A. Hunter. Constructing argument graphs with deductive arguments: A tutorial. *Argument & Computation*, 5(1):5–30, 2014.

[14] A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2021.

[15] Y. M. Bishop, S. E. Fienberg, and P. W. Holland. *Discrete Multivariate Analysis: Theory and Practice*. Springer Science & Business Media, 2007.

[16] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19:173–209, 2009.

[17] E. Black, N. Maudet, and S. Parsons. Argumentation-based dialogue. *Handbook of Formal Argumentation*, 2, 2021.

[18] M. Bofill, M. Garcia, J. Suy, and M. Villaret. MaxSAT-based scheduling of B2B meetings. In *Proceedings of the International Conference on Integration of AI and OR Techniques in Constraint Programming (CPAIOR)*, pages 65–73, 2015.

[19] S. Bogomolov, D. Magazzeni, A. Podelski, and M. Wehrle. Planning as model checking in hybrid domains. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[20] R. Bommasani, D. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[21] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial intelligence*, 93(1-2):63–101, 1997.

[22] G. Boole. *An Investigation of the Laws of Thought: On which are Founded the Mathematical Theories of Logic and Probabilities*. Dover, 1854.

[23] R. Borgo, M. Cashmore, and D. Magazzeni. Towards providing explanations for AI planner decisions. *arXiv preprint arXiv:1810.06338*, 2018.

[24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1877–1901, 2020.

[25] M. C. Budán, M. L. Cobo, D. C. Martinez, and G. R. Simari. Proximity semantics for topic-based abstract argumentation. *Information Sciences*, 508:135–153, 2020.

[26] R. M. Byrne and C. R. Walsh. Contradictions and counterfactuals: Generating belief revisions in conditional inference. In *Proceedings of the Annual Conference of the Cognitive Science Society (CogSci)*, pages 160–165, 2002.

[27] J. M. Carroll and J. R. Olson. Mental models in human-computer interaction. *Handbook of Human-Computer Interaction*, pages 45–65, 1988.

[28] M. Cashmore, A. Collins, B. Krarup, S. Krivic, D. Magazzeni, and D. Smith. Towards explainable ai planning as a service. In *ICAPS Workshop on Explainable AI Planning*, 2019.

[29] M. Cashmore, M. Fox, and E. Giunchiglia. Planning as quantified boolean formula. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 217–222, 2012.

[30] M. Cashmore, M. Fox, D. Long, and D. Magazzeni. A Compilation of the Full PDDL+ Language into SMT. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 79–87, 2016.

[31] M. Cashmore, D. Magazzeni, and P. Zehtabi. Planning for hybrid systems via satisfiability modulo theories. *Journal of Artificial Intelligence Research*, 67:235–283, 2020.

[32] U. Chajewska and J. Y. Halpern. Defining explanation in probabilistic systems. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, page 62–71, 1997.

[33] T. Chakraborti and S. Kambhampati. (When) can AI bots lie? In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pages 53–59, 2019.

[34] T. Chakraborti, A. Kulkarni, S. Sreedharan, D. E. Smith, and S. Kambhampati. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 86–96, 2019.

[35] T. Chakraborti, S. Sreedharan, S. Grover, and S. Kambhampati. Plan explanations as model reconciliation – an empirical study. In *Proceedings of the IEEE/ACM International Conference on Human-Robot Interaction (HRI)*, pages 258–266, 2019.

[36] T. Chakraborti, S. Sreedharan, and S. Kambhampati. The emerging landscape of explainable automated planning & decision making. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4803–4811, 2020.

[37] T. Chakraborti, S. Sreedharan, Y. Zhang, and S. Kambhampati. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 156–163, 2017.

[38] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

[39] M. T. Chi, N. De Leeuw, M.-H. Chiu, and C. LaVancher. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439–477, 1994.

[40] W. Chi, S. Chien, and J. Agrawal. Scheduling with complex consumptive resources for a planetary rover. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 348–356, 2020.

[41] A. Collins, D. Magazzeni, and S. Parsons. Towards an argumentation-based approach to explainable planning. In *ICAPS Workshop on Explainable AI Planning*, pages 39–43, 2019.

[42] S. A. Cook. The complexity of theorem-proving procedures. In *ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 1971.

[43] K. J. W. Craik. *The Nature of Explanation*. 1943.

[44] J. Crawford and A. Baker. Experimental results on the application of satisfiability algorithms to scheduling problems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1092–1097, 1994.

[45] A. Creswell, M. Shanahan, and I. Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[46] K. Čyras, X. Fan, C. Schulz, and F. Toni. Assumption-based argumentation: Disputes, explanations, preferences. *Journal of Logics and Their Applications*, 4(8), 2017.

[47] K. Čyras, M. Lee, and D. Letsios. Schedule explainer: An argumentation-supported tool for interactive explanations in makespan scheduling. In *AAMAS Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 243–259, 2021.

[48] K. Čyras, D. Letsios, R. Misener, and F. Toni. Argumentation for explainable scheduling. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[49] K. Čyras, A. Rago, E. Albini, P. Baroni, and F. Toni. Argumentative XAI: A survey. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4392–4399, 2021.

[50] K. Čyras and F. Toni. ABA+ assumption-based argumentation with preferences. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 553–556, 2016.

[51] J. Davies and F. Bacchus. Solving MaxSAT by solving a sequence of simpler SAT instances. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*, pages 225–239, 2011.

[52] M. Davis, G. Logemann, Donald, and Loveland. A Machine Program for Theorem Proving. *Communications of the ACM*, 5(7):394–397, 1962.

[53] J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

[54] L. De Moura and N. Bjørner. Z3: An efficient SMT solver. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systemsof (TACAS)*, pages 337–340, 2008.

[55] L. De Moura and N. Bjørner. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77, 2011.

[56] L. De Raedt and K. Kersting. Probabilistic inductive logic programming. In *Probabilistic Inductive Logic Programming*, pages 1–27. 2008.

[57] J. P. Delgrande. A knowledge level account of forgetting. *Journal of Artificial Intelligence Research*, 60:1165–1213, 2017.

[58] E. Demirović, N. Musliu, and F. Winter. Modeling and solving staff scheduling with partial weighted MaxSAT. *Annals of Operations Research*, 275:79–99, 2019.

[59] L. A. Dennis and N. Oren. Explaining BDI agent behaviour through dialogue. *Autonomous Agents and Multi-Agent Systems*, 36(2):29, 2022.

[60] D. Deutsch. *The Beginning of Infinity: Explanations that Transform the World*. Penguin UK, 2011.

[61] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[62] K. Dieussaert, W. Schaeken, W. De Neys, and G. d'Ydewalle. Initial belief state as a predictor of belief revision. *Cahiers de Psychologie Cognitive/Current Psychology of Cognition*, 2000.

[63] C. Domshlak, J. Hoffmann, and A. Sabharwal. Friends or foes? On planning as satisfiability and abstract CNF encodings. *Journal of Artificial Intelligence Research*, 36:415–469, 2009.

[64] Y. Dong, H. Su, J. Zhu, and B. Zhang. Improving interpretability of deep neural networks with semantic information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4306–4314, 2017.

[65] H. T. Dung and T. C. Son. On model reconciliation: How to reconcile when robot does not know human's model? In *Proceedings of the International Conference on Logic Programming (ICLP)*, pages 27–48, 2022.

[66] A. S. d'Avila Garcez, L. C. Lamb, and D. M. Gabbay. *Neural-Symbolic Learning Systems*. Springer, 2009.

[67] H. Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of Neurosciences*, 20(4):155, 2013.

[68] R. Eifler and J. Hoffmann. Iterative planning with plan-space explanations: A tool and user study. *arXiv preprint arXiv:2011.09705*, 2020.

[69] T. Eiter and G. Kern-Isberner. A brief survey on forgetting from a knowledge representation and reasoning perspective. *Künstliche Intelligenz*, 33(1):9–33, 2019.

[70] R. Elio. How to disbelieve p$\rightarrow$ q: Resolving contradictions. In *Proceedings of the Annual Conference of the Cognitive Science Society (CogSci)*, pages 315–320, 1989.

[71] R. Elio and F. J. Pelletier. Belief change as propositional update. *Cognitive Science*, 21(4):419–460, 1997.

[72] M. A. Falappa, G. Kern-Isberner, and G. R. Simari. Explanations, belief revision and defeasible reasoning. *Artificial Intelligence*, 141(1-2):1–28, 2002.

[73] X. Fan and F. Toni. Agent strategies for ABA-based information-seeking and inquiry dialogues. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 324–329, 2012.

[74] X. Fan and F. Toni. On computing explanations in argumentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1496–1502, 2015.

[75] E. Fermé and S. O. Hansson. *Belief Change: Introduction and Overview*. Springer, 2018.

[76] D. Fierens, G. V. den Broeck, B. G. I. Thon, and L. D. Raedt. Inference in probabilistic logic programs using weighted cnf's. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 211–220, 2011.

[77] D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.

[78] J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

[79] M. Fox and D. Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, pages 61–124, 2003.

[80] M. Fox and D. Long. Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research*, 27:235–297, 2006.

[81] M. Fox, D. Long, and D. Magazzeni. Explainable planning. *arXiv preprint arXiv:1709.10256*, 2017.

[82] M. Friedman. Explanation and scientific understanding. *The Journal of Philosophy*, 71(1):5–19, 1974.

[83] A. D. Garcez and L. C. Lamb. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.

[84] P. Gärdenfors. Rules for rational changes of belief. *Philosophical Essays Dedicated to Lennart Aqvist on his Fiftieth Birthday*, 34:88–101, 1982.

[85] P. Gärdenfors. Epistemic importance and minimal changes of belief. *Australasian Journal of Philosophy*, 62(2):136–157, 1984.

[86] P. Gärdenfors. Belief revisions and the ramsey test for conditionals. *The Philosophical Review*, 1986.

[87] P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. The MIT press, 1988.

[88] P. Gärdenfors. The dynamics of belief systems: Foundations vs. coherence theories. *Revue Internationale de Philosophie*, pages 24–46, 1990.

[89] P. Gärdenfors, H. Rott, D. Gabbay, C. Hogger, and J. Robinson. Belief revision. *Computational Complexity*, 1995.

[90] A. Gerevini, A. Saetti, and I. Serina. An approach to temporal planning and scheduling in domains with predictable exogenous events. *Journal of Artificial Intelligence Research*, 25:187–231, 2006.

[91] M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL – the planning domain definition language. Technical Report TR-98-003, Yale Center for Computational Vision and Control, 1998.

[92] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018.

[93] D. H. Glass. How good is an explanation? *Synthese*, 201(2):53, 2023.

[94] M. Göbelbecker, T. Keller, P. Eyerich, M. Brenner, and B. Nebel. Coming up with good excuses: What to do when no plan can be found. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 81–88, 2010.

[95] I. J. Good. Weight of evidence, corroboration, explanatory power, information and the utility of experiments. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 22(2):319–331, 1960.

[96] I. J. Good. Corroboration, explanation, evolving probability, simplicity and a sharpened razor. *The British Journal for the Philosophy of Science*, 19(2):123–143, 1968.

[97] T. F. Gordon. An inquiry dialogue system. *Artificial Intelligence and Law*, 2:239–292, 1994.

[98] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Survey*, 51(5):93:1–93:42, 2018.

[99] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. XAI—explainable artificial intelligence. *Science Robotics*, 4(37), 2019.

[100] C. L. Hamblin. *Fallacies*. Methuen and Co. Ltd., 1970.

[101] C. L. Hamblin. Mathematical models of dialogue. *Theoria*, 37(2):130–155, 1971.

[102] S. O. Hansson. Belief contraction without recovery. *Studia Logica*, 50:251–260, 1991.

[103] S. O. Hansson. Kernel contraction. *The Journal of Symbolic Logic*, 59(3):845–859, 1994.

[104] S. O. Hansson. A survey of non-prioritized belief revision. *Erkenntnis*, 50(2-3):413–427, 1999.

[105] G. Harman. *Change in View: Principles of Reasoning*. The MIT Press, 1986.

[106] S. Haspeslagh, T. Messelis, G. V. Berghe, and P. De Causmaecker. An efficient translation scheme for representing nurse rostering problems as satisfiability problems. In *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART)*, pages 303–310, 2013.

[107] M. Helmert. Decidability and undecidability results for planning with numerical state variables. In *Proceedings of the Conference on Artificial Intelligence Planning Systems (AIPS)*, pages 44–53, 2002.

[108] M. Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[109] C. G. Hempel and P. Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175, 1948.

[110] T. A. Henzinger. The theory of hybrid automata. In *Verification of Digital and Hybrid Systems*, pages 265–292. 2000.

[111] D. Hitchcock and D. Hitchcock. Some principles of rational mutual inquiry. *On Reasoning and Argument: Essays in Informal Logic and on Critical Thinking*, pages 313–321, 2017.

[112] A. Holzinger, A. Carrington, and H. Müller. Measuring the quality of explanations: The system causability scale (scs). *KI-Künstliche Intelligenz*, 34(2):193–198, 2020.

[113] R. Howey, D. Long, and M. Fox. VAL: automatic plan validation, continuous effects and mixed initiative planning using PDDL. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 294–301, 2004.

[114] A. Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47–81, 2013.

[115] A. Ignatiev, A. Morgado, and J. Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 428–437, 2018.

[116] A. Ignatiev, F. Pereira, N. Narodytska, and J. Marques-Silva. A SAT-based approach to learn explainable decision sets. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR)*, pages 627–645, 2018.

[117] A. Ignatiev, A. Previti, M. Liffiton, and J. Marques-Silva. Smallest MUS extraction with minimal hitting set dualization. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*, pages 173–182, 2015.

[118] P. Jackson. Computing prime implicates. In *Proceedings of the ACM Annual Computer Science Conference (CSC)*, pages 65–72, 1992.

[119] W. James. Pragmatism's conception of truth. *Journal of Philosophy, Psychology and Scientific Methods*, 4(6):141–155, 1907.

[120] M. Janota, I. Lynce, and J. Marques-Silva. Algorithms for computing backbones of propositional formulae. *AI Communications*, 28(2):161–177, 2015.

[121] H. Johnson. Processes of successful intentional forgetting. *Psychological Bulletin*, 116(2):274, 1994.

[122] P. N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness.* 1983.

[123] P. N. Johnson-Laird, V. Girotto, and P. Legrenzi. Reasoning from inconsistency to consistency. *Psychological Review*, 111(3):640, 2004.

[124] B. Juba, H. S. Le, and R. Stern. Safe learning of lifted action models. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 379–389, 11 2021.

[125] D. Kahneman. Thinking, fast and slow, 2011.

[126] S. Kambhampati. A classification of plan modification strategies based on coverage and information requirements. In *AAAI Spring Symposium Series*, 1990.

[127] S. Kambhampati. Challenges of human-aware AI systems: AAAI presidential address. *AI Magazine*, 41(3):3–17, 2020.

[128] S. Kambhampati, K. Valmeekam, L. Guan, M. Verma, K. Stechly, S. Bhambri, L. P. Saldyt, and A. B Murthy. Position: LLMs can't plan, but can help planning in LLM-modulo frameworks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 22895–22907, 2024.

[129] R. Karia, D. Dobhal, D. Bramblett, P. Verma, and S. Srivastava. ∀uto∃val: Autonomous assessment of LLMs in formal synthesis and interpretation tasks, 2024.

[130] A.-H. Karimi, B. Schölkopf, and I. Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 353–362, 2021.

[131] V. Karthik, S. Sreedharan, S. Sengupta, and S. Kambhampati. RADAR-X: An interactive interface pairing contrastive explanations with revised plan suggestions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 16051–16053, 2021.

[132] D. Kasenberg, R. Thielstrom, and M. Scheutz. Generating explanations for temporal logic planner decisions. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 449–458, 2020.

[133] H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 1991.

[134] H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 1991.

[135] H. Kautz, D. McAllester, and B. Selman. Encoding plans in propositional logic. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 374–384, 1996.

[136] H. Kautz and B. Selman. Planning as satisfiability. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 359–363, 1992.

[137] F. C. Keil. Explanation and understanding. *Annual Review of Psychology*, 57:227–254, 2006.

[138] G. Kern-Isberner. The principle of conditional preservation in belief revision. In *Proceedings of the International Symposium on Foundations of Information and Knowledge Systems*, pages 105–129, 2002.

[139] S. Khemlani and P. Johnson-Laird. Cognitive changes from explanations. *Journal of Cognitive Psychology*, 25(2):139–146, 2013.

[140] J. Kim. *Plan Summarization for Decision Support in Human Team Planning*. PhD thesis, Massachusetts Institute of Technology, 2020.

[141] J. Kim, C. Muise, A. Shah, S. Agarwal, and J. Shah. Bayesian inference of linear temporal logic specifications for contrastive explanations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5591–5598, 2019.

[142] A. Kimmig, B. Demoen, L. De Raedt, V. S. Costa, and R. Rocha. On the implementation of the probabilistic logic programming language problog. *Theory and Practice of Logic Programming*, 11(2-3):235–262, 2011.

[143] P. Kitcher. Explanatory unification. *Philosophy of Science*, 48(4):507–531, 1981.

[144] J. Kohlas. Probabilistic argumentation systems: A new way to combine logic with probability. *Journal of Applied Logic*, 1(3-4):225–253, 2003.

[145] M. Koshimura, H. Nabeshima, H. Fujita, and R. Hasegawa. Solving open job-shop scheduling problems by SAT encoding. *IEICE Transactions on Information and Systems*, 93(8):2316–2318, 2010.

[146] A. Kumar, S. L. Vasileiou, M. Bancilhon, A. Ottley, and W. Yeoh. Vizxp: A visualization framework for conveying explanations to users in model reconciliation problems. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 701–709, 2022.

[147] S. Kundu and S. Acharyya. Stochastic local search approaches in solving the nurse scheduling problem. In *Proceedings of the International Confernece on Computer Information Systems - Analysis and Technologies (CISIM)*, pages 202–211.

[148] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1675–1684, 2016.

[149] P. Langley. Explainable agency in human-robot interaction. In *AAAI Fall Symposium Series*, 2016.

[150] P. Langley. Varieties of explainable agency. In *ICAPS Workshop on Explainable AI Planning*, 2019.

[151] M. Lauria, J. Elffers, J. Nordström, and M. Vinyals. CNFgen: A generator of crafted benchmarks. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 464–473, 2017.

[152] H. J. Levesque. A knowledge-level account of abduction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1989.

[153] I. Levi. Subjunctives, dispositions and chances. In *Synthese*. 1978.

[154] C. M. Li and F. Manya. MaxSAT, hard and soft constraints. In *Handbook of Satisfiability*, pages 903–927. IOS press, 2021.

[155] M. H. Liffiton, A. Previti, A. Malik, and J. Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, 2016.

[156] M. H. Liffiton and K. A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning*, 40(1):1–33, 2008.

[157] F. Lin. On strongest necessary and weakest sufficient conditions. *Artificial Intelligence*, 128(1-2):143–159, 2001.

[158] T. Lombrozo. The structure and function of explanations. *Trends in Cognitive Sciences*, 10:464–470, 2006.

[159] T. Lombrozo. Simplicity and probability in causal explanation. *Cognitive Psychology*, 55(3):232–257, 2007.

[160] K. Lu, A. Grover, P. Abbeel, and I. Mordatch. Pretrained transformers as universal computation engines. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, page 7628–7636, 2022.

[161] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, page 4768–4777, 2017.

[162] C. Lutz and F. Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 989–995, 2011.

[163] M. C. Magnaguagno, R. Fraga Pereira, M. D. Móre, and F. R. Meneguzzi. Web planner: A tool to develop classical planning domains and visualize heuristic state-space search. In *Knowledge Engineering Tools and Techniques for AI Planning*, pages 209–227, 2020.

[164] J. Marques-Silva. Computing minimally unsatisfiable subformulas: State of the art and future directions. *Journal of Multiple-Valued Logic & Soft Computing*, 19, 2012.

[165] J. Marques-Silva, F. Heras, M. Janota, A. Previti, and A. Belov. On computing minimal correction subsets. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 615–622, 2013.

[166] R. T. McCoy, S. Yao, D. Friedman, M. Hardy, and T. L. Griffiths. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*, 2023.

[167] D. McDermott. The 1998 AI planning systems competition. *AI Magazine*, 21(2):35–35, 2000.

[168] H. Mercier and D. Sperber. Why do humans reason? Arguments for an argumentative theory. *Behavioral and Brain Sciences*, 34(2):57–74, 2011.

[169] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[170] B. Moulin, H. Irandoust, M. Bélanger, and G. Desbordes. Explanation and argumentation capabilities: Towards the creation of more persuasive agents. *Artificial Intelligence Review*, 17(3):169–222, 2002.

[171] K. Myers. Metatheoretic plan summarization and comparison. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 182–192, 2006.

[172] V. Nguyen, S. L. Vasileiou, T. C. Son, and W. Yeoh. Explainable planning using answer set programming. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 662–666, 2020.

[173] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[174] N. Oren, K. van Deemter, and W. W. Vasconcelos. Argument-based plan explanation. In *Knowledge Engineering Tools and Techniques for AI Planning*, pages 173–188. 2020.

[175] S. Palan and C. Schitter. Prolific.ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27, 2018.

[176] A. J. Parkes. Clustering at the phase transition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 340–345, 1997.

[177] S. Parsons, M. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 394–401, 2002.

[178] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[179] J. Pinto and I. Grossmann. A logic-based approach to scheduling problems with resource constraints. *Computers & Chemical Engineering*, 21(8):801–818, 1997.

[180] G. D. Plotkin. *A Structural Approach to Operational Semantics.* Aarhus University, 1981.

[181] G. Politzer and L. Carles. Belief revision and uncertain reasoning. *Thinking & Reasoning*, 7(3):217–234, 2001.

[182] C. Powell and A. Riccardi. Abstract argumentation for explainable satellite scheduling. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2022.

[183] A. Pozanco, F. Mosca, P. Zehtabi, D. Magazzeni, and S. Kraus. Explaining preference-driven schedules: the expres framework. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 710–718, 2022.

[184] H. Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21(2):163–188, 2006.

[185] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526, 1978.

[186] A. Previti and J. Marques-Silva. Partial MUS enumeration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 818–825, 2013.

[187] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[188] A. Rago, H. Li, and F. Toni. Interactive explanations by conflict resolution via argumentative exchanges. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 582–592, 2023.

[189] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.

[190] J. Renkens, A. Kimmig, G. Van den Broeck, and L. De Raedt. Explanation-based approximate weighted model counting for probabilistic logics. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2490–2496, 2014.

[191] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144, 2016.

[192] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[193] N. Robinson, C. Gretton, D. N. Pham, and A. Sattar. SAT-based parallel planning using a split representation of actions. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2009.

[194] S. F. Roselli, K. Bengtsson, and K. Åkesson. SMT solvers for job-shop scheduling problems: Models comparison and performance evaluation. In *International Conference on Automation Science and Engineering (CASE)*, pages 547–552, 2018.

[195] S. Rosenthal, S. P. Selvaraj, and M. M. Veloso. Verbalization: Narration of autonomous robot experience. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 862–868, 2016.

[196] D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.

[197] H. Rott. Two dogmas of belief revision. *The Journal of Philosophy*, 97(9):503–522, 2000.

[198] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

[199] W. C. Salmon. Statistical explanation. *Statistical Explanation and Statistical Relevance*, pages 29–87, 1971.

[200] D. Shterionov, J. Renkens, J. Vlasselaer, A. Kimmig, W. Meert, and G. Janssens. The most probable explanation for probabilistic logic programs with annotated disjunctions. In *Proceedings of the International Conference on Logic Programming (ICLP)*, pages 139–153. 2015.

[201] M. Shvo, T. Q. Klassen, and S. A. McIlraith. Towards the role of theory of mind in explanation. In *AAMAS International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 75–93, 2020.

[202] T. C. Son, V. Nguyen, S. L. Vasileiou, and W. Yeoh. Model reconciliation in logic programs. In *Proceedings of the European Conference on Logics in Artificial Intelligence (JELIA)*, pages 393–406, 2021.

[203] T. A. Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Kongelige Danske Videnskabernes Selskab*, 5:1–34, 1948.

[204] D. Sperber and D. Wilson. *Relevance: Communication and cognition*, volume 142. Harvard University Press Cambridge, MA, 1986.

[205] S. Sreedharan, T. Chakraborti, and S. Kambhampati. Handling model uncertainty and multiplicity in explanations via model reconciliation. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 518–526, 2018.

[206] S. Sreedharan, T. Chakraborti, and S. Kambhampati. The emerging landscape of explainable automated planning & decision making. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4803–4811, 2020.

[207] S. Sreedharan, T. Chakraborti, and S. Kambhampati. Foundations of explanations as model reconciliation. *Artificial Intelligence*, 301:103558, 2021.

[208] S. Sreedharan, A. O. Hernandez, A. P. Mishra, and S. Kambhampati. Model-free model reconciliation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 587–594, 2019.

[209] S. Sreedharan, A. Kulkarni, and S. Kambhampati. *Explainable Human-AI Interaction: A Planning Perspective*. Springer Nature, 2022.

[210] S. Sreedharan, S. Srivastava, and S. Kambhampati. Hierarchical expertise level modeling for user specific contrastive explanations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4829–4836, 2018.

[211] S. Sreedharan, S. Srivastava, and S. Kambhampati. Using state abstractions to compute personalized contrastive explanations for AI agent behavior. *Artificial Intelligence*, 301:103570, 2021.

[212] S. Sreedharan, S. Srivastava, D. Smith, and S. Kambhampati. Why can't you do that HAL? Explaining unsolvability of planning tasks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1422–1430, 2019.

[213] J. Svegliato, S. B. Nashed, and S. Zilberstein. Ethically compliant sequential decision making. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 11657–11665, 2021.

[214] Y. Tang, S. L. Vasileiou, and W. Yeoh. Does your AI agent get you? A personalizable framework for approximating human models from argumentation-based dialogue traces. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[215] J. C. Teze, L. Godo, and G. I. Simari. An approach to improve argumentation-based epistemic planning with contextual preferences. *International Journal of Approximate Reasoning*, 151:130–163, 2022.

[216] P. Thagard. Explanatory coherence. *Behavioral and Brain Sciences*, 12(3):435–467, 1989.

[217] R. Thorstad and P. Wolff. What causal illusions might tell us about the identification of causes. In *Proceedings of the Annual Conference of the Cognitive Science Society (CogSci)*, pages 991–996, 2016.

[218] X. Tian, H. H. Zhuo, and S. Kambhampati. Discovering underlying plans based on distributed representations of actions. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1135–1143, 2016.

[219] M. Ulbricht and R. Baumann. If nothing is accepted–repairing argumentation frameworks. *Journal of Artificial Intelligence Research*, 66:1099–1145, 2019.

[220] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati. On the planning abilities of large language models-a critical investigation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 75993–76005, 2023.

[221] J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

[222] H. Van Ditmarsch, A. Herzig, J. Lang, and P. Marquis. Introspective forgetting. *Synthese*, 169(2):405–423, 2009.

[223] B. C. van Fraassen. *The Scientific Image*. Oxford University Press, 1980.

[224] S. L. Vasileiou, A. Kumar, W. Yeoh, T. C. Son, and F. Toni. Dialectical reconciliation via structured argumentative dialogues. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2024.

[225] S. L. Vasileiou, A. Previti, and W. Yeoh. On exploiting hitting sets for model reconciliation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6514–6521, 2021.

[226] S. L. Vasileiou, B. Xu, and W. Yeoh. A logic-based framework for explainable agent scheduling problems. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 2402–2410, 2023.

[227] S. L. Vasileiou and W. Yeoh. Please: Generating personalized explanations in human-aware planning. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 2411–2418, 2023.

[228] S. L. Vasileiou and W. Yeoh. Human-aware belief revision: A cognitively inspired framework for explanation-guided revision of human models. *arXiv preprint arXiv:2405.19238*, 2024.

[229] S. L. Vasileiou and W. Yeoh. TRACE-cs: A synergistic approach to explainable course scheduling using llms and logic. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[230] S. L. Vasileiou, W. Yeoh, A. Previti, and T. C. Son. On generating monolithic and model reconciling explanations in probabilistic scenarios. *arXiv preprint arXiv:2405.19229*, 2024.

[231] S. L. Vasileiou, W. Yeoh, T. C. Son, A. Kumar, M. Cashmore, and D. Magazzeni. A logic-based explanation generation framework for classical and hybrid planning problems. *Journal of Artificial Intelligence Research*, 73:1473–1534, 2022.

[232] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31:841, 2017.

[233] C. R. Walsh and P. Johnson-Laird. Changing your mind. *Memory & Cognition*, 37(5):624–631, 2009.

[234] D. Walton and E. C. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY press, 1995.

[235] Y. Wang, Y. Zhang, Y. Zhou, and M. Zhang. Knowledge forgetting in answer set programming. *Journal of Artificial Intelligence Research*, 50:31–70, 2014.

[236] J.-P. Watson, J. C. Beck, A. Howe, and L. D. Whitley. Problem difficulty for tabu search in job-shop scheduling. *Artificial Intelligence*, 143(2):189–217, 2003.

[237] D. Wilson and D. Sperber. *Relevance Theory*. Blackwell, 2002.

[238] M. S. Winslett. *Reasoning about Action using a Possible Models Approach*. 1988.

[239] J. Woodward. *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, 2003.

[240] Z. Zahedi, A. Olmo, T. Chakraborti, S. Sreedharan, and S. Kambhampati. Towards understanding user preferences for explanation types in model reconciliation. In *Proceedings of the IEEE/ACM International Conference on Human-Robot Interaction (HRI)*, pages 648–649, 2019.

[241] P. Zehtabi, A. Pozanco, A. Bolch, D. Borrajo, and S. Kraus. Contrastive explanations of centralized multi-agent optimization solutions. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2024.

[242] Q. Zhang, Y. N. Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8827–8836, 2018.

[243] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, and S. Kambhampati. Plan explicability and predictability for robot task planning. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1313–1320, 2017.

[244] Y. Zhang and Y. Zhou. Knowledge forgetting: Properties and applications. *Artificial Intelligence*, 173(16-17):1525–1537, 2009.

[245] H. Zhou, B. Gu, X. Zou, Y. Li, S. S. Chen, P. Zhou, J. Liu, Y. Hua, C. Mao, X. Wu, et al. A survey of large language models in medicine: Progress, application, and challenge. *arXiv preprint arXiv:2311.05112*, 2023.

[246] T. Y. Zhuo, Y. Huang, C. Chen, and Z. Xing. Red teaming ChatGPT via jailbreaking: Bias, robustness, reliability and toxicity. *arXiv preprint arXiv:2301.12867*, 2023.

Appendix A

Chapter 3

A.1 Human-Subject Study: Comprehension Questions

Q1. *What caused the error(s) in your plan?* (Multiple Choice)

- I made a mistake in my plan.
- Wrong information provided in domain description.
- Missing information in domain description.
- There were no errors in my plan.

Q2. *Given your plan and the explanation provided, why is your plan invalid? Please be as descriptive as possible.* (Open-ended)

Q3. *Do you feel you understand what information was different in the domain Rob gave you? If yes, what was that information?* (Open-ended)

Q4. *What are the corrections needed to your plan to make it achieve the goal, with the new information in mind?* (Open-ended)

Q5. *Where do you think the error in Rob's domain description was?* (Multiple choice)

- In the description of the actions and their preconditions.
- In the description of the states (start state or goal state).

Q6. *If applicable, identify areas with wrong or missing preconditions by clicking on the corresponding region. Double click to unselect.* (Users are shown a selection area where they can click on various actions.)

Q7. *If applicable, identify areas with wrong or missing start states by clicking on the corresponding region. Double click to unselect.* (Users are shown a selection area where they can click on various states.)

These questions ensured that the participants had to think about what the explanation meant, and hence allow us to see if they really understood it.

To evaluate the user responses, we scored them for each question, where the maximum score that can be achieved is 8 points. For the open-ended questions (Q2, Q3, and Q4), we manually read through the answers and assigned a correct and incorrect flag. For the other questions, we had an answer key to check against the user responses. All questions except Q5 are worth 1 point. Q5 is worth 2 points if participants only select the correct answer, 1 point if they select both answers, and 0 otherwise.

Appendix B

Chapter 6

B.1 Human User Study

Figure B.1 depicts the 5x4 grid that was shown to all participants in the study. Recall that the participants were informed that the robot moved from location B2 to A3 and then made a decision for which an explanation was generated.

B.1.1 Explanation Generation

The explanations for the participants were generated from the following knowledge base, i.e., the robot's knowledge base:

$$\mathrm{KB}_a = \{\Phi, \Omega, \neg\Phi \vee \neg\Omega \vee \Lambda, \neg\Lambda \vee \Psi, \neg\Psi \vee C\}$$

with vocabulary

$$\begin{aligned}\mathcal{V}_{\mathrm{KB}_a} = \{&\Phi : \text{rock-at(A3)}\\ &\Lambda : \text{sample-rock(A3)}\\ &\Omega : \text{handempty}\\ &\Psi : \text{have-analysis(rock)}\\ &C : \text{communicate-data(rock)}\}\end{aligned}$$

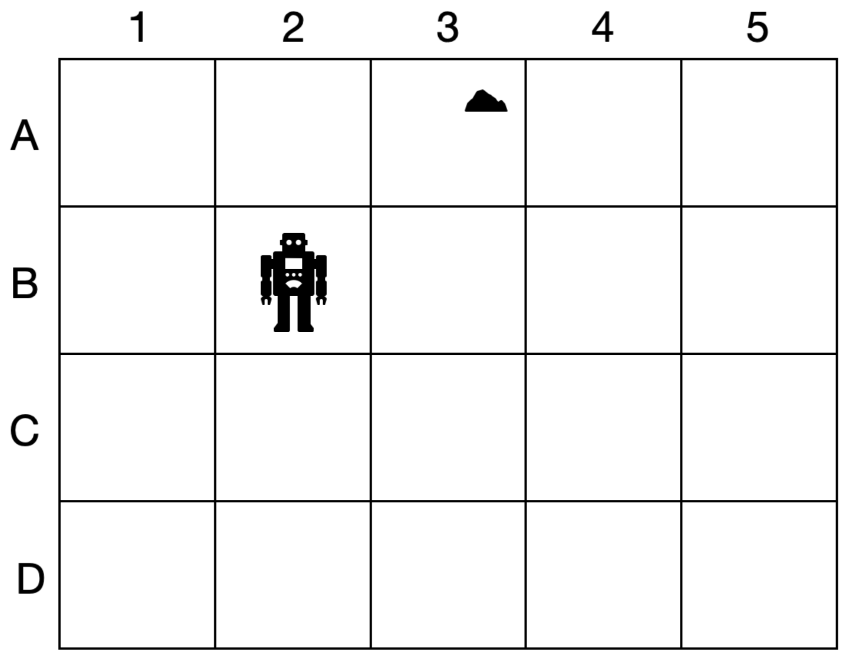


Figure B.1: The 5x4 grid shown to all participants.

The vocabulary $\mathcal{V}_{\mathrm{KB}_a}$ describes the meaning of the symbols used in the formulae of KB_a. The explanandum for the study was $\varphi = C$, where $\mathrm{KB}_a \models C$.

The participants were divided into three vocabulary pairs, with each pair consisting of a treatment group and a control group. The pairs received the following vocabularies:

$$\textit{Pair 1: } \mathcal{V}_{h_1} = \{\Phi\}$$
$$\textit{Pair 2: } \mathcal{V}_{h_2} = \{\Phi, \Lambda\}$$
$$\textit{Pair 3: } \mathcal{V}_{h_2} = \{\Phi, \Lambda, \Omega\}$$

The treatment groups in each pair received a personalized explanation w.r.t. their vocabulary, where we used an upper bound $\mathcal{UB} = 4$. Specifically, the personalized explanations for each treatment group are as follows:

Treatment group 1:

$$\begin{aligned}\epsilon_{t_1} &= \mathcal{F}(\mathrm{KB}_a, \{\Lambda, \Omega, \Psi\}) \\ &= \{\Phi, \neg\Phi \vee C\}\end{aligned}$$

Treatment group 2:

$$\begin{aligned}\epsilon_{t_2} &= \mathcal{F}(\mathrm{KB}_a, \{\Omega, \Psi\}) \\ &= \{\Phi, \neg\Phi \vee \Lambda, \neg\Lambda \vee C\}\end{aligned}$$

Treatment group 3:

$$\begin{aligned}\epsilon_{t_3} &= \mathcal{F}(\mathrm{KB}_a, \{\Psi\}) \\ &= \{\Phi, \Omega, \neg\Phi \vee \neg\Omega \vee \Lambda, \neg\Lambda \vee C\}\end{aligned}$$

The control groups in each pair received the same generic explanation, i.e.:

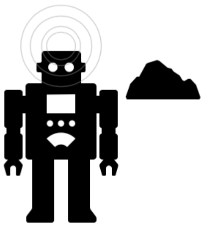
Control groups (for all pairs):

$$\begin{aligned}\epsilon_{c} &= \mathcal{F}(\mathrm{KB}_a, \{\emptyset\}) \\ &= \{\Phi, \Omega, \neg\Phi \vee \neg\Omega \vee \Lambda, \neg\Lambda \vee \Psi, \neg\Psi \vee C\}\end{aligned}$$

Figure B.2 shows the natural language translation of the explanations shown to the groups in each pair.

B.1.2 Results

We now present all of the results and analysis thereof. Upon seeing the explanations, the groups were asked to evaluate the explanations by answering the following questions:

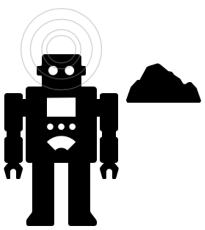


Φ
rock-at(A3)

Treatment Group Explanation:
You told me to move to location A3, but because of Φ I communicated that data to the station.

Control Group Explanation:
You told me to move to location A3, but because of Φ and Ω, I executed Λ. Then, because I had done Λ, I resulted in doing Ψ. And because I had done Ψ, I communicated that data to the station.

(a)

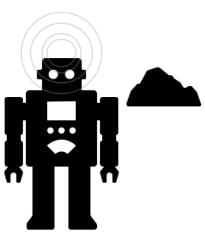


Φ	Λ
rock-at(A3)	sample-rock(rock,A3)

Treatment Group Explanation:
You told me to move to location A3, but because of Φ, I executed Λ. Then, because I had done Λ, I communicated that data to the station.

Control Group Explanation:
You told me to move to location A3, but because of Φ and Ω, I executed Λ. Then, because I had done Λ, I resulted in doing Ψ. And because I had done Ψ, I communicated that data to the station.

(b)



Φ	Ω	Λ
rock-at(A3)	handempty	sample-rock(rock,A3)

Treatment Group Explanation:
You told me to move to location A3, but because of Φ and Ω, I executed Λ. Then, because I had done Λ, I communicated that data to the station.

Control Group Explanation:
You told me to move to location A3, but because of Φ and Ω, I executed Λ. Then, because I had done Λ, I resulted in doing Ψ. And because I had done Ψ, I communicated that data to the station.

(c)

Figure B.2: The explanations shown to the groups in each vocabulary pair, where (a) are the explanations for pair 1 $\mathcal{V}_{h1}$, (b) for pair 2 $\mathcal{V}_{h2}$, and (c) for pair 3 $\mathcal{V}_{h3}$

Q1. **The explanation helped me understand the robot's decision to communicate the data.** (Likert-type: 1: *Strongly disagree* - 5 : *Strongly agree*)

Q2. **I am satisfied with the robot's explanation about how it behaved.** (Likert-type: 1: *Strongly disagree* - 5 : *Strongly agree*)

Q3. **I feel that the explanation of how the robot behaved has sufficient detail.** (Likert-type: 1: *Strongly disagree* - 5 : *Strongly agree*)

Q4. **I feel that the explanation of how the robot behaved is complete.** (Likert-type: 1: *Strongly disagree* - 5 : *Strongly agree*)

Q5. **How useful do you find the robot's explanation for understanding its behavior?** (Likert-type: 1: *Not useful at all* - 5 : *Extremely useful*)

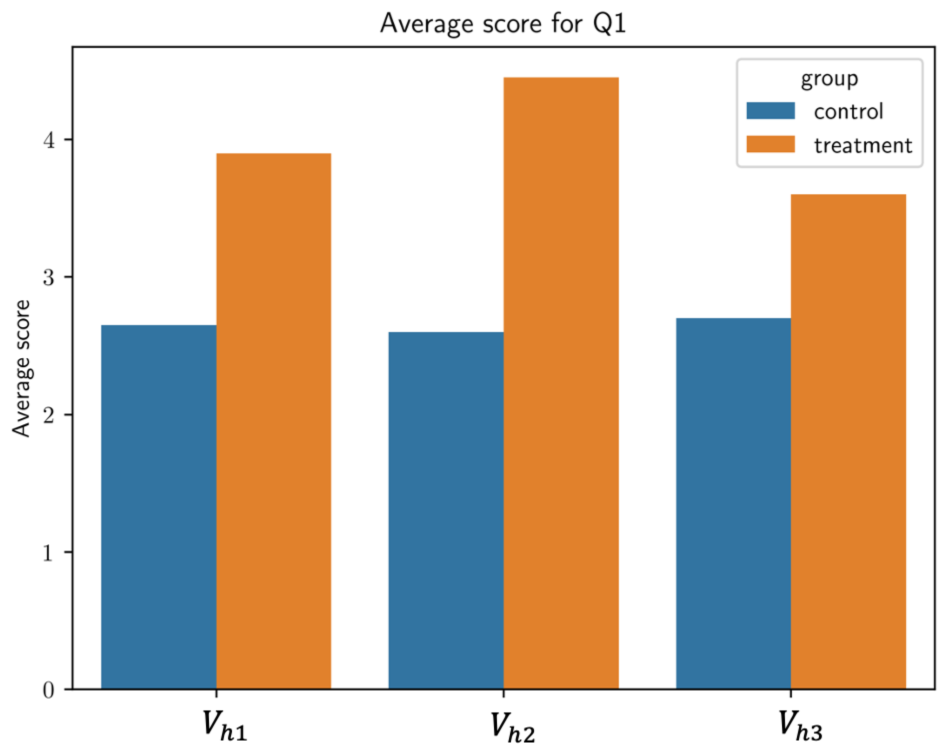
Q6. **How confident are you in your understanding of the explanation?** (Likert-type: 1: *Not confident at all* - 5 : *Extremely confident*)

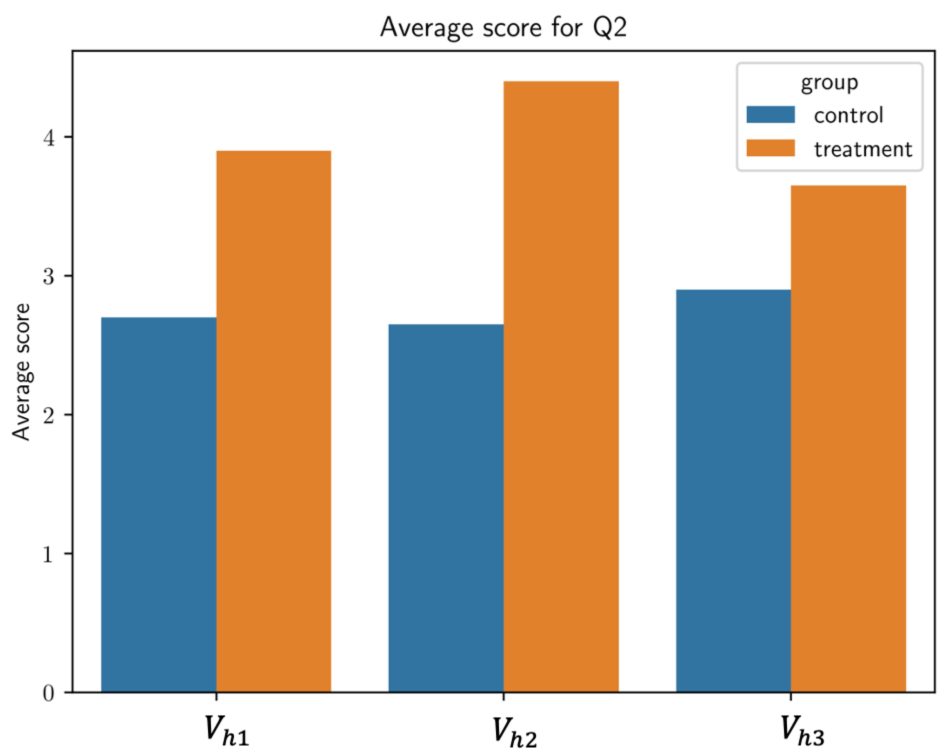
Q7. **How confident are you in your ability to explain the robot's behavior (based on its explanation) to someone else?** (Likert-type: 1 : *Not confident at all* - 5 : *Extremely confident*)

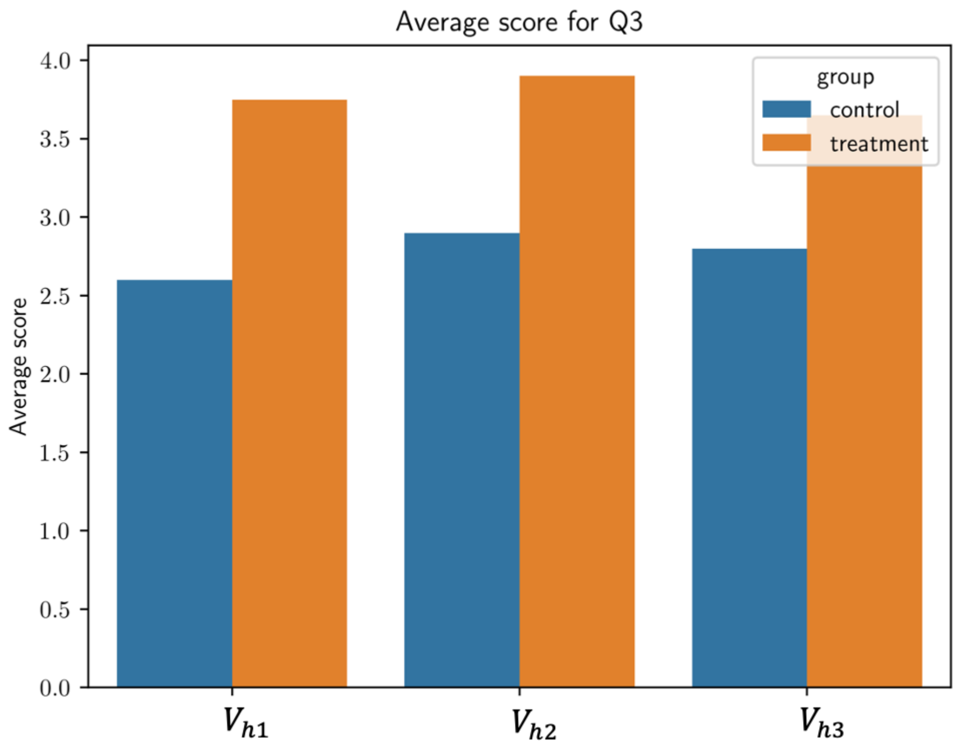
Q8. **Do you think having access to a vocabulary of task-specific terms helped improve your understanding of the explanation?** (Yes, No, Maybe)

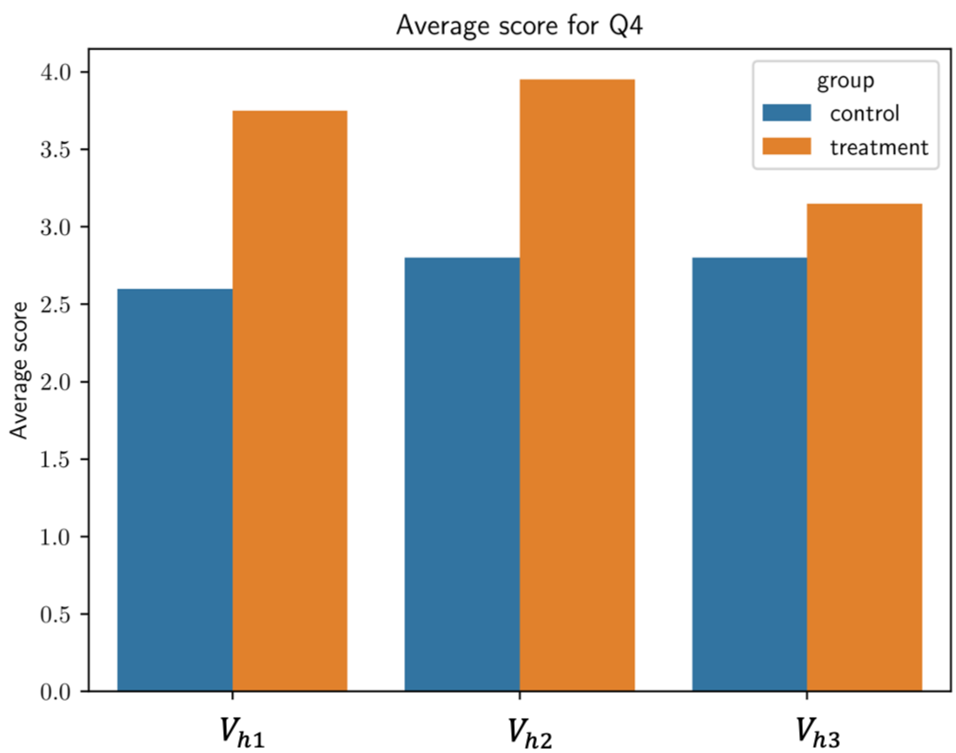
Q9. **In future interactions with AI agents, would you prefer personalized explanations or generic ones?** (Personalized explanations, generic explanations)

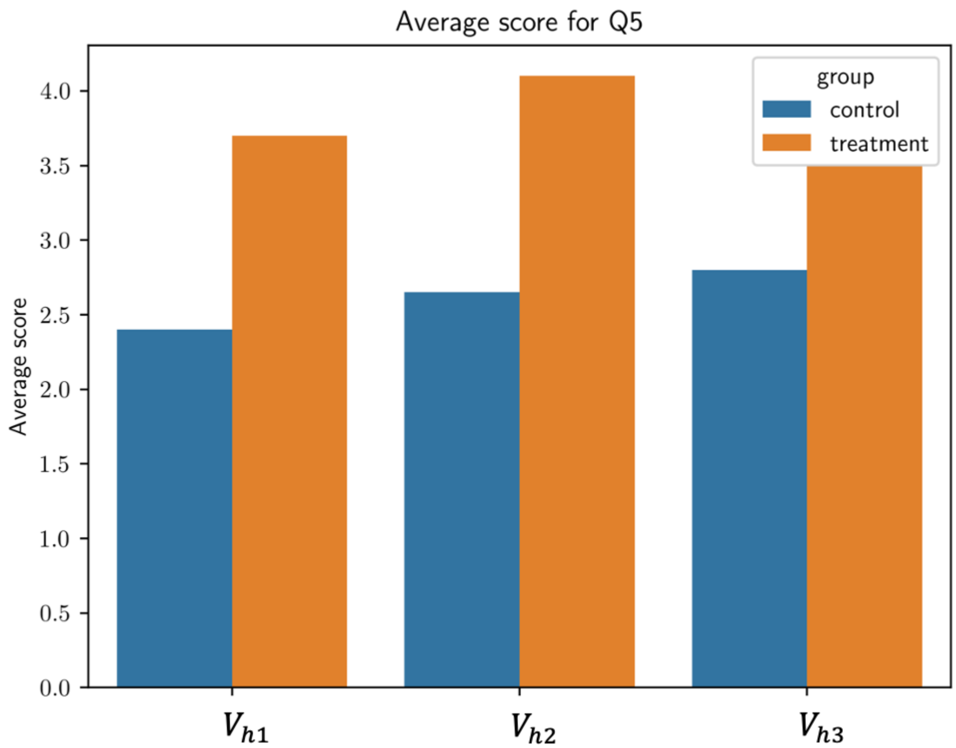
The distributions of each questions are shown in the figures below.

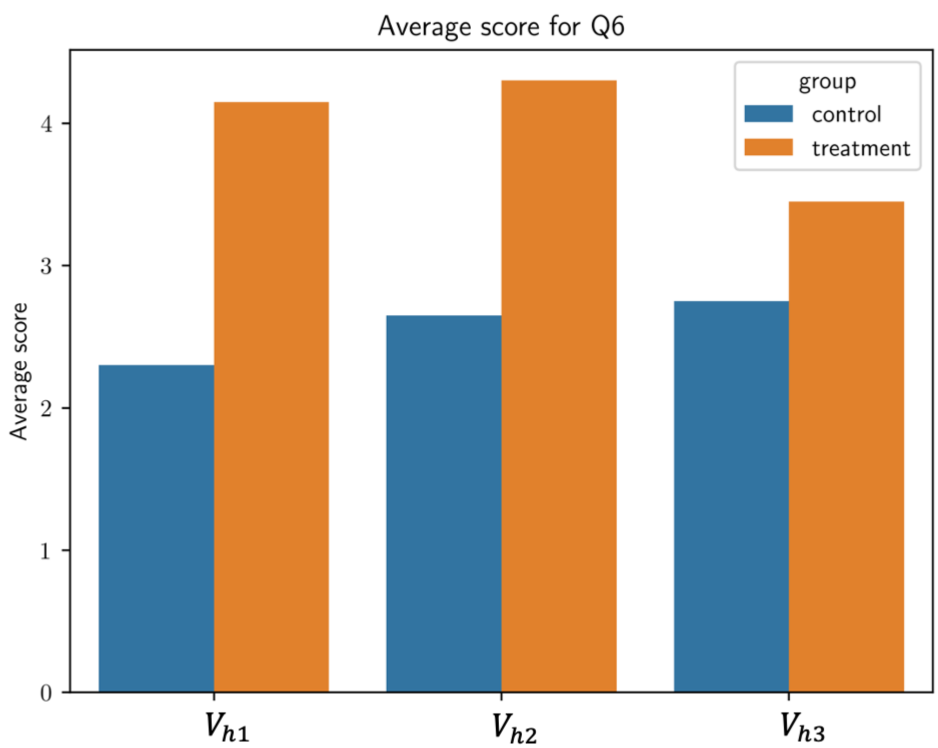

Average score for Q1
Average score
0
1
2
3
4
group
control
treatment
V_{h1}
V_{h2}
V_{h3}

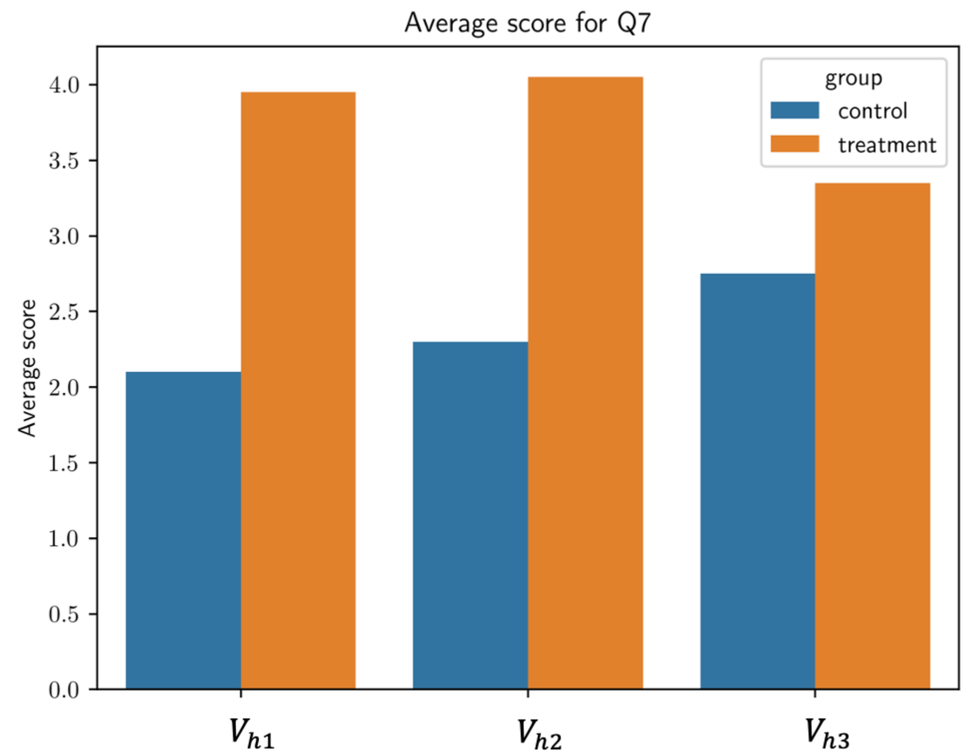

Average score for Q2
Average score
0
1
2
3
4
group
control
treatment
V_{h1}
V_{h2}
V_{h3}

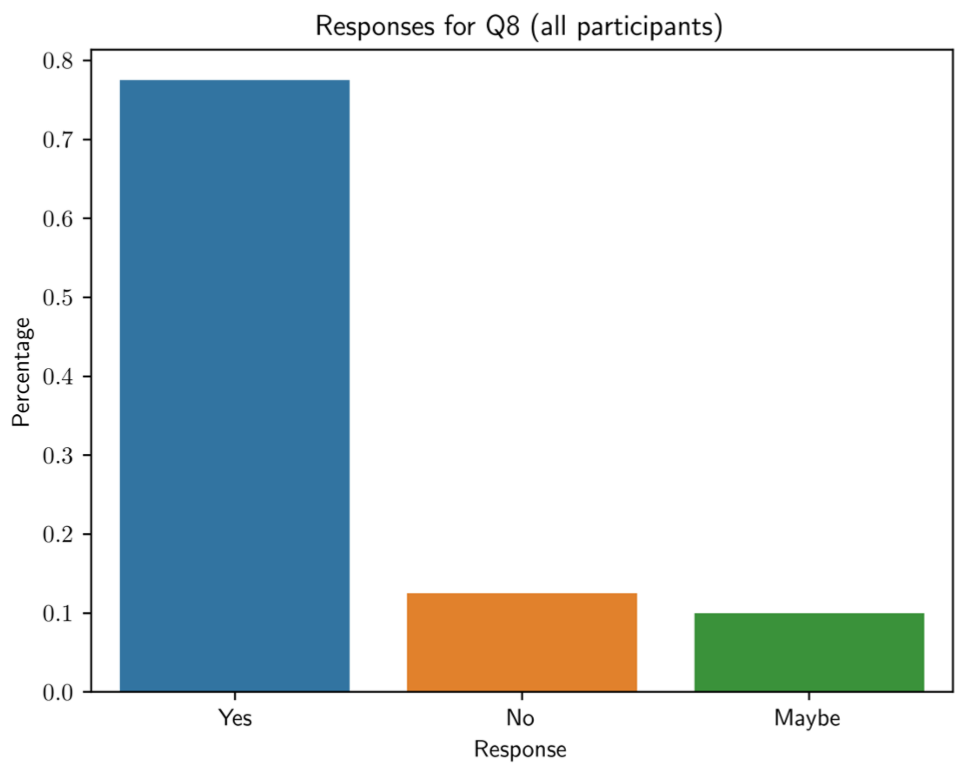

Average score for Q3
group
control
treatment
Average score
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
V_{h1}
V_{h2}
V_{h3}

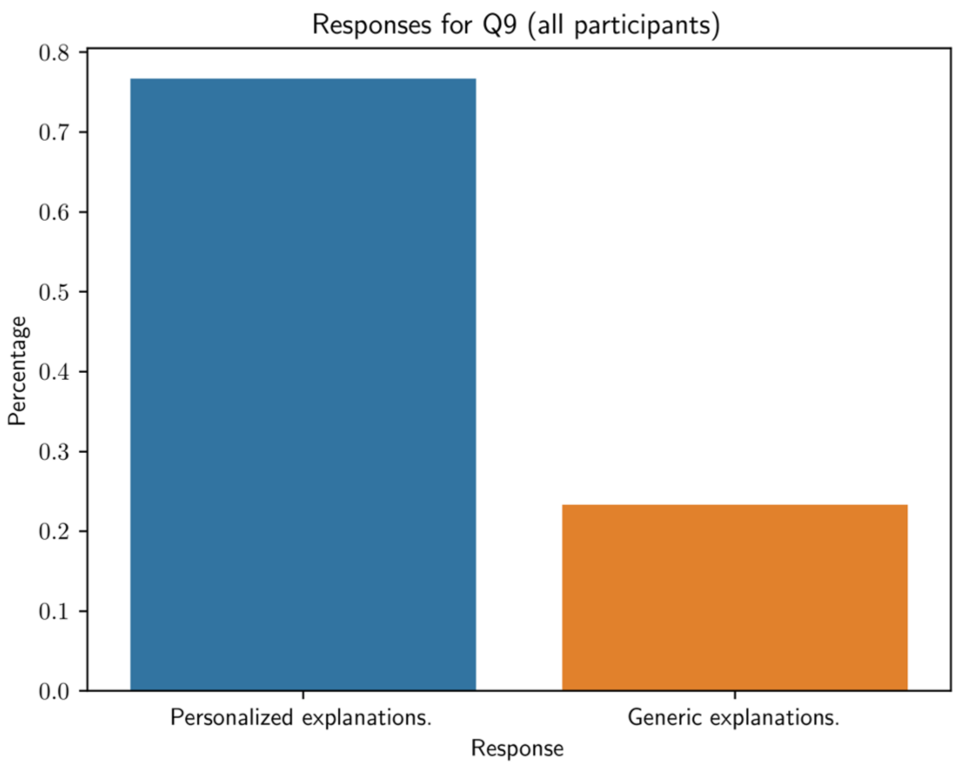

Average score for Q4
group
control
treatment
Average score
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
V_{h1}
V_{h2}
V_{h3}


Average score for Q5
group
control
treatment
Average score
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
V_{h1}
V_{h2}
V_{h3}


Average score for Q6
group
control
treatment
Average score
0
1
2
3
4
V_{h1}
V_{h2}
V_{h3}


Average score for Q7
group
control
treatment
Average score
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
V_{h1}
V_{h2}
V_{h3}


Responses for Q8 (all participants)
Percentage
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Yes
No
Maybe
Response


Responses for Q9 (all participants)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Percentage
Personalized explanations.
Generic explanations.
Response

Appendix C

Chapter 7

C.1 Human-Subject Study

Demographics: Overall, 97 participants completed the study. All participants were proficient in English and had at leas an undergraduate education. Out of the 97 participants, 59 identified as female, 35 as male, and 3 as other.

C.1.1 Study Details

All participants first received the information depicted in Figure C.3. Afterwards, the participants answered two attention check questions, they were divided into two groups:

- **Single-Shot (SSR)**: Group 1 received a single-shot model reconciliation explanation, where the human model was assumed to include the information provided during the scenario's introduction. The explanations were computed using a state-of-the-art solver by Vasileiou et al. [2021].
- **DR-Arg**: Group 2 interacted with DR-Arg's explanations, choosing from four unique questions (counterarguments to Roomie's responses) in a game-like format. They could continue asking questions or decide to end the interaction.

Figures C.4 to C.9 show some of the interactions the DR-Arg users had with Roomie, while Figure C.10 shows the interaction the SSR users had with Roomie.

C.1.2 Study Questions and More Results

In the DR-Arg group, participant engagement varied, leading us to further classify this group for analysis. Specifically, we divided the DR-Arg participants into two subgroups based on their interaction depth:

- **DR-Arg$_{\textbf{Single}}$**: This subgroup is comprised of participants who chose to end the interaction after only one question.
- **DR-Arg$_{\textbf{Multi}}$**: This subgroup is comprised of participants who engaged with more than one question.

This classification allowed us to evaluate the impact of deeper interaction on comprehension and satisfaction.

After the participants in each group interacted with Roomie, they were all asked to answer the questions below. The answers to these questions can be seen in Figures C.1 and C.2.

Comprehension questions:

Q1. *Why did Roomie not have an internet connection?*

1. Hardware lock. (**Correct answer**)
2. Cable not connected properly.
3. Wifi was not working.
4. Docking station failure.

Q2. *Why were there issues even though you paid for the full package?*

1. License expired. (**Correct answer**)
2. Service required.
3. Roomie was set up incorrectly.
4. Roomie does not support uneven floors.

Q3. *Why was there a flashing light next to the internet port?*

1. Battery was low. (**Correct answer**)
2. Internet port was in use.
3. Roomie was malfunctioning.
4. Roomie's connection was high-speed.

Q4. *Why did the app say Roomie was connected to the internet?*

1. Roomie was connected to the internet.
2. App was outdated. (**Correct answer**)
3. All cables were securely connected.
4. Roomie was malfunctioning.

Likert-type questions:

Q1. *Roomie's explanations were easy to understand.*

1: *Strongly disagree* - 5 : *Strongly agree*

Q2. *I understood all the issues with Roomie.*

1: *Strongly disagree* - 5 : *Strongly agree*

Q3. *I would have liked to ask more questions to improve my understanding.*

1: *Strongly disagree* - 5 : *Strongly agree*

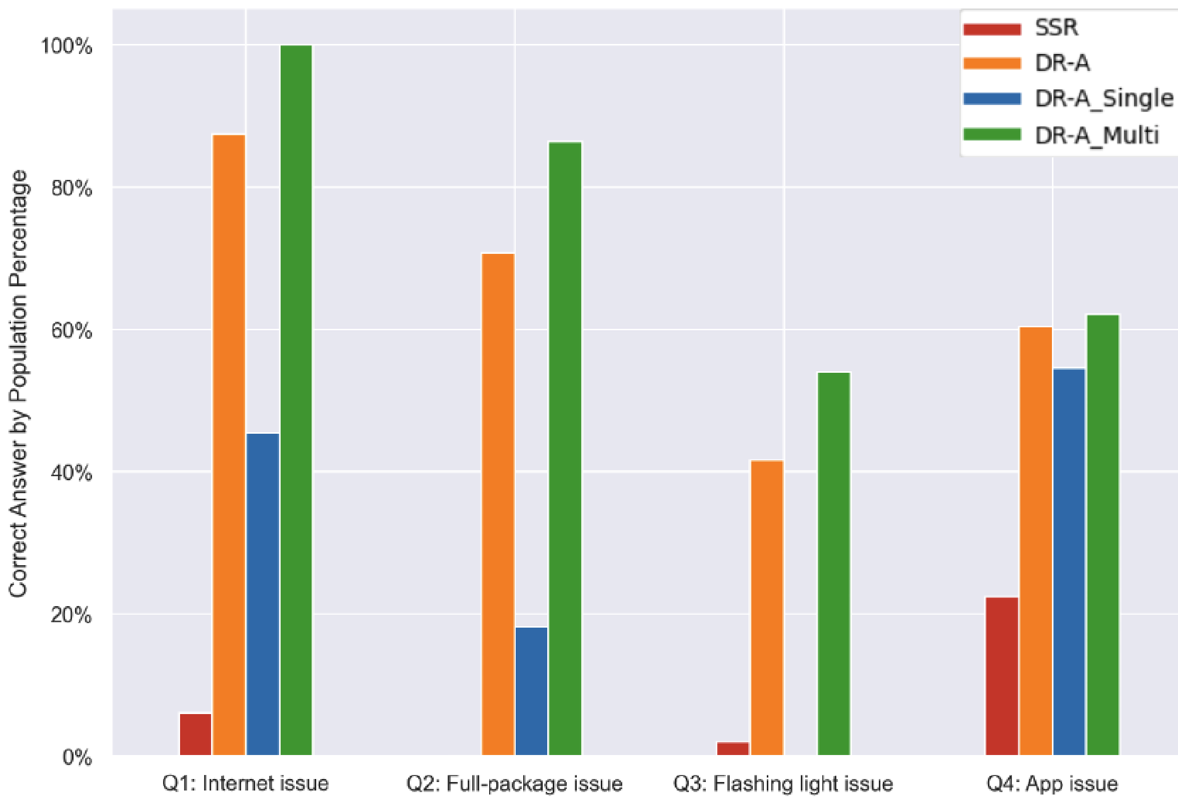


Figure C.1: Distribution of answers to comprehension questions.

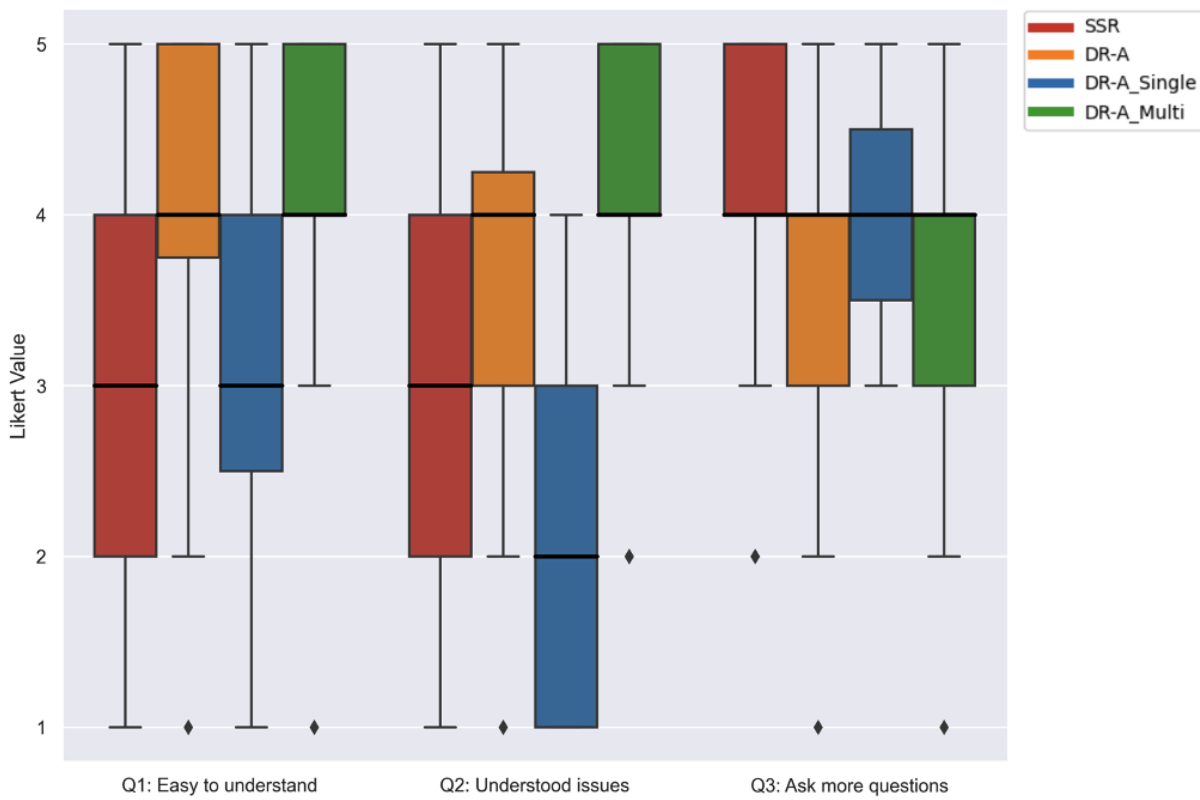


Figure C.2: Distribution of answers to Likert-type questions.

Introducing **Roomie --** Your AI Home Robot

Last month, you got a brand new Roomie robot, an all-in-one AI-powered house cleaner. The product specifications looked amazing. So, you purchased the full package online. However, because you were busy, you did not find the time needed to set it up immediately.

Today, a month later, you finally found the time to set it up. Roomie is a smart robot that can hold conversations with you and clean your house. It came with a docking station that could be used to charge Roomie as well as connect Roomie to the internet when plugged into an internet port. It also came with an app that allows you to set up tasks and see the current status of Roomie.

You then unbox Roomie, plug it into its docking station, connect its internet port and your router with an internet cable, and open the app. However, there is a red icon that says "DISCONNECTED". No matter what you do, you are unable turn the icon off.

You notice the following things:

- The app status says "connected".
- The phone has internet access, and the internet cable is securely connected to both Roomie's internet port and your router.
- There is a flashing light next to Roomie's internet port suggesting that it is in use.

Figure C.3: Introductory information when beginning the user study.

Chatting with Roomie

Since nothing else is working, you try to use the **Q&A mode** on Roomie, which lets the robot interact with you and answer your questions.

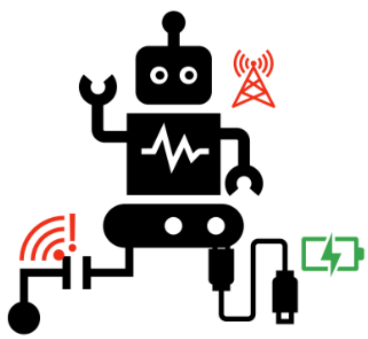
You then ask:

"Why is the "DISCONNECTED" icon on?"

After a few seconds, Roomie responds:

"The DISCONNECTED icon is on because I'm plugged into the docking station and I do not have access to the internet."

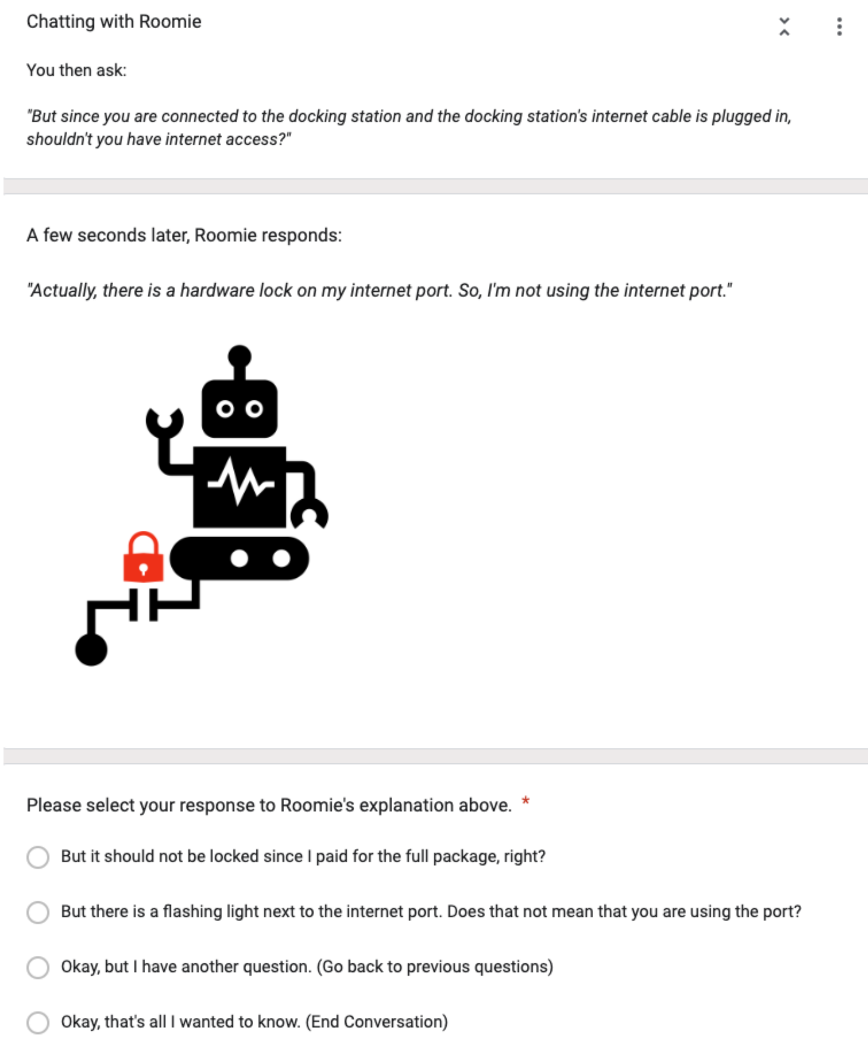
It prompts you to give a response.



Please select your response to Roomie's explanation above. *

- ◯ But since you are connected to the docking station and the docking station's internet cable is plugged in, ...
- ◯ But my app status says that you are connected to the internet.
- ◯ But it is impossible for you to be plugged into the docking station and be disconnected at the same time....
- ◯ Okay, that's all I wanted to know. (End Conversation)

Figure C.4: (DR-Arg) Initial interaction: starting query, response (support) to query, and follow-up questions (refute).



Chatting with Roomie

You then ask:

"But since you are connected to the docking station and the docking station's internet cable is plugged in, shouldn't you have internet access?"

A few seconds later, Roomie responds:

"Actually, there is a hardware lock on my internet port. So, I'm not using the internet port."

Please select your response to Roomie's explanation above. *

- ◯ But it should not be locked since I paid for the full package, right?
- ◯ But there is a flashing light next to the internet port. Does that not mean that you are using the port?
- ◯ Okay, but I have another question. (Go back to previous questions)
- ◯ Okay, that's all I wanted to know. (End Conversation)

Figure C.5: (DR-Arg) Second interaction: Refutation to user response, and follow-up questions.

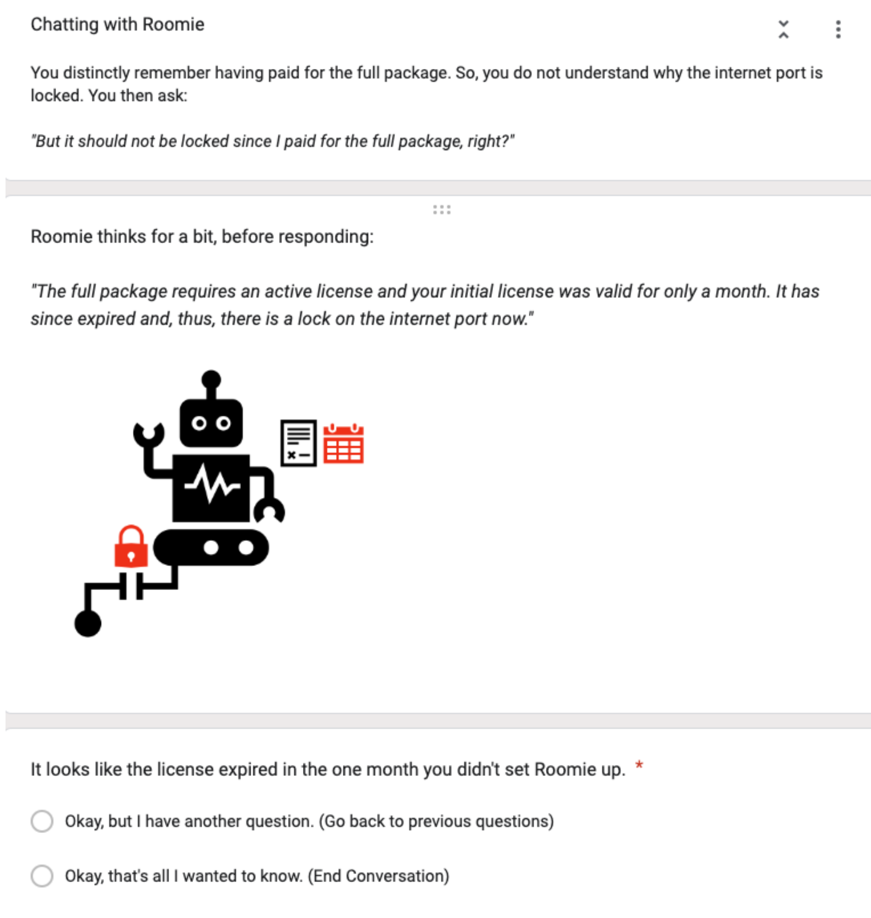


Figure C.6: (DR-Arg) Third interaction: Refutation to user response, and follow-up questions.

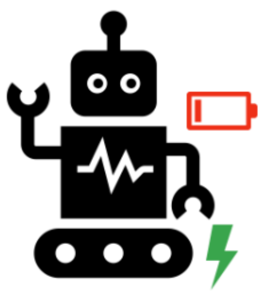
Chatting with Roomie

You see the light flashing next to the internet port. So, it must be in use. You then ask:

"But there is a flashing light next to the internet port. Does that not mean that you are using the port?"

Roomie doesn't take long before responding:

"My battery is low. That light flashes when either my battery is low or when the internet port is in use."



Looks like the flashing light was misleading, and it was warning that the battery is low. *

- Okay, but I have another question. (Go back to previous questions)
- Okay, that's all I wanted to know. (End Conversation)

Figure C.7: (DR-Arg) Fourth interaction: Refutation to user response, and follow-up questions.

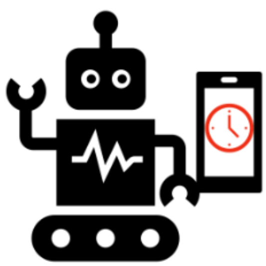
Chatting with Roomie

You think Roomie should be connected because the app status says it is connected. You thus say:

"But my app status says that you are connected to the internet."

Roomie beeps a couple of times before responding:

"Your app version is outdated. So, the app status is wrong."



Looks like you need to update the app to see the correct status. *

- Okay, but I have another question. (Go back to previous questions)
- Okay, that's all I wanted to know. (End Conversation)

Figure C.8: (DR-Arg) Fifth interaction: Refutation to user response, and follow-up questions.

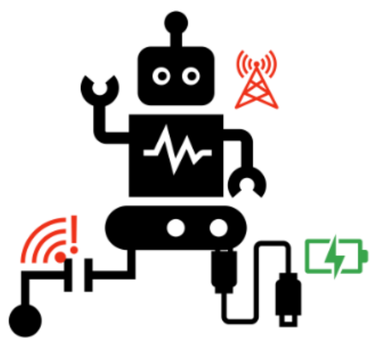
Chatting with Roomie

You don't think it makes sense because Roomie cannot be disconnected when it is clearly plugged into the base. You then say:

"But it is impossible for you to be plugged into the docking station and be disconnected at the same time. So, you should be connected."

Roomie takes a second, before responding:

"I am charging. That's why I know I am plugged into the docking station. I know I do not have internet access because I can not connect to the server on the cloud. So, I am plugged in and disconnected from the internet at the same time."



Please select your response to Roomie's explanation above. *

- ◯ But my app status says that you are connected to the internet.
- ◯ But, since you are connected to the docking station and the docking station's internet cable is plugged in,...
- ◯ Okay, but I have another question. (Go back to previous questions)
- ◯ Okay, that's all I wanted to know. (End Conversation)

Figure C.9: (DR-Arg) Sixth interaction: Refutation to user response, and follow-up questions.

Questioning Roomie

Since nothing else is working, you try to use the **Explain mode** on Roomie, which lets the robot interact with you and answer your question.

You then ask:

"Why is the "DISCONNECTED" icon on?"

After a few seconds, Roomie responds:

"The DISCONNECTED icon is on because I'm plugged into the docking station and I do not have access to the internet."

"Additionally, you have some misconceptions that I would like to clarify. First, being plugged into the docking station does not mean that I have internet access. Also, your app's status wrongly says that I am connected. Finally, even though the internet port is in use, it does not mean that I have internet access."

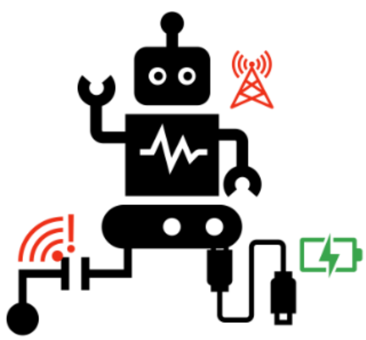


Figure C.10: (SSR) Initial query and response.

Appendix D

Chapter 8

D.1 Human-Subject Experiments

D.1.1 Experiment 1

Our first experiment looked at the three problem types described above and was aimed at providing some empirical data on whether people seek explanations that tend to invoke non-minimal changes to their beliefs, that is, do people seek explanations that resolve categorical or conditional propositions? The participants main task was to explain the inconsistencies presented to them, and we examined the revisions implied by their explanations.

Participants and Design

We recruited 62 participants from the online crowdsourcing platform Prolific [175] across diverse demographics, with the only filter being that they are fluent in English. The participants carried out three different problems of each of three types (Type I, Type II, and Type III). The propositions were taken from common, everyday events including subjects such as economics and psychology. The conditional propositions in all problems were selected to be highly plausible and interpretable, similar to those in the high-plausibility category used by Politzer *et al.* [181].

Each participant was given the following instructions:

> *You will be presented with a series of everyday common scenarios. In each scenario, you will be presented with information from two or three different speakers*

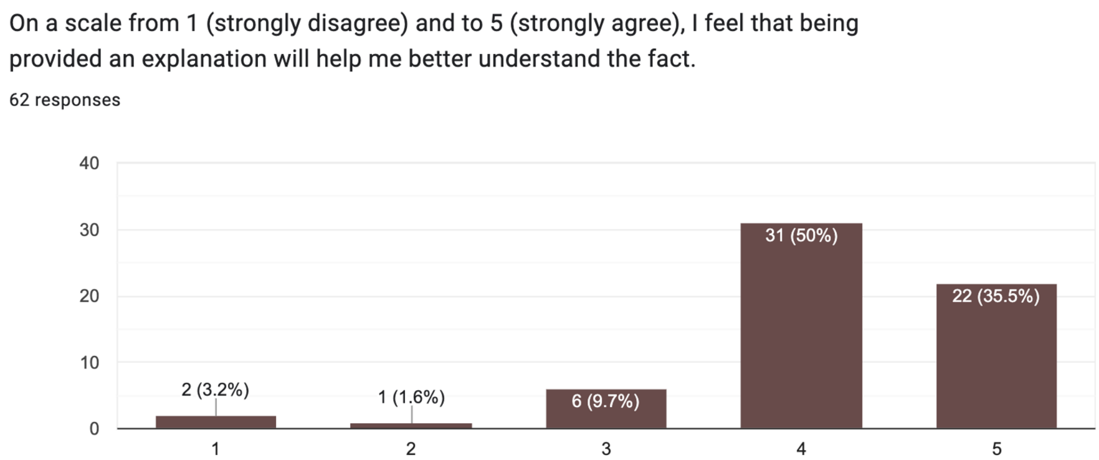


Figure D.1: Distribution of Responses to Likert-type Question.

talking about some specific things. You will then be given some additional information that you know, for a fact, to be true. Your task, in essence, is to explain what is going on.

We gave the following instructions to all participants:

- ***Read carefully:*** *For each scenario, read all the information very carefully.*
- ***Explain****: Think about how to explain the fact. In other words, ask yourself: why does the fact conflict with the information provided by the speakers? Answer in your own words.*
- ***No Right or Wrong Answers****: This study aims to understand your personal thought process. There are no right or wrong answers. Choose what feels most accurate to you.*
- ***Pace Yourself****: While there's no strict time limit, try to spend a reasonable amount of time on each scenario—neither rushing through nor overthinking too much.*

Afterwards, the participants saw the following scenarios and questions:

Scenario 1:

- S_1: *If a drink contains sugar, then it gives you energy.*

- S_2: *This drink contains sugar.*
- **Fact**: *In fact, it doesn't give you energy.*

Why does the drink not give you energy?

Scenario 2:

- S_1: *If sales go up, then profits improve.*
- S_2: *The sales went up.*
- **Fact**: *In fact, the profits did not go up.*

Why did the sales not go up?

Scenario 3:

- S_1: *If people have a fever, then they have a high temperature.*
- S_2: *Maria had a fever.*
- **Fact**: *In fact, Maria did not have a high temperature.*

Why did Maria not have a high temperature?

Scenario 4:

- S_1: *If there is very loud music, then it is difficult to have a conversation.*
- S_2: *If there is very loud music, then the neighbors complain.*
- S_3: *The music was loud.*
- **Fact**: *In fact, the neighbors did not complain.*

Why did the neighbors not complain?

Scenario 5:

- S_1: *If people are worried, then they find it difficult to concentrate.*
- S_2: *If people are worried, then they have insomnia.*
- S_3: *Alice was worried.*
- **Fact**: *In fact, Alice did not find it difficult to concentrate.*

Why did Alice not find it difficult to concentrate?

Scenario 6:

- S_1: *If you follow this diet, then you lose weight.*
- S_2: *If you follow this diet, then you have a good supply of iron*
- S_3: *John followed this diet.*
- **Fact**: *In fact, John did not lose weight.*

Why did John not lose weight?

Scenario 7:

- S_1: *If someone is very kind to you, then you like that person.*
- S_2: *If someone is very kind to you, then you are kind in return.*
- S_3: *Jocko is very kind to Kristen.*
- **Fact**: *In fact, Kristen did not like Jocko, and she were not kind in return.*

Why did Kristen not like Jocko and was not kind to him?

Scenario 8:

- S_1: *If a match is struck, then it produces light.*
- S_2: *If a match is struck, then it gives off smoke.*
- S_3: *Mary struck a match.*
- **Fact**: *In fact, the match produced no light, and it did not give off smoke.*

Why did the match produce no light and gave off no smoke?

Scenario 9:

- S_1: *If people are nervous, then their hands shake.*
- S_2: *If people are nervous, then they get butterflies in their stomach.*
- S_3: *Patrick was nervous.*
- **Fact**: *In fact, Patrick's hands did not shake, and he didn't get butterflies in his stomach.*

Why did Patrick's hands not shake and he didn't get butterflies in his stomach?

After going through all nine scenarios, the participants were asked the following two questions:

Q1: *Describe in your own words how you approached explaining what was going on. Was there a specific reason why you chose to retain or discard certain information?*

Q2: *On a scale from 1 (strongly disagree) and to 5 (strongly agree), I feel that being provided an explanation will help me better understand the fact.*

Figure D.1 shows the distribution of the Likert question (Q2).

D.1.2 Experiment 2

Building on the findings of Experiment 1, which demonstrated a strong tendency among participants to resolve inconsistencies through non-minimal revisions, in this experiment, we look at how people actually revise their beliefs when they are given an explanation. This experiment is also relevant to human-aware AI systems, where AI agents provide explanations to human users.

Participants and Design

We recruited 60 participants from the Prolific platform with the same requirements as before. In this follow-up study, rather than having the participants generate their own explanations, they were presented with some of the most plausible explanations created by participants in Experiment 1, and then asked to describe how they would revise their information in light of the given explanation. To ensure that they will not discard the explanation, they were informed that the explanation is trustworthy. Unlike in Experiment 1, however, the participants were only shown problems of Type II and III. The reason is that these problem types contain more information (e.g., two conditionals and one categorical proposition), and thus it is easier to measure if their revisions are minimal or not.

The scenarios the participants saw can be seen below:

Scenario 1:

- S_1: *If there is very loud music, then it is difficult to have a conversation.*
- S_2: *If there is very loud music, then the neighbors complain.*
- S_3: *The music was loud.*
- **Fact**: *In fact, the neighbors did not complain.*
- **Explanation:** *Explanation: If the neighbors are away on vacations, then very loud music does not lead to complaints.*

Scenario 2:

- S_1: *If people are worried, then they find it difficult to concentrate.*
- S_2: *If people are worried, then they have insomnia.*

- S_3: *Alice was worried.*
- **Fact**: *In fact, Alice did not find it difficult to concentrate.*
- **Explanation:** *If people have effective coping strategies, then they may still be able to concentrate despite being worried.*

Scenario 3:

- S_1: *If you follow this diet, then you lose weight.*
- S_2: *If you follow this diet, then you have a good supply of iron*
- S_3: *John followed this diet.*
- **Fact**: *In fact, John did not lose weight.*
- **Explanation:** *If people have metabolic imbalances, then following a particular diet may not result in weight loss.*

Scenario 4:

- S_1: *If someone is very kind to you, then you like that person.*
- S_2: *If someone is very kind to you, then you are kind in return.*
- S_3: *Jocko is very kind to Kristen.*
- **Fact**: *In fact, Kristen did not like Jocko, and she were not kind in return.*
- **Explanation:** *If people have had negative past experiences with someone, then they may not like that person or reciprocate kindness despite the person being kind to them.*

Scenario 5:

- S_1: *If a match is struck, then it produces light.*
- S_2: *If a match is struck, then it gives off smoke.*

- S_3: *Mary struck a match.*
- **Fact**: *In fact, the match produced no light, and it did not give off smoke.*
- **Explanation:** *If the match is wet, then it will neither produce light nor give off smoke.*

Scenario 6:

- S_1: *If people are nervous, then their hands shake.*
- S_2: *If people are nervous, then they get butterflies in their stomach.*
- S_3: *Patrick was nervous.*
- **Fact**: *In fact, Patrick's hands did not shake, and he didn't get butterflies in his stomach.*
- **Explanation:** *If individuals have practiced stress-management techniques, then they may not exhibit shaky hands or butterflies in the stomach when nervous.*

After each single scenario, the participants answered the following question:

Describe in your own words how you will revise the information. Was there a specific reason why you chose to retain or discard information from the speakers? To be brief, you can write: keep S_1*, discard* S_1*, alter* S_1*, and so on (if you alter, please describe how).*