Theory of Mind Reasoning in Explanation, Plan Recognition, and Assistance: Theory and Practice

by

Maayan Shvo

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Department of Computer Science
University of Toronto

Theory of Mind Reasoning in Explanation, Plan Recognition, and Assistance: Theory and Practice

Maayan Shvo
Doctor of Philosophy

Department of Computer Science
University of Toronto
2023

Abstract

With the rapid advancements in the field of artificial intelligence (AI) in recent years, one of the longstanding goals of the field – building AI systems that are able to interact with other agents in social settings – may appear within reach. However, present-day AI systems do not yet demonstrate the social cognitive skills needed to robustly operate in the real world, and in particular in complex social settings involving humans. Research on humans has identified that Theory of Mind – the ability to attribute mental states (e.g., beliefs, desires, and intentions) to oneself and to others – is a precursor to the development of social cognitive abilities that enable most humans to successfully navigate complex social settings. It follows, then, that augmenting AI systems with Theory of Mind capabilities could improve upon their existing social cognitive abilities and get us closer to achieving AI's aforementioned longstanding goal. In this dissertation, we present a body of work that investigates the role of Theory of Mind in explanation, plan recognition, and assistance and takes steps towards the betterment of social cognitive abilities held by present and future AI systems. Moreover, in much of this dissertation we appeal to the computational paradigm of epistemic planning, which allows us to address or circumvent some of the challenges involved with augmenting AI systems with Theory of Mind. Appealing to epistemic planning allows us to develop computational solutions and demonstrate the feasibility and usefulness of Theory of Mind in explanation, plan recognition, and assistance. Finally, this dissertation employs these computational solutions and

integrates them within a robotic system that demonstrates social cognitive abilities in complex real-world settings, as well as a potential to benefit humans with whom it interacts.

להלה ואמנון שבו

To Hila and Amnon Shvo

Acknowledgements

This journey, which has come to a close with this dissertation, began with a single digital step – an email sent to a professor at my alma mater, whom I did not know, but whom I knew had connections at the University of Toronto. In that email, I asked the professor if she might be willing to connect me with an AI research lab in hopes of gaining experience in the field during my upcoming undergraduate exchange semester. That email led me to Professor Sheila McIlraith, my (future) PhD advisor who connected me with her graduate student and postdoc and facilitated my involvement in a research project. And with that, Sheila warmly welcomed me (in typical Sheila fashion), a fledgling undergrad with no experience or knowledge in the field, to the lab, inviting me to meetings and encouraging me to present the project's findings when it concluded.

The aforementioned graduate student and postdoc were Alberto Camacho and Rick Valenzano who generously contributed their time and expertise to mentor me. Our project was my very first foray into the field and into research more generally. I'd like to thank Rick, Alberto, and Sheila for that opportunity. I am especially grateful for that experience as it led to a second one in the form of an internship at IBM Research, mentored by one of Sheila's former students – Shirin Sohrabi. Shirin took a chance on me based on the project I had done with Rick and Alberto, and hired me as an intern. The summer at the Yorktown Heights IBM Research lab was a delight – I met friends and got to do fundamental research in plan recognition (see Chapter 4 :-)) and take my next couple of steps in research, under Shirin's expert mentorship. Thank you, Shirin, for being a wonderful mentor and co-author for years after the internship had ended. Also, thank you for introducing me to plan recognition and for showing me that it is possible to do cutting-edge research outside of academia (and even publish it).

The workshop publication resulting from the IBM internship led to my first academic event – The AAAI 2017 Workshop on Plan, Activity, and Intent Recognition. Sarah Keren and Reuth Mirsky, two of the co-organizers of the workshop, being the wonderful human beings that they are as well as fellow Hebrew speakers, immediately adopted me and eased the stress accompanying my first academic talk. Thank you, Reuth and Sarah, for your friendship and mentorship over the years, as well as for various discussions on plan recognition and beyond. I look forward to future BIU and Technion visits. I also want to thank Rick Freedman for his friendship and for numerous research discussions that took place over the years.

Following the formative summer at IBM Research, I began a Master's degree in

AI at Utrecht University. My time in Utrecht deepened my interest in symbolic and logic-based AI at a time when deep learning was all one could hear about. I am immensely grateful to John-Jules Meyer for the connection we share and for piquing my interest in multi-agent systems and epistemic reasoning. I also want to mention Sophie Horsman, who was part of my Master's cohort. My friendship with Sophie was forged by logical proofs and kidney-matching algorithms, as well hummus and okonomiyaki. Sophie, thank you for your friendship and for the many discussions on the challenges of doing what this dissertation set out to do (Horsman, 2019). My time in Utrecht concluded with the start of my thesis work, supervised by Mehdi Dastani and Sheila McIlraith. Mehdi, thank you for your mentorship, kindness, and confidence in me. It's always a pleasure to meet up (be it in Utrecht or Montreal) and I look forward to future meetings and exchanges.

This brings us to 2017 and a frigid December day in Toronto, marking the beginning of my second chapter in The Six. After completing my Master's thesis as an international visiting graduate student at the University of Toronto, I went on to begin my PhD under Sheila's supervision in September 2018.

Sheila, thank you for opening your heart and lab all those years ago to an undergraduate exchange student lacking any experience. Thank you for this journey of discovery and mental edification. I learned from you to separate the 'what' from the 'how' and I consider that to be one of the most valuable lessons I've learned, applicable to so many facets of life. Thank you for challenging me, leading to a process of reflection and improvement. Thank you also for allowing me to explore, unconstrained, my research passions and for deepening my interest in empathy, epistemic reasoning, and Theory of Mind through numerous discussions that amplified my excitement about these fascinating topics. I look forward to our future work together. Perhaps most importantly, I wanted to thank you for being a kind and empathetic human being, who cares so deeply about people in your life (and outside of it), be they your family, your friends, your students, or your peers. Lastly, thank you for apologizing for the nasty Canadian winter, but know that, though I appreciate the genuine gesture, it's not your fault!

I had the fortune of doing my PhD in a lab filled with kindness and support, an atmosphere fostered by Sheila over the years. I would like to thank my labmates – Toryn Klassen, Alberto Camacho, León Illanes, Rodrigo Toro Icarte, Pashootan Vaezipoor, Andrew Li, and Keiran Paster – for their friendship and support over the years.

Alberto, you helped me take my very first steps in research and did so with your

typical curiosity and kindness. Ever since, you've given me an abundance of support and valuable advice. Thank you for the above and for your friendship.

Rodrigo, you have a kind presence that exudes warmth and deep thought. Thank you for making me smile every time I entered the lab with your excited yet soft 'Maayan!'. I always admired your ability to be laser-focused on whatever's occupying your mind, as well as your ability to intently listen and outwardly process whatever it is you heard. Thank you for your friendship, for laughing at my jokes, and for listening and discussing research with me.

Pashootan, you're a cool and kind human. I always enjoy the surprising twists and turns of our conversations and I look forward to exploring food and music with you in the future, be it in Toronto or in Montreal.

Toryn, you have co-authored with me the vast majority of the work on which this dissertation builds. I have learned so much from you: about logic, about critically examining one's work and finding chinks in its armor (disjunction anyone?), and about comics, to name a few things. There were times when you challenged me that made me feel anxious, but your challenges (almost) always led to the work being stronger, to another chink fixed. I believe (and I hope that you believe as well) that I am a better thinker and researcher thanks to our journey together. Thank you for all of the above and for your friendship.

I also want to thank Christian Muise for interesting discussions and for patiently answering my practical and theoretical RP-MEP questions throughout the years. Christian, the day will come when we will finally co-author a paper together!

Thank you to my doctoral committee members, Drs. Rick Valenzano and Daniel Wigdor, for your continued support and guidance throughout this process, and to my doctoral examination committee members, Drs. Scott Sanner and Ron Petrick, for taking on this role and facilitating an engaging discussion during my final oral examination (well past midnight for some members).

I also want to thank my support network of family and friends who all contributed in one way or another to this journey.

To the Gralnick family, thank you for welcoming me so warmly into your homes and lives and for always being so supportive of my work.

Ziggy, thank you for your friendship and for all the wonderful philosophical discussions since that 2016 AGI conference. I'm so glad we ended up in cognate fields and found a research project in the space between our respective research.

Liad, you're a true friend and I cherish our relationship. Thank you for your support throughout the years. Honestly, without your high-school study groups, I

would not be here. Here's to many more years of friendship.

Shai, thank you for joining the family and for being a great, intentional listener who truly cares about what others say, and always manages to deepen the conversation.

To my beloved family, Mom, Dad, and Shir, I couldn't have done this without you.

Shir, are we grown-ups already? As I said in September 2018, let's give it a few more years shall we? As you know, I admire your kindness and empathy and your journey as a psychologist often made me reflect on my own empathy-related work that appears in this dissertation. You're curious, inquisitive, and intelligent and I'm always moved when you ask me about my research out of genuine interest in me and what I'm doing. Thank you.

To my parents, Hila and Amnon Shvo, this dissertation (and I) would not be here but for you. Your parenting and love have forged my intellectual curiosity and desire to dig deeper. Every book and even every video and computer game have helped on this journey of curiosity and exploration. I have always felt supported by you, never feeling like a goal was out of reach. I acknowledge the immense privilege that this entails, and hold commensurate gratitude to you for enabling that. I love you very much.

Tara, I love you. As your grandma Helen once told us, we make music together. When I first came into the world of AI, I had the romantic aspiration of combining psychology and computer science to create technology that benefits people. Ever since that Disney Research project, I've been excited about the psychologically-minded music we are able to make and I'm doubly excited about the onion-y music that's in the works. Thank you for being agent Tara in all of my examples. Thank you for listening to me go on (and on) about my research (and only falling asleep once or twice). Thank you for reading and sometimes editing my drafts. Thank you for improving my emotional intelligence over the course of our relationship. And finally, thank you for always being so proud of me. I believe the best is yet to come for us ♡

Contents

Chapter 1

Introduction

"וחווית רעך כמוהו"

– Rolnick (2013)

A longstanding goal of artificial intelligence (AI) research has been to build and deploy AI systems that are able to interact with other agents (be they human or other AI systems) in real-world social settings. To effectively interact with and benefit multiple heterogeneous agents in such settings, these systems should be able to consider the unique *perspectives* held by other agents, in a manner reflective of the Hebrew phrase that opened this chapter: *"experience others as they would experience themselves"* (Rolnick, 2013). As will be demonstrated by this dissertation, considering another's perspective is necessary in a variety of social reasoning tasks including effectively communicating with other agents and reasoning about their behavior, and (perhaps proactively) offering them assistance.

While most humans do not 'experience others as they would themselves' at every turn, they have the capacity to do so (Davies & Stone, 1995; Michlmayr, 2002). Indeed, most humans begin to perform perspective taking of this nature in a variety of social settings at a very young age (e.g., Warneken & Tomasello, 2006), facilitated by advanced social cognitive abilities. Research (e.g., Baron-Cohen et al., 1985; Baron-Cohen, 1991, 1997) shows that an important precursor to the development of these abilities in humans is *Theory of Mind* – the ability to attribute mental states (e.g., beliefs, desires, and intentions) to oneself and to others (Premack & Woodruff, 1978). One hallmark of Theory of Mind (and a litmus test for its presence (Wimmer & Perner, 1983)) is the ability to attribute false beliefs to others. Indeed, the Sally-Anne false-belief task has long been used to test the emergence and development of Theory of Mind in children (Baron-Cohen et al., 1985). As part of the task, testees

are told a story and asked to answer a number of questions pertaining to the beliefs of the characters in the story. Let us consider a story inspired by the Sally-Anne task, involving a robot and two humans:

> *While cleaning the kitchen, Rob the robot notices Alice and Bob in the room with it, observing moreover that Alice put away some groceries, including a fresh stock of coffee beans, which she placed in some cabinet. After Alice leaves the room, the robot notices Bob make himself coffee and return the coffee beans to a different cabinet than the one from which he had taken them (and where Alice had initially put them). The robot, using its Theory of Mind, must now conclude that Alice holds a false belief pertaining to the location of the coffee beans, since she wasn't in the room when Bob moved them. If Alice were to return to the room and, say, take out a mug and boil water, the robot could reason that she plans to make coffee and assist her by communicating that the coffee beans were moved (or by bringing the coffee beans to her).*

Even this mundane situation elucidates that if AI systems possess Theory of Mind capabilities, this could improve upon their existing social cognitive abilities and further benefit humans with whom they interact. Moreover, research has shown that AI systems demonstrating Theory of Mind are perceived as having a better service quality (Söderlund, 2022) and as more socially intelligent (Sturgeon et al., 2019), which may in turn lead to greater trust in and acceptance of such technology (e.g., De Ruyter et al., 2005; Mou et al., 2020). However, while many approaches have been proposed for augmenting AI systems with Theory of Mind and the ability to reason about mental states (e.g., Bratman, 1987; Cohen & Levesque, 1990; Grosz & Kraus, 1996; Scassellati, 2002; Trafton et al., 2005; Berlin et al., 2006; Breazeal et al., 2009; Sindlar et al., 2009; Breazeal et al., 2010; Clair et al., 2011; Baker et al., 2011; Nikolaidis & Shah, 2012; Talamadupula et al., 2014; Leyzberg et al., 2014; Zhao et al., 2015; Andersen, 2015; Devin & Alami, 2016; Nikolaidis et al., 2017; Görür et al., 2017; Panisson et al., 2018; Rabinowitz et al., 2018; Bühler & Weisswange, 2018; Jara-Ettinger, 2019; Brooks & Szafir, 2019; Sarkadi et al., 2019; Dissing & Bolander, 2020; Bühler & Weisswange, 2020; Gervits et al., 2020; Liberman, 2020; Buckingham et al., 2020a; Cuzzolin et al., 2020; Bühler et al., 2021; Bolander et al., 2021; Oguntola et al., 2021; Li et al., 2021; Bühler, 2022; Williams et al., 2022; Sclar et al., 2022; Aru et al., 2022; Montes et al., 2022; Erdogan et al., 2022; Cohen & Galescu, 2023; Wu et al., 2023a), present-day AI systems do not yet demonstrate the social cognitive

skills needed to robustly operate in complex social settings involving multiple agents, and in particular humans.

Reasons for this deficit are many and varied and include a number of challenges (e.g., Horsman, 2019). One major challenge for augmenting AI systems with Theory of Mind is *representing and estimating the mental states of others.* Firstly, an AI system (or a human for that matter) can never[1] gain direct access to the mental states of humans with whom it is interacting. While such mental states – comprising, for example, beliefs, goals, and plans – may be estimated via communication or observation, the resulting estimated state will always be but a proxy for the human's actual mental state. Moreover, the mental state of different agent types may be represented and stored in widely different ways (e.g., as formulae in a knowledge base; parameters of a neural network; or stored in a human brain). Nevertheless, having sufficiently accurate estimations of other agents' mental states is crucial in various settings, as will be demonstrated by this dissertation.

A second challenge is *reasoning about how other agents assimilate information into their existing set of beliefs.* A large body of work has studied belief change in agents, possibly in the context of incorrect and partial beliefs (e.g., Alchourrón et al., 1985; Darwiche & Pearl, 1997). While deciding what constitutes the 'best' way to model belief change is difficult, doing so in the context of multiple agents and in dynamic settings involving action and change adds layers of complexity and remains relatively underexplored.

A third challenge is *higher-order reasoning about mental states.* Past research has found that humans do not excel at reasoning about the deeply nested mental states of other agents (e.g., Mary thought that Bob thought that Andrew thought that Mary thought that ...) (e.g., Miller et al., 1970; Dunn, 1991; Camerer et al., 2004). As will be discussed later on in this dissertation, such reasoning is also challenging for certain approaches in AI. In particular, these approaches must pay a hefty computational price when performing reasoning about nested mental states and limitations put in place to mitigate this cost can lead to failures in Theory of Mind reasoning.

In this dissertation, we present a body of work that investigates the role of Theory of Mind in a number of reasoning tasks and takes steps towards the betterment of social cognitive abilities held by present and future AI systems. The central approach involves adopting Theory of Mind as a conceptual core that provides a lens through

[1] One can imagine how a direct transfer of an agent's mental state could be performed between two AI systems via some common representation. However, at this time, the same process cannot occur for a human's mental state.

which to look at reasoning tasks. In particular, we focus in this dissertation on the following reasoning tasks: *explanation*, *plan recognition* (the task of inferring the plan and goal of an observed agent based on observations (Schmidt et al., 1978; Kautz & Allen, 1986)) and *assistance*. These problems underscore many social settings where multiple agents interact, as will be demonstrated in this dissertation. We provide additional context for these tasks when overviewing the dissertation in the next section.

To realize Theory of Mind-based computational solutions to these reasoning tasks, in much of this dissertation we appeal to the computational paradigm of *epistemic planning* (Cohen, 1978; Perrault et al., 1978; Cohen & Perrault, 1979; Perrault & Allen, 1980; Allen & Perrault, 1980; Cohen et al., 1981; Sadek et al., 1997; Bacchus & Petrick, 1998; Petrick & Bacchus, 2002; Gmytrasiewicz & Doshi, 2005; Bolander & Andersen, 2011; Kominis & Geffner, 2015; Muise et al., 2015b; Baral et al., 2017; Engesser et al., 2017; Huang et al., 2017; Liu & Liu, 2018; Le et al., 2018; Bolander et al., 2018; Fabiano et al., 2020; Engesser & Miller, 2020; Buckingham et al., 2020b; Fabiano et al., 2021; Singh & Khemani, 2020; Wan et al., 2021; Muise et al., 2021; Izmirlioglu et al., 2022; Cooper et al., 2021; Hu et al., 2022; Belle et al., 2022). Epistemic planning enriches automated planning (Ghallab et al., 2004) – the task of selecting a goal-leading plan based on a high-level description of the world – with reasoning about agents' beliefs and knowledge and builds on decades of research on automated planning, epistemic logic (Fagin et al., 1995), and knowledge representation and reasoning (Levesque, 1986). In Chapter 2 we will survey epistemic planning literature and discuss the particular approach to epistemic planning to which we appeal in this dissertation.

Importantly, since the epistemic planning techniques we appeal to are symbolic and logic-based, they are highly attractive as they offer provably correct solutions. This stands in contrast with approaches that draw solely from machine learning techniques to augment AI systems with Theory of Mind reasoning (e.g., Rabinowitz et al., 2018; Nematzadeh et al., 2020; Oguntola et al., 2021; Aru et al., 2022; Nguyen et al., 2023), as well as machine learning models called large language models (LLMs) (Vaswani et al., 2017; Lee & Toutanova, 2018; Brown et al., 2020) that have been purported by some to have emergent, Theory of Mind reasoning (e.g., Kosinski, 2023; Bubeck et al., 2023), with evidence against these claims including lack of robustness of these capabilities (Sap et al., 2022; Ullman, 2023). In Chapter 7 (Section 7.3) we elaborate on this latter point and discuss the relation of LLMs to the work in this dissertation. Either way, generating provably correct solutions using symbolic and

logic-based approaches is important in a myriad of settings, including those where mistakes can be very costly and where solutions should be verifiable. Furthermore, epistemic planning affords inherently inspectable and meaningful representations of agents' mental states, as well as the ability to trace the reasoning performed by agents in multi-agent environments in the context of beliefs and knowledge. As argued by Kambhampati et al. (2022), such symbolic representations – that make sense to humans – are paramount to human-AI interaction that benefits and prioritizes the humans in the loop. Moreover, epistemic planning techniques are able to address some of the aforementioned challenges involved in augmenting AI systems with Theory of Mind reasoning, while circumventing others by making various assumptions that enable reasoning in rich and complex settings.

Finally, as suggested by the title of this document, this dissertation comprises two components: theory and practice. In the final technical chapter of this dissertation, we describe how we employed the computational solutions developed in the earlier chapters and integrated them within a robotic system that demonstrates social cognitive abilities in complex real-world settings. While our robotic integration is, by design, restricted to only work in certain settings, it nevertheless puts our developed theory into practice and showcases the potential of the work presented in this dissertation. With that, our thesis statement is as follows:

> **Thesis Statement.** *Augmenting AI systems with Theory of Mind reasoning is feasible and useful for explanation, plan recognition, and assistance, and the application of such systems in real-world settings has the potential to benefit humans interacting with them.*

In what follows, we provide an overview of this dissertation that supports the above thesis. This overview includes a brief summary of each chapter, in addition to its contributions and associated publications.

1.1 Contributions and Dissertation Overview

Below, we present an overview of the contributions in each of the chapters that follow and moreover relate the various chapters to one another. Chapter 2 includes relevant background on epistemic planning. Chapters 3 to 6 are the main chapters of this dissertation. For each of these chapters, we present a brief summary of its content, contributions, and associated publications. Lastly, Chapter 7 includes the overall conclusions.

Chapter 2: Epistemic Planning

This chapter provides background on epistemic planning, which is the computational paradigm to which we appeal in Chapters 4–6. We begin by discussing automated planning (which epistemic planning augments), continue with an exposition of epistemic logic (upon which epistemic planning builds) and conclude with a discussion of epistemic planning, focusing on the particular flavor of epistemic planning we appeal to in this dissertation.

Chapter 3: The Role of Theory of Mind in Explanation

Miller (2019) states that *"explanations are social — they are a transfer of knowledge, presented as part of a conversation or interaction, and are thus presented relative to the explainer's beliefs about the [beliefs of the recipient of the explanation]."* Indeed, a large body of previous work – from the social sciences to AI – has observed that Theory of Mind capabilities are central to providing an *explanation* to another agent or when explaining that agent's behavior. In Chapter 3, we embrace this view of explanation and explore the role of Theory of Mind in explanation with a view to addressing the diverse needs of explanation in AI. To this end, we build and expand upon previous work by providing an account of explanation in terms of the beliefs of agents and the mechanism by which agents revise their beliefs given possible explanations. We further identify a set of desiderata for explanations that utilize Theory of Mind. These desiderata inform our belief-based account of explanation. With that, the main contributions of Chapter 3 are the following:

1. We identify a set of desiderata for explanations that utilize Theory of Mind.
2. We present a belief-based account of explanation, whose design is informed by the aforementioned desiderata.
3. We prove a number of theorems pertaining to various properties of our account of explanation and various explanation types therein.

Chapter 3 builds upon our EXTRAAMAS 2020 publication (Shvo et al., 2020a).

Chapter 4: Epistemic Plan Recognition

As mentioned, Chapter 3 elucidates the importance of Theory of Mind in explanation and presents a general account of explanation applicable to a myriad of settings. Chapters 4–6 narrow their attention and focus it on Theory of Mind reasoning in

the context of action and change. In particular, in Chapter 4 we are interested in predicting agents' plans and goals given observations about the environment and agents' behavior. As mentioned earlier in this chapter, this is known as the *plan recognition* problem, which can be seen as an exercise in the observing agent's Theory of Mind. That is, in order to explain the observed behavior of the actor (the observed agent), the observer attributes to the latter various mental states – beliefs, plans, and goals. We submit that in some cases, for plan recognition to be effective and complete, it must appeal to a notion of epistemics – beliefs and knowledge – to (1) recognize epistemic goals, where the observed agent is trying to achieve some state of knowledge or belief; and (2) model the observer, and its knowledge of the observed agent, as first-class elements of the plan recognition process. To this end, we formalize the notion of *epistemic plan recognition*, which builds on and brings together epistemic planning and plan recognition. Our epistemic plan recognition specification appeals to an epistemic logic framework to represent agent beliefs. To realize our specification, we cast the epistemic plan recognition problem as an epistemic planning problem, whose solutions can be generated using existing epistemic planning tools. Finally, we evaluate our approach by utilizing and comparing existing epistemic planners on a set of epistemic plan recognition problems. With that, the main contributions of Chapter 4 are the following:

1. We propose and formalize the notion of epistemic plan recognition, which adds an important dimension to the recognition process by appealing to a notion of epistemics to allow for the recognition of epistemic goals and to model the observer and its knowledge of the actor as first-class elements of the recognition process.
2. We propose a computational realization of epistemic plan recognition as epistemic planning, which synthesizes formalisms and computational techniques from both epistemic planning and plan recognition and enables the use of existing planning tools.
3. We evaluate our approach on a set of epistemic plan recognition problems, using two epistemic planning domains and four epistemic planners.
4. We evaluate the impact of the veracity of the observer's beliefs on goal recognition accuracy.

Chapter 4 builds upon our AAMAS 2020 publication (Shvo et al., 2020b).

Chapter 5: Discrepancy Resolution via Theory of Mind

As mentioned, in Chapter 4 we investigate the role of Theory of Mind in plan recognition and allow an observing agent to reason about the plan and goal of other agents. However, as identified by Pollack (1986), *"inferring another agent's plan means figuring out what actions she 'has in mind,' and she may well be wrong about the effects of those intended actions."* Indeed, for a plan to achieve some goal – to be valid – a set of sufficient and necessary conditions must hold. In dynamic settings, agents (including humans) may come to hold false beliefs about these conditions and, by extension, about the validity of their plans or the plans of other agents. Since different agents often believe different things about the world and about the beliefs of other agents, discrepancies may occur between agents' beliefs about the validity of plans. In Chapter 5, we explore how agents can use their Theory of Mind to resolve such discrepancies by acting in the environment. We appeal to an epistemic logic framework to allow agents to reason over other agents' nested beliefs, and demonstrate how epistemic planning tools can be used to resolve discrepancies regarding plan validity in a number of domains. Lastly, we conduct a study that showcases the ability of our approach to resolve misconceptions held by humans. Importantly, the discrepancy resolution formulation proposed in Chapter 5 builds on ideas from Chapter 3 by satisfying the majority of desiderata for explanations that utilize Theory of Mind proposed in that chapter. With that, the main contributions of Chapter 5 are the following:

1. We propose a formulation of discrepancy resolution that appeals to a multi-agent epistemic logic.
2. We present an algorithm that resolves discrepancies via epistemic planning and establish its soundness.
3. We demonstrate that epistemic planning tools can be used to resolve discrepancies via different modalities (i.e., by communicating with other agents or making physical changes in the environment) in various domains and evaluate the impact of the depth of nested belief on the runtime of our algorithm.
4. We conduct a user study which indicates that our approach can effectively resolve misconceptions held by humans pertaining to plan validity.

Chapter 5 is based on our ICAPS 2022 publication (Shvo et al., 2022c).

Chapter 6: Proactive Robotic Assistance via Theory of Mind

Advanced social cognitive skills enhance the effectiveness of human-robot interactions (e.g., Dautenhahn, 2007). Moreover, as discussed earlier in this chapter, research shows that an important precursor to the development of these abilities is Theory of Mind. To this end, in Chapter 6 we endow robots with Theory of Mind abilities and propose a Theory of Mind-based approach to proactive robotic assistance that integrates the epistemic planning-based techniques presented in Chapters 4 and 5. In particular, our approach allows a robot to recognize another agent's plan and goal and resolve any discrepancies pertaining to the validity of the other agent's plan by acting in the environment. Our evaluation shows that robots implementing our approach and demonstrating Theory of Mind are measurably more helpful and perceived by humans as possessing greater Theory of Mind capabilities compared to robots with a deficit in Theory of Mind. With that, the main contributions of Chapter 6 are the following:

1. We present an algorithm that integrates the epistemic planning-based techniques presented in Chapters 4 and 5 to enable proactive robotic assistance via Theory of Mind.
2. We implement our algorithm and integrate it with a humanoid robot (Pepper (Pandey & Gelin, 2018)), combining various perception techniques with Theory of Mind reasoning.
3. We conduct a study evaluating how participants perceive the Theory of Mind capabilities of a robot employing our proposed approach for Theory of Mind-based proactive assistance.
4. We utilize a quantified metric of a robot's helpfulness (Freedman et al., 2020) to measure the efficacy of our method in a set of simulations across various domains.

Chapter 6 is based on our IROS 2022 publication (Shvo et al., 2022b) and parts of our approach to proactive robotic assistance were inspired by our Humanizing AI workshop @ IJCAI 2019 publication (Shvo & McIlraith, 2019).

Chapter 7: Conclusion

Finally, we conclude with a discussion of the body of work presented in this dissertation and its potential broader impact, and reflect briefly on interesting avenues for future research.

1.2 Concluding Remarks

Research shows that an important precursor to the development of advanced social cognitive abilities in humans is Theory of Mind. While augmenting AI systems with Theory of Mind reasoning could improve their ability to robustly navigate social settings with multiple agents, present-day AI systems do not yet demonstrate such capabilities. In this dissertation, we present a body of work that investigates the role of Theory of Mind in a number of reasoning tasks and takes steps towards the betterment of social cognitive abilities held by present and future AI systems. Moreover, in much of this dissertation we appeal to the computational paradigm of epistemic planning, which allows us to address or circumvent some of the challenges involved with augmenting AI systems with Theory of Mind. In Chapters 3–5, we investigate the role of Theory of Mind in explanation, plan recognition, and assistance and in Chapter 6 we describe how the computational solutions developed in the earlier chapters may be employed and integrated within a robotic system that demonstrates social cognitive abilities in complex real-world settings. Finally, in Chapter 7 we conclude by discussing the work presented in this dissertation and its potential broader impact.

Chapter 2

Epistemic Planning

In this chapter we provide background on epistemic planning, which is the computational paradigm to which we appeal in Chapters 4–6. We begin by discussing automated planning (which epistemic planning augments), continue with an exposition of epistemic logic (upon which epistemic planning builds) and conclude with a discussion of epistemic planning, focusing on the particular formalism of epistemic planning we appeal to in this dissertation.

2.1 Automated Planning

Automated planning (also referred to as AI planning or simply planning) is a branch of AI that deals with sequential decision-making problems in dynamic environments (Ghallab et al., 2004). A planning system, or planner, is an algorithm that receives a formal description of a task as input and constructs a *plan* – a sequence of actions – based on this description. We will soon define this idea formally. Automated planning has been applied in a variety of domains, including robotics (e.g., to enable robots to navigate and manipulate objects in their environment, allowing them to perform tasks such as assembling products); manufacturing (e.g., to optimize the use of resources in manufacturing processes and help reduce waste); transportation (e.g., to help reduce congestion and improve the efficiency of transportation systems); and computer games (e.g., to design intelligent game characters that can make strategic decisions and adapt to changing game environments).

Planning problems can often be formalized as *labeled transition systems*, where the labels represent actions that can be taken in a modeled environment. Moreover, some common assumptions can be made about the specific properties of a given transition system, which can lead to different classes of planning problems. One particular

class of planning problems is *classical planning* (also known as STRIPS[1] planning) which involves deterministic actions, a complete and fully observable environment, and conjunctive goals (informally, conjunctive goals consist of multiple subgoals that must all be achieved in order for the overall goal to be considered satisfied).

A common way to represent classical planning problems (to which we appeal here) is in set-theoretic terms. In particular, using such a representation a state is defined as a set of *fluent atoms* (propositions whose truth value can change over time) that are true in that state. Moreover, following the *closed world assumption* (Reiter, 1981; Lifschitz, 1985), fluent atoms that do not appear in the state are assumed to be false. States can therefore be thought of as databases of facts that currently hold true in the world and actions are then operations on these databases, removing and adding facts from and to the database.

To align with the flavor of epistemic planning to which we appeal later on in the dissertation, here we are interested in classical$^+$ planning problems which are classical planning problems augmented with ADL (Pednault, 1989) features, most notably here *conditional effects of actions* and *disjunctive goals*. Conditional effects describe the way that an action can change the state of the system under certain conditions and are used to describe the effects of an action that are conditional on the truth values of certain fluent atoms. For example, consider the action 'pick up block' that is intended to pick up a block from a table and place it in the robot's hand. The action might have the following (informally described) conditional effects:

- if the fluent atom 'block on table' is true and the fluent atom 'robot holding block' is false, then set the fluent atom 'block on table' to false and the fluent atom 'robot holding block' to true
- if the fluent atom 'block on table' is false, then do not change the truth values of any fluent atoms
- if the fluent atom 'robot holding block' is true, then do not change the truth values of any fluent atoms

Conditional effects are a useful tool for representing the effects of actions that have different outcomes depending on the context in which they are performed.

Next, a disjunctive goal is comprised of individual goal conditions and is considered to be achieved if at least one of the individual goal conditions is achieved. For example,

[1]The Stanford Research Institute Problem Solver (STRIPS) was a planning system developed by Fikes & Nilsson (1971). Its input language, also known as STRIPS, is frequently used to describe classical planning tasks.

consider a planning problem in which the goal is to either go to the movies or go to the zoo. The disjunctive goal, in this case, might be (informally) represented as 'go to movies OR go to zoo'. This goal can be achieved by either going to the movies or going to the zoo. With that, we move to formally define classical$^+$ planning problems by adapting Muise et al.'s (2021) definition, which includes conditional effects, to also include disjunctive goals.

Definition 2.1 (Classical$^+$ Planning Problem (building on Muise et al. (2021))**.** *A* ***classical$^+$ planning problem*** *consists of a tuple $\langle \mathcal{F}, \mathcal{I}, G, O \rangle$ where $\mathcal{F}$ is a set of fluent atoms, $\mathcal{I}$ is the initial state, G is the goal, and O is a set of operators. A complete state s is a subset of $\mathcal{F}$ with the interpretation that fluent atoms not in s are false (equivalently, this can be seen as a conjunction between the fluent atoms found in s and the negated fluent atoms not found in s). A partial state s is similarly a subset of $\mathcal{F}$, but has the interpretation that fluent atoms not found in s can take on any value (thus equivalent to a conjunction of just the fluent atoms in s). $\mathcal{I}$ is a complete state while G is a disjunction of partial states (possibly a single disjunct, in which case a goal is simply a partial state, as in a 'regular' classical planning problem). Every operator $o \in O$ is a tuple $\langle Pre_o, eff_o^+, eff_o^- \rangle$, and we say that o is applicable in the state s iff $Pre_o \subseteq s$. The set eff_o^+ (resp. eff_o^-) contains conditional effects describing the fluent atoms that should be added (resp. removed) from the state when applying the operator. Finally, every conditional effect in eff_o^+ or eff_o^- is of the form $(C \rightarrow l)$ where C is the condition for the effect and l is a fluent that is the result of the effect. The condition C consists of a tuple $\langle C^+, C^- \rangle$ where C^+ is the set of fluent atoms that must hold in s and C^- the set of fluent atoms that must not hold in s. A conditional effect $(\langle C^+, C^- \rangle \rightarrow l)$ fires in state s if $C^+ \subseteq s$ and $C^- \cap s = \emptyset$. Assuming o is applicable in s, and $eff_o^+(s)$ (respectively, $eff_o^-(s)$) are the positive (resp. negative) conditional effects that fire in state s, the state of the world s' after applying o is defined as follows:*

$$s' = s \setminus \{l \mid (C \rightarrow l) \in eff_o^-(s)\} \cup \{l \mid (C \rightarrow l) \in eff_o^+(s)\}$$

Given a classical$^+$ planning problem $\langle \mathcal{F}, \mathcal{I}, G, O \rangle$, a solution is a sequence of operators (a plan) $\pi = a_1 \dots a_n$, $a_i \in O$, where π is applicable in $\mathcal{I}$ (i.e., a_1 is applicable in $\mathcal{I}$, a_2 is applicable in the state of the world after applying a_1 and so on) and there exists a disjunct G_i of G (where G_i is a partial state) such that $G_i \subseteq s_\pi$ where s_π is the state of the world after sequentially applying the operators in π. In this dissertation, we assume that the cost of a plan is equivalent to its length (i.e., the

number of actions in a plan), and optimal plans are therefore the shortest plans. In addition, we say that a classical$^+$ planning domain is a tuple, $\langle \mathcal{F}, O \rangle$, comprising a set of fluent atoms, $\mathcal{F}$, and a set of operators, O. Moreover, in automated planning, actions and operators have historically been two distinct concepts, both used to represent the ways in which a system can change or affect its environment. However, in this dissertation, we use the terms slightly loosely. In particular, we use operators in the context of classical$^+$ planning and actions in the context of epistemic planning. Finally, Appendix B.1.1 provides an example of a classical$^+$ planning problem in the context of the discrepancy resolution approach presented in Chapter 5.

2.2 Epistemic Logic

As discussed in Chapter 1, we are interested in augmenting AI systems with Theory of Mind. Paramount to an agent's ability to use its Theory of Mind, is its ability to represent and reason about the mental states of other agents, including their beliefs and knowledge about their environment and about other agents. In this dissertation we appeal to *multi-agent epistemic logic* (Fagin et al., 1995) to facilitate such representation and reasoning. Seminal work on epistemic logic was done by Von Wright (1951), with Hintikka (1965) later helping epistemic logic gain recognition in the logic and philosophy communities by proposing the possible world semantics, which will be discussed shortly. Importantly, multi-agent epistemic logic allows for reasoning about nested belief and knowledge in environments comprising multiple agents. Such environments are of particular interest to us in this dissertation.

Epistemic logics most commonly take the form of *modal logics* that are used to formalize reasoning about knowledge and belief. In modal logic, modalities are operators that act on propositions and change the way they are evaluated or interpreted. In the case of epistemic logic, the modalities K and B change the way propositions are evaluated based on whether they are *known* or *believed* to be true. While there is no one accepted definition for an agent's knowing some proposition, some common interpretations of knowledge are for the proposition to be true and to be believed by the agent (Fagin et al., 1995); while others require the agent to also have a good reason to believe the proposition (i.e., justified true belief) (Steup, 2007). Belief, on the other hand, is less demanding and importantly allows for agents to believe things which are not true in the world. As mentioned in Chapter 1, a hallmark of Theory of Mind reasoning is the ability to attribute to other agents false beliefs. As such, we are interested here in modeling belief and therefore appeal to the belief modality

B as discussed below. We note also that while 'epistemic' typically refers to knowledge while 'doxastic' typically refers to belief, in this dissertation we use the term 'epistemic' slightly loosely and informally to refer to both.

In what follows, we formalize the multi-agent modal logic to which we appeal in this dissertation. Let Ag be a finite set of agents and $\mathcal{P}$ be a finite set of fluent atoms. ϕ and ψ are used to represent formulae. $\top$ and $\bot$ represent *true* and *false*, respectively. The language $\mathcal{L}$ of multi-agent modal logic is generated by the following BNF:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi' \mid B_i\phi$$

where $p \in \mathcal{P}$, $i \in Ag$, $\phi \in \mathcal{L}$ and $B_i\phi$ means that *"agent i believes ϕ."* For instance, let the fluent atom *cold* be part of the set of fluent atoms $\mathcal{P}$. The formula *cold* would then mean *"it is cold"*. Moreover, B_i*cold* would mean that *"agent i believes that it is cold."* However, this *multi-agent* framework allows us to go beyond the beliefs of a single agent. In particular, we can express the beliefs of agent i about the beliefs of agent j about whether or not it is cold – B_iB_j*cold*. Such nesting can be arbitrarily deep where, for example, agent i believes that agent j believes that agent k ... believes that it is cold – $B_iB_jB_k \ldots$ *cold*. Lastly, agents' beliefs may disagree. For instance, Bob may believe that it is not cold while Mary may believe that it is cold while also believing that Bob believes that it is not cold – $B_{\text{Bob}}\neg$*cold* $\wedge$ B_{Mary}*cold* $\wedge$ $B_{\text{Mary}}B_{\text{Bob}}\neg$*cold*. Throughout this dissertation, we will see how the richness of this framework allows us to model a myriad of interesting settings involving Theory of Mind reasoning.

The semantics for formulae in $\mathcal{L}$ is given by Kripke structures (Fagin et al., 1995) which are triplets, $M = \langle W, R, V\rangle$, containing a set of worlds, accessibility relations between the worlds for each of the agents ($R = \{R_i \mid i \in Ag\}$), and a valuation map, $V\colon W \rightarrow 2^{\mathcal{P}}$. The set of worlds an agent i (at world $w \in W$) considers possible is given by M and the accessibility relations in R_i pertaining to w. R_i is a binary relation on W and is a subset of $W \times W$. A formula ϕ is true in a world w of a Kripke structure $M = \langle W, R, V\rangle$, written $M, w \models \phi$, under these conditions:

$M, w \models p$ for a fluent atom p, *iff* $p \in V(w)$,
$M, w \models \neg\phi$, *iff* $M, w \not\models \phi$,
$M, w \models \phi \wedge \psi$, *iff* both $M, w \models \phi$ and $M, w \models \psi$,
$M, w \models B_i\phi$, *iff* $M, w' \models \phi \quad \forall w' \in W$ s.t. $R_i(w, w')$

Intuitively, agent i is said to believe ϕ iff ϕ holds in all words the agent considers possible. Moreover, ϕ is satisfiable if there is a Kripke structure M and a world w of

M s.t. $M, w \models \phi$. ϕ is said to entail ψ, written $\phi \models \psi$, if for any Kripke structure M, $M, w \models \phi$ entails $M, w \models \psi$.

As discussed by Fagin et al. (1995), the constraints imposed on Kripke structures, particularly those related to the connections between possible worlds, play a crucial role in determining (1) whether knowledge or belief is being modeled and (2) the specific properties of knowledge or belief that are under consideration. Here we are interested in modeling belief and in particular we also wish to model a certain set of properties of belief. To this end, we assume a number of constraints on Kripke structures to achieve this (Fagin et al., 1995). Specifically, we assume that Kripke structures are:

Serial $(\forall w \, \exists v \, R(w\,, v))$,
Transitive $(R(w\,, v) \wedge R(v\,, u) \Rightarrow R(w\,, u))$, and
Euclidean $(R(w\,, v) \wedge R(w\,, u) \Rightarrow R(v\,, u))$,

with the resulting properties of belief:

K - Distribution $(B_i\phi \wedge B_i(\phi \Rightarrow \psi) \Rightarrow B_i\psi)$
D - Consistency $(B_i\phi \Rightarrow \neg B_i \neg \phi)$
4 - Positive Introspection $(B_i\phi \Rightarrow B_i B_i \phi)$
5 - Negative Introspection $(\neg B_i\phi \Rightarrow B_i \neg B_i \phi)$

This is the KD45$_n$ system (n is the number of agents in the environment) that is defined by these properties of belief. Importantly, these properties of belief dictate what inferences agents can make. For example, the consistency axiom (D) means that if an agent believes ϕ, then it necessarily does not believe the negation of ϕ. The epistemic planner we use in this dissertation enforces these properties which allows us to model interesting settings with multiple agents performing epistemic reasoning.

2.3 Epistemic Planning

Armed with knowledge of automated planning and epistemic logic, we can turn our attention to the marriage of these two, namely *epistemic planning* (Sadek et al., 1997; Bacchus & Petrick, 1998; Petrick & Bacchus, 2002; Van Der Hoek & Wooldridge, 2002; Bolander & Andersen, 2011; Kominis & Geffner, 2015; Muise et al., 2015b; Baral et al., 2017; Engesser et al., 2017; Huang et al., 2017; Kominis & Geffner, 2017; Liu & Liu, 2018; Le et al., 2018; Bolander et al., 2018; Petrick et al., 2019; Fabiano et al., 2020; Engesser & Miller, 2020; Buckingham et al., 2020b; Fabiano et al., 2021; Liberman

et al., 2020; Singh & Khemani, 2020; Belardinelli & Rendsvig, 2021; Singh & Khemani, 2021; Wan et al., 2021; Muise et al., 2021; Izmirlioglu et al., 2022; Cooper et al., 2021; Hu et al., 2022; Belle et al., 2022; Burigana et al., 2022; Burigana & Fabiano, 2022; Buckingham, 2023; Wu et al., 2023b; Hu et al., 2023). As with automated planning, epistemic planning is also concerned with constructing a goal-achieving plan, but does so in the context of agents' belief or knowledge about the state of the world. Moreover, whereas the goals being pursued by the agent in automated planning are *ontic* – related to changing the state of the world – in epistemic planning goals can also be *epistemic*, where agents strive to change their own (or another agent's) state of belief or knowledge.

As discussed by Bolander (2017), most works in epistemic planning can be placed in two categories: *syntactic* (e.g., Muise et al., 2015b; Huang et al., 2017) or *semantic* (e.g., Bolander & Andersen, 2011; Le et al., 2018), with the latter comprised predominantly of works based on Dynamic Epistemic Logic (DEL). The syntactic approach represents states as knowledge bases (KBs) which are sets of logical formulae, while the semantic approach represents states as semantical objects (Kripke structures), and manipulates them directly. Bolander (2017) offers a 'gentle' introduction to the latter thread of work. In this dissertation we appeal to a syntactic approach to epistemic planning (Muise et al., 2021) that, by making syntactical restrictions on formulae, is computationally attractive while still affording impressive expressivity, lending itself to a variety of scenarios involving complex Theory of Mind reasoning.

One of the seminal works in epistemic planning (that also belongs to the syntactic camp) was done by Petrick & Bacchus (1998; 2002). Petrick & Bacchus propose a knowledge-based approach to planning with incomplete information and sensing, realized by the single-agent epistemic planner PKS (Planning with Knowledge and Sensing). PKS's knowledge state is represented by five different knowledge bases, each modeling a specific type of knowledge (e.g., a *know whether* knowledge base and a knowledge base encoding the system's state of knowledge about the *value* of various 'variables'). Moreover, PKS models STRIPS-style actions that modify the different knowledge bases, and follows a syntactic approach to epistemic planning in that it manipulates the formulae in its different knowledge bases. The system uses a forward chaining approach to find plans. The approach is efficient as a result of the restricted expressiveness of the knowledge that can be represented. PKS has been applied to web service composition where the system's actions are driven by its knowledge (e.g., whether or not there exists a flight from Toronto to Calgary on Tuesday) (Martínez & Lespérance, 2005). Recently, PKS has been applied to social

human robot interaction settings (e.g., a robotic bartender (Petrick & Foster, 2020) and a social robot for healthcare (Foster & Petrick, 2020; Foster et al., 2020)).

Further, Huang et al. (2017) propose a syntactic epistemic planning approach and a corresponding epistemic planner (MEPK). Specifically, the knowledge base in Huang et al.'s work comprises formulae in a normal form called Alternating Cover Disjunctive Formula (ACDF). Huang et al. propose an algorithm that syntactically manipulates these ACDFs (for belief revision and belief update) and also produces formulae in the normal form. Huang et al.'s epistemic planner is highly expressive and can encode disjunctive belief and perform reasoning in the KD45_n belief system discussed in the previous section.

As mentioned, Muise et al.'s (2015b; 2021) approach to epistemic planning (RP-MEP) is also syntactic, however unlike PKS and MEPK it leverages classical planning tools, as will be discussed in Section 2.3.1. In the next section we elaborate on our chosen epistemic planning formalism and the reasons for choosing it in our investigations of Theory of Mind reasoning.

2.3.1 RP-MEP

As mentioned earlier, epistemic planning (like automated planning) is concerned with constructing a goal-achieving plan, but does so in the context of agents' belief or knowledge about the state of the world, and where goals can be epistemic. Here we are moreover interested in modeling settings with *multiple* agents and therefore appeal to multi-agent epistemic planning (MEP).

Definition 2.2 (MEP Problem)**.** *A* **Multi-agent Epistemic Planning Problem** *is a tuple* $\langle Q, \mathcal{I}, G\rangle$ *where* $Q = \langle \mathcal{P}, \mathcal{A}, Ag\rangle$ *is the* ***domain*** *comprising a set of fluent atoms* $\mathcal{P}$*, a set of actions* $\mathcal{A}$*, and a set of agents* Ag*, together with the problem instance description comprising the initial state,* $\mathcal{I} \in \mathcal{L}$*, and the goal condition* $G \in \mathcal{L}$*, where* $\mathcal{L}$ *is the language of multi-agent modal logic corresponding to* $\mathcal{P}$ *and* Ag*, presented in Section 2.2.*

$\mathcal{A}$ is a set of actions where each action $a \in \mathcal{A}$ is a tuple $\langle \text{Pre}, \{(\gamma_1, \epsilon_1), ..., (\gamma_k, \epsilon_k)\}\rangle$, where $\text{Pre} \in \mathcal{L}$ is the precondition of a (written $\text{Pre}(a)$), $\gamma_i \in \mathcal{L}$ is the condition of a conditional effect, and $\epsilon_i \in \mathcal{L}$ is the effect of a conditional effect.

The general plan existence problem in MEP is undecidable (Bolander & Andersen, 2011; Bolander et al., 2020). To circumvent this, a number of decidable and expressive fragments of epistemic planning have been investigated. Here we turn to one such fragment and a special case of MEP, namely ***R****estricted* ***P****erspectival-****MEP*** (RP-MEP

(Muise et al., 2015b, 2021)). In particular, (1) RP-MEP operates over syntactically *restricted* formulae, (2) reasoning in RP-MEP is done from the *perspective* of a single agent (the root agent, $\star \in Ag$), and (3) an upper bound is set on the depth of nested belief. We will expand upon these points shortly.

In this dissertation, we appeal to the RP-MEP setting (as well as make use of the RP-MEP planner, which will be discussed later on) for a number of reasons. First, in the previous chapter we noted that one of the challenges of augmenting AI systems with Theory of Mind reasoning is reasoning about nested mental states. The RP-MEP formalism circumvents this challenge by setting the aforementioned upper bound on nested belief. Moreover, we will see later on in this dissertation that many interesting use cases of Theory of Mind reasoning only require reasoning about shallowly nested belief, in which case the upper bound of nested belief can remain low which correspondingly keeps the planner's runtime low. Second, to compute solutions for epistemic planning problems, RP-MEP encodes an RP-MEP problem as a classical$^+$ planning problem and augments actions in the domain with conditional effects that enforce KD45$_n$'s induced properties of belief. We define this classical encoding later on in this section. By leveraging this compilation, RP-MEP (and, by extension, our developed computational solutions in Chapters 4–6) can benefit from advances in classical planning research without any changes to RP-MEP's inner workings. Third, RP-MEP supports reasoning over the KD45$_n$ belief system which allows us to model complex scenarios involving multiple agents and false belief. As discussed, one hallmark of Theory of Mind is the ability to attribute false beliefs to others and, as will be demonstrated by this dissertation, it is crucial for AI systems to perform such reasoning in many real-world social settings.

PEKBs, RMLs, and the RP-MEP Definition

The syntactically restricted objects over which RP-MEP operates are called Proper Epistemic Knowledge Bases (PEKBs) (Lakemeyer & Lespérance (2012) building on Liu et al. (2004) and Levesque (1998) with further work by Muise et al. (2015a)). *A PEKB is a set of restricted formulae, called restricted modal literals (RMLs)* (Lakemeyer & Lespérance, 2012). An RML is obtained from the following grammar:

$$\phi ::= p \mid B_i\phi \mid \neg\phi$$

where $p \in \mathcal{P}$ and $i \in Ag$. $\mathcal{P}$ is a set of fluent atoms and Ag is a set of agents, as defined in Section 2.2. The maximum number of nested belief modalities determines

the depth of an RML. For instance, $B_i\phi$ has a depth of 1 in addition to the depth of ϕ, where ϕ may hold additional belief modalities. Formally, the depth of an RML is defined by Muise et al. as: $depth(p) = 0$ for $p \in \mathcal{P}$, $depth(\neg\phi) = depth(\phi)$ and $depth(B_i\phi) = 1 + depth(\phi)$. In addition, a conjunction of RMLs is defined as a set of RMLs, where the set of all RMLs with bounded depth d for a group of agents Ag is denoted as $\mathcal{L}_{RML}^{Ag,d}$. In this way, RP-MEP sets an upper bound on the depth of nested belief to facilitate efficient computation.

Lastly, the state of the system is represented by some set of RMLs, i.e., a PEKB. As mentioned, reasoning in RP-MEP is done from the perspective of a single agent which we refer to as the root agent. Therefore, the PEKB state of the system represents the mental model of the root agent that perceives an environment that includes all other agents. Importantly, as noted by Muise et al. (2021): *"All reasoning is from the perspective of this single agent. The fluents that are true in a state correspond to the RMLs that the [root] agent believes, while the fluents that are false correspond to the RMLs that the agent does not believe. Action execution, then, is predicated on the [root] agent believing that the preconditions are satisfied. Similarly, the mental model of the [root] agent is updated according to the effects of an action."* Recall one of the challenges for augmenting AI systems with Theory of Mind reasoning, discussed in Chapter 1, namely representing the mental states of agents. RP-MEP's solution is to leverage PEKBs to represent agents' mental states. As will be discussed later on in the dissertation, the reduced expressivity of PEKBs (that affords computational benefits) is restrictive at times (e.g., they do not allow for disjunctive belief). However, RP-MEP still allows us to model a wide array of interesting social settings involving the (nested) mental states of multiple agents.

We can now formally specify an RP-MEP problem, a special case of Definition 2.2, based on Muise et al. (2021). First, recall the set of actions $\mathcal{A}$ in Definition 2.2. We modify that definition slightly and say that an RP-MEP action is a tuple $\langle \text{Pre}, \{(\gamma_1, \epsilon_1), ..., (\gamma_k, \epsilon_k)\}\rangle$, where Pre and each γ_i are PEKBs (sets of RMLs), and each ϵ_i is a single RML.

Definition 2.3 (RP-MEP Problem)**.** *For depth bound d and the root agent $\star \in Ag$, an* **RP-MEP Problem** *is a tuple $\langle Q, \mathcal{I}, G\rangle$ where $Q = \langle \mathcal{P}, \mathcal{A}, Ag\rangle$ is the* ***RP-MEP domain*** *comprising a set of fluent atoms $\mathcal{P}$, a set of RP-MEP actions $\mathcal{A}$ (henceforth actions), and a set of agents Ag, together with the problem instance description comprising the PEKB (a set of RMLs) representing the initial state, $\mathcal{I}$, and the goal condition G, also a PEKB. Moreover, every RML is from the perspective of $\star \in Ag$,*

i.e., from the following set:

$$\{B_\star \phi \mid \phi \in \mathcal{L}^{Ag,d}_{RML}\}$$

For instance, the initial state $\mathcal{I}$ may contain the RML $B_\star B_j \mathit{cold}$ which means that the root agent believes that agent j belives that it is cold. For readability, definitions in Chapters 4–6 that make use of Definition 2.3 do not include the depth d which is assumed to be part of the input. Moreover, we use $\mathcal{I}$ to denote an initial PEKB state and S to denote an arbitrary PEKB state. Moreover, when talking about a tuple $\vec{v} = \langle v_1, \ldots, v_n \rangle$ of agents from Ag, we may use $B_{v_1 \ldots v_n}$ or $B_{\vec{v}}$ to stand for the belief operator sequence $B_{v_1} \ldots B_{v_n}$. In case $\vec{v}$ is empty, $B_{\vec{v}} p$ represents p. We use a tuple of agents rather than a set since we wish to be able to differentiate between, for example, $B_{v_1} B_{v_2} B_{v_3} p$ and $B_{v_1} B_{v_3} B_{v_2} p$, i.e., between 'agent v_1 believes that agent v_2 believes that agent v_3 believes p' and 'agent v_1 believes that agent v_3 believes that agent v_2 believes p'. Lastly, we will see Definition 2.3 used extensively in Chapters 4-6, which will serve to illustrate the theoretical concepts discussed in this chapter.

Action Progression in RP-MEP

Another challenge mentioned in the previous chapter is reasoning about how other agents assimilate information into their existing set of beliefs. RP-MEP addresses this by defining a *progression operator* that allows the planner to update the root agent's beliefs about the beliefs of other agents in the context of change and action. While there is no one accepted way to model how agents' beliefs change, RP-MEP's solution allows us to model complex settings and track the evolution of agents' beliefs from the perspective of the root agent. In Chapters 4-6, we make heavy use of this progression operator.

Progression is of course not specific to RP-MEP but rather an operation over a state representation (e.g., a database, a knowledge base as represented by a set of formulae), together with a transition system (perhaps represented as a set of STRIPS operators or a PDDL domain theory (McDermott et al., 1998)), that takes a state and an operator as input and returns the resulting state after the application of the operator (e.g., Lin & Reiter, 1997). It is an important concept in automated planning in that the information about the resulting state can be used to direct the planning algorithm's search for states that satisfy the specified goal condition.

Here we appeal to the progression operator defined by Muise et al. (2021) that, rather than progressing arbitrary formulae in $\mathcal{L}$, operates over PEKBs. Once again, operating over PEKBs affords appealing computational complexity. Muise et al.'s

definition builds on two important operators – *belief update* and *belief erasure*. These operators guide the process of modifying an agent's beliefs. This might involve adding new beliefs, revising existing beliefs, or eliminating beliefs that are no longer supported. In the context of Muise et al.'s progression operator, this means modifying the root agent's beliefs (about the world and the beliefs of other agents) based on the effects of actions. First, we follow Muise et al. (2021) and say that given some PEKB state S, $\overline{S}$ is the PEKB that contains the negation of every RML in S, i.e., $\overline{S} = \{\neg\phi \mid \phi \in S\}$.

Definition 2.4 (Belief Update and Erasure in PEKBs (Muise et al. 2022, Definition 3))**.** *Given PEKB states P and Q, we define $P \blacklozenge Q$ and $P \diamond Q$ as the belief erasure and belief update of P and Q respectively, as follows:*

$$P \blacklozenge Q = P \setminus Q$$
$$P \diamond Q = (P \blacklozenge \overline{Q}) \cup Q$$

Belief update of P with Q is defined as the standard 'forget (erase) then conjoin': anything in P disagreeing with Q is forgotten, then everything from Q is added.

The belief update and belief erasure operators for PEKBs presented in Definition 2.4 have been defined and shown to be polynomial time operations by Miller & Muise (2016). As mentioned, building on the definitions of these operators, Muise et al. (2021) define the progression operator, PROG, in the RP-MEP setting.

Definition 2.5 (PROG (Muise et al. 2022, Definition 4))**.** *Given a PEKB state ϕ and an action $a = \langle Pre, \{(\gamma_1, \epsilon_1), ..., (\gamma_k, \epsilon_k)\}\rangle$, the **progression** of ϕ with respect to action a, a PEKB state labelled $\text{PROG}(a, \phi)$, is*

$$\text{PROG}(a, \phi) = (\phi \blacklozenge (R \cup U)) \diamond Q$$
$$Q = \bigcup_{\substack{(\gamma_i, \epsilon_i) \\ 1 \le i \le k}} \{\psi \mid \gamma_i \subseteq \phi \text{ and } \epsilon_i \models \psi\}$$
$$R = \bigcup_{\substack{(\gamma_i, \neg\epsilon_i) \\ 1 \le i \le k}} \{\psi \mid \gamma_i \subseteq \phi \text{ and } \epsilon_i \models \psi\}$$
$$U = \bigcup_{\substack{(\gamma_i, \epsilon_i) \\ 1 \le i \le k}} \{\neg\psi \mid \overline{\gamma_i} \cap \phi = \emptyset \text{ and } \neg\epsilon_i \models \neg\psi\}$$

where $\blacklozenge$ and $\diamond$ are the belief erasure and belief update operators presented in Defini-

tion 2.4. In case the action a is not executable in ϕ, i.e. $\phi \not\models Pre(a)$[2]*, $\text{PROG}(a, \phi)$ is undefined. Q defines the set of literals to be added, R defines the set of literals to be deleted, and U defines the set of uncertain firing literals to be deleted (we elaborate upon this shortly).*

We use the shorthand $\text{PROG}([a_1, ..., a_n], \phi)$ or $\text{PROG}(\pi, \phi)$ to denote the progression of ϕ with respect to a sequence of actions, a *plan*, $\pi = [a_1, ..., a_n]$. Using this shorthand, we say that a plan π solves an RP-MEP problem $\langle Q, \mathcal{I}, G \rangle$ *iff* $\text{PROG}(\pi, \mathcal{I}) \models G$.

Uncertain Firing

Uncertain firing occurs when an agent is unsure whether a conditional effect is true (denoted $\overline{\gamma_i} \cap \phi = \emptyset$) and should therefore not believe the effect but must also not believe the opposite. Concretely, in Muise et al.'s words: *"If an agent is unsure, then it should not believe the effect (unless it was already true), but must admit that it could be true. Therefore, it must not believe the opposite, and we should remove [the opposite of the effect] and anything that follows from it deductively."* For instance, consider the following example given by Muise et al. (2021): say that there is a conditional effect for the action of agent 1 sharing their secret that stipulates that if agent 2 believes that agent 1 is trustworthy, then agent 2 would believe agent 1's secret. In the context of uncertain firing, if agent 2 is unsure about agent 1's trustworthiness, agent 1 should not come to believe the secret is true, but it should also no longer believe that the secret is false. The use of uncertain firing in Definition 2.5 ensures that, following the execution of actions, agents' beliefs are modified while reflecting this intuition.

Conditioned Mutual Awareness

RP-MEP allows us to model another important facet of Theory of Mind reasoning, namely reasoning about whether or not other agents are 'aware' of changes occurring in the environment and, correspondingly, whether those agents' beliefs have changed based on this awareness. For example, agents may be 'aware' that an action has been performed if they are in the same location in which the action is performed. Recall the example given in Chapter 1. After Bob returns the coffee beans to a different cabinet, since the robot believes that Alice was not in the kitchen when Bob performed the action, it should reason that Alice was not 'aware' of Bob's action (and her beliefs were therefore not updated). To achieve such reasoning, Muise et al.

[2]Conversely, we say that an action a is executable in PEKB S *iff* $S \models \text{Pre}(a)$.

(2015b; 2021) propose a mechanism for *conditioned mutual awareness* that adds an awareness condition to actions in the RP-MEP domain.

The RP-MEP planner implements this mechanism by automatically generating additional conditional effects from an action's existing set of conditional effects to ensure that the awareness condition is satisfied (e.g., only agents that are in the same location in which the action is performed, are 'aware' of the action). Lastly, conditioned mutual awareness also handles higher-order Theory of Mind reasoning. For instance, in the example given in Chapter 1, after observing Bob moving the coffee beans to the other cabinet, the robot (via conditioned mutual awareness) will (informally) believe that Bob believes that Alice is not aware of the action that he performed. While agents cannot hold beliefs about awareness in the RP-MEP formalism, the previous sentence concretely means that the robot believes that Bob's beliefs about Alice's beliefs about the location of the coffee beans haven't changed following Bob's action. We will see additional concrete examples of conditioned mutual awareness in the coming chapters.

RP-MEP Classical Encoding

As mentioned, to compute solutions for RP-MEP problems, RP-MEP encodes an RP-MEP problem as a classical$^+$ planning problem. In what follows, we define this encoding, following Muise et al. (2021).

Definition 2.6 (Classical Encoding of an RP-MEP Problem (following Muise et al. (2021)))**.** *Let $\mathcal{B}_i$ and $\mathcal{N}_i$ be functions that map agent i's positive and, respectively, negative beliefs from a PEKB S to the respective fluents in the classical$^+$ planning domain:*

$$\mathcal{B}_i(S) = \{l_\phi \mid B_i\phi \in S\}$$

$$\mathcal{N}_i(S) = \{\neg l_\phi \mid \neg B_i\phi \in S\}$$

Given an RP-MEP problem, $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, G\rangle$, we define the propositional encoding as the tuple $\langle \mathcal{F}, \mathcal{I}', G', O\rangle$ such that:

$$\mathcal{F} \stackrel{def}{=} \{l_\phi \mid \phi \in \mathcal{L}^{Ag,d}_{RML}\}$$

$$\mathcal{I}' \stackrel{def}{=} \mathcal{B}_\star(\textsc{Closure}(\mathcal{I}))$$

$$G' \stackrel{def}{=} \mathcal{B}_\star(G)$$

where $\star \in Ag$ is the root agent. Closure *is a closure procedure that deduces a new set of RMLs from an existing one under the KD_n axioms (Muise et al., 2021, Definition 7). For every action $\langle Pre(a), \langle eff_a^+, eff_a^- \rangle\rangle$ in $\mathcal{A}$, we have a corresponding operator $\langle Pre_o, \langle eff_o^+, eff_o^- \rangle\rangle$ in O such that:*

$$Pre_o \stackrel{def}{=} \mathcal{B}_\star(Pre(a))$$

$$eff_o^+ \stackrel{def}{=} \{(\langle \mathcal{B}_\star(\gamma_i), \overline{\mathcal{N}_\star(\gamma_i)} \rangle \rightarrow l_\phi) \mid (\gamma_i, \mathcal{B}_\star\phi) \in eff_a^+\}$$

$$eff_o^- \stackrel{def}{=} \{(\langle \mathcal{B}_\star(\gamma_i), \overline{\mathcal{N}_\star(\gamma_i)} \rangle \rightarrow l_\phi) \mid (\gamma_i, \neg\mathcal{B}_\star\phi) \in eff_a^-\}$$

As mentioned, Muise et al. augment actions in the domain with conditional effects that enforce the properties of belief discussed in Section 2.2. In particular, they use the closure procedure described above to do so (Muise et al., 2021, Sections 4.2 & 4.3).

The following definitions will be used in Chapter 5, where we utilize RP-MEP's encoding to resolve discrepancies pertaining to the validity of agents' plans.

Definition 2.7. *Given a PEKB S, the classical encoding S' of S is*

$$S' \triangleq \textsc{Closure}(\mathcal{B}_\star(S) \cup \mathcal{N}_\star(S))$$

Definition 2.8 (Classical encoding function)**.** *Given an RP-MEP problem $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, G\rangle$, a root agent $\star \in Ag$, the corresponding set of all RMLs with bounded depth d, $\mathcal{L}_{RML}^{Ag,d}$, and an RML $\phi \in \mathcal{L}_{RML}^{Ag,d}$, we say that $\mathcal{C}(\phi) = \mathcal{B}_\star\phi$.*

Definition 2.9 (Classical decoding function)**.** *Given an RP-MEP problem $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, G\rangle$, a root agent $\star \in Ag$, the corresponding classical$^+$ planning problem per Definition 2.6, $\langle \mathcal{F}, \mathcal{I}', G', O\rangle$, the corresponding set of all RMLs with bounded depth d, $\mathcal{L}_{RML}^{Ag,d}$, and a fluent atom $l_\phi \in \mathcal{F}$, we say that $\mathcal{D}(l_\phi) = \phi$, $\phi \in \mathcal{L}_{RML}^{Ag,d}$.*

2.4 Concluding Remarks

In this chapter, we provided background on automated planning, epistemic logic, and their marriage – epistemic planning. We moreover provided an overview of the flavor of epistemic planning we have chosen to leverage in this dissertation – RP-MEP – and discussed why it was chosen. The background in this chapter will serve as the technical foundation for the work presented in Chapters 4–6.

Chapter 3

The Role of Theory of Mind in Explanation

> *"Explanations are social — they are a transfer of knowledge, presented as part of a conversation or interaction, and are thus presented relative to the explainer's beliefs about the [beliefs of the recipient of the explanation]"*
>
> – Miller (2019)[p. 3]

As Miller (2019) notes, explanation has been studied for millennia by philosophers, with the aim of establishing what constitutes an explanation and what the function and structure of explanations are (e.g., Peirce, 1878; Hempel & Oppenheim, 1948). More recently, social and cognitive psychologists focused on how humans explain the behavior of others and, more generally, how humans generate explanations and determine their quality (e.g., Hilton, 1990; Antaki & Leudar, 1992; Slugoski et al., 1993; Malle et al., 2000). Within AI, explanation has also been researched extensively with early work including a variety of logic-based and probabilistic approaches to abductive inference or so-called *inference to the best explanation* including the early works of Pople (1973), Charniak & McDermott (1985), Poole (1989), and Levesque (1989). The recent resurgence of interest in explanation within AI is largely in the guise of so-called *Explainable AI* (XAI), which is motivated by the need to provide human-interpretable explanations for decision making in black-box classification and decision-making systems based on machine and deep learning (e.g., Doshi-Velez & Kim, 2017; Samek et al., 2017; Gunning et al., 2019). We elaborate on the body of extant work in Section 3.6.

Drawing from their own experience, the reader would likely agree that explanations are often generated by an agent to explain to itself some observed phenomenon. For

instance, the problem of plan recognition (Kautz & Allen, 1986) is to abduce an observed agent's plan and goal (e.g., making a cup of coffee) given observations about the environment and the observed agent's behavior (e.g., the observed agent walked into the kitchen, turned on the kettle, and obtained a mug). In other words, an agent observing another agent's behavior generates an explanation – in this context, a hypothesized plan and goal the observed agent is likely pursuing – which explains the observed agent's behavior. We revisit this interesting problem in Chapter 4.

In Chapter 1, we posited that in order to effectively communicate with other agents, AI systems should be able to consider the unique *perspective* held by each agent in the environment. Therefore, of particular interest to us in this chapter is that explanations may also be *social*, given by one agent to another agent (or agents), requiring the former (the *explainer*) to use their Theory of Mind and consider the mental state of the latter (the *explainee*). We henceforth use explainer and explainee in reference to the provider and recipient of the explanation, and explanandum in reference to the thing to be explained. Consider the following narrative by way of illustration.

> *Mary, Bob, and Tom are housemates sharing a house. While Tom was away on a business trip, Mary and Bob noticed a hole in the roof of their house and called a handyman to fix it. Before the handyman could come, however, it rained during the night and the floor got wet. Bob, who sleeps in a windowless room, did not notice the rain. Tom, who just got back from his trip that day, noticed the rain but did not know about the hole in the roof. Mary saw Tom return to the house at night and so knew that Tom knew that it had rained. In the morning, when trying to explain the wet floor to Bob, Mary tells him that it had rained during the night and when explaining to Tom she tells him that she and Bob had discovered a hole in the roof (adding that the handyman will arrive the next day).*

Clearly, Mary tailored her explanations to each of her housemates, believing the information she was providing to them was sufficient to explain the wet floor in their respective mental states. Mary's ability to do this stems from her Theory of Mind. In humans, the use of Theory of Mind in explanation has been demonstrated empirically by Slugoski et al. (1993) via a set of experiments where human participants gave different explanations to different explainees, based on the beliefs of the explainers about the beliefs of the explainees. Of course, the participants' as well as Mary's explanations are only as good as their and her ability to model the mental states of

the explainee (in Mary's case, her housemates) and how they will alter their mental states in light of the explanation. Mary's beliefs about Bob and Tom's beliefs, or her belief about how each of them revises their beliefs, may well be wrong, in which case her explanations to them may fail to explain why the floor is wet. We investigate this matter further and discuss the notion of the adequacy of the explainer's beliefs about the explainee's beliefs later in this chapter.

Furthermore, in this chapter we build on the shoulders of previous scholarly work to investigate the role of Theory of Mind in explanation with a view to addressing the diverse needs of explanation in AI and XAI. In particular, we commence our investigation by articulating a number of desiderata for accounts of explanation that employ Theory of Mind. These desiderata, in turn, inform the design of a belief-based account of explanation that highlights the role of Theory of Mind in this context.

Main Contributions

- We identify a set of desiderata for explanations that utilize Theory of Mind.
- We present a belief-based account of explanation, whose design is informed by the aforementioned desiderata.
- We prove a number of theorems pertaining to various properties of our account of explanation and various explanation types therein.

Relationship to Published Work

This chapter builds upon our EXTRAAMAS 2020 publication (Shvo et al., 2020a), with an augmented discussion of the complexity of explanation generation and comprehension, as well as an enhanced related work section.

Chapter Structure

In Section 3.1 we identify a set of desiderata for explanations that utilize Theory of Mind. These desiderata inform a set of design choices for a belief-based account of explanation which we present in Section 3.2. In Section 3.3 we discuss the criteria by which the quality of an explanation captured by our account can be evaluated. In Section 3.4 we demonstrate how, in the absence of an explicit prompt to be explained, our account allows the explainer to simulate the explainee's mental state and identify discrepancies that warrant explanation. In Section 3.5 we show how our account allows for the modeling of the ignorance and misconceptions of an explainer

pertaining to the mental state of an explainee and how these may affect the quality of explanation. In Section 3.6 we survey related, extant work. Finally, we offer some concluding remarks in Section 3.7.

3.1 Desiderata for Theory of Mind in Explanation

We begin our investigation by reflecting on the key components that support an agent in imputing mental states to itself and others, reasoning about how the provision of new information is assimilated into an agent's existing set of beliefs, and the circumstances under which such information constitutes an explanation for the explainee. To this end, we identify a set of desiderata that inform our account of explanation in the sections to follow.

multi-agent: the account must be conceived in a multi-agent setting to support representation of the beliefs of one or more explainer and explainee. This is important because in many situations, as discussed previously, there are multiple agents involved who have different beliefs.

agent-type agnostic: the account must support a myriad of different agent types whose beliefs may be internally represented, inspectable, and revisable in diverse ways. For example, the agent's beliefs may be stored in a human brain or in, for instance, the parameters of a neural network or formulae in a knowledge base. Consequently, an account of explanation that supports this desideratum can better capture the role of Theory of Mind in explanation for a myriad of different types of agents.

belief based: the account must model the possibly false or simply incomplete beliefs of explainers and explainees. This is illustrated by our example involving Mary, Bob, and Tom, where each agent holds disparate beliefs about the hole in the roof, the overnight rain, and the wet floor.

reason about the beliefs of others: the account must allow an explainer to reason about the explainee's beliefs when providing the latter with an explanation since, due to their possibly differing beliefs, an explanation for the explainer may not be an explanation for the explainee. This is also illustrated by our example involving Mary, Bob, and Tom, where Mary, the explainer, does not herself require an explanation (since she has complete information in that par-

ticular situation) but can reason that her housemates (the explainees) require one, due to their disparate beliefs.

support belief revision: the account must enable the explainer to consider how an explanation is assimilated by the explainee, and in particular how the latter revises their beliefs given potential explanations which may be inconsistent with their current beliefs. Importantly, incorporating a belief revision operator in an account of explanation moves beyond the expansion of existing beliefs to include an explanation, as is the case in most logical treatments of explanation. These aforementioned treatments of explanation disallow explanations that are inconsistent with the explainee's current beliefs, as will be discussed in the next section.

explanations can refer to beliefs: the account must allow for explanations that themselves refer to beliefs. To illustrate why this is useful, consider that the explainer might explain their having not told the explainee the location of a party by saying that the explainer believed that the explainee knew the location.

While previous work has addressed some of these desiderata, in this chapter we propose a belief-based account of explanation in terms of epistemic states of agents that satisfies all of the aforementioned desiderata by employing a number of crucial building blocks relating to these desiderata. Lastly, while the list above may not be exhaustive, we nonetheless believe that it enumerates the salient facets of Theory of Mind reasoning required of accounts of explanation.

3.2 A Belief-Based Account of Explanation

In this chapter we appeal to logics of belief to provide a belief-based account of explanation in the context of Theory of Mind. Many logical accounts of explanation assume the existence of a knowledge base—a logical axiomatization of the domain in terms of a set of formulae (e.g., Brachman & Levesque, 2004). With such a knowledge base in hand, a popular logic-based characterization of explanation is in terms of abduction as follows.

Definition 3.1 (Abductive Explanation (after Poole (1990))). *Given a logical theory, T, and an explanandum O, E explains O given a theory T if $T \cup E \models O$ and $T \cup E$ is consistent.*

Here we make no such commitment to the representation of beliefs in terms of a set of logical formulae. Rather, in order to capture the diversity of human and machine explainers and explainees, our account finds its origins in works that attributed agents with mental states in the form of epistemic states (with seminal work by Gärdenfors (1988) and later notable work by Levesque (1989), Boutilier & Becher (1995), Chajewska & Halpern (1997), and Halpern & Pearl (2005)). In contrast, in Chapters 4–6, in order to develop epistemic planning-based computational solutions, we make commitments about agents' beliefs as well as the properties of those beliefs, that impact the inferences agents can make. In particular, as discussed in Chapter 2, we appeal to the multi-agent modal logic KD45$_n$ and make commitments pertaining to the representation of beliefs (i.e., agents' mental states are represented by PEKBs). In this chapter we raise the level of abstraction to present a general belief-based account of explanation, appropriate for a wide array of settings and agent types. For this reason, in what follows the exposition is similar but different to our exposition in Chapter 2.

3.2.1 Mental States as Epistemic States

We employ the notion of an epistemic state, e, or in the case of multiple agents, a collection of epistemic states, $\vec{e}$, to capture the beliefs of agents. These are used to provide the semantics for the language below.

We will suppose that we have a finite set of agents, $Ag = \{1, 2, \ldots, n\}$, and a set of propositional symbols $\mathcal{P}$. We define a language

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid B_i\varphi \mid [\varphi]_i\varphi \tag{3.1}$$

where $p \in \mathcal{P}$ and $i \in Ag$. We introduce $\bot$ as an abbreviation for $(p \wedge \neg p)$ for an arbitrary $p \in \mathcal{P}$. The intended meaning of $B_i\varphi$ is that agent i believes φ, and the intended meaning of $[\alpha]_i\varphi$ is that after agent i revises their beliefs by α, φ is true. We assume that our epistemic states are such that we can say that a formula φ is true in e when φ is believed.

To be clear, although we use formulae to describe what is believed, an epistemic state is not in general *defined* as a set of formulae, nor required to be represented internally as one. For a conventional example, e might be a set of possible worlds with accessibility relations and so on (akin to typical representations in modal logic, discussed in Section 2.2). However, we also allow for epistemic states to take very different forms. For example, one might want to model limited reasoning capabilities

in some manner to avoid the so-called problem of logical omniscience (Stalnaker, 1991), in which agents unrealistically believe all the deductive consequences of their beliefs. We might also wish for our epistemic states to be realized in a human brain or in terms of a computer program, such as a neural network.

Furthermore, we assume we have a *revision operator* $*$ so that $e * \alpha$ is another epistemic state, the result of revising by α. We will use * in defining the semantics for the $[\alpha]_i$ operator. Much as we have not committed to a particular structure for epistemic states, we will not commit to a particular revision operator. A large body of work has studied belief change in agents where belief revision typically concerns belief change in a static environment, possibly in the context of incorrect and partial beliefs. Amongst the most popular guidelines for belief revision are the AGM postulates (Alchourrón et al., 1985), and the DP postulates (Darwiche & Pearl, 1997) (for iterated revision). We will not require that our $*$ satisfies these properties except where noted. Similarly to the situation with our epistemic states, we might want our revision operator to be realized in terms of a computer program or human reasoning. As discussed in Chapter 1, one of the challenges of augmenting AI systems with Theory of Mind is reasoning about how other agents assimilate information into their existing set of beliefs. Indeed, while deciding what constitutes the 'best' way to model belief change is difficult, doing so in the context of multiple agents and in dynamic settings involving action and change adds layers of complexity and remains relatively under explored.

While epistemic states assign a truth value to any formula in our language – the language given by the grammar in (3.1) – that value indicates whether the formula is believed by the agent in question, not whether it is actually true. From an objective point of view, the formulae whose truth values we can determine are from the subset of the language consisting of formulae which are concerned only with beliefs. We define this subset of formulae below:

Definition 3.2 (Agent Formula)**.** *An agent formula is one in which no atomic symbol appears outside the scope of a belief operator, i.e., a formula ϕ of the form*

$$\phi ::= B_i\varphi \mid \neg\phi \mid (\phi \wedge \phi) \mid [\varphi]_i\phi \tag{3.2}$$

where φ is any (possibly non-agent) formula.

We assign truth values to agent formulae with a collection of epistemic states $\vec{e} = e_1, \ldots, e_n$ (corresponding to the different agents) according to the satisfaction relation $\models$ below.

- $\vec{e} \models B_i\varphi$ iff φ is true in e_i
- $\vec{e} \models \neg\phi$ iff $\vec{e} \not\models \phi$
- $\vec{e} \models (\phi \wedge \psi)$ iff $\vec{e} \models \phi$ and $\vec{e} \models \psi$
- $\vec{e} \models [\alpha]_i\phi$ iff $\langle e_1, \ldots, (e_i * \alpha), \ldots, e_n\rangle \models \phi$

Note that the semantics of the $[\alpha]_i$ operator is defined using the revision operator $*$.

Given this abstract framework for talking about beliefs, we can define explanations. The lack of commitment to the form of the epistemic state and revision operator is important because it affords us the ability to model a diversity of agents. In so doing, for the definitions of explanation that follow, the explainer will have beliefs about the other agents' beliefs and about their revision operators, and the effectiveness of the explainer's explanations for any particular agent will rely on the fidelity of those beliefs.

3.2.2 Characterizing Explanations

Definition 3.3 (Explanation)**.** *Given epistemic states $\vec{e}$, we say that α explains β for agent i if $\vec{e} \models [\alpha]_i(B_i\beta \wedge \neg B_i\bot)$.*

Notation: For notational convenience, we define $Expl(i, \alpha, \beta)$ as an abbreviation for $[\alpha]_i(B_i\beta \wedge \neg B_i\bot)$.

That is, α explains β if revising by α makes agent i believe β while still having consistent beliefs, i.e., agent i does not believe $(p \wedge \neg p)$ for an arbitrary $p \in \mathcal{P}$. If agent i is not logically omniscient, requiring i to not believe $\bot$ may not prevent i's beliefs from being inconsistent in some subtler way. For example, i might both believe p and believe $\neg p$, even though it does not believe $(p \wedge \neg p)$. Note that (with respect to revising by non-modal formulae) if revision of agent i's epistemic state satisfies the AGM postulates (see Appendix A for exposition), then the result of revision will be inconsistent only if either the agent initially had inconsistent beliefs, or if α itself is inconsistent.

Intuitively, our definition of explanation allows for more explanations than the traditional account in Definition 3.1. For one thing, we allow explanations to refer to modal operators and thus to agents' beliefs. Even without that, though, an important difference is that our definition is in terms of belief revision and so allows for an explanation that isn't consistent with the agent's initial beliefs. Our account

conceptually builds upon prior accounts of explanation defined relative to belief revision such as those of Boutilier & Becher (1995) and Nepomuceno-Fernández et al. (2017). Our account goes beyond this body of work by addressing a multi-agent setting, as well as focusing on agents using their Theory of Mind to provide explanations to other agents, rather than focusing on a single agent obtaining an explanation for itself. In Section 3.6 we elaborate on this point. To make the comparison to the traditional account of explanation more explicit, consider defining an epistemic state e_i as a propositional theory T, as in the following theorem.

Theorem 3.4. *Suppose that e_i is defined as being a propositional theory T, and that the formulae e_i makes true are defined to be the logical consequences of T (these are restricted to the non-modal subset of our language). Suppose furthermore that the revision operator $*$ on e_i satisfies the AGM postulates with respect to non-modal formulae. Then for non-modal formulae α and β, $\vec{e} \models Expl(i, \alpha, \beta)$ if $T \cup \{\alpha\}$ is consistent and $T \cup \{\alpha\} \models \beta$.*

Proof. Because $T \cup \{\alpha\}$ is consistent, by the AGM "vacuity" postulate, $T * \alpha$ is equal to the expansion of T by α, that is, the closure of $T \cup \{\alpha\}$. Therefore, $T * \alpha \models \beta$. □

However, we may also get further explanations. In the circumstances described by Theorem 3.4, if $T \cup \{\beta\}$ is inconsistent, then Definition 3.1 would say there are no explanations of β given the theory T, while there may be formulae that an agent with epistemic state T can revise by that would make them believe β.

It is also possible to talk in the language about agents' beliefs about $Expl(i, \alpha, \beta)$, i.e., about whether α explains β for agent i.

Definition 3.5 (Subjective Explanation)**.** *Given epistemic states $\vec{e}$, we say that α explains β for agent j from agent i's perspective, if $\vec{e} \models B_i Expl(j, \alpha, \beta)$.*

Note that when we say that something is from the perspective of agent i, we mean that that something is in accordance with agent i's beliefs. We invoke the term 'perspective' both for readability as well as to align with the motivation discussed in Chapter 1.

Example 3.6. *We formalize our example from the beginning of this chapter. We assume that Mary, Bob, and Tom all believe (and believe that the other agents believe)* $\text{Rain} \wedge \text{HoleInRoof} \rightarrow \text{WetFloor}$. *Furthermore,*

$$Ag = \{\text{Mary, Bob, Tom}\} \tag{3.3}$$

$$\vec{e} \models B_{\text{Mary}}\text{WetFloor} \wedge B_{\text{Mary}}\text{HoleInRoof} \wedge B_{\text{Mary}}\text{Rain} \tag{3.4}$$

$$\vec{e} \models B_{\text{Mary}}B_{\text{Bob}}\neg\text{WetFloor} \wedge B_{\text{Mary}}B_{\text{Bob}}\neg\text{Rain} \wedge B_{\text{Mary}}B_{\text{Bob}}\text{HoleInRoof} \tag{3.5}$$

$$\vec{e} \models B_{\text{Mary}}B_{\text{Tom}}\neg\text{WetFloor} \wedge B_{\text{Mary}}B_{\text{Tom}}rain \wedge B_{\text{Mary}}B_{\text{Tom}}\neg\text{HoleInRoof} \tag{3.6}$$

$$\vec{e} \models B_{\text{Mary}}Expl(\text{Bob}, \text{Rain}, \text{WetFloor}) \tag{3.7}$$

$$\vec{e} \models B_{\text{Mary}}Expl(\text{Tom}, \text{HoleInRoof}, \text{WetFloor}) \tag{3.8}$$

We also assume that the agents are able to draw at least simple inferences (and each knows that the others will) and their belief revision operators behave in a sensible way (and each knows that the others' operators do so).

For instance, entailments (3.7) and (3.8) mean that (per Definition 3.5) Rain explains WetFloor for Bob from Mary's perspective, and HoleInRoof explains WetFloor for Tom from Mary's perspective, respectively.

The $\approx$ Relation Between Epistemic States

We define a relation $\approx$ that can be understood intuitively as equating two epistemic states, e_i and e_j. For $e_i \approx e_j$ to hold, the internal structures of the states e_i and e_j need not be the same, but they must support the same beliefs as each other, and must continue to do so after any sequence of revisions. Formally, we say that $e_i \approx e_j$ if

- $\forall\varphi\ \vec{e} \models B_i\varphi$ *iff* $\vec{e} \models B_j\varphi$
- and for any sequence of formulae $\alpha_1, \ldots, \alpha_k$, we have that $\forall\varphi\ \vec{e} \models [\alpha_1]_i \cdots [\alpha_k]_i B_i\varphi$ iff $\vec{e} \models [\alpha_1]_j \cdots [\alpha_k]_j B_j\varphi$

Theorem 3.7. *Given epistemic states $\vec{e}$ and explanandum β, if $e_i \approx e_j$ it then follows that for all α, $\vec{e} \models Expl(i, \alpha, \beta)$ iff $\vec{e} \models Expl(j, \alpha, \beta)$.*

Proof. Note that $\vec{e} \models Expl(i, \alpha, \beta)$ iff $\vec{e} \models [\alpha]_i B_i\beta$ and $\vec{e} \models [\alpha]_i \neg B_i\bot$, and similarly for agent j. The result follows from the definition of $\approx$. □

That is, when $e_i \approx e_j$, an objective explanation for the former is also an objective explanation for the latter. Therefore, agent i, acting as the explainer, need not employ its Theory of Mind and reason about agent j's beliefs in order to generate explanations for the latter. However, the fact that $e_i \approx e_j$ does not mean that e_i holds accurate beliefs pertaining to how e_j revises its beliefs. Thus, while any α that

explains β may be an objective explanation for both agents i and j, agent i need not necessarily *believe* that α is an explanation for j. Nonetheless, $e_i \approx e_j$ is quite strong, as illustrated by the following theorem.

Theorem 3.8. *Suppose e_j supports positive and negative introspection – i.e., $\vec{e} \models (B_j\varphi \equiv B_jB_j\varphi) \wedge (\neg B_j\varphi \equiv B_j\neg B_j\varphi)$. Then if $e_i \approx e_j$, agent i will have correct beliefs about j's beliefs, i.e., $\vec{e} \models (B_j\varphi \equiv B_iB_j\varphi) \wedge (\neg B_j\varphi \equiv B_i\neg B_j\varphi)$.*

Proof. If agent j believes φ, then we'll have that $\vec{e} \models B_jB_j\varphi$ (by positive introspection) and then $\vec{e} \models B_iB_j\varphi$ (because $e_i \approx e_j$). Similarly, if agent j disbelieves φ, then $\vec{e} \models B_j\neg B_j\varphi$ (by negative introspection) and so $\vec{e} \models B_i\neg B_j\varphi$. □

Inconsistency Resolving Explanations

In some cases, an explanation need not cause the explanandum to be entailed by the epistemic state, but rather cause it to be *possible* in the epistemic state. This type of explanation is similar to Boutilier & Becher's (1995) *might explanation*.

Definition 3.9 (Inconsistency-resolving Explanation)**.** *Given epistemic states $\vec{e}$, we say that α explains the possibility of β for agent i if $\vec{e} \models [\alpha]_i\neg B_i\neg\beta$.*

This is a weaker form of explanation but important in various settings such as when an agent is attempting to find an explanation that will allow the behavior of another agent or in consistency-based diagnosis, where the agent is attempting to identify the abnormal components in a system that allow for the observed behavior of the system.

Proposition 3.10. *Given epistemic states $\vec{e}$ and explanandum β, then for all α, if $\vec{e} \models Expl(i, \alpha, \beta)$ it then follows that α is an inconsistency-resolving explanation for β for agent i, assuming that $\vec{e} \models [\alpha]_i\big((B_i\beta \wedge B_i\neg\beta) \rightarrow B_i\bot\big)$, i.e., that the agent can perform enough reasoning to notice the inconsistency in believing both β and $\neg\beta$.*

This follows straightforwardly from Definitions 3.3 and 3.9.

Explanations Involving Agent Beliefs

Importantly, an explainer can utilize its Theory of Mind to generate explanations pertaining to the mental states of other agents, such as their beliefs or goals.

Example 3.11. *Let us reconsider our example where this time, after Mary explains* WetFloor *to Bob, he asks her why Tom doesn't know* WetFloor*. That is, the explanandum β is $\neg B_{\text{Tom}}$WetFloor. A possible explanation is then $B_{\text{Tom}}\neg$HoleInRoof, assuming Bob believes B_{Tom}Rain.*

3.2.3 Explanations Involving Multiple Agents

An interesting setting that is straightforwardly captured by our framework is one in which an explainer (or explainers) is attempting to explain multiple (possibly disparate) explanandums to multiple explainees.

Definition 3.12. *Given epistemic states $\vec{e}$ and explanandums β_j, β_k, $\ldots\beta_l$, we say that α explains β_j, β_k, $\ldots\beta_l$ from agent i's perspective for agents j, k, $\ldots l$, respectively, if $\vec{e} \models B_i Expl(j, \alpha, \beta_j) \wedge B_i Expl(k, \alpha, \beta_k) \wedge \ldots \wedge B_i Expl(l, \alpha, \beta_l)$.*

Consider a collaborative card game (e.g., Hanabi (Bard et al., 2020)) where a certain player is attempting to make different players (each with a unique epistemic state) understand different things with a single piece of information about another player's cards, publicly announced to all players. The explaining player should therefore find an α that explains different explanandums for the different players, given the explaining player's beliefs about the other players' beliefs.

Example 3.13. *In a simpler setting such as our running example, if Mary is trying to explain* WetFloor *to Bob and Tom at the same time, the explanation α could be* Rain $\wedge$ HoleInRoof*, where the explanandum for both Bob and Tom is* WetFloor*.*

Privacy

Our framework can also capture a notion of privacy. For example, the explainer (agent i) may want to generate an explanation α that explains the explanandum β to some agents (agent j) but not to others (agent k):

$$\vec{e} \models B_i Expl(j, \alpha, \beta) \wedge B_i \neg(Expl(k, \alpha, \beta))$$

Example 3.14. *If Mary, for some reason, wants only Bob to entail* WetFloor*, the explanation α could be Rain in which case Bob will entail* WetFloor *but Tom will not. One can imagine parent #1 wanting to explain something to parent #2 such that their child does not understand.*

A significant amount of research has been conducted on privacy in AI (e.g., Such et al., 2014). While this is an intriguing topic, it is not the primary focus of this dissertation. However, the paradigms of epistemic planning and epistemic logic that we utilize in our work are well-suited for modeling privacy concepts (e.g., He & Liu, 2020).

Multiple Explainers and 'Nested' Explanations

In some cases, there may be multiple explainers trying to explain an explanandum β to an explainee. For example, agents i and j may want to find an α that explains β for agent k:

$$\vec{e} \models B_i Expl(k, \alpha, \beta) \wedge B_j Expl(k, \alpha, \beta)$$

Definition 3.12 can be straightforwardly extended to capture this setting. Finally, agent i may want to find an α that i believes that agent j believes is an explanation for agent k:

$$\vec{e} \models B_i B_j Expl(k, \alpha, \beta)$$

3.3 "Best" Explanations for Whom?

An explanadum can typically be explained by a variety of different explanations, but it is often the case that an agent *prefers* one explanation to another relative to some set of criteria. Indeed, there is a large body of previous work (e.g., Levesque, 1989; Lipton, 1990; Boutilier & Becher, 1995) that outlines criteria for defining preference orderings over explanations. In the context of multiple agents, we have seen that what constitutes an explanation for one agent, may not constitute an explanation for another. This observation extends to the notion of preferred explanations – what's good in the eyes of the explainer may not be good for the explainee, or for all explainees. Here we explore the issue of preferred explanations in the context of Theory of Mind.

For each agent in the set of agents Ag, we define a binary preference relation $\prec$ over explanations such that $\prec_i$ is the preference relation for agent i.

Definition 3.15 (Preferred Explanation)**.** *Given epistemic states $\vec{e}$ and explanandum β, if α and α' both explain β for agent i and $\alpha \preceq_i \alpha'$, we say that α is at least as*

preferred as α' for agent i. $\alpha \prec_i \alpha'$ denotes that α is strictly preferred to α' for agent i.

Similarly, we use $\alpha \preceq_{i,j} \alpha'$ to denote that agent i believes that α is at least as preferred as α' for agent j.

Definition 3.16 (Optimal Explanation)**.** *Given epistemic states $\vec{e}$ and explanandum β, α is an optimal explanation for β with respect to $\prec_i$ iff α explains β for agent i and there does not exist an explanation α' for β for agent i such that $\alpha' \prec_i \alpha$.*

Hilton (1990) posits that an explanation given by one agent to another is a form of conversation and should therefore adhere to Grice's (1975) maxims which he proposed as part of a model for effective cooperative conversation. In what follows, we discuss a number of criteria for preferred explanations and relate them to Grice's maxims.

Truthfulness

Grice's first maxim is the **quality** maxim, according to which one must not provide information (e.g., to the explainee) that she believes to be false.

Definition 3.17 (Subjectively Truthful Explanation)**.** *Given epistemic states $\vec{e}$ and an explanandum β, α is a subjectively truthful explanation for agent j from the perspective of agent i iff $\vec{e} \models B_i Expl(j, \alpha, \beta) \wedge B_i \alpha$.*

That is, agent i believes that α is an explanation for agent j, and moreover believes α.

Example 3.18. *In our example, Mary may tell Bob that Tom poured water all over the floor, thereby explaining* WetFloor*. However, since Mary does not believe that Tom did such a thing, it would not be a subjectively truthful explanation from Mary's perspective.*

Minimality

According to Grice's **quantity** and **relation** maxims, one must provide information that is relevant, sufficiently informative, and no more informative than needed. In a Theory of Mind context, the sufficiency of information is defined relative to the explainer's beliefs about the explainee's epistemic state and the explainer should therefore find the *minimal* explanation relative to the explainee's epistemic state. A large body of work concerned with explanation has discussed a minimality property which an explanation should satisfy (e.g., Reiter, 1987; Levesque, 1989; Hobbs

et al., 1993). For example, Levesque (1989) defines a syntactic simplicity relation between explanations wherein an explanation is *simpler* than another if it contains fewer propositional letters. Minimal explanations in the semantic sense may be defined relative to a set of possible explanations as those that are implied by all other explanations.

Plausibility

Grice's **quality** maxim also dictates that one should not provide information that is not supported by evidence. When applying this maxim to the beliefs of the explainee, an explainer may wish to consider how likely an explanation is from the point of view of the former. For instance, in our example it is more likely that Bob will accept the fact that it had rained last night as an explanation over the highly unlikely explanation according to which Alan Turing came to visit in the middle of the night and accidentally poured water all over the floor.

Therefore, the likelihood of an explanation is an important preference criterion when explaining to ourselves and to others. In the quantitative case, Pearl (2014) defines a *most probable explanation* while in a qualitative setting the *plausibility* of explanations may be defined where the most plausible explanations are those that require the 'least' change in the explainee's epistemic state (e.g., Quine & Ullian, 1978; Boutilier & Becher, 1995). One way by which to measure change in the epistemic state and the plausibility of an explanation in the qualitative case is via *belief entrenchment*. Intuitively, the more entrenched some belief is, the more an agent is 'reluctant' to revise it. Thus, the most plausible explanations may be defined as those that change less entrenched beliefs (Gärdenfors, 1988; Grove, 1988; Van der Hoek & Meyer, 1992; Boutilier & Becher, 1995).

Complexity

According to Grice's **manner** maxim, one should try to be as clear, as brief, and as orderly as one can in what one says, and avoid ambiguity. Inspired by this maxim, we posit that when generating an explanation for an agent, it is important to consider how *complex* it will be for that agent to understand the explanation, i.e., to incorporate the explanation into their existing knowledge base such that it explains the explanandum.

Approaching explanation with a computational lens, we are interested in the complexity of reasoning required by the explainee to properly (i.e., as intended by the explainer) understand the explanation. Here we consider two sources of complexity

– the first stems from the explainee having to reason about nested beliefs, while the second stems from the depth of reasoning and inferential capabilities required of the explainee.

Depth of Nested Belief. It has been empirically shown that it is difficult for humans to reason about deeply nested beliefs of other agents (e.g., Miller et al., 1970; Dunn, 1991; Camerer et al., 2004). Complexity concerns also exist in AI systems, where reasoning about nested beliefs can be computationally prohibitive. For instance, in the coming chapters we leverage the epistemic planning system RP-MEP (Muise et al., 2015b) whose runtime is significantly impacted by the depth of nested belief. Thus, a less complex explanation (and more preferred in some cases) may be one that contains fewer levels of nested belief.

Depth of Reasoning. Often, the recipient of an explanation, be they a human or an artificial agent, is not logically omniscient, i.e., knows all the logical consequences of their knowledge base. A large body of work has explored the use of logics which limit the 'depth' of the reasoning needed to arrive at some logical consequence (e.g., Konolige, 1985; Patel-Schneider, 1985; Levesque, 1984; Vardi, 1986; Fagin & Halpern, 1987; Lakemeyer, 1994; Delgrande, 1995; Kaplan & Schubert, 2000; Chen et al., 2018). One approach to formalizing the required depth of reasoning is to stratify beliefs into *belief levels*, where the first belief level only comprises elements of the knowledge base that are written down expressly (e.g., Liu et al., 2004; Lakemeyer & Levesque, 2013, 2014; Klassen et al., 2015; Schwering & Lakemeyer, 2016; Lakemeyer & Levesque, 2016). Higher belief levels draw conclusions based on conclusions drawn in the previous belief level.

An explainer with Theory of Mind capabilities should be able to reason about the depth of reasoning required of the explainee. Then, the complexity of an explanation may be defined, for example, with respect to the belief level in which the explainee can arrive at the logical consequences needed for the explanation to be properly assimilated. Reasoning about the complexity of an explanation is crucial when generating explanations for a diversity of agents. For instance, a more explicit explanation (i.e., one that requires few or no reasoning steps on the part of the explainee) may be desirable when the explainee is enduring significant stress (e.g., a time sensitive search and rescue setting) or living with various cognitive impairments.

3.4 Explainer-Explainee Discrepancies

To this point our account of explanation has assumed the existence of an explanandum, β, that is in need of explanation for a particular agent. However, in the absence of such a prompt, the explainer may use her Theory of Mind to put herself in the explainee's shoes, so to speak, and to identify *discrepancies* between the beliefs of the explainee and those of the explainer, or perhaps in the case of multiple agents, to identify discrepancies between the beliefs of two agents that the explainer can resolve via an explanation. For instance, in our example, while Mary believes that the floor is wet, she also believes that Bob believes that the floor is not wet. Discrepancies can also arise from inconsistencies between an agent's beliefs and observations in the world. Such discrepancies are common prompts for explanation in the case of diagnosis (e.g., Reiter, 1987; Boutilier & Becher, 1995).

Definition 3.19 (Discrepancy)**.** *Given epistemic states $\vec{e}$, β is a discrepancy between e_i and e_j iff $\vec{e} \models B_i\beta \wedge B_j\neg\beta$.*

That is, agent i believes β while agent j believes $\neg\beta$. The beliefs of agents pertaining to discrepancies can also be represented in our framework.

Definition 3.20 (Subjective Discrepancy)**.** *Given epistemic states $\vec{e}$, β is a discrepancy between e_i and e_j from the perspective of agent i iff $\vec{e} \models B_i(B_i\beta \wedge B_j\neg\beta)$.*

Example 3.21. *In our example, while Mary believes* WetFloor*, she believes that Bob believes that the floor is not wet (i.e., $\vec{e} \models B_{\text{Mary}}(B_{\text{Mary}}\text{WetFloor} \wedge B_{\text{Bob}}\neg\text{WetFloor})$). Thus,* WetFloor *is a discrepancy between Bob and Mary's respective epistemic states from Mary's perspective.*

Definition 3.22 (Subjective Discrepancy-resolving Explanation)**.** *Given epistemic states $\vec{e}$ and a discrepancy β between e_i and e_j from the perspective of agent i, we say that α is a discrepancy-resolving explanation for agent j for β from agent i's perspective if $\vec{e} \models B_i[\alpha]_j\neg B_j\neg\beta$.*

Example 3.23. *A discrepancy-resolving explanation for* WetFloor *for Bob from Mary's perspective is* Rain*.*

Note that Definition 3.22 appeals to the weaker inconsistency-resolving explanation defined in Definition 3.9. Thus, the explainer need not find an α that it believes will allow the explainee to entail the discrepancy. Rather, α should resolve the discrepancy by explaining its possibility.

We cast agent i as the explainer and agent j as the explainee, and distinguish between two types of subjective discrepancies: (1) where β is a discrepancy between e_i and e_j from the explainer's perspective; and (2) where β is a discrepancy between e_i and e_j from the explainee's perspective. In (1), as discussed, the explainer (e.g., Mary) may provide a discrepancy-resolving explanation for β (e.g., Rain). However, for (2), in order to provide such an explanation, the explainer must *believe* that the explainee believes that there exists a discrepancy between e_i and e_j. If the explainer's beliefs about the explainee's beliefs are incomplete or incorrect, the former may not recognize that such a discrepancy exists (we elaborate on this in Section 3.5).

In Chapter 5 we revisit the notion of a discrepancy in the context of the validity of agent plans. In particular, we propose a discrepancy resolution framework to enable agents to resolve discrepancies they perceive between their beliefs and other agents' beliefs pertaining to plan validity.

The Explainer as a Mediator

The definition of a subjective discrepancy (Definition 3.20) can be straightforwardly generalized to capture a setting where agent i believes that there exists a discrepancy between e_j and e_k:

$$\vec{e} \models B_i(B_j\beta \wedge B_k\neg\beta)$$

Agent i may also believe that agent j believes that α is an explanation for β for agent k, while also believing that α is not in fact a valid explanation for agent k due to the discrepancy between the epistemic states of agents j and k:

$$\vec{e} \models B_i(B_j Expl(k, \alpha, \beta) \wedge \neg Expl(k, \alpha, \beta))$$

Using Definition 3.12, agent i may explain the discrepancy to agents j and k.

Finally, the notion of discrepancy discussed here can be extended to encode other, possibly richer notions of discrepancy including the degree to which the epistemic states of two agents are discrepant. Relatedly, Krause & Vossen (2020) proposed a taxonomy of direct (e.g., commands or questions) and indirect (e.g., confusion or uncertainty) triggers of explanation. Here we leveraged our general account of explanation and proposed an abstract trigger defined by discrepancies between the mental states of the explainer and explainee. Richer notions of discrepancy could capture a diversity of explanation triggers (including various affective markers (Petrick

& Hill, 2019)).

3.5 The (In)Adequacy of the Explainer's Beliefs

The explainer is limited by the accuracy of its beliefs about the explainee's beliefs and reasoning capabilities. Specifically, the explainer's beliefs about the explainee's beliefs and reasoning capabilities must be accurate 'enough' – *adequate* – for the explainer to generate 'good' explanations with respect to the explainee.

Definition 3.24 (Adequacy)**.** *Given epistemic states $\vec{e}$ and explanandum β, we say that agent i's epistemic state e_i is adequate with respect to agent j iff for all α, $\vec{e} \models B_i Expl(j, \alpha, \beta)$ iff $\vec{e} \models Expl(j, \alpha, \beta)$.*

That is, if agent i's epistemic state is adequate with respect to agent j and β, then it can generate all explanations (for β) for agent j that are also explanations for agent j in its actual epistemic state, e_j.

Theorem 3.25. *Given epistemic states $\vec{e}$, explanandum β and $\preceq_{i,j}, \preceq_j$, agent i's perspective of agent j's preference relation and agent j's actual preference relation, respectively, if $\preceq_{i,j} = \preceq_j$ and e_i is adequate with respect to agent j and β, then for all α, α is an optimal explanation for agent j from agent i's perspective with respect to $\preceq_{i,j}$ iff α is an optimal explanation for agent j with respect to $\preceq_j$.*

That is, when e_i is adequate with respect to agent j and when agent i's beliefs about agent j's preference relation are correct, the optimal explanation for agent j from the perspective of agent i is also the optimal obejctive explanation for agent j. The proof follows straightforwardly from Definitions 3.16 and 3.24.

3.5.1 Sources of (In)Adequacy

Since most agents do not have a perfect image of another agent's mental state, an agent's beliefs about another agent may be inadequate for a myriad of reasons, including the inaccuracy of an agent's beliefs about the beliefs of other agents and about the way in which other agents revise their beliefs and perform entailment. In what follows, we focus on a setting where an agent holds inadequate beliefs about another agent's beliefs and illustrate using our running example.

Example 3.26. *Returning to our example, assume that Mary forgot that Bob found the hole with her and so she now falsely believes that Bob believes that there is no*

hole in the roof (i.e., $\vec{e} \models B_{\text{Mary}} B_{\text{Bob}} \neg \text{HoleInRoof}$*). Mary will therefore believe that* Rain $\wedge$ HoleInRoof *is the minimal explanation for Bob, relative to an intuitive measure of minimality. Notice, however, that in her explanation, Mary is conveying more information than is needed for Bob to entail* WetFloor*, thereby violating Grice's quantity maxim.*

Example 3.27. *Now consider that Mary falsely believes that Bob believes that it had rained and that there is no hole in the roof (perhaps she confused him with Tom!). Mary will therefore believe that* HoleInRoof *is an explanation for Bob. However,* $\vec{e} \not\models \mathit{Expl}(\text{Bob}, \text{HoleInRoof}, \text{WetFloor})$ *since Bob does not believe* Rain*. This time, Mary has violated the quantity maxim by not providing enough information for Bob to entail* WetFloor*.*

Example 3.28. *Mary now falsely believes that Bob believes* WetFloor *(i.e.,* $\vec{e} \models B_{\text{Mary}} B_{\text{Bob}} \text{WetFloor}$*) and so does not provide him with an explanation, believing he does not require one. In this case, while* WetFloor *is an objective discrepancy between Bob and Mary's epistemic states, it is not a discrepancy from Mary's perspective due to her false beliefs.*

Addressing Inadequacy

It is possible to mitigate for the inadequacy of the explainer's beliefs in a variety of ways. For example, it may be beneficial for the explainer to attempt to refine its beliefs about the beliefs of the explainee when explanations are not understood by the explainee. To this end, the explainer could try to gather additional pertinent information by acting in the world (e.g., querying the explainee). Additionally, Sreedharan et al. (2019) proposed a learning technique which enables an explainer to learn a simple model of an explainee and decide, based on the learned model, what information would constitute a good explanation. Lastly, Sreedharan et al. (2018) show how an explainer may generate explanations that are applicable to a set of possible explainee models which arise as the consequence of explainer uncertainty pertaining to the explainee's model.

While we emphasized the importance of the explainer modeling the beliefs of the explainee, our general account could in theory support the explainee, perhaps compensating for the explainer's inadequate beliefs, reasoning about the beliefs of the explainer to understand a given explanation that might otherwise be construed as inadequate. For example, consider Chandrasekaran et al.'s (2017) discussion of

a 'Theory of AI's Mind' where a human attempting to better understand a black-box decision making system can do so by familiarizing themselves with the system's capabilities, peculiarities, and shortcomings.

Finally, in Chapter 4 we revisit the notion of adequacy in the context of the interesting problem of plan recognition, where an observing agent attempts to infer an observed agent's plan and goal given observations. In particular, plan recognition systems are limited by the veracity and completeness of the observer's knowledge and beliefs and we discuss how an imperfect observer lacking complete observations may wrongly infer a plan or goal or fail to discriminate between a number of possible hypotheses.

3.6 Related Work

There is a rich literature on the topic of explanation. In this section, we cover closely related work. In particular, we discuss (1) existing work that highlighted the importance of tailoring an explanation to the mental model of the explainee; (2) accounts of explanation that utilized epistemic states or belief revision; and (3) related work in Explainable AI Planning (XAIP).

3.6.1 Giving an Explanation, Not a Soliloquy

Chakraborti et al. (2017) argue that AI systems should move beyond soliloquies that pertain only to their own model, and instead provide explanations that take into account the mental state of the explainee. Indeed, a large body of work has highlighted a similar desideratum for explanation. In the 80s and 90s, formal accounts of explanation such as those proposed by Gärdenfors (1988) and Chajewska & Halpern (1997) observed that an explanation for one agent may not serve as an explanation for another, and the explainer must therefore tailor an explanation to an explainee given the latter's beliefs. Within the space of user modeling and dialogue, and also set in the 80s and 90s, Weiner's (1980) BLAH system and Cawsey's (1991) EDGE system both tailor explanations to the presumed user model.

More recently, researchers have leveraged belief-desire-intention (BDI) architectures (Bratman, 1987) as a natural framework for explanations reflecting Theory of Mind. Such software architectures can enable an explainer to explicitly represent its own beliefs, desires, and intentions, as well as those of an explainee, and to relate explanations to its own beliefs and goals or those of the explainee (e.g., Harbers et al.,

2012; Kaptein et al., 2017).

Most recently, Westberg et al. (2019) have posited that incorporating various points of view on Theory of Mind from the cognitive sciences will facilitate the creation of agents better suited to communicate and explain themselves to the humans with whom they are interacting. Additionally, Miller (2019) has surveyed a substantial body of work and has also emphasized the importance of the explainer's ability to tailor an explanation to the explainee, using its understanding of the latter's mind. Moreover, Ehsan et al. (2021) have investigated the impact of an explainee's prior AI knowledge on how they perceive explanations given by an AI system. Their work exemplifies an additional dimension of an explainee's mental state and belief revision mechanism to which the explainer must be sensitive. Finally, Dazeley et al. (2021) define so-called levels of explanation for XAI: zero-order, first-order, second-order, and Nth-order explanations. The types of explanations captured by our belief-based account fall under their higher-order levels of (social) explanation. While Dazeley et al.'s work captures important insights that go beyond the scope of this chapter, we propose a formal and general belief-based account of explanation in terms of agents' epistemic states.

In much of this chapter we have been relating our Theory of Mind characterization of explanation in the context of English-like statements (e.g., Mary *telling* Bob that it had rained last night). However, if we turn to the broad endeavour of XAI that helped motivate the work in this chapter, we note that an explanation can take on many different forms other than human-interpretable language (e.g., a set of weights in a neural network, select pixels, a gesture, a heightening of intensity in a region of an image).

Indeed, at its core an explanation is something that is conveyed by the explainer to the explainee (e.g., by telling, demonstrating, visualizing, etc.) in order to justify the latter's belief in some explanandum. For example, by constructing a heat-map from a medical image, an otherwise black-box decision-making algorithm can highlight for the explainee the pixels that have most strongly supported its classification decision (Montavon et al., 2018), thereby allowing the explainee to assimiliate this explanation into their beliefs and better interpret the system's decision.

However, many heat-map based XAI approaches will present the same explanation (i.e., a heat-map) to all explainees. In contrast, Akula et al. (2022) exemplify the importance of Theory of Mind in explainable image recognition. In particular, Akula et al. develop an approach to generate explanations for Convolutional Neural Networks (LeCun et al., 1998) via an iterative dialogue process wherein the mental

state of the explainee is taken into account. Akula et al. moreover show that human trust in the system is increased when using their Theory of Mind-based approach, compared to standard attention- or heatmap-based approaches that do not tailor explanations to the human explainee.

While a subset of the body of work surveyed in this section has explicitly acknowledged the importance of Theory of Mind in explanation, the rest, by highlighting the need to move beyond soliloquies and provide tailored explanations, can be seen to do so when viewed through the lens of our work. The work presented in this chapter builds on the shoulders of this extant body of work and offers a principled investigation of the role of Theory of Mind in explanation. Our formal account of explanation aspires to address the diverse needs of explanation in AI by including a belief revision component within a multi-agent belief-based framework, as well as by proposing a number of desiderata for explanation that utilizes Theory of Mind.

3.6.2 Use of Epistemic States and Belief Revision in Explanation

As previously discussed, we are not the first to propose an account of explanation in terms of the epistemic state of an agent. Levesque (1989) presents a knowledge-level account of abduction based on the epistemic state of an agent. Levesque provides a generic definition of explanation that does not commit to a specific type of agent belief. Then, building on his seminal work on a logic of implicit and explicit belief (Levesque, 1984), Levesque shows how such different formal models of belief lead to different forms of abductive inference and resultant explanations. Boutilier & Becher (1995) similarly appeal to epistemic states to characterize the beliefs of an agent, employing belief revision to allow for explanations that are inconsistent with the epistemic state of the explainee. Prior to the works of Levesque and Boutilier & Becher, Gärdenfors (1988) proposed a model of explanation where explanations are defined relative to the epistemic states of agents. While Gärdenfors's account is probabilistic, the models proposed by Levesque and Boutilier & Becher are qualitative. We share the use of epistemic states with all three works, the appeal to qualitative criteria with Levesque and Boutilier & Becher, and the recognition of the importance of belief revision with Boutilier & Becher. Nevertheless, these works all characterize explanation with respect to a single agent, providing no account of the distinct beliefs of the explainee *and* explainer.

Furthermore, Nepomuceno-Fernández et al. (2017) proposed an account of ex-

planation that also recognizes the importance of a revision operator and the use of epistemic states. However, while their Dynamic Epistemic Logic (DEL) based framework can capture multiple agents, their focus remains on an agent's task of obtaining an abductive explanation for itself, rather than for other agents.

Lastly, Halpern & Pearl (2005) proposed a structural model of explanation selection based on the epistemic state of the explainee. In their work, the explainee's epistemic state comprises a set of situations the explainee considers possible and an explanation is then meant to remove some of these possible situations such that the cause of some explanandum may be uniquely identified. More recently, Miller (2021) extended Halpern & Pearl's approach to include *contrastive* explanations which are given relative to some counterfactual (e.g, in response to the question '*Why P rather than Q?*'). Halpern & Pearl and Miller, however, do not discuss some of the necessary elements of Theory of Mind in explanation we discuss in this chapter, such as the notions of explainer-explainee discrepancies, the adequacy of the explainer's beliefs, and explanations comprising epistemic components such as the nested beliefs of agents.

3.6.3 XAIP

The subfield of XAI known as Explainable AI Planning (XAIP) (e.g., Fox et al., 2017; Hoffmann & Magazzeni, 2019; Chakraborti et al., 2020; Carreno et al., 2021; Lindsay & Petrick, 2021) is a special case of the general task of explanation generation where different agents may disagree not only about the state of the world but also about the dynamics that govern it. Within XAIP, a growing body of extant work has promoted the view (also espoused in this chapter) that there is need to consider the (possibly incomplete and incorrect) beliefs held by the explainee when explaining agents' plans. As mentioned earlier, Chakraborti et al. (2017) have termed explanations that do not consider the explainee's perspective *soliloquies* and morever argued that planning agents offering explanations should eschew soliloquies and instead consider the possibly disparate model held by the explainee. To realize these desiderata, Chakraborti et al. formulated the *model reconciliation problem*, with a large body of work continuing this line of research (Chakraborti et al., 2020; Sreedharan et al., 2021). Intuitively, the process of model reconciliation involves the AI system reasoning about (its understanding of) the human's model and suggesting changes to it to conform with the system's model and consequently make the AI system's plan optimal from the perspective of the human (i.e., with respect to the human's reconciled

model).

Moreover, and related to the work in this chapter, Vasileiou et al. (Vasileiou et al., 2021a; Vasileiou & Yeoh, 2022; Vasileiou et al., 2022) investigate a logic-based approach to model reconciliation proposed a logic-based framework for model reconciliation that operates over two knowledge bases – of an explainer and an explainee.

In addition, Sreedharan et al. (2019) (and later Dung & Son (2022) in the context of logic programs) demonstrate how the model reconciliation paradigm, proposed by Chakraborti et al. (2017), can be generalized to address the important case where the explainee's model of the explainer's planning model is not explicitly known or not provided in a declarative form. Our work captures some of the insights in Sreedharan et al.'s work, in addition to incorporating the notions of epistemic states and belief revision, which in turn allows us to draw inspiration from the rich body of previous work in the field where these ideas originated. Chapter 5 will focus more heavily on XAIP and provide a deeper dive into extant work.

3.7 Concluding Remarks

In this chapter, we discussed the role of Theory of Mind in explanation. We moreover argued that the use of Theory of Mind in explanation holds the promise of producing high-quality explanations that are tailored to the beliefs of the explainee, in the context of the beliefs (and ignorance) of the explainer. In particular, we identified a set of desiderata for explanation that utilizes Theory of Mind. These desiderata informed our proposed belief-based account of explanation. Key features of this account are the appeal to epistemic states to capture the mental states of *both* the explainer and explainee, and the use of the explainee's belief revision to assimilate explanations. Further, we formalized and discussed the notion of a discrepancy as a property that allows the explainer to anticipate and provide explanations without prompting. We also presented properties relating to the adequacy of the explainer's beliefs with respect to providing an explanation. Our technical contributions are the theorems and propositions within this chapter, as well as all definitions excluding Definitions 3.1 and 3.2.

There are several take-aways from this chapter that are worth highlighting. Explanations need not be consistent with an agent's beliefs. As such, contrary to most logical treatments of explanation, characterizations of explanation should involve a belief revision component, and not just the expansion of existing beliefs to include an explanation. Further, by providing a belief-based account of explanation that charac-

terizes mental states in terms of epistemic states, and by allowing for epistemic states and revision operators to be realized in a diversity of forms from standard logical accounts, to computer programs, neural networks or human brains, we can capture the mental states of a myriad of different types of agents. Consequently, by characterizing explanations in terms of the explainer's beliefs about the explainee's beliefs and revision operator, we can capture the role of Theory of Mind in explanation for a myriad of different types of agents.

As mentioned in Section 3.6.3, XAIP is a special case of the general task of explanation generation where different agents may disagree not only about the state of the world but also about the dynamics that govern it (e.g., actions in a planning model). While this chapter presented a general characterization of explanation which, by design, is not restricted to settings where such dynamics are involved, Chapters 4, 5, and 6 will operate within the context of dynamical systems and allow for reasoning about action and change in the context of agents' beliefs. In particular, in Chapter 4 we revisit the notion of adequacy introduced in this chapter in the context of the interesting problem of plan recognition. Furthermore, Chapter 5 will build on ideas from the present chapter by revisiting the notion of a discrepancy and proposing a formulation that allows agents to resolve discrepancies pertaining to the validity of plans. Moreover, discrepancies may be resolved in this formulation via explanations that satisfy the majority of desiderata proposed in Section 3.1. In this way, the present chapter lays some of the conceptual foundations for the chapters that follow. Finally, this chapter did not offer an implementation of the belief-based account of explanation presented in Section 3.2. Instead, our aim in this chapter was to elucidate the role of Theory of Mind in explanation rather than navigate the computational challenges of augmenting AI systems with Theory of Mind, discussed in Chapter 1. The following chapters will appeal to the computational paradigm of epistemic planning and develop computational solutions that address some of these challenges while circumventing others.

Chapter 4

Epistemic Plan Recognition

> *"After all, inferring another agent's plan means figuring out what actions they 'have in mind,' and they may well be wrong about the effects of those intended actions".*
>
> – Pollack (1986)[p. 3]

Chapter 3 focused on social explanations of some abstract explanandum, given by an explainer to an explainee where, importantly, the explainee's perspective is taken into account in the explanation process. However, explanations are often generated by an agent to explain to *itself* some observed phenomenon (e.g., other agents' behavior). For instance, the problem of plan recognition (Schmidt et al., 1978; Cohen et al., 1981; Kautz & Allen, 1986; Appelt & Pollack, 1992; Carberry, 2001) is to infer an observed agent's plan and goal (e.g., making a cup of coffee) given observations about the environment and the observed agent's behavior (e.g., the observed agent walked into the kitchen, turned on the kettle, and obtained a mug). In other words, an agent observing another agent's behavior generates an explanation – a hypothesized plan and goal the observed agent is likely pursuing – which explains the observed agent's behavior.

Plan recognition has a long history and was originally seen as an intersection of psychology and AI (Schmidt et al., 1978). Early approaches used plan libraries to match observations to a particular plan (e.g., Kautz & Allen, 1986), while later work cast plan recognition as a form of parsing using formal grammars (e.g., Vilain, 1990; Pynadath & Wellman, 2000; Geib & Goldman, 2009, 2011). Recent research has dispensed with plan libraries by using AI planning to perform plan recognition (e.g., Ramírez & Geffner, 2010; Sohrabi et al., 2016). Later in this chapter, we discuss how we appeal to this latter approach in our proposed computational approach. Extant

work in plan recognition will be discussed more extensively in Section 4.6.

In this chapter we focus on plan recognition and transition from an explainer, an explainee, and an explanandum in Chapter 3, to an observer, an actor (the observed agent), and observations about the actor's behavior and the state of the world that must be explained. Plan recognition can be seen as an exercise in the observer's Theory of Mind. That is, in order to explain the observed behavior of the actor, the observer attributes to the latter various mental states - beliefs, plans, and goals. The recognition process is thus inherently *epistemic* in that it is determined by the beliefs of the observer about the beliefs, plans, and goals of the actor. We therefore take the view (espoused by Pollack (1986) and others) that plan recognition can necessitate representing and reasoning about the potentially distinct beliefs of the observer and the actor, and in particular that the observer may be required to assume the perspective of the actor in order to effectively recognize what the actor is doing. We further observe that the goals being pursued by the actor may be *ontic* – related to changing the state of the world, *epistemic* – related to changing its own (or another agent's) state of belief or knowledge, or perhaps both.

To advance this view of plan recognition, in this chapter we introduce and formalize the notion of *epistemic plan recognition* in which we explicitly represent and reason about agents' (nested) beliefs about the world, and the effects of actions on the world and on the beliefs of other agents. While Chapter 3 offered a general account of explanation without an associated computational realization, the remainder of the dissertation will narrow its attention and focus it on Theory of Mind reasoning in the context of action and change. Moreover, in this chapter and the two chapters that follow we appeal to the computational paradigm of epistemic planning (see extensive discussion in Chapter 2) and develop computational solutions that augment AI systems with Theory of Mind capabilities. In this chapter, by appealing to epistemic planning tools that build on a rich multi-agent epistemic logic framework we are able to: (1) model the observer and its knowledge of the actor's beliefs and capabilities as first-class elements of the plan recognition process, and (2) recognize plans with epistemic as well as ontic goals, or their combination. Modeling the observer is important as it enables reasoning about its own, and the actor's, beliefs, ignorance, and misconceptions relating to its environment and to the beliefs and capabilities of other agents.

Main contributions

- We propose and formalize the notion of epistemic plan recognition, which adds an important dimension to the plan recognition process by appealing to a notion of epistemics.
- We propose a computational realization of epistemic plan recognition as epistemic planning, which synthesizes formalisms and computational techniques from both epistemic planning and plan recognition.
- We propose and discuss the notion of adequacy of the observer's beliefs.
- We evaluate our approach on a set of epistemic plan recognition problems, using four epistemic planners.
- We evaluate the impact of the veracity of the observer's beliefs on goal recognition accuracy.

Relationship to Published Work

This chapter builds upon our AAMAS 2020 publication (Shvo et al., 2020b). In addition, the work in this chapter relates to ideas presented in the author's Master's thesis (Shvo et al., 2018). Relative to (Shvo et al., 2018), this chapter presents a more focused definition of epistemic plan recognition (rather than multi-agent epistemic plan recognition); adds a discussion of the adequacy of the observer's beliefs; and presents the results of an evaluation involving the recognition of epistemic goals and four epistemic planners. Moreover, relative to both (Shvo et al., 2020b) and (Shvo et al., 2018), this chapter represents a better theoretical understanding of epistemic plan recognition and practical experience with the epistemic planning tools that are at the core of our computation and evaluation.

Chapter Structure

In Section 4.1, we propose and formalize the notion of epistemic plan recognition. In Section 4.2, we discuss the notion of the adequacy of the observer's beliefs in plan recognition. In Section 4.3, we propose a computational realization of epistemic plan recognition as epistemic planning. In Section 4.4, we present the results of our experimental evaluation. Finally, in Sections 4.5 & 4.6 we discuss the work presented in this chapter, including some limitations and possible extensions, and survey related extant work.

4.1 Epistemic Plan Recognition

In this section, we propose and formalize the notion of epistemic plan recognition. The task of plan recognition is to infer a plan and goal that account for the observed behavior of an actor. A plan recognition problem comprises a number of important components: an actor and observer; a set of possible goals the actor may be pursuing; and a sequence of observations. As mentioned, in this chapter we appeal to the plan recognition as planning computational paradigm (discussed later on in Section 4.3). To align with this approach, we first define (following Ramírez & Geffner (2009)) a (non-epistemic) plan recognition problem in a classical$^+$ planning setting, before transitioning to the epistemic planning setting.

Definition 4.1 (Plan Recognition Problem (after Ramírez & Geffner (2009))). *A* ***plan recognition problem*** *is a tuple* $\langle \mathcal{F}, \textsc{Ops}, \mathcal{I}, \mathcal{G}, O \rangle$*, where* $\langle \mathcal{F}, \textsc{Ops} \rangle$ *is a classical*$^+$ *planning domain (as defined in Section 2.1*[1]*),* $\mathcal{I}$ *is the initial state,* $\mathcal{G}$ *is a set of possible goals (each goal* $G \in \mathcal{G}$ *is a partial state, as defined in Definition 2.1), and* $O = o_1, ..., o_m$*, is a sequence of observations, where* $o_i \in \textsc{Ops}$*.*

In Definition 4.1, each observation $o_i \in O$ is an operator drawn from the set of operators Ops (e.g., O could include an observation of the actor turning on the kettle). Later on in this section, we will augment the notion of observations to also include properties of state, in addition to operators. Next, while the set of possible goals $\mathcal{G}$ may in general be infinite, it is typical in the plan recognition literature to utilize domain knowledge to prune $\mathcal{G}$ such that it only contains relevant hypotheses about the actor's presumed goal. Lastly, a solution to a plan recognition problem is, intuitively, the conjectured plan and goal that the actor is pursuing, given the observations in O, where the conjectured goal is drawn from the set of possible goals $\mathcal{G}$ (e.g., the actor's goal is to make coffee). We will soon define this formally in the epistemic setting.

As argued in the beginning of this chapter, a general account of plan recognition should provide a means of modeling the beliefs of the observer, including its beliefs about the actor's beliefs. To this end, in defining the epistemic plan recognition problem we build on the multi-agent modal logic KD45$_n$ (discussed in Chapter 2, Section 2.2). We moreover appeal to the RP-MEP formalism (discussed in Chapter 2, Section 2.3) to represent the beliefs of the observer, the actor, and possibly also those

[1]To avoid confusion with the notation for the sequence of observations, O, we use the notation Ops to denote the set of operators in a classical$^+$ planning domain.

of other agents, as well as to facilitate reasoning about action, change and goals (both ontic and epistemic).

More concretely, to transition from Definition 4.1 to a definition of *epistemic* plan recognition, we (1) enable representation of and reasoning about the potentially distinct beliefs of the observer and the actor; and (2) enable modeling of *epistemic goals* the actor may be pursuing in the set of possible goals, $\mathcal{G}$. To achieve (1), as mentioned, we appeal to the RP-MEP formalism to represent the beliefs of the observer, the actor, and possibly also those of other agents. Recall that an RP-MEP problem includes the set of agents, Ag. We include the observer (Obs $\in Ag$) and the actor (Act $\in Ag$) in Ag. Recall further that, as discussed in Chapter 2, the first 'P' in RP-MEP stands for perspectival since in the RP-MEP setting all planning is done from the perspective of a root agent, $\star \in Ag$. In the epistemic plan recognition setting, the observer takes the role of the root agent and so agent Obs is $\star$. In Section 4.5 we discuss this modeling choice further. Next, to achieve (2), each goal $G \in \mathcal{G}$ in the set of possible goals is a PEKB. This allows us to model epistemic goals that the actor may be pursuing.

Finally, observing not only what an actor does, but also the properties of state that may be influencing their decision making, is critical to epistemic plan recognition and to plan recognition more generally. To this end, in defining the epistemic plan recognition problem, we model observations in O such that they comprise both actions and properties of the state of the system. In particular, a sequence of observations is a sequence of tuples $(\alpha_1, \phi_1), \ldots, (\alpha_m, \phi_m)$. Each α_i corresponds to the observation of an action, $a \in \mathcal{A}$, and each $\phi_i \in \mathcal{P}$ corresponds to the observation of some properties of state immediately following the execution of α_i. ϕ_i is a conjunction of fluent atoms drawn from $\mathcal{P}$. For example, the observation (*leave*(John, Room1), *empty*(Room1)) signifies that the observer had observed that the room is empty after observing John leaving the room. In cases where only properties are observed, α_i is empty. In cases where an action is observed but no state properties are, ϕ_i is $\top$, i.e., *true*.

Definition 4.2 (Epistemic Plan Recognition Problem)**.** *An* **epistemic plan recognition problem** *is a tuple* $\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$*, where* $\langle \mathcal{P}, \mathcal{A}, Ag \rangle$ *is an RP-MEP domain (*Obs $\in Ag$ *is the root agent),* $\mathcal{I}$ *is the initial state,* $\mathcal{G}$ *is a set of possible goals (each goal* $G \in \mathcal{G}$ *is a PEKB), and* $O = o_1, ..., o_m$*, is a sequence of observations. Each observation* o_i *is a pair* (α_i, ϕ_i) *comprising zero or one observed actions,* $\alpha_i \in \mathcal{A}$*, together with* ϕ_i*, a conjunction of fluent atoms drawn from* $\mathcal{P}$*, or* $\top$*.*

Recall that we are interested in finding a plan and goal that account for the given sequence of observations, O. The following definition – formalizing what it means

for a plan to account for a sequence of observations – is based on the definitions of Ramírez & Geffner (2010), Sohrabi et al. (2010), and Sohrabi et al. (2016).

Definition 4.3 (Satisfaction of a Sequence of Observations)**.** *Given an epistemic plan recognition problem,* $\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$*, a plan* $\pi = a_1, \ldots, a_n$ **satisfies** *observations* $O = (\alpha_1, \phi_1), \ldots, (\alpha_m, \phi_m)$ *if there is a non-decreasing function* f *mapping the observation indices* $j = 1, ..., m$ *into plan indices* $i = 1, ..., n$ *such that* $a_{f(j)} = \alpha_j$ *(trivially satisfied when* α_j *is empty), and* $\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}) \models B_{\textsc{Obs}}\phi_j$ *for* $j = 1, \ldots, m$*.*

Intuitively, a plan π satisfies a sequence of observations O if π *embeds* the observed actions α_i in the (α_i, ϕ_i) pairs in O (Ramírez & Geffner, 2010). Moreover, to satisfy the state observations ϕ_i, we require that in the progression of the initial state by a relevant prefix of the presumed plan, the observer should believe the corresponding state observation. We will shortly illustrate this concept using an example. Finally, using the notion of satisfaction we can characterize a solution to an epistemic plan recognition problem.

Definition 4.4 (Epistemic Plan Recognition Solution)**.** *Given an epistemic plan recognition problem,* $\langle \mathcal{P}, \mathcal{A}, \mathcal{D}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$*, a* ***solution*** *is a pair* (π, G)*, where* $G \in \mathcal{G}$ *is a goal and* π *is a plan that satisfies* O*.*

That is, a solution to an epistemic plan recognition problem is the goal and plan the observing agent conjectures the observed agent is following. To maintain generality of Definition 4.4, the relationship between the plan component π and the goal component G is left unconstrained. Later in this section we discuss a possible relationship between the two, characterized by the observer's beliefs about whether or not G holds following π's execution and the observer's beliefs about whether or not the actor will believe that G holds following π's execution. Moreover, in Section 4.3 our proposed approach to computing solutions to epistemic plan recognition problems constrains this relationship to obtain desirable solutions.

4.1.1 Example

Consider a scenario where Alice (denoted as $\textsc{Act} \in Ag$) has the goal of bringing a cake to her friend Eve's party and an observer ($\textsc{Obs} \in Ag$) is trying to infer Alice's plan and goal. Alice is observed going into the store, leaving it with a bag, and heading to the party. We model this scenario as an epistemic plan recognition problem:

$Ag = \{\text{Obs}, \text{Act}\}$

$\mathcal{I} \models B_{\text{Obs}}at(\text{Act}, \text{Home}) \wedge B_{\text{Obs}}B_{\text{Act}}at(\text{Act}, \text{Home}) \wedge B_{\text{Obs}}B_{\text{Act}}\neg at(\text{Cake}, \text{Party}) \wedge$
$B_{\text{Obs}}loc(\text{Party},\text{Address1}) \wedge B_{\text{Obs}}B_{\text{Act}}loc(\text{Party},\text{Address1})$

$O = (enter(\text{Act}, \text{Store}),\ in(\text{Act}, \text{Store})),$
$(leave(\text{Act}, \text{Store}), \neg in(\text{Act}, \text{Store}) \wedge holding(\text{Act}, \text{Bag})),$
$(goTo(\text{Act}, \text{Address1}),\ at(\text{Act}, \text{Address1}))$

$\mathcal{G} = \{at(\text{Cake}, \text{Home}),\ at(\text{Cake}, \text{Work}),\ at(\text{Cake}, \text{Party})\}$

A possible solution to this epistemic plan recognition problem is the pair (π, G) where Alice's presumed goal G is $at(\text{Cake}, \text{Party})$. It is easy to see in this toy example that $at(\text{Cake}, \text{Party})$ is indeed the most likely goal out of the three possible 'cake transporting' goals in $\mathcal{G}$, given the observations in O. Alice's presumed plan π is

$$
\begin{aligned}
&[takeBus(\text{Act}, \text{Home}, \text{Store}),\\
&\ enter(\text{Act}, \text{Store}),\\
&\ buy(\text{Act}, \text{Cake}),\\
&\ putIn(\text{Act}, \text{Cake}, \text{Bag}),\\
&\ leave(\text{Act}, \text{Store}),\\
&\ goTo(\text{Act}, \text{Address1})].
\end{aligned}
$$

Importantly, the presumed plan π satisfies the sequence of observations in O per Definition 4.3. In particular, let f be a non-decreasing function f mapping the observation indices $j = 1, 2, 3$ into plan indices $i = 1, ..., 6$ such that $a_{f(j)} = \alpha_j$, and $\text{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}) \models B_{\text{Obs}}\phi_j$ for $j = 1, 2, 3$. Recall that α_j and ϕ_j are the action component and the state component of the observation, respectively. Recall also that $\text{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I})$ uses the progression operator, prog, to construct a formula that describes the state that results from executing the sequence of actions $[a_1, \ldots, a_{f(j)}]$ in the initial state $\mathcal{I}$ (see discussion of prog in Chapter 2, Section 2.3.1).

We have that:

$$
\begin{aligned}
&a_{f(1)} = a_2 = \alpha_1 = \mathit{enter}(\textsc{Act}, \mathit{Store})\\
&\textsc{prog}([a_1, a_2], \mathcal{I}) \models B_{\textsc{Obs}}\mathit{in}(\textsc{Act}, \text{Store})\\
&a_{f(2)} = a_5 = \alpha_2 = \mathit{leave}(\textsc{Act}, \text{Store})\\
&\textsc{prog}([a_1, \ldots, a_5], \mathcal{I}) \models B_{\textsc{Obs}}(\neg\mathit{in}(\textsc{Act}, \text{Store}) \wedge \mathit{holding}(\textsc{Act}, \text{Bag}))\\
&a_{f(3)} = a_6 = \alpha_3 = \mathit{goTo}(\textsc{Act}, \text{Address1})\\
&\textsc{prog}([a_1, \ldots, a_6], \mathcal{I}) \models B_{\textsc{Obs}}\mathit{at}(\textsc{Act}, \text{Address1}).
\end{aligned}
$$

The Relationship Between π and G

Let us now consider the relationship between the plan component π and the goal component G in the solution (π, G), characterized by the observer's beliefs about whether or not G holds following π's execution and the observer's beliefs about whether or not the actor will believe that G holds following π's execution. In our example, the following holds:

$$
\begin{aligned}
&\textsc{prog}(\pi, \mathcal{I}) \models B_{\textsc{Obs}}\mathit{at}(\text{Cake, Party})\\
&\textsc{prog}(\pi, \mathcal{I}) \models B_{\textsc{Obs}}B_{\textsc{Act}}\mathit{at}(\text{Cake, Party}).
\end{aligned}
$$

That is, following π's execution the observer will believe that the goal at(Cake, Party) holds. Moreover, following π's execution the observer will believe that *the actor* believes that the goal at(Cake, Party) holds.

However, Pollack (1986) identified that *"inferring another agent's plan means figuring out what actions she 'has in mind,' and she may well be wrong about the effects of those intended actions"*. To align with Pollack's view, in defining a solution to an epistemic plan recognition problem in Definition 4.4 we distinguish between the actor's presumed plan π and the underlying intent of the plan, i.e., the presumed goal G. This allows us to distinguish between the observer's beliefs about the goal G and the observer's beliefs about the actor's beliefs about the goal G, following the execution of the plan π.

To make this point concrete, we build on our party example. Consider that this time the observer believes that the party's address is Address1 and moreover believes that Alice *falsely* believes that the address is Address2, i.e.,

$$
\mathcal{I} \models B_{\textsc{Obs}}\mathit{loc}(\text{Party,Address1}) \wedge B_{\textsc{Obs}}B_{\textsc{Act}}\mathit{loc}(\text{Party,Address2}).
$$

In Section 4.3 we will see how solutions to epistemic plan recognition problems may be computed. For now, since the observer believes that Alice has a false belief about the party's location, let us assume that Alice's presumed plan π is the same as before except for the last action, which is now

$$goTo(\textsc{Act}, \text{Address2}).$$

The following now holds pertaining to Alice's presumed plan π and goal G (i.e., $at(\text{Cake}, \text{Party})$):

$$\textsc{prog}(\pi, \mathcal{I}) \not\models B_{\textsc{Obs}}\, at(\text{Cake}, \text{Party})$$
$$\textsc{prog}(\pi, \mathcal{I}) \models B_{\textsc{Obs}} B_{\textsc{Act}}\, at(\text{Cake}, \text{Party}).$$

That is, following π's execution the observer will *not* believe that the goal $at(\text{Cake}, \text{Party})$ holds. Moreover, following π's execution the observer will believe that *Alice* believes that the goal $at(\text{Cake}, \text{Party})$ holds.

While we focus in this chapter on the recognition task, in Chapter 6 we will build on the techniques proposed in this chapter and Chapter 5 and show how an assistive observer is able to recognize Alice's plan, and subsequently help her 'correct course' by resolving the *discrepancy* between its beliefs and its beliefs about Alice's beliefs pertaining to the address of the party and correspondingly about the validity of Alice's plan.

Recognizing Epistemic Goals

Recall that the actor may be trying to achieve an epistemic goal, i.e., some state of knowledge or belief and moreover that the observer may wish to infer the actor's epistemic goal. For example, consider that Alice wants to surprise her friend Eve and so does not want her to know that she has brought cake to the party. Of course, to surprise Eve, Alice *does* want her to know she has brought cake to the party, but only at the appropriate time. While this kind of interesting temporal reasoning is outside the scope of this dissertation, see for example an investigation of the role of logic in formalizing magic tricks by Smith et al. (2016), and related work that connects these insights to goal recognition (Masters et al., 2021). Of relevance is also the large body of work on deception and its logical formalization (e.g., Sakama, 2015).

Next, to represent Alice's 'surprise Eve with a cake' goal in $\mathcal{G}$, we include Eve as

an agent in Ag, i.e.,

$$\neg B_{Eve}\, at(\text{Cake, Party}) \wedge at(\text{Cake, Party}) \subseteq \mathcal{G}.$$

The observer initially believes that Alice believes that Eve is at the party, i.e.,

$$\mathcal{I} \models B_{\text{Obs}} B_{\text{Act}}\, at(\text{Eve, Party}).$$

Moreover, recall that the action

$$putIn(\textsc{Act}, \text{Cake, Bag})$$

is part of Alice's presumed plan, π. The *putIn* action has a number of conditional effects, including

$$(\top, B_{\text{Obs}} hidden(Obj))$$
and
$$(\top, B_{\text{Obs}} B_{\text{Act}} hidden(Obj)).$$

Intuitively, this means that the observer believes (and also believes that Alice believes) that whatever object was put in the bag is hidden[2]. Therefore, following the *putIn* action, the observer's beliefs about Alice's beliefs include

$$B_{\text{Obs}} B_{\text{Act}} hidden(\text{Cake}),$$
$$B_{\text{Obs}} B_{\text{Act}} \neg hidden(\text{Bag}),$$
and
$$B_{\text{Obs}} B_{\text{Act}} \neg hidden(\textsc{Act}),$$

assuming the observer previously believed that neither the bag nor Alice are hidden. Next, following the execution of $goTo(\textsc{Act}, \text{Address1})$, the observer's beliefs about

[2]For more details on how agents' 'awareness' of actions in modelled in the RP-MEP formalism, see discussion in Section 2.3.1.

Alice's beliefs include

$$B_{\text{OBS}} B_{\text{ACT}}\, at(\text{Cake, Party}),$$
$$B_{\text{OBS}} B_{\text{ACT}}\, at(\text{ACT, Party}),$$
and
$$B_{\text{OBS}} B_{\text{ACT}}\, at(\text{Bag, Party}).$$

and since the bag and Alice are not hidden, the observer believes that Alice believes that every agent $i \in Ag$ at the party now believes that Alice and the bag are at the party, i.e.,

$$B_{\text{OBS}} B_{\text{ACT}} B_i\, at(\text{ACT, Party}) \wedge B_{\text{OBS}} B_{\text{ACT}} B_i\, at(\text{Bag, Party}).$$

However, the observer believes that Alice does not believe the agents at the party have changed their beliefs about $at(\text{Cake, Party})$ since she believes $hidden(\text{Cake})$. Finally, the observer believes that Alice believes that Eve is at the party,

$$B_{\text{OBS}} B_{\text{ACT}}\, at(\text{Eve, Party}),$$

and that Eve initially does not believe $at(\text{Cake, Party})$. Therefore, the observer also believes that Alice believes that Eve does not believe that the cake is at the party, i.e.,

$$B_{\text{OBS}} B_{\text{ACT}} \neg B_{Eve}\, at(\text{Cake, Party})$$

following the execution of Alice's presumed plan π. Importantly and evidently, to recognize epistemic goals more generally and Alice's epistemic goal in particular (i.e., figuring out that Alice wishes to surprise Eve), an observer must reason not only about an actor's beliefs about the world, but also about her nested beliefs pertaining to other agents (e.g., Eve's) beliefs. This supports our claim that a general account of plan recognition should provide a means of modeling the (nested) beliefs of the observer.

4.2 The (In)Adequacy of the Observer's Beliefs

Epistemic plan recognition, like any plan recognition system, is limited by the veracity and completeness of the observer's knowledge and beliefs – the set of possible goals it

contemplates, the sequence of observations it observes, and, importantly, the veracity and completeness of the observer's beliefs about the environment and the actor's beliefs. epistemic plan recognition is also limited by how discriminable the goals and plans under consideration are in the context of the observations and the model. An imperfect observer lacking complete observations may wrongly infer a plan or goal or fail to discriminate between a number of possible hypotheses. In particular, the quality of plan recognition can suffer if observations are noisy (e.g., as a result of a faulty sensor) or missing altogether as observed and addressed by Sohrabi et al. (2016). Finally, the observer's beliefs about the actor's beliefs must be sufficiently complete and accurate for the observer to robustly infer the actor's plan and goal. We elaborate on this point in the rest of this section.

Recall that an epistemic plan recognition problem is cast from the perspective of the observer who is also the root agent $\star \in Ag$. Let $\mathcal{I}^*$ and $\mathcal{A}^*$ denote the observer's beliefs about the world and the set of actions such that $\mathcal{I}^*$ and $\mathcal{A}^*$ are perfectly accurate with respect to the actor's *actual* beliefs (informally, the observer can be thought here to be perfectly 'mind reading' the actor). Finally, $\textsc{prog}^*(\pi, \mathcal{I}^*)$ denotes that plan π is progressed using $\mathcal{I}^*$ and $\mathcal{A}^*$.

Definition 4.5 (Adequacy)**.** *Given an epistemic plan recognition problem* $R = \langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$ *and the observer's 'perfect' beliefs about the actor's actual beliefs* $Z = (\mathcal{I}^*, \mathcal{A}^*)$*, we say that* $\mathcal{I}$ *and* $\mathcal{A}$ *are* ***adequate*** *with respect to* (R, Z) *iff for every goal* $G \in \mathcal{G}$ *and plan* π *that satisfies* O*, the following condition holds:*

$$\textsc{prog}(\pi, \mathcal{I}) \models B_{\textsc{Obs}} B_{\textsc{Act}} G$$
$$\textit{iff}$$
$$\textsc{prog}^*(\pi, \mathcal{I}^*) \models B_{\textsc{Obs}} B_{\textsc{Act}} G.$$

That is, if the observer's beliefs about the actor's beliefs are adequate, then following the execution of any plan π that satisfies O and achieves some goal $G \in \mathcal{G}$, the observer's beliefs about the actor's beliefs about G will be 'aligned' with the actor's actual beliefs about G.

4.2.1 Addressing Inadequacy of the Observer's Beliefs

In our example, if the observer's beliefs about Alice's beliefs are not adequate and it falsely believes $B_{\textsc{Act}} loc$(Party, Address1) (whereas Alice *actually* believes loc(Party,Address2)), then Alice's presumed plan π (which involves heading to Ad-

dress1 because of the observer's beliefs about Alice's beliefs) will not 'align' with Alice's actual plan of heading to Address2, since that is where she actually believes the party to be taking place. Indeed, with respect to Alice's presumed plan π we have that in this case

$$\textsc{prog}(\pi, \mathcal{I}) \models B_{\textsc{Obs}} B_{\textsc{Act}} G \quad \text{and}$$
$$\textsc{prog}^*(\pi, \mathcal{I}^*) \not\models B_{\textsc{Obs}} B_{\textsc{Act}} G$$

and the condition in Definition 4.5 therefore does not hold.

Inadequacy may be addressed in a myriad of ways, including the modulation of the observer's confidence in its predictions in the face of perceived irrationality (i.e., deviation from expected optimality). Masters & Sardina (2019) identified that deviation from expected optimality may stem from a number of possible reasons including a deceptive actor and the observer's misconceptions regarding the actor's beliefs. Inadequacy of $\mathcal{I}$ and $\mathcal{A}$ falls under the latter case where the observed irrationality may be due to the inadequacy of the observer's beliefs about the actor's beliefs. While in our computation and experimentation we assume the observer's beliefs about the actor's beliefs are adequate, future work can relax this assumption and employ Masters & Sardina's *rationality measure* (that quantifies the agent's deviation from the observer's expectation of optimality) to allow the observer to self-modulate its confidence in its predictions.

In addition to self-modulating its confidence, the observer may strive to improve the adequacy of $\mathcal{I}$, $\mathcal{A}$, and the observations at its avail. For example, it may be beneficial for the observer to attempt to refine its beliefs about the actor's beliefs when perceiving a low degree of rationality in the actor's behavior. Even in cases where the actor is observed to follow (expected) optimal behavior, improving the adequacy of the observer's beliefs may be needed when the actor's goals and plans are wrongly inferred or are indiscriminable (e.g., due to limited observations). To this end, the observer could, for instance, query the actor or sense the world in order to gain additional information. Indeed, there is a growing body of extant work concerned with endowing the observer with agency to act in the world in order to disambiguate the hypothesis space (Bisson et al., 2011; Keren et al., 2014; Mirsky et al., 2016, 2018; Amato & Baisero, 2019; Keren et al., 2020; Shvo & McIlraith, 2020; Gall, 2021; Gall et al., 2021)

Further, Pollack (1986) discusses inference techniques by which an observer can infer an actor's beliefs (and plans) given dialogue utterances formed by the latter. The

observer may also refine its beliefs about the actor's beliefs by employing machine learning techniques (e.g., Liao et al., 2007; Pereira et al., 2019b). Finally, when $\mathcal{A}$ is inadequate – when the observer's beliefs about the actor's beliefs about the underlying planning model do not align with the actor's acutal beliefs – the observer may hold false beliefs about the actor's beliefs following the execution of an action. This could be because, for example, a conditional effect of an action does not accurately reflect how the actor's beliefs change after the execution of that action. Pereira et al. (2019a) explore goal recognition in such settings where the observer holds an incomplete or incorrect model of the actor's action theory.

4.3 Computation

Recall that a solution to an epistemic plan recognition problem is a pair (π, G) where π is a plan that satisfies the sequence of observations O and $G \in \mathcal{G}$ is a goal. A number of different criteria have been proposed in the past to select a plan and goal that 'best' align with O, including the simplicity of the plan and the likelihood of a goal given the observations. Our general specification supports a diversity of criteria and computational realizations and here we propose one such realization of epistemic plan recognition as epistemic planning by appealing to the *plan recognition as planning* paradigm, proposed by Ramírez & Geffner (2009). Ramírez & Geffner transform the plan recognition problem into a planning problem, allowing for the use of off-the-shelf planning tools to solve the recognition problem. In what follows, we describe a transformation of epistemic plan recognition to epistemic planning; a way by which to compute a probability distribution over the set of possible goals given the transformed planning problem; and an algorithm that uses these techniques to compute a solution to an epistemic plan recognition problem.

4.3.1 Transformation to Epistemic Planning

Plan recognition belongs to a class of problems that calls upon a dynamical system model to account for system behavior, observed over a period of time (Sohrabi et al., 2011). Another task that belongs to this class of problems is diagnosis (e.g., Reiter, 1987; Boutilier & Becher, 1995; McIlraith, 1998), where the agent is attempting to identify the abnormal components in a system that allow for the observed behavior of the system. Similarly to the aforementioned work on plan recognition to which we appeal in this chapter (Ramírez & Geffner, 2009), some work on diagnosis has also

proposed methods to compile away sequences of observations and has investigated the relationship of this task to planning (Sohrabi et al., 2010). Diagnosis and plan recognition are therefore symbolically analogous when cast as instances of the general task of generating an explanation for observed behavior with respect to a model of the behavior of a dynamical system (Sohrabi et al., 2011). As explanation generation tasks, these two tasks only differ in the nature of observations (agent behavior and state observations in plan recognition and the possibly aberrant behavior of, for example, an electromechanical device in diagnosis), the nature of explanations (a sequence of actions that conjecture faulty events in diagnosis and the actor's presumed plan and goal in plan recognition), and the nature of the underlying dynamical system.

In this chapter, we view epistemic plan recognition as one such explanation generation task where the underlying dynamical system importantly involves the beliefs of multiple agents and a mechanism that dictates how those beliefs evolve. The transformation of an epistemic plan recognition problem to an RP-MEP problem we propose in this chapter is inspired by the classical planning-based transformations proposed by Ramírez & Geffner (2010) and by Sohrabi et al. (2016) who later modified Ramírez & Geffner's approach. In particular, while Ramírez & Geffner's transformation handles only action observations, Sohrabi et al.'s extension allows for state observations as well. Since our epistemic plan recognition formulation leverages observations comprising both observed actions and properties of state, we build on both Ramírez & Geffner's and Sohrabi et al.'s work. It should be noted that in the context of diagnosis, the work of Sohrabi et al. (2010) handles temporally extended observations of both actions and properties of state. Their formulation can thus encode complex observations, beyond what is afforded by the fully-ordered sequences of observations used in this chapter.

While Ramírez & Geffner cast the classical plan recognition problem as a classical planning problem, we realize a computational solution to the epistemic plan recognition problem by appealing to epistemic planning and utilizing epistemic planners. Intuitively, Ramírez & Geffner's approach compiles the observations in the sequence of observations O into the planning domain, thereby forcing all generated plans that solve the transformed planning problem to *satisfy* O. In the epistemic plan recognition setting, we define a correspondence between a given epistemic plan recognition problem and an RP-MEP problem by augmenting the set of actions, $\mathcal{A}$, with *explain* actions that allow the planner to explain the observations. Further, the set of fluent atoms, $\mathcal{P}$, is augmented with special bookkeeping predicates, p_i, l_{α_i}, and p_{α_i}, for each observation $o_i \in O$ that are made true when o_i is explained. These predi-

cates ensure that the order of the observation sequence O is respected by all plans that solve the transformed RP-MEP problem. The set $\mathcal{P}$ is also augmented with p_{init} which is set to be true initially. Formally, given an epistemic plan recognition problem, $\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$, and some goal $G \in \mathcal{G}$, we create an RP-MEP problem, $\langle \langle \mathcal{P}', \mathcal{A}', Ag \rangle, \mathcal{I}', G' \rangle$, such that:

- $\mathcal{P}' = \mathcal{P} \cup \{p_i \mid (\alpha_i, \phi_i) \in O\} \cup \{l_{\alpha_i} \mid (\alpha_i, \phi_i) \in O\} \cup \{p_{init}\}$
- $\mathcal{A}' = \mathcal{A} \cup \mathcal{A}_{\text{explain}}$
  - $\mathcal{A}_{\text{explain}} = \{\langle B_{\text{OBS}}(l_{\alpha_i} \wedge p_{i-1}), \{(\top, B_{\text{OBS}} p_{\alpha_i})\}\rangle \mid (\alpha_i, \phi_i) \in O\} \cup$ $\{\langle B_{\text{OBS}}(\phi_i \wedge p_{\alpha_i}), \{(\top, B_{\text{OBS}} p_i)\}\rangle \mid (\alpha_i, \phi_i) \in O\}$
  - We add $(\top, B_{\text{OBS}} l_{\alpha_i})$ and $B_{\text{OBS}} p_{i-1}$ to the conditional effects and precondition, respectively, of every action in $\mathcal{A}$ that corresponds to an observed action α_i that appears in an observation $(\alpha_i, \phi_i) \in O$ (where α_i is not empty).
- $\mathcal{I}' = \mathcal{I} \wedge B_{\text{OBS}} p_{init}$
- $G' = B_{\text{OBS}} B_{\text{ACT}} G \wedge B_{\text{OBS}} p_m$, where the special predicate p_m corresponds to the last observation in O, o_m

Note that for every observation $o_i \in O$, we augment $\mathcal{A}$ with two explain actions, i.e., one action belonging to the set

$$\{\langle B_{\text{OBS}}(l_{\alpha_i} \wedge p_{i-1}), \{(\top, B_{\text{OBS}} p_{\alpha_i})\}\rangle \mid (\alpha_i, \phi_i) \in O\}$$

and the other action belonging to the set

$$\{\langle B_{\text{OBS}}(\phi_i \wedge p_{\alpha_i}), \{(\top, B_{\text{OBS}} p_i)\}\rangle \mid (\alpha_i, \phi_i) \in O\}.$$

We do so to ensure that the action component of o_i, α_i, is accounted for before the state component of o_i, ϕ_i, is explained. We do this since ϕ_i is assumed to be observed immediately *following* the execution of α_i. If α_i is empty, the preconditions of the corresponding explain action will be p_{i-1} rather than $l_{\alpha_i} \wedge p_{i-1}$. Similarly, if $\phi_i = \top$ then the preconditions of the corresponding explain action will be p_{α_i} rather than $\phi_i \wedge p_{\alpha_i}$. The precondition of the explain action in $\mathcal{A}_{\text{explain}}$ corresponding to the first observation in O is set to be $B_{\text{OBS}}(l_{\alpha_i} \wedge p_{init})$.

The order in which the observations are explained is enforced by the precondition p_{i-1} which only allows an observation $(\alpha_i, \phi_i) \in O$ to be explained after all the

observations in O which precede it have been explained. Further, the transformation ensures that if some plan π' achieves G' then the observer will believe that the actor believes G following the execution of π'. Importantly, the observer need not itself believe that G holds when G' is achieved. This allows us to distinguish between the observer's beliefs about the goal G and the observer's beliefs about the actor's beliefs about the goal G, as discussed in Section 4.1.

Epistemic Plan Recognition to RP-MEP Transformation is Solution-Preserving

We use $\langle Q', \mathcal{I}', G' \rangle$ as shorthand for the RP-MEP problem $\langle \langle \mathcal{P}', \mathcal{A}', Ag \rangle, \mathcal{I}', G' \rangle$, which is the result of the transformation described above. We moreover say that a plan π solves an RP-MEP problem $\langle Q, \mathcal{I}, G \rangle$ *iff* $\textsc{prog}(\pi, \mathcal{I}) \models G$ (as mentioned in Chapter 2). In addition, we say that an action a is executable in PEKB S *iff* $S \models \text{Pre}(a)$. With the correspondence described above, solutions to the transformed RP-MEP problem, $\langle Q', \mathcal{I}', G' \rangle$, capture precisely the solutions to the corresponding RP-MEP problem within the epistemic plan recognition problem that satisfy O, as stated by the following theorem.

Theorem 4.6. *Given an epistemic plan recognition problem, $\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$, some goal $G \in \mathcal{G}$ and the corresponding transformed RP-MEP problem, $\langle Q', \mathcal{I}', G' \rangle$, we have that:*
***(1)** If π is a plan that satisfies O and solves the RP-MEP problem $\langle \langle \mathcal{P}, \mathcal{A}, Ag \rangle, \mathcal{I}, B_{\text{OBS}} B_{\text{ACT}} G \rangle$ then there exists a plan π' that solves $\langle Q', \mathcal{I}', G' \rangle$ such that π can be reconstructed straightforwardly from π' by removing the explain actions from π'.*
***(2)** If π' is a plan that solves $\langle Q', \mathcal{I}', G' \rangle$ then there exists a plan π that solves $\langle \langle \mathcal{P}, \mathcal{A}, Ag \rangle, \mathcal{I}, B_{\text{OBS}} B_{\text{ACT}} G \rangle$ and satisfies O such that π can be reconstructed straightforwardly from π' by removing the explain actions from π'.*

Proof:
(1) We prove that if π is a plan that satisfies O and solves $\langle \langle \mathcal{P}, \mathcal{A}, Ag \rangle, \mathcal{I}, B_{\text{OBS}} B_{\text{ACT}} G \rangle$ then there exists a plan π' that solves $\langle Q', \mathcal{I}', G' \rangle$ such that π can be reconstructed straightforwardly from π' by removing the explain actions from π'. Since π satisfies O, there is a function f mapping the observation indices $j = 1, ..., m$ into plan indices $i = 1, ..., n$ such that $a_{f(j)} = \alpha_j$, and $\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}) \models B_{\text{OBS}} \phi_j$ for $j = 1, \ldots, m$. We show by induction that for every observation $o_j \in O$, the two explain actions in Q' corresponding to o_j can be inserted into π after the action $a_{f(j)}$ such that the two explain actions are executable in $\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I})$.

Base case: for o_1, the two corresponding explain actions are executable *iff*

$$\textsc{prog}([a_1, \ldots, a_{f(1)}], \mathcal{I}') \models B_{\textsc{Obs}}(l_{\alpha_1} \wedge p_{init})$$

and

$$\textsc{prog}([a_1, \ldots, a_{f(1)}], \mathcal{I}') \models B_{\textsc{Obs}}(\phi_1 \wedge p_{\alpha_1})$$

hold, respectively. From the definition of satisfaction, $a_{f(1)} = \alpha_1$ and since $(\top, B_{\textsc{Obs}} l_{\alpha_1})$ has been added to the conditional effects of the action in $\mathcal{A}$ corresponding to $a_{f(1)}$, we have that the first explain action is executable. For the second explain action, we have that

$$\textsc{prog}([a_1, \ldots, a_{f(1)}], \mathcal{I}') \models B_{\textsc{Obs}} \phi_1$$

and hence the second action is also executable (p_{α_1} will hold after the execution of the first explain action).

Inductive step: we now prove that for some observation $o_j \in O$, the two corresponding explain actions are executable. The two actions are executable *iff*

$$\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}') \models B_{\textsc{Obs}}(l_{\alpha_j} \wedge p_{j-1})$$

and

$$\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}') \models B_{\textsc{Obs}}(\phi_j \wedge p_{\alpha_j})$$

hold, respectively. By the induction hypothesis, we have that the two explain actions corresponding to o_{j-1} are executable and have been added to π after the action $a_{f(j-1)}$. Hence, by the construction of the explain actions, we have that

$$\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}') \models B_{\textsc{Obs}} p_{j-1}$$

holds, where the prefix of π, $[a_1, \ldots, a_{f(j)}]$, includes the explain actions corresponding to observations $o_1, \ldots, o_{j-1}$. The rest of the argument for the applicability of the two explain actions corresponding to o_j is identical to that made for o_1. Thus, we have that the two explain actions corresponding to o_j are executable. Since we have proved by induction that for any observation in O, the two corresponding explain actions are

executable, this also holds for o_m, the last observation. Therefore, we have that

$$\textsc{prog}([a_1, \ldots, a_{f(m)}], \mathcal{I}') \models B_{\textsc{Obs}} p_m$$

where $[a_1, \ldots, a_{f(m)}]$ is the plan π with the inserted explain actions. We call this plan π'. The inserted explain actions in π' only serve bookkeeping purposes and thus π' still solves $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, B_{\textsc{Obs}} B_{\textsc{Act}} G\rangle$ (since π solves $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, B_{\textsc{Obs}} B_{\textsc{Act}} G\rangle$). Putting everything together, we have that

$$\textsc{prog}(\pi', \mathcal{I}') \models B_{\textsc{Obs}} B_{\textsc{Act}} G \wedge B_{\textsc{Obs}} p_m$$

and therefore π' achieves G' and solves $\langle Q', \mathcal{I}', G'\rangle$. Finally, since we were able to straightforwardly add the explain actions to the plan π to obtain π', we are able to equally straightforwardly remove them from π' to obtain π.

(2) We prove that if π' is a plan that solves $\langle Q', \mathcal{I}', G'\rangle$ then there exists a plan π that solves $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, B_{\textsc{Obs}} B_{\textsc{Act}} G\rangle$ and satisfies O such that π can be reconstructed straightforwardly from π' by removing the explain actions from π'. Since π' achieves G', we have that

$$\textsc{prog}(\pi', \mathcal{I}) \models B_{\textsc{Obs}} B_{\textsc{Act}} G \wedge B_{\textsc{Obs}} p_m$$

and in particular that

$$\textsc{prog}(\pi', \mathcal{I}) \models B_{\textsc{Obs}} B_{\textsc{Act}} G$$

to prove that π' satisfies O, we need to show that there is a function f mapping the observation indices $j = 1, ..., m$ into plan indices $i = 1, ..., n$ such that $a_{f(j)} = \alpha_j$, and $\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}) \models B_{\textsc{Obs}} \phi_j$ for $j = 1, \ldots, m$. Since $\pi' = a_1, \ldots, a_n$ achieves $B_{\textsc{Obs}} p_m$, by the construction of the transformed RP-MEP problem it must also hold that

$$\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}') \models B_{\textsc{Obs}} p_j$$

for every $o_j \in O$. p_j can only hold if the two explain actions corresponding to o_j are executed. For that to happen, both actions must be executable, i.e.,

$$\textsc{prog}([a_1, \ldots, a_{f(j)}], \mathcal{I}') \models B_{\textsc{Obs}}(l_{\alpha_j} \wedge p_{j-1})$$

and

$$\text{PROG}([a_1, \ldots, a_{f(j)}], \mathcal{I}') \models B_{\text{OBS}}(\phi_j \wedge p_{\alpha_j})$$

must hold. It is clear that the conditions for satisfaction hold for o_j. Since this holds for every observation $o_j \in O$, we have that π' satisfies O. Finally, the explain actions in π' are only used for bookkeeping and do not change the state of the world. Thus, they can be removed to obtain π. π solves $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, B_{\text{OBS}} B_{\text{ACT}} G\rangle$ and satisfies O and so we are done. □

4.3.2 Computing $P(G|O)$

We build on the approach proposed by Ramírez & Geffner (2010), using epistemic planners instead of classical planners, and compute the probability distribution over $\mathcal{G}$, $P(G|O)$, given the transformed RP-MEP problem, $\langle Q', \mathcal{I}', G'\rangle$. To compute the probability distribution, two plans are generated for every goal $G \in \mathcal{G}$ – one where the planner is forced to satisfy O by achieving

$$G' = B_{\text{OBS}} B_{\text{ACT}} G \wedge B_{\text{OBS}} p_m$$

and one where the planner is forced to *not* satisfy O, by achieving

$$G' = B_{\text{OBS}} B_{\text{ACT}} G \wedge B_{\text{OBS}} \neg p_m.$$

We then define Δ as the difference between the costs of these two plans (subtracting the latter from the former) and use Δ to compute the probability of a goal. Formally, Bayes' Rule is used to compute

$$P(G|O) = \alpha P(O|G) P(G) \tag{4.1}$$

where α is a normalization constant and $P(G)$ is the prior probability of G, which we assume in this work to be uniform across $\mathcal{G}$. Finally, assuming a Boltzmann distribution following Ramírez & Geffner (2010), we have that

$$P(O|G) \approx \frac{e^{-\beta\Delta}}{1 + e^{-\beta\Delta}} \tag{4.2}$$

where β is a positive constant. Ramírez & Geffner assume a soft rationality postulate according to which *G is a better predictor of O when Δ is smaller*. Therefore, when

computing the distribution $P(G|O)$ in this way, the most likely goals will be those that minimize Δ. They moreover say that a goal G is a perfect predictor of O when all plans that achieve G satisfy O, since in that case $\Delta = -\infty$. This is because, in this case, the cost of achieving G and not satisfying O is infinite (∞). Computing Δ in this case means subtracting ∞ from a finite number (i.e., the cost of achieving G and satisfying O). Note that the actor's rationality is assumed with respect to the *observer's* model of the former. When the observer's beliefs about the actor's beliefs are not adequate, the quality of inferences may consequently suffer (as is also the case when Δ is computed with sub-optimal planners, as we will see in Section 4.4).

Ramírez & Geffner (2010) discuss additional assumptions underlying their probabilistic account of plan recognition (e.g., when the actor is pursuing a goal G, it is more likely to follow cheaper plans than more expensive ones) and conclude that these assumptions ensure that their account yields a consistent posterior distribution over the set of possible goals $\mathcal{G}$, given the observations. We make the same assumptions here and so the distribution produced by our account of epistemic plan recognition is also consistent over the set of possible goals $\mathcal{G}$.

Finally, Sohrabi et al. (2016) have shown that top-k planning (e.g., Katz et al., 2018, 2019, 2020) and diverse planning (e.g., Nguyen et al., 2012; Katz & Sohrabi, 2020) can be used to compute a posterior probability distribution, $P(\pi|O)$, directly over the set of plans that satisfy the observations and achieve some goal from the set of possible goals. As opposed to a typical planning problem, where a single plan is found, in top-k planning the planner attempts to find the k 'best' plans, according to some cost metric, while in diverse planning the objective is to find a set of plans that are at least some distance away from each other (where distance can be computed in a variety of ways). Further, Ramírez & Geffner's approach can theoretically be extended to compute the probability distribution directly over plans $P(\pi|O)$. In this way, we may compute the likelihood of a plan, rather than a goal, given the observations.

4.3.3 Computing a Solution to an Epistemic Plan Recognition Problem

In this section we present an algorithm that leverages off-the-shelf epistemic planners and uses the techniques described in Sections 4.3.1 and 4.3.2 to compute a solution to an epistemic plan recognition problem. In particular, in Line 3 of Algorithm 1, for each goal $G \in \mathcal{G}$, the function TransformEpistemicPlanRecognitionToRPMEP

Algorithm 1

1: **procedure** SolveEpistemicPlanRecognitionProblem($\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$) - Given an epistemic plan recognition problem $R = \langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$, return ($\pi$,$G$), where $G \in \mathcal{G}$ is a goal and π is a plan such that π satisfies O and $\textsc{prog}(\pi, \mathcal{I}) \models B_{\text{Obs}} B_{\text{Act}} G$.
2: **for each** $G \in \mathcal{G}$ **do**
3: $\langle Q', \mathcal{I}', G' \rangle \leftarrow$ TransformEpistemicPlanRecognitionToRPMEP(R, G)
4: $\Delta \leftarrow$ ComputeDelta($\langle Q', \mathcal{I}', G' \rangle$)
5: $P(G|O) \leftarrow$ ComputeProbability(Δ)
6: **end for**
7: $G^m \leftarrow \arg\max_{G \in \mathcal{G}}(P(G|O))$
8: $\pi \leftarrow$ RetrieveAssociatedPlan(G^m)
9: **return** (π, G^m)
10: **end procedure**

transforms an epistemic plan recognition problem R and a goal $G \in \mathcal{G}$ to an RP-MEP problem, as described in Section 4.3.1. In Line 4, ComputeDelta runs a planner twice on the transformed RP-MEP problem – once with $G' = B_{\text{Obs}} B_{\text{Act}} G \wedge B_{\text{Obs}} p_m$ and once with $G' = B_{\text{Obs}} B_{\text{Act}} G \wedge B_{\text{Obs}} \neg p_m$. In Line 5, ComputeProbability uses the cost difference between the plans produced in Line 4, Δ, to compute a posterior probability $P(G|O)$ for $G \in \mathcal{G}$, using Equation 4.2. In Line 7, we select the goal G^m with the highest posterior probability $P(G|O)$ (we randomly break ties between goals). In Line 8, the function RetrieveAssociatedPlan retrieves the plan π (forced to satisfy O) that was generated in Line 4 in order to compute Δ for G^m. Lastly, π and G^m are returned in Line 9.

Assumptions

In both our computation and experimentation (except in Section 4.4.3, where explicitly noted) we assume the following:

- The observer's beliefs about the actor's beliefs are adequate.
- The actor is assumed to only pursue one goal $G^* \in \mathcal{G}$ at a given time. We refer to this goal in the next section as the actor's *hidden* goal, as it is not known to the observer.
- We assume that there exists a plan π such that π satisfies O and $\textsc{prog}(\pi, \mathcal{I}) \models B_{\text{Obs}} B_{\text{Act}} G^*$.

Algorithm 1 computes solutions from a subset of epistemic plan recognition solutions where the plan component π has the added property of the observer believing

that the actor believes G following π's execution, as stated by the following theorem.

Theorem 4.7. *Given an epistemic plan recognition problem, $\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$, and a sound and complete epistemic planning algorithm, Algorithm 1 returns a pair (π, G) where $G \in \mathcal{G}$ is a goal and π is a plan such that π satisfies O and $\textsc{prog}(\pi, \mathcal{I}) \models B_{\text{Obs}} B_{\text{Act}} G$ holds, if such a solution exists.*

Proof: In Line 9, Algorithm 1 returns (π, G^m), where $G^m \in \mathcal{G}$. π solves the transformed RP-MEP problem corresponding to the given epistemic plan recognition problem and G^m, and it follows from Theorem 4.6 that π satisfies O and moreover that $\textsc{prog}(\pi, \mathcal{I}) \models B_{\text{Obs}} B_{\text{Act}} G^m$ holds. □

4.4 Experimental Evaluation

In this section, we present the results of an experimental evaluation of our implemented algorithm. In our evaluation, we set out to:

1. Demonstrate that existing epistemic planners can straightforwardly be used to solve epistemic plan recognition problems and recognize epistemic goals pursued by an actor
2. Compare the performance of existing epistemic planners (both optimal and satisficing) in terms of computation time and their ability to assign the highest probability to the hidden goal the actor is pursuing
3. Evaluate the impact of the required depth of nested belief on the runtime of Algorithm 1
4. Evaluate the impact of the (in)adequacy of the observer's beliefs about the actor's beliefs on the quality of goal recognition inferences

4.4.1 Experimental Setup - Recognizing Epistemic Goals

While Algorithm 1 produces a solution to an epistemic plan recognition problem comprising both a goal G and a plan π that satisfies O, we focus here on the goal recognition task - inferring the actor's hidden goal $G^* \in \mathcal{G}$. To this end, we compute $P(G|O)$ in Line 5 of Algorithm 1 to determine the likelihood of goals in $\mathcal{G}$ given the sequence of observations and evaluate whether G^* was assigned the highest probability. In this section we are interested in recognizing an actor's *epistemic goal*. To this end, we ran Algorithm 1 on two adapted epistemic planning domains:

- In the *Grapevine*[3] (GV(n,m)) domain, n guests are in a villa with m rooms. Initially, each guest has her own secret to share with other guests. In this modified version, only one agent (the actor) can freely move between the rooms and share her secrets with the guests in the room she is currently in. Possible epistemic goals include misconception (an agent holds a false belief about someone else's belief), a universal spread of information (e.g., all agents obtaining all secrets), and a partial spread of information (e.g., some guests obtaining some (or no) secrets).

- In the *Selective-Communication*[4] (SC(n,m)) domain, there are n agents distributed amongst m rooms along a corridor, and one agent (the actor) can freely move between neighboring rooms. After an agent shares information, all agents in the room and in neighboring rooms believe the information. The agent must first find out the information before being able to share it with other agents. Possible epistemic goals include agents believing some or all of the information.

To create an epistemic plan recognition problem instance, $\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$, we populated a set of possible *epistemic* goals, $\mathcal{G}$, for each planning instance in the problem set (in total, we had 10 planning instances for GV and 10 for SC), where $\mathcal{G}$ comprised the hidden goal $G^* \in \mathcal{G}$ (the hidden goal pursued by the actor) and 6 other possible goals (in total, $|\mathcal{G}| = 7$ for all problem instances). We generated the sequence of observations, O, for each problem instance by sampling hidden optimal plans for the hidden goal G^* concatenated to the RP-MEP domain within the epistemic plan recognition problem. Concretely, an optimal epistemic planner was tasked with solving $\langle \langle \mathcal{P}, \mathcal{A}, Ag \rangle, \mathcal{I}, B_{\text{OBS}} B_{\text{ACT}} G^* \rangle$. When sampling an action from a hidden optimal plan, the corresponding observation (α_i, ϕ_i) comprised the sampled action $\alpha_i \in \mathcal{A}$. In addition, $\phi_i \in \mathcal{P}$ (the observed properties of states immediately following α_i's execution) was obtained by randomly sampling 30% of fluent atoms $p \in \mathcal{P}$ in the state resulting from executing the prefix of the hidden optimal plan ending in α_i. Concretely, observations in the GV and SC domains included the share and move actions performed by the actor, as well as state observations comprising agent locations (since the epistemic effects of the share action are not observable).

Moreover, for each problem instance we varied the percentage of observations sampled from the hidden optimal plan: 10%, 40%, 70%, and 100%. This is common

[3]Based on the original domain, found in (Muise, 2021c)
[4]Based on the original domain, found in (Muise, 2021b)

practice in the evaluation of plan and goal recognition systems. Intuitively, when fewer observations are given to the system, it is harder to accurately infer the actor's hidden goal and plan. Finally, to evaluate the impact of the required depth of nested belief, d, on the runtime of Algorithm 1 for the different planners, we varied the value of d for each problem instance: $d = 1$ and $d = 3$.

We ran Algorithm 1 for each problem instance using the latest version of 4 off-the-shelf planners – RP-MEP (Muise et al., 2015b), MEPK (Huang et al., 2017), EFP 1.0 (Le et al., 2018), and EFP 2.0 (Fabiano et al., 2020). In Line 4, to compute Δ with RP-MEP (which encodes the RP-MEP problem as a classical planning problem) we call the Fast Downward planner (Helmert, 2006) with an admissible heuristic twice for each goal, configuring the planner to only compute optimal plans. MEPK and the EFP planners are all optimal planners and we ran each of them twice for each goal in each problem instance to find Δ. We ran all planners on a 3.4GHz Intel Core i5 machine with 16 GB RAM. Timeout was 30 minutes.

4.4.2 Results - Recognizing Epistemic Goals

Table 4.1 summarizes the results for the GV and SC domains, offering a comparison between the four different optimal epistemic planners we used. Each row in the table is an average over ten epistemic plan recognition problem instances, with a varying percentage of observations sampled from the hidden optimal plan: 10%, 40%, 70%, and 100%. As mentioned, we also vary d, the required depth of nested belief modalities in the problem instance. The T column represents the average time it took to run Algorithm 1 and solve the epistemic plan recognition problem instances. For example, it took RP-MEP, on average, 3.87 seconds to solve problem instance in the $GV(2,3)$ setting, with 70% of sampled observations and $d = 1$.

The Q column represents the quality of the solution, i.e., the fraction of problems where the hidden goal G^* is among the most likely goals, given the observations. Q values are obtained using the probability $P(G|O)$ computed in Line 5 of Algorithm 1. For example, $Q = 0.89$ signifies that in 89% of problem instances, the hidden goal was amongst the goals found to be most likely[5]. The variances ranged 0-1.37 and 2.12-3.91 for the Q and T values (for a particular row and planner), respectively. TO and n/a signify a timeout and inability to model the problem, respectively. In particular, we were not able run EFP 1.0 and PG-EFP (which will be discussed shortly) on $GV(4,5)$ problems. This is because, in order to prevent memory overflow, the EFP 1.0 and

[5]The subset of goals from $\mathcal{G}$ that are tied for the highest assigned posterior probability is considered the set of most likely candidates for G^*, the actor's hidden goal.

			RP-MEP		MEPK		EFP 1.0		EFP 2.0	
	%O	d	T	Q	T	Q	T	Q	T	Q
$GV(2,3)\|L=4$	10	1	1.72	1	0.12	1	1.49	1	0.83	1
	40	1	2.92	1	0.13	1	11.91	1	2.79	1
	70	1	3.87	1	0.14	1	37.82	1	3.59	1
	100	1	4.76	1	0.22	1	58.12	1	5.19	1
	10	3	549.24	1	0.31	1	1.49	1	0.83	1
	40	3	572.87	1	0.39	1	11.91	1	2.79	1
	70	3	591.43	1	0.41	1	37.82	1	3.57	1
	100	3	601.76	1	0.46	1	58.12	1	5.17	1
$GV(4,5)\|L=11$	10	1	576.23	0.89	25.76	0.89	n/a	n/a	84.12	0.89
	40	1	584.86	1	54.43	1	n/a	n/a	97.34	1
	70	1	590.21	1	85.35	1	n/a	n/a	89.53	1
	100	1	598.19	1	TO	TO	n/a	n/a	114.67	1
	10	3	739.89	0.89	26.89	0.89	n/a	n/a	85.81	0.89
	40	3	753.91	1	57.12	1	n/a	n/a	95.81	1
	70	3	772.83	1	88.18	1	n/a	n/a	89.91	1
	100	3	801.79	1	TO	TO	n/a	n/a	113.04	1
$SC(8,4)\|L=7$	10	1	67.12	0.85	0.36	0.85	324.89	0.85	154.82	0.85
	40	1	75.98	1	0.48	1	395.21	1	161.92	1
	70	1	77.63	1	0.61	1	457.21	1	188.36	1
	100	1	79.31	1	0.71	1	503.36	1	191.78	1
	10	3	912.54	0.85	9.54	0.85	324.89	0.85	152.55	0.85
	40	3	925.08	1	14.21	1	395.21	1	161.43	1
	70	3	949.21	1	18.48	1	457.21	1	188.36	1
	100	3	964.76	1	24.01	1	503.36	1	187.03	1

Table 4.1: Comparison between four optimal epistemic planners (RP-MEP, MEPK, EFP 1.0, and EFP 2.0) in the Grapevine (GV) and Selective-Communication (SC) domains. Each row describes averages over ten epistemic plan recognition problems, where the columns stand for % of actions sampled (%O), required depth of nested belief (d), average time in seconds to solve problem instance (T), average quality measuring fraction of problems where the hidden goal is among the most likely (Q). TO and n/a signify a timeout and inability to model the problem, respectively. $|\mathcal{G}| = 7$ in all problems.

PG-EFP planners put a limitation on the number of fluents in the initial state that are not known by all agents and $GV(4,5)$ instances required a number of fluents greater than that allowed by these planners. EFP 2.0 eschews this limitation which is why we were able to run it on $GV(4,5)$ problems.

The value Q is 1 in most rows. That is, the hidden goal G^* was found to be most likely in all problem instances. This is expected when sampling observations from optimal hidden plans and aligns with Ramírez & Geffner's (2010) results. Note, however, that in some cases where $\%O$ is 10, $Q < 1$. Recall that to obtain $P(G|O)$,

we first compute Δ, the cost difference between a plan that satisfies O and a plan that does not. As discussed by Ramírez & Geffner, it may be the case that for some goal $G' \in \mathcal{G}$, where $G' \neq G^*$, there does not exist a plan that does not satisfy O (more precisely in the rows where $\%O$ is 10, a sampled 10% of O), but there does exist such a plan for G^* (albeit not an optimal one). Therefore, the value of Δ for G' will be *smaller* $(-\infty)$ than the value of Δ for G^* (some finite number) and thus G' will be deemed more likely. This relates to the discussion of adequacy in Section 4.2 where we posited that plan recognition systems are limited by the veracity and completeness of the observer's knowledge and beliefs which include the sequence of observations – in this case, only 10% of the observation sequence is available, which hurts recognition accuracy.

With respect to our **first and second objectives**, the results present a comparison between existing epistemic planners and demonstrate that these planners can indeed be used to solve epistemic plan recognition problems and accurately recognize an actor's epistemic goals. With respect to our **third objective**, Table 4.1 demonstrates a clear impact of the depth of nested belief, d, on Algorithm 1's runtime – when d grows, so does the runtime. This impact, however, is influenced by the planner used in Algorithm 1. Indeed, the performance, in seconds, of the planners with respect to d is consistent with Le et al.'s (2018) investigation of the impact of d on planner runtime. For instance, when $d = 3$ EFP 1.0's and 2.0's performance is not affected whereas RP-MEP slows down considerably. This is because the representation used by the EFP systems is based on accessibility relations and thus increasing the depth of nested belief has little impact compared to RP-MEP's compilation based approach. Lastly, as shown by Fabiano et al. (2020), EFP 2.0 outperforms EFP 1.0 across all domains. As discussed by Fabiano et al., this performance boost is due to improvements to the epistemic state representation as well as the transition function in the updated implementation of the EFP planner. We will discuss some limitations of our approach and existing epistemic planners in Section 4.5.

Satisficing Epistemic Planners

To compare with the optimal epistemic planners, we ran all GV and SC experiments with RP-MEP, coupled with a *satisficing* configuration of Fast Downward (Helmert, 2006). We also experimented with PG-EFP, a satisficing heuristic search planner that employs a heuristic derived from an epistemic planning graph (Le et al., 2018). While in our experiments EFP 1.0 and EFP 2.0 leveraged breadth first search to obtain optimal plans, plans produced by PG-EFP are not guaranteed to be optimal.

			RP-MEP (S)		PG-EFP	
	%O	d	T	Q	T	Q
$GV(2,3)\|L=4$	10	1	0.84	0.37	1.53	0.72
	40	1	1.04	0.42	11.43	0.73
	70	1	1.57	0.63	28.43	0.82
	100	1	1.82	0.85	64.02	0.90
	10	3	60.42	0.34	1.54	0.74
	40	3	102.43	0.48	11.38	0.73
	70	3	96.49	0.69	28.43	0.77
	100	3	170.42	0.88	64.02	0.92
$GV(4,5)\|L=11$	10	1	168.42	0.33	n/a	n/a
	40	1	152.89	0.39	n/a	n/a
	70	1	172.44	0.54	n/a	n/a
	100	1	162.91	0.71	n/a	n/a
	10	3	250.37	0.32	n/a	n/a
	40	3	248.11	0.45	n/a	n/a
	70	3	302.78	0.58	n/a	n/a
	100	3	296.54	0.74	n/a	n/a
$SC(8,4)\|L=7$	10	1	5.42	0.33	345.12	0.81
	40	1	2.21	0.41	199.47	0.71
	70	1	1.94	0.64	239.65	0.93
	100	1	3.05	0.78	240.17	0.91
	10	3	256.24	0.35	345.12	0.81
	40	3	225.41	0.39	200.52	0.73
	70	3	249.28	0.65	345.11	0.93
	100	3	297.53	0.86	232.77	0.91

Table 4.2: Comparison between two satisficing epistemic planners (PG-EFP and a satisficing congifuration of RP-MEP (RP-MEP (S))) in the Grapevine (GV) and Selective-Communication (SC) domains. Each row describes averages over ten epistemic plan recognition problems, where the columns stand for % of actions sampled (%O), required depth of nested belief (d), average time in seconds to solve problem instance (T), average quality measuring fraction of problems where the hidden goal is among the most likely (Q). n/a signifies an inability to model the problem. $|\mathcal{G}| = 7$ in all problems.

The results are reported in Table 4.2.

In the case of RP-MEP, the satisficing planner was, on average, much faster than the optimal planner we used. However, the accuracy (Q) decreased significantly (particularly with incomplete observations) since the satisficing planner often generated highly suboptimal plans. In turn, the quality of inferences suffered as Δ was computed with low-quality plans, which differed greatly from the optimal ground truth.

As for PG-Efp, we observed a reduction in computation time in some cases (consistent with Le et al.'s results). Interestingly, the accuracy did not suffer greatly

as the planner typically found optimal plans. Indeed, Ramírez & Geffner (2010) have shown that satisficing classical planners can be successfully used in plan recognition to greatly reduce computation time, without significantly hurting accuracy. Thus, the epistemic plan recognition as epistemic planning approach will benefit from future research on satisficing epistemic planners.

4.4.3 When the Observer Has Inadequate Beliefs

In Section 4.2 we identified conditions under which an observer's beliefs about the actor's beliefs are adequate and discussed the importance of adequacy in the epistemic plan recognition context. While we have assumed so far in this section that the observer has adequate beliefs, in this section we relax this assumption and report on our evaluation of the impact of the veracity of the observer's beliefs on goal recognition accuracy. Consider the following scenario which is inspired by Talamadupula et al.'s (2014) work, set in a search and rescue setting. In our scenario, Eve (the actor) is in need of assistance from either Alice or Bob (the observer). The three agents are each initially located in a different cell on a grid. Since Alice moved to a different location unbeknownst to Eve, the latter holds a false belief pertaining to Alice's location. Bob believes that Alice moved and also believes that Eve is not 'aware' of the move. Lastly, Eve is observed to be heading from her initial location to some location which is on the way to where she believes Alice is.

We modelled this scenario as an epistemic plan recognition problem and performed a set of simulations by varying the parameters of this problem (grid size and agent locations), resulting in 20 problem instances. $\mathcal{G}$ contains all cells on the grid. The observations were sampled from Eve's hidden plan which consists of her making her way to where she falsely believes Alice is (and thus Alice's incorrect location is Eve's hidden goal, $G^* \in \mathcal{G}$). We created two copies of each problem instance, where either:

- Bob the observer holds **adequate** beliefs about Eve's beliefs, or
- Bob the observer holds **inadequate** beliefs about Eve's beliefs.

To create problem instances where Bob's beliefs are not adequate, we modified the initial state $\mathcal{I}$ such that Bob believes that Eve believes Alice's true location, when in fact she does not. We ran both copies of each problem instance using Algorithm 1 with RP-MEP to compute $P(G|O)$ over $\mathcal{G}$. As shown in Table 4.3, 'adequate' Bob was able to assign the highest probability to Eve's hidden goal (getting to where she believes Alice is) in most cases, compared to 'inadequate' Bob, who performed poorly.

Observer Type	$\%O = 10$	$\%O = 30$
Observer w/ Inadequate Beliefs	10	15
Observer w/ Adequate Beliefs	90	100

Table 4.3: **When the observer's beliefs are inadequate, the quality of goal recognition inferences suffers.** Comparison of an observer with adequate beliefs about the actor's beliefs and one with inadequate beliefs, on 20 problem instances. The values represent the percentage of problems in which the actor's hidden goal was assigned the highest probability given the observations, relative to the percentage of total observations ($\%O$).

With respect to our **fourth objective**, the results demonstrate that when Bob's beliefs about Eve's beliefs are inadequate, the quality of inferences indeed suffers. Further, suppose Bob is trying to infer whether Eve is looking for him or for Alice. 'Inadequate' Bob would infer that given the observations it is more likely that Alice is heading towards his location (one of the goals in $\mathcal{G}$) than towards Alice's *actual* location which is in the opposite direction of Eve's trajectory and is thus assigned a lower probability. However, 'adequate' Bob's beliefs about Eve's beliefs allow him to reason that the observations put Eve on the optimal path to where *she believes* Alice is located. Thus, Bob can reason that Eve is looking for Alice and that her presumed plan will work 'from her perspective'.

Finally, we note that, more generally, an 'inadequate' observer can do much better or worse than 'inadequate' Bob in our simulations, depending on the relevance of what they (do not) know. For example, if the observer has a false belief about the color of the actor's socks, that probably will not greatly impact recognition accuracy. However, if the observer has a false belief about the actor's belief about another agent's location (as is the case in our search & rescue simulations), there may be severe consequences when trying to recognize the actor's goal (as evident by Table 4.3). In Chapter 6, we build on our work in this chapter and evaluate the impact of the veracity of an observer's beliefs on its ability to *assist* the actor in a myriad of domains.

4.5 Discussion

In this section we discuss the work presented in this chapter, including some limitations and possible extensions.

Scalability/Expressivity Tradeoff

While the results of our experimentation are promising, they do not come without limitations. Firstly, the runtime complexity of Algorithm 1 is dominated by the two calls to the chosen epistemic planner. For instance, Huang et al.'s planner (MEPK) uses a satisfiability solving algorithm which has an exponential time complexity. Furthermore, the encoding process in Muise et al.'s planner (RP-MEP) generates an exponential number of fluents when classically encoding the problem. Our approach calls an epistemic planner twice for each goal and will therefore benefit from advances in epistemic planning including faster computation and satisficing planners.

As mentioned in the discussion of adequacy in Section 4.2, plan recognition systems are limited by the veracity and completeness of the observer's beliefs. Even if the observer's beliefs about the actor's beliefs are adequate, the planner used may not be sound and complete, in which case it may not generate all plans the actor can generate. Indeed, a challenge of epistemic planning (as well as conformant and contingent planning, which involve planning under uncertainty) is the issue of belief space representation to make planning computationally feasible, often at the expense of planner completeness and applicability.

For example, conformant and contingent planners often appeal to belief state approximations such as 0-approximation (Son & Baral, 2001) and related work (e.g., Palacios & Geffner, 2007). In epistemic planning, syntactic restrictions or approximations (e.g., the syntactically restricted PEKBs used by RP-MEP) may restrict applicability or limit completeness of planners and are traded off against scalability. For instance, if the depth of nested belief is restricted to 2, formulae such as $B_{John}B_{Alice}\phi$ may appear in the knowledge base but $B_{John}B_{Alice}B_{Bob}\phi$ may not, which may prevent the observer from inferring the actor's plan or goal. Another such syntactic restriction is the exclusion of disjunctive belief, as is done in our computation and experiments and in PEKBs more generally. Indeed, extant work (e.g., Cooper et al., 2016; Miller et al., 2016) has shown that representing that an agent *knows whether* ϕ (a restricted form of disjunction) can facilitate inference about that agent's plans and moreover that such inference would be more difficult or even not possible without representing the know whether modality. It is therefore important, in the epistemic plan recognition setting, to consider the trade-off between computational feasibility and the expressivity of the chosen fragment of logic.

As mentioned, we appeal to the multi-agent modal logic KD45_n to model the beliefs of agents in the environment. While RP-MEP and MEPK can handle KD45_n,

EFP 1.0 operates over $S5$, which cannot address false beliefs[6]. Nonetheless, the epistemic planning benchmarks used in our evaluation do not require the modeling of false beliefs and for this reason we were able to compare the KD45 planners with EFP 1.0. More recently, and as mentioned in Section 4.4, Fabiano et al. (2020) optimized and refactored the original EFP implementation and developed EFP 2.0, which is also included in our evaluation. This implementation does permit some form of false belief and has been compared to the RP-MEP system by Muise et al. (2021).

Online Recognition and Runtime Optimization

In our experimentation, we focused on a setting where recognition is done post-hoc. Future work will explore computational approaches that are better geared towards an online recognition setting, where the observer is attempting to recognize a plan that is in-progress (e.g., Freedman & Zilberstein, 2017; Vered et al., 2018). Further, we may utilize existing landmark-based approaches to goal recognition (e.g., Pereira et al., 2017; Vered et al., 2018) to compute a probability distribution over $\mathcal{G}$ given the classical planning problem that results from Muise et al.'s compilation. These approaches have been shown to be much faster than Ramírez & Geffner's approach, while achieving similar recognition accuracy (Pereira et al., 2017). While the landmark-based approaches offer attractive computation time, it no longer views RP-MEP (and epistemic planners more generally) as a black box. Additionally, the landmark-based approaches are geared towards goal recognition and thus do not also infer a plan that explains the inferred goal as is done in the plan recognition as planning approach we employ in this chapter.

Decentralized Multi-agent Setting

The epistemic plan recognition model in this work captures a fixed observer which is adequate in many settings such as tutoring systems and conversational systems. Like centralized vs distributed control, this model affords computational advantages over a more general setting. Our account of epistemic plan recognition could be extended to a decentralized multi-agent setting where each agent is both an actor and an observer and holds epistemic plan recognition capabilities, with a view to active collaboration or adversarial interactions.

[6]We note that the language upon which EFP 1.0 builds, $m\mathcal{A}^*$ (Baral et al., 2012; Pham et al., 2023), is highly expressive and the EFP 1.0 implementation only utilizes a fragment of it.

Obfuscating Actor

In cases where the actor is aware of being observed, modeling their beliefs about the observer can be useful as these could affect the actor's behavior. For example, an actor keen on obfuscating its goal or plan might purposefully generate an ambiguous plan that is predicated on the actor's beliefs about the observer's beliefs (Keren et al., 2016; Masters & Sardina, 2017; Kulkarni et al., 2019). This idea is also investigated by Masters et al. (2021) who relax the prevalent assumption of observer *infallibility* and propose a model that can be used by an adversarial actor to exploit the fallibility of an observer – its confirmation bias, selective attention and memory decay – for the purpose of strategic deception. Our epistemic plan recognition specification can be extended in the future to accommodate various approaches to plan and goal obfuscation (or legibility (e.g., Chakraborti et al., 2019)).

Is the Observer Really a First-Class Citizen in the Recognition Process?

Given the formulation of epistemic plan recognition presented in this chapter (where the observer is the root agent), it is not entirely accurate to claim that the observer is a first-class citizen in the recognition process since all reasoning is done relative to its beliefs. This is a reasonable modeling choice (indeed, in Chapter 6 we illustrate this claim in the context of an assistive robotics application) unless we wish to model the plan recognition problem from the perspective of a third party that is neither the observer nor the actor. In a general multi-agent epistemic plan recognition setting, this might be desired. For example, we could have an observer who is observing an observer and an actor (who might themselves be an observer). We leave such interesting settings to future work.

4.6 Related Work

Plan recognition, as a field of research, has a long history and was originally seen as an intersection of psychology and AI (Schmidt et al., 1978). Early accounts of plan recognition largely utilized (possibly hierarchical) plan libraries in order to best match a sequence of observations to a particular plan (e.g., Kautz & Allen, 1986). Conceptually related, a large body of work cast the problem of plan recognition as a form of parsing, using a formal grammar capturing possible plans the actor may be pursuing (e.g., Vilain, 1990; Pynadath & Wellman, 2000; Geib & Goldman, 2009, 2011). In recent years, a number of comprehensive surveys of the past, present, and

future of the field have come out (e.g., Sukthankar et al., 2014; Mirsky et al., 2021; Van-Horenbeke & Peer, 2021).

Another trajectory of research in the field has dispensed with the need for a plan library by casting the plan recognition problem as a planning problem and leveraging advances in AI planning research to perform plan recognition (e.g., Ramírez & Geffner, 2010; Sohrabi et al., 2016). For further reading, see the comprehensive survey on plan recognition as planning approaches by Meneguzzi & Pereira (2021). As discussed in Section 4.3, it is this line of work we appeal to when computing solutions to epistemic plan recognition problems via epistemic planning.

In this chapter we appealed to a notion of epistemics and posited that a general account of plan recognition should provide a means of explicitly modeling the beliefs of the observer about the world and about other agents (most notably the actor) and enable the recognition of epistemic goals. Most previous work in plan recognition, however, has not satisfied our posited desiderata for general accounts of plan recognition. Indeed, Masters & Vered's (2021) comprehensive examination of implicit and explicit assumptions in goal recognition reveals that most works do not satisfy our first desideratum by not distinguishing between the models of the observer and the actor. In such cases, as aptly pointed out by Masters & Vered, the observer can simply ask itself *"If I were performing [the observed] actions, what would I be trying to achieve?"* (Masters & Vered, 2021), rather than reasoning about the actor's unique perspective. There are, however, notable exceptions that have partially satisfied our posited desiderata and we discuss these exceptions in the remainder of this section. For instance, Talamadupula et al. (2014) integrate plan recognition, belief modeling, and AI planning in a human-robot teaming scenario. While their work is indeed conceptually related to ours (and has inspired early iterations of our work, as well as some of the experiments in our evaluation (see Section 4.4.3)), and while it models aspects of agent beliefs, it does not address the recognition of epistemic goals nor does it utilize epistemic planning tools. Moreover, since our epistemic plan recognition formulation appeals to a rich multi-agent epistemic logic framework, it is able to naturally handle reasoning about the arbitrarily nested beliefs of multiple agents, which Talamadupula et al.'s approach cannot model.

Plan recognition has also been studied within the vast body of work on Belief-Desire-Intention (BDI) agents and architectures (Bratman, 1987), where agent beliefs are explicitly modelled (e.g., Sindlar et al., 2008). However, BDI approaches have typically required agent plans to be specified in advance. Pre-defined libraries are especially prohibitive when agents have misconceptions and generate invalid plans

(Pollack, 1986). Instead, enabled by the advent of epistemic planning research, we appeal to the flexibility of generative epistemic planning techniques to generate plans that are used to recognize the actor's plan and goal.

Moreover, the body of work on Bayesian Theory of Mind (Baker et al., 2011; Baker & Tenenbaum, 2014) proposes a Bayesian approach to modeling Theory of Mind reasoning. Importantly, this body of work distinguishes between the beliefs of the observer and those of the actor. In addition, the Bayesian Theory of Mind approach allows an observer to improve its adequacy of the actor's beliefs and even preferences. Baker et al. (2011) present a domain where a hungry agent has a set of preferences over three different food trucks. Only two trucks are on campus on a given day and, initially, the agent can only see one truck and must round a corner to determine the identity of the remaining truck. Based on the agent's actions, i.e., whether she returns to the first food truck or eats at the more distant truck, her preferences may be inferred. Further, Baker et al. (2011) conducted a study with human participants that elucidated the importance of jointly reasoning about other agents' beliefs and desires when attempting to attribute mental states. Lastly, more recent work extended the Bayesian Theory of Mind framework to model *boundedly rational* agents who may have mistaken goals, plans, and actions (Alanqary et al., 2021). While the motivation behind our work and the body of work on Bayesian Theory of Mind overlaps, the formalisms used are different, in addition to our framework's ability to recognize epistemic goals.

Further, Persiani & Hellström (2021) address an important setting where the actor's model is not known apriori and must be created by the observer via Theory of Mind reasoning. Rabkina et al. (2020) have also appealed to Theory of Mind (building on Rabkina et al.'s (2017) computational model of Analogical Theory of Mind) to do goal recognition. In particular, Analogical Theory of Mind allows an agent to learn through observation about the mental states of other agents. Persiani & Hellström's and Rabkina et al.'s works relate to the discussion of adequacy in this chapter and demonstrate how an observer can use prior assumptions and observations about the actor's behavior to improve the adequacy of its beliefs about the actor's beliefs and planning model.

Finally, as mentioned, Pollack (1986) espoused the view that plan recognition can necessitate distinguishing between the beliefs of the observer and the actor (see also (Cohen & Perrault, 1979; Perrault & Allen, 1980)). Pollack introduces a plan inference model in a collaborative dialogue setting where an observer is tasked with inferring the actor's plans, intentions, and beliefs from natural language dialogue

queries. Her work includes a fascinating discussion of the importance of recognizing different types of invalidities of queries (e.g., ill-formed and incoherent queries) and how these invalidities must shape the observer's response to the query. Importantly, such invalidities arise as a result of a discrepancy between the beliefs of the observer and the beliefs of the observer about the beliefs of the actor. While her work assumes a mapping from natural language utterances to beliefs and does not leverage planning techniques, Pollack's work and insights will certainly prove important in future dialogue systems that are sensitive to the beliefs of the user (e.g., as envisioned by Cohen (2020) and others).

4.7 Concluding Remarks

In this chapter, we have introduced the notion of epistemic plan recognition, which appeals to a rich epistemic logic framework to model the observer in the plan recognition setting, represent agent beliefs, and allow for the recognition of epistemic goals. We proposed a computational realization of epistemic plan recognition as epistemic planning that enables the use of existing planning tools. Finally, we performed an experimental evaluation of our approach on a set of epistemic plan recognition problems by utilizing existing epistemic planners.

While we focused in this chapter on the recognition task, plan recognition is typically not an end in itself. In Chapter 6 we will build on the techniques proposed in this chapter and show how an assistive robot is able to recognize another agent's plan and goal and subsequently assist that agent by resolving discrepancies between its beliefs and the other agent's beliefs pertaining to the *validity* of the other agent's presumed plan and goal. Chapter 5 will introduce techniques that allow for such discrepancy resolution.

Chapter 5

Discrepancy Resolution via Theory of Mind

In Chapter 4, we investigated the role of Theory of Mind in plan recognition and allowed an observing agent to reason about the plan and goal of other agents. However, as identified by Pollack (1986), *"inferring another agent's plan means figuring out what actions they 'have in mind,' and they may well be wrong about the effects of those intended actions"*. Indeed, while *"planning is the art of thinking before acting"* (Haslum, 2014), a problem with thinking before acting is that the validity of the resultant plan is predicated on *beliefs* about the way the world is, rather than ground truth, and even if those beliefs are correct at the time of planning (and they may not be!), the actual state of the world may change prior to plan execution, invalidating the plan, sometimes unbeknownst to various agents. Moreover, agents may perceive discrepancies between their own beliefs and other agents' beliefs about the validity of plans (e.g., Alice believes that Bob's plan is not valid but that *he* believes it is).

In this chapter we wish to allow agents to contemplate each others' plans, realize when agents hold misconceptions about the validity of their plans or the plans of other agents, and resolve discrepancies pertaining to the validity of these plans by acting in the environment (e.g., communicating with another agent). For example, a robot could communicate to its human teammate that the conditions necessary to the success of her plan do not hold or, alternatively, the robot could act in the world to ensure that those conditions hold.

To contemplate another agent's beliefs and plans, agents must employ their Theory of Mind. To enable agents to employ their Theory of Mind, we appeal to epistemic logic and propose a framework that allows agents to identify and resolve discrepancies between their beliefs and the beliefs of others regarding plan validity. Importantly,

our framework allows agents to be aware of and reason about the mental states of human counterparts and offer assistance by resolving perceived discrepancies.

Indeed, recent work in Explainable AI Planning (XAIP) has stressed the need to consider the possibly incomplete and incorrect perspective of other agents when resolving misconceptions pertaining to various properties of plans (e.g., optimality and validity). For instance, the *model reconciliation* literature has investigated how to enable planning agents to resolve discrepancies between the planning models of the planning agent and the observing human(s) (Sreedharan et al., 2021). In Section 5.5 we survey the body of related extant work.

Our work goes beyond extant work by supporting a unique variety of settings requiring complex Theory of Mind reasoning. In particular, the expressive nature of our framework supports (1) nested belief attribution (e.g., in order to resolve Mary's misconception about Bob's beliefs about the validity of his plan, Alice may inform Mary that Bob holds a false belief about some fact relevant to the plan's success); and (2) reasoning about threats to the achievement of *epistemic goals* (e.g., if Bob's epistemic goal is for Mary to know ϕ without Eve knowing ϕ and Bob's robot teammate knows that – unbeknownst to Bob – Eve is within earshot, then the robot could inform Bob of Eve's proximity). As we did in Chapter 4, to realize our approach we establish a relationship between our proposed formulation of discrepancy resolution and epistemic planning (see extensive discussion in Chapter 2).

Main contributions

- We propose a formulation of discrepancy resolution that appeals to a multi-agent epistemic logic.
- We present an algorithm that resolves discrepancies via epistemic planning and establish its soundness.
- We demonstrate that epistemic planning tools can be used to resolve discrepancies via different modalities in various domains and evaluate the impact of the depth of nested belief on the runtime of our algorithm.
- We conduct a user study which indicates that our approach can effectively resolve misconceptions held by humans pertaining to plan validity.

Relationship to Published Work

This chapter is based on our ICAPS 2022 publication (Shvo et al., 2022c).

Chapter Structure

In Section 5.1, we propose a formulation of discrepancy resolution that appeals to a multi-agent epistemic logic. In Section 5.2, we present an algorithm that resolves discrepancies via epistemic planning and establish its soundness. In Section 5.3, we demonstrate that epistemic planning tools can be used to resolve discrepancies via different modalities (i.e., with epistemic communicative and/or ontic world-altering actions) in various domains and evaluate the impact of the depth of nested belief on the runtime of our algorithm. In Section 5.4, we discuss a user study we conducted which indicates that our approach can effectively resolve misconceptions held by humans pertaining to plan validity. Finally, in Section 5.5 we survey related extant work.

5.1 Resolving Discrepancies

In this section, we propose a formulation of discrepancy resolution for plan validity. Our formulation, similarly to the epistemic plan recognition formulation presented in Chapter 4, appeals to the multi-agent modal logic KD45$_n$ (discussed in Chapter 2, Section 2.2) to represent the beliefs of different agents. To do so in a dynamic setting, we moreover appeal to the RP-MEP formalism (discussed in Chapter 2, Section 2.3). Next, for a plan to achieve some goal – to be valid – a set of sufficient and necessary conditions must hold. Since all definitions and computation in the RP-MEP formalism are from the perspective of a so-called root agent, $\star \in Ag$, we care about the root agent's beliefs about whether or not the aforementioned set of sufficient and necessary conditions holds.

Definition 5.1 (Plan Validity)**.** *Given an RP-MEP domain $\langle \mathcal{P}, \mathcal{A}, Ag \rangle$ where $\star \in Ag$ is the root agent, a state S, and a goal G, the root agent believes a plan π is valid for achieving (or simply achieves) G from S if* $\textsc{prog}(\pi, S) \models B_\star G$.

We want a formula $\textsc{valid}(\pi, G)$ that captures plan validity in the sense that $\textsc{prog}(\pi, S) \models B_\star G$ if and only if $S \models B_\star \textsc{valid}(\pi, G)$. We characterize $\textsc{valid}(\pi, G)$ by appealing to regression rewriting (Waldinger, 1977; Reiter, 2001; Fritz & McIlraith, 2007; Rintanen, 2008), a form of pre-image computation that takes a formula and an action and returns the condition that is necessary to hold in the current state for the formula to hold in the state resulting from performing the action. For example, if the (informally specified) formula is 'robot is in room B' and the action is 'robot move from room A to room B', the regression would return 'robot is in room A' as (at least

part of) the condition that must hold in the current state for the formula to be true in the resulting state, following the execution of the action. Regression can be applied repeatedly to compute the condition that must be true in the initial state for the goal to hold in the state resulting from the execution of the actions in a sequential plan. Here we suppose that we have (given an RP-MEP domain) a *regression* operator REG which maps a formula ϕ and action sequence π to a formula $\textsc{reg}(\pi, \phi)$ which satisfies the property that for any state S, $S \models B_\star \textsc{reg}(\pi, \phi)$ if and only if $\textsc{prog}(\pi, S) \models B_\star \phi$. Finally, we say that $S \models B_\star \textsc{valid}(\pi, G)$ if and only if $S \models B_\star \textsc{reg}(\pi, G)$.

Importantly, since $\textsc{valid}(\pi, G)$ is a formula, we can talk about the root agent's beliefs about other agents' beliefs about it, *which we can interpret as indicating the root agent's beliefs about those agents' beliefs about whether π is a valid plan.* For instance, if $B_{\star,j}\textsc{valid}(\pi, G)$ holds in some state S, we say that the root agent believes that agent j believes that π is a valid plan for achieving goal G from state S. The root agent can also perceive *discrepancies* between its beliefs and its beliefs about the beliefs of other agents (about the beliefs of other agents) about formulae (e.g., a plan validity formula $\textsc{valid}(\pi, G)$).

Definition 5.2 (Discrepancy)**.** *Given an RP-MEP domain $\langle \mathcal{P}, \mathcal{A}, Ag \rangle$ where $\star \in Ag$ is the root agent, agent $j \in Ag$, and a (possibly empty) tuple $\vec{v} = \langle v_1, \ldots, v_n \rangle$ of agents in Ag, we say that a formula ϕ is a **discrepancy** perceived by the root agent in state S between its beliefs and those of agent j (about the beliefs of agent v_1 about the beliefs of ... about the beliefs of agent v_n) if one of the following conditions is entailed by S:*

1. $B_{\star,j,\vec{v}}\phi \wedge B_{\star,\vec{v}}\neg\phi$
2. $B_{\star,j,\vec{v}}\neg\phi \wedge B_{\star,\vec{v}}\phi$

We will be interested in discrepancies about formulae like $\textsc{valid}(\pi, G)$, i.e., in discrepancies about the validity of plans, and in enabling the root agent to resolve such discrepancies by acting in the environment. To this end, we cast the task of resolving a discrepancy perceived by the root agent between its beliefs and those of agent j as an *epistemic goal*, where the root agent needs to either change j's beliefs to align with its own, or change its own beliefs to align with j's beliefs. The following definition is of a plan that achieves this goal and ensures that in the end, a formula ϕ will *not* be a discrepancy perceived by the root agent between its beliefs and those of agent j (about the beliefs of agent v_1 about the beliefs of ... about the beliefs of agent v_n).

Definition 5.3 (Discrepancy Resolving Plan)**.** *Given an RP-MEP domain $\langle \mathcal{P}, \mathcal{A}, Ag \rangle$ where $\star \in Ag$ is the root agent, agent $j \in Ag$, a (possibly empty) tuple $\vec{v} = \langle v_1, \ldots, v_n \rangle$ of agents in Ag, initial state $\mathcal{I}$, and some formula ϕ, a* ***discrepancy resolving plan*** *for $\langle \mathcal{P}, \mathcal{A}, Ag \rangle$, j, $\vec{v}$, and ϕ is a plan π' such that the following holds:*

$$\begin{aligned} \textsc{prog}(\pi', \mathcal{I}) \models \\ & (B_{\star,j,\vec{v}}\phi \wedge B_{\star,\vec{v}}\phi) \vee \\ & (B_{\star,j,\vec{v}}\neg\phi \wedge B_{\star,\vec{v}}\neg\phi). \end{aligned}$$

As we are interested here in resolving discrepancies pertaining to agents' beliefs about the validity of plans, we define a special case of a discrepancy resolving plan that ensures that in the end, $\textsc{valid}(\pi, G)$ will *not* be a discrepancy perceived by the root agent between its beliefs and those of agent j (about the beliefs of agent v_1 about the beliefs of ... about the beliefs of agent v_n).

Definition 5.4 (Plan Validity Discrepancy Resolving Plan)**.** *Given an RP-MEP domain $Q = \langle \mathcal{P}, \mathcal{A}, Ag \rangle$ where $\star \in Ag$ is the root agent, agent $j \in Ag$, a (possibly empty) tuple $\vec{v} = \langle v_1, ..., v_n \rangle$ of agents in Ag, initial state $\mathcal{I}$, a plan π, and a goal G, a* ***plan validity discrepancy resolving plan*** *for $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G \rangle$ is a plan π' such that the following holds:*

$$\begin{aligned} \textsc{prog}(\pi', \mathcal{I}) \models \\ & (B_{\star,j,\vec{v}}\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\textsc{valid}(\pi, G)) \vee \\ & (B_{\star,j,\vec{v}}\neg\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\neg\textsc{valid}(\pi, G)). \end{aligned}$$

We henceforth refer to plan validity discrepancy resolving plans simply as discrepancy resolving plans. There are many ways to resolve a discrepancy, some of which are trivial or undesirable. For instance, suppose that the root agent is a planning system trying to explain the validity of its own plan, π, to agent j (Bob, the human user of the system). The system believes that π is valid while believing also that Bob believes that π is not valid. In this case, a valid discrepancy resolving plan, π', would be for the system to render π invalid which resolves the discrepancy – the system now believes that π is not valid and also believes that Bob believes it is not valid. However, this is an undesirable[1] solution since the system's intention was to

[1]Interestingly, there are cases where it is desirable for an agent to make *another* agent's plan invalid. For example, suppose that a homeowner has an incorrect belief about the security of their own home – that a plan to break into the home would fail. It would be useful for the homeowner's

convince Bob of π's validity. We therefore often wish to resolve discrepancies under certain conditions by constraining the discrepancy resolution epistemic goal specified in Definition 5.4.

Definition 5.5 (Constrained Discrepancy Resolving Plan)**.** *Given an RP-MEP domain $Q = \langle \mathcal{P}, \mathcal{A}, Ag \rangle$ where $\star \in Ag$ is the root agent, agent $j \in Ag$, a (possibly empty) tuple $\vec{v} = \langle v_1, ..., v_n \rangle$ of agents in Ag, initial state $\mathcal{I}$, a plan π, a goal G, and a logical formula Φ representing additional constraints, a* ***constrained discrepancy resolving plan*** *π' for $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G \rangle$ is a plan such that the following holds:*

$$\begin{aligned} &\textsc{prog}(\pi', \mathcal{I}) \models \\ &\quad [(B_{\star,j,\vec{v}}\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\textsc{valid}(\pi, G)) \vee \\ &\quad (B_{\star,j,\vec{v}}\neg\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\neg\textsc{valid}(\pi, G))] \wedge \Phi. \end{aligned}$$

Note that as written here, Φ imposes a constraint on what states the plan can end in. It might also be desirable to constrain the plan trajectory (e.g., to require some condition holds throughout the entire plan), as has been explored in the literature on temporally extended goals (e.g., Bacchus & Kabanza, 1998; De Giacomo & Vardi, 2000; Baier & McIlraith, 2006). Our definition could be extended to do that, though we will not pursue that further in this dissertation.

Each of the following definitions specifies different conditions involving Φ, resulting in two conceptually distinct ways to resolve discrepancies: (1) by changing the root agent's beliefs about agent j's beliefs about plan validity to align with the root agent's beliefs about plan validity, and (2) by changing the root agent's beliefs about plan validity to align with the root agent's beliefs about j's beliefs about plan validity.

Definition 5.6 (Root-agent-aligned Discrepancy Resolving Plan)**.** *Given an RP-MEP domain $Q = \langle \mathcal{P}, \mathcal{A}, Ag \rangle$ where $\star \in Ag$ is the root agent, agent $j \in Ag$, a (possibly empty) tuple $\vec{v} = \langle v_1, ..., v_n \rangle$ of agents in Ag, initial state $\mathcal{I}$, a plan π, and a goal G, a* ***root-agent-aligned discrepancy resolving plan*** *π' for $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G \rangle$ is a constrained discrepancy resolving plan where the constraint Φ satisfies these conditions:*

1. *If $\mathcal{I} \models B_{\star,\vec{v}}\textsc{valid}(\pi, G)$ then*
   $\Phi \models B_{\star,j,\vec{v}}\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\textsc{valid}(\pi, G)$.
2. *If $\mathcal{I} \models B_{\star,\vec{v}}\neg\textsc{valid}(\pi, G)$ then*
   $\Phi \models B_{\star,j,\vec{v}}\neg\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\neg\textsc{valid}(\pi, G)$.

domestic robot to make the home-break-in plan invalid.

While discrepancy resolving plans need not contain only actions performed by the root agent, one useful form of root-agent-aligned discrepancy resolving plans involves the root agent communicating salient information to agent j either *implicitly* (e.g., by opening a box in front of agent j, demonstrating that it is unlocked) or *explicitly* (e.g., by telling agent j the box is unlocked). It is worth mentioning in this context the work by Sreedharan et al. (2020a) who investigated how a planning agent can explain its plan via either implicit or explicit communication. For example, moving through a narrow corridor will cause the human to revise her beliefs about the robot's capabilities (and hence about the action definitions).

Root-agent-aligned discrepancy resolving plans are important for a variety of settings. For instance, recall the aforementioned undesirable discrepancy resolving plan π' that rendered the planning system's plan invalid. While π' is a valid discrepancy resolving plan, it is not a valid root-agent-aligned discrepancy resolving plan. Therefore, to avoid such undesirable solutions in the plan explanation setting, we would generate root-agent-aligned discrepancy resolving plans, thus preserving the validity of the plan π. In contrast, discrepancies can be resolved by changing the root agent's beliefs to align with (the root agent's beliefs about) j's beliefs.

Definition 5.7 (Agent-j-aligned Discrepancy Resolving Plan)**.** *Given an RP-MEP domain $Q = \langle \mathcal{P}, \mathcal{A}, Ag \rangle$ where $\star \in Ag$ is the root agent, agent $j \in Ag$, a (possibly empty) tuple $\vec{v} = \langle v_1, ..., v_n \rangle$ of agents in Ag, initial state $\mathcal{I}$, a plan π, and a goal G, an **agent-j-aligned discrepancy resolving plan** π' for $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G \rangle$ is a constrained discrepancy resolving plan where the constraint Φ satisfies these conditions:*

1. *If $\mathcal{I} \models B_{\star,j,\vec{v}}\text{VALID}(\pi, G)$ then*
   $\Phi \models B_{\star,j,\vec{v}}\text{VALID}(\pi, G) \wedge B_{\star,\vec{v}}\text{VALID}(\pi, G)$.
2. *If $\mathcal{I} \models B_{\star,j,\vec{v}}\neg\text{VALID}(\pi, G)$ then*
   $\Phi \models B_{\star,j,\vec{v}}\neg\text{VALID}(\pi, G) \wedge B_{\star,\vec{v}}\neg\text{VALID}(\pi, G)$.

One form of agent-j-aligned discrepancy resolving plans involves the root agent changing the environment to align with j's beliefs. For example, the root agent could place some item where it believes agent j falsely believes it to be, in order to make j's plan valid. Such plans facilitate assistance which does not require coordination or communication with agent j. For instance, in Chapter 6 we enable an assistive robot to generate such plans and offer proactive assistance to a human, without coordinating with the latter.

Finally, Definitions 5.4–5.7 do not consider the plans of the other agents in Ag. Therefore, it is possible that a valid discrepancy resolving plan will introduce new

discrepancies pertaining to the validity of other agents' plans (e.g., making some agent's plan invalid while they believe it is valid). Φ can be specified appropriately such that discrepancy resolving plans preserve the validity of other agents' plans (from the root agent's perspective). In particular, our definitions could be extended to include, instead of a single plan π, a set of plans Π comprising plans 'belonging' to the various agents in Ag, and Φ could then relate to plans from Π.

5.1.1 Example

Using an example, we illustrate the concepts discussed in this section. Consider a search and rescue scenario with three agents, Alice (virtual assistant or robot who is also the root agent), Bob (human), and Mary (human), where Bob's goal is to obtain a particular medical kit (MedKit1). Moreover, Alice believes that Bob falsely believes that MedKit1 is in room A (Alice herself believes that the medical kit is in room B). Finally, Alice believes that Mary falsely believes that Bob believes that MedKit1 is in room B. We partially model this scenario:

$$Ag = \{\text{Alice, Mary, Bob}\} \tag{5.1}$$
$$\mathcal{I} \models B_{\text{Alice}}\, at(\text{Bob, HallWay}) \tag{5.2}$$
$$\mathcal{I} \models B_{\text{Alice}}\, at(\text{MedKit1, RoomB}) \tag{5.3}$$
$$\mathcal{I} \models B_{\text{Alice}} \neg at(\text{MedKit1, RoomA}) \tag{5.4}$$
$$\mathcal{I} \models B_{\text{Alice,Mary}}\, at(\text{MedKit1, RoomB}) \tag{5.5}$$
$$\mathcal{I} \models B_{\text{Alice,Bob}}\, at(\text{MedKit1, RoomA}) \tag{5.6}$$
$$\mathcal{I} \models B_{\text{Alice,Bob}} \neg at(\text{MedKit1, RoomB}) \tag{5.7}$$
$$\mathcal{I} \models B_{\text{Alice,Mary,Bob}}\, at(\text{MedKit1,RoomB}) \tag{5.8}$$
$$\mathcal{I} \models B_{\text{Alice,Mary,Bob}} \neg at(\text{MedKit1,RoomA}) \tag{5.9}$$

We refer to Alice as Alice rather than $\star$ for readability. Let us assume that Bob's goal G is *holding*(Bob,MedKit1) and that Alice predicts[2] that Bob's plan to achieve G is

$$[move(\text{Bob,HallWay,RoomA}),\ pickUp(\text{Bob,MedKit1,RoomA})].$$

We refer to Alice's prediction about Bob's plan as π_{AliceBob}. Moreover, let us assume

[2]Plan recognition techniques, such as those discussed in Chapter 4, can be used to predict or recognize other agents' plans. In Chapter 6 we present an approach that integrates the epistemic plan recognition techniques introduced in the previous chapter with the discrepancy resolution techniques introduced in this chapter. See Section 5.3.4 for a discussion.

that Alice can reason that Mary predicts that Bob's plan is

$$[move(\text{Bob,HallWay,RoomB}), pickUp(\text{Bob,MedKit1,RoomB})].$$

We refer to Alice's prediction about Mary's prediction about Bob's plan as $\pi_{\text{AliceMaryBob}}$. The actions in π_{AliceBob} are:

$$\begin{aligned}
&move(\text{Bob,HallWay,RoomA}) = \langle at(\text{Bob, HallWay}),\\
&\quad \{(\top, at(\text{Bob, RoomA})), (\top, \neg at(\text{Bob, HallWay}))\}\rangle\\
&pickUp(\text{Bob,MedKit1,RoomA}) =\\
&\quad \langle at(\text{MedKit1, RoomA}) \wedge at(\text{Bob, RoomA}),\\
&\qquad \{(\top, holding(\text{Bob, MedKit1})), (\top, \neg at(\text{MedKit1, RoomA}))\}\rangle
\end{aligned}$$

Actions in $\pi_{\text{AliceMaryBob}}$ are identical with RoomB replacing RoomA. Agents are 'aware' that an action has been performed if they are in the same location in which the action is performed (for more details on how agents' perception of actions is modelled in the RP-MEP formalism, see discussion in Section 2.3.1). For example, if Bob picks up MedKit1 in RoomA and Mary is also there, then Mary will believe that Bob is holding MedKit1. Next, suppose that $\textsc{valid}(\pi_{\text{AliceBob}}, G)$ and $\textsc{valid}(\pi_{\text{AliceMaryBob}}, G)$ are

$$\begin{aligned}
&at(\text{MedKit1, RoomA}) \wedge at(\text{Bob, HallWay})\\
&\text{and}\\
&at(\text{MedKit1, RoomB}) \wedge at(\text{Bob, HallWay}),
\end{aligned}$$

respectively. That is, for π_{AliceBob} to be valid, Bob must initially be in the hallway and MedKit1 must be in RoomA. Given entailments (5.2)-(5.9) (and assuming that Alice believes that all agents believe (that all agents believe) $at(\text{Bob, HallWay})$), the following holds pertaining to agents' beliefs about the validity of π_{AliceBob} and

$\pi_{\text{AliceMaryBob}}$:

$$\mathcal{I} \models B_{\text{Alice}} \neg \text{VALID}(\pi_{\text{AliceBob}}, G) \tag{5.10}$$
$$\mathcal{I} \models B_{\text{Alice,Bob}} \text{VALID}(\pi_{\text{AliceBob}}, G) \tag{5.11}$$
$$\mathcal{I} \models B_{\text{Alice,Mary,Bob}} \neg \text{VALID}(\pi_{\text{AliceBob}}, G) \tag{5.12}$$
$$\mathcal{I} \models B_{\text{Alice}} \text{VALID}(\pi_{\text{AliceMaryBob}}, G) \tag{5.13}$$
$$\mathcal{I} \models B_{\text{Alice,Bob}} \neg \text{VALID}(\pi_{\text{AliceMaryBob}}, G) \tag{5.14}$$
$$\mathcal{I} \models B_{\text{Alice,Mary,Bob}} \text{VALID}(\pi_{\text{AliceMaryBob}}, G) \tag{5.15}$$

Alice perceives in $\mathcal{I}$ a number of discrepancies between her beliefs and those of Bob and Mary pertaining to plan validity. In particular, $\text{VALID}(\pi_{\text{AliceBob}}, G)$ is a discrepancy perceived by Alice between her beliefs and her beliefs about Bob's beliefs, where $\vec{v}$ is empty (entailments (5.10) and (5.11)). One possible (root-agent-aligned) discrepancy resolving plan is then

$$\begin{aligned}
&\pi' = [\mathit{inform}(\text{Alice,Bob,}\neg \mathit{at}(\text{MedKit1,RoomA}))], \text{such that} \\
&\text{PROG}(\pi', \mathcal{I}) \models \\
&B_{\text{Alice,Bob}} \neg \text{VALID}(\pi_{\text{AliceBob}}, G) \wedge B_{\text{Alice}} \neg \text{VALID}(\pi_{\text{AliceBob}}, G).
\end{aligned}$$

The grounded inform action in π' is modelled as follows:

$$\begin{aligned}
&\mathit{inform}(\text{Alice,Bob,}\neg \mathit{at}(\text{MedKit1,RoomA})) = \\
&\quad \langle B_{\text{Alice}} \neg \mathit{at}(\text{MedKit1, RoomA}), \\
&\qquad \{(\top, B_{\text{Alice,Bob}} \neg \mathit{at}(\text{MedKit1, RoomA}))\}\rangle.
\end{aligned}$$

Modeling the inform action in this way enforces truthful communication, since its precondition is that Alice believe $\neg \mathit{at}$(MedKit1, RoomA). Moreover, we assume that Alice believes that other agents find Alice's communications trustworthy. Relatedly, see discussion of trust in epistemic planning by Fabiano et al. (2021). An interesting avenue for future work is the integration of trust into our discrepancy resolution framework.

The plan π' consists of Alice informing Bob that MedKit1 in not in RoomA. This resolves Alice's perceived discrepancy about the validity of π_{AliceBob}. That is, Alice believes that after Bob learns that MedKit1 is not in RoomA, he will believe that π_{AliceBob} is not valid. In Section 5.2 we discuss how to leverage epistemic planning to compute discrepancy resolving plans.

The root-agent-aligned discrepancy resolving plan π' aligns (Alice's beliefs about) Bob's beliefs with Alice's beliefs via a communication action. In contrast, the agent-j-aligned discrepancy resolving plan π'' resolves Alice's perceived discrepancy pertaining to π_{AliceBob} by aligning Alice's beliefs with (Alice's beliefs about) Bob's beliefs. Assuming Alice is initially in the hallway,

$$\begin{aligned}
\pi'' = [&\mathit{move}(\text{Alice,HallWay,RoomB}),\\
&\mathit{pickUp}(\text{Alice,MedKit1,RoomB}),\\
&\mathit{move}(\text{Alice,RoomB,RoomA}),\\
&\mathit{dropOff}(\text{Alice,MedKit1,RoomA})], \text{such that}\\
&\textsc{prog}(\pi'', \mathcal{I}) \models\\
&B_{\text{Alice,Bob}}\textsc{valid}(\pi_{\text{AliceBob}}, G) \ \wedge \ B_{\text{Alice}}\textsc{valid}(\pi_{\text{AliceBob}}, G).
\end{aligned}$$

Intuitively, Alice aligns the environment with Bob's beliefs by bringing MedKit1 from where it actually is (RoomB), to where Bob *believes* it to be (RoomA). Therefore, after Alice performs these actions, both Alice and Bob (from Alice's perspective) will believe that π_{AliceBob} is valid, thereby resolving the discrepancy. In many real-world settings, it may either be undesirable or even not possible for agents to resolve discrepancies by means other than communication. For instance, if Alice were a virtual assistant, then it is likely she would only be able to communicate information to other agents (π'). However, if Alice were a robot, as is the case in Chapter 6, she could perhaps resolve Bob's discrepancy by executing π''. In Section 5.3 we show examples of these two discrepancy resolution 'modalities' in various domains.

There is also a *'higher-order' discrepancy* in our example. In particular, $\textsc{valid}(\pi_{\text{AliceBob}}, G)$ is a discrepancy perceived by Alice between her beliefs and those of Mary about Bob's beliefs, where $\vec{v}$ is $\langle\text{Bob}\rangle$. That is, while Alice believes that Bob believes that π_{AliceBob} is valid (entailment (5.11)), she also believes that Mary believes that Bob believes that π_{AliceBob} is not valid (entailment (5.12)). This is because of Mary's false belief about Bob's beliefs about MedKit1's location (entailments (5.8) and (5.9)). A possible discrepancy resolving plan is

$$\begin{aligned}
&\pi''' = [\mathit{inform}(\text{Alice,Mary,}B_{\text{Bob}}\mathit{at}(\text{MedKit1,RoomA}))],\\
&\text{such that}\\
&\textsc{prog}(\pi''', \mathcal{I}) \models\\
&B_{\text{Alice,Mary,Bob}}\textsc{valid}(\pi_{\text{AliceBob}}, G) \wedge B_{\text{Alice,Bob}}\textsc{valid}(\pi_{\text{AliceBob}}, G).
\end{aligned}$$

Alice believes that after Mary learns that Bob believes that MedKit1 is in RoomA, Mary will believe that Bob believes that π_{AliceBob} is valid (which resolves Alice's perceived discrepancy).

5.2 Computing Discrepancy Resolving Plans

As discussed previously, epistemic planning combines automated planning and reasoning over the beliefs and knowledge of agents. In this section we present an algorithm that computes (unconstrained) discrepancy resolving plans using epistemic planning tools and establish the soundness of our algorithm with a theorem. We appeal to the epistemic planning system RP-MEP (Muise et al., 2015b, 2021) which was discussed in Chapter 2 (Section 2.3) and utilized in the previous chapter to solve plan recognition problems. As discussed in Section 2.3, to compute solutions for RP-MEP problems, Muise et al.'s planning system encodes an RP-MEP problem as a classical$^+$ planning problem and augments actions in the domain with conditional effects that enforce the KD45 axioms (discussed in Chapter 2, Section 2.2). The resulting classical$^+$ planning problem can then be given to an off-the-shelf classical planner that supports conditional effects. We appeal to this classical encoding of RP-MEP problems in our computation.

Algorithm 2

1: **procedure** ResolveDiscrepancy($\langle Q, \mathcal{I}, j, \vec{v}, \pi, G\rangle$) Given a tuple $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G\rangle$, return a plan π'.
2: $\pi' \leftarrow [\ ]$
3: $\langle \mathcal{F}, \mathcal{I}', O, \pi_c, G_c\rangle \leftarrow$ ClassicallyEncodeRP-MEPProblem$(Q, \mathcal{I}, \pi, G)$
4: $\phi \leftarrow$ ComputePlanValidityFormula$(\mathcal{F}, O, \pi_c, G_c)$
5: $G' \leftarrow$
$$\bigvee_{\phi_d \in \text{DNF}(\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_{\star,\vec{v}}\mathcal{D}(\phi_{dc})) \right) \vee$$
$$\bigvee_{\phi_d \in \text{DNF}(\neg\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_{\star,\vec{v}}\mathcal{D}(\phi_{dc})) \right)$$
6: $\pi' \leftarrow$ CallClassicalPlanner$(\langle \mathcal{F}, \mathcal{I}', G', O\rangle)$
7: **return** π'
8: **end procedure**

Our algorithm for discrepancy resolution is shown in Algorithm 2. Algorithm 2 accepts as input a tuple $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G\rangle$, where $Q = \langle \mathcal{P}, \mathcal{A}, Ag\rangle$ is an RP-MEP domain, $j \in Ag$ is an agent, $\vec{v}$ is a tuple of agents in Ag, $\mathcal{I}$ is an initial state, π is a plan, and G is a goal, and returns a discrepancy resolving plan for it. Ideally, we

would compute the validity formula for plan π and goal G, $\textsc{valid}(\pi, G)$, and provide the epistemic goal specified in Definition 5.4,

$$(B_{\star,j,\vec{v}}\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\textsc{valid}(\pi, G)) \vee$$
$$(B_{\star,j,\vec{v}}\neg\textsc{valid}(\pi, G) \wedge B_{\star,\vec{v}}\neg\textsc{valid}(\pi, G)),$$

to an epistemic planner. However, owing to the restricted nature of PEKBs discussed in Section 2.3.1, RP-MEP cannot directly solve such goals due to the disjunction in the goal expression and the possible disjunction in $\textsc{valid}(\pi, G)$.

Instead, we appeal to RP-MEP's encoding, which allows us to compute the validity formula and formulate the discrepancy resolution goal in a *classical*$^+$ *planning* setting. To this end, in Line 3, the ClassicallyEncodeRP-MEPProblem function returns a classical encoding of the initial state $\mathcal{I}$, RP-MEP domain Q, the plan π and goal G, $\langle \mathcal{F}, \mathcal{I}', O, \pi_c, G_c \rangle$. In the classical encoding, $\mathcal{F}$ is a set of fluent atoms representing each RML in the domain Q, $\mathcal{I}'$ is the classically encoded initial state $\mathcal{I}$, O is a set of classically encoded operators corresponding to the set of actions $\mathcal{A}$ in Q, π_c contains operators from O (corresponding to actions in $\mathcal{A}$ from π), and G_c corresponds to G and is expressed using fluent atoms from $\mathcal{F}$. We implement this function using RP-MEP's machinery which does not require any modification. Section 2.3.1 in Chapter 2 provides details of the encoding as well as the definitions of $\mathcal{C}()$ and $\mathcal{D}()$, which are mapping functions from RMLs in Q to fluent atoms in $\mathcal{F}$ (and vice versa).

Next, given the classical$^+$ planning domain $\langle \mathcal{F}, O \rangle$, the plan π_c, and the goal G_c, in Line 4 the ComputePlanValidityFormula function returns the formula $\phi = \textsc{valid}_c(\pi_c, G_c)$, where $\textsc{valid}_c$ is the validity formula in a classical$^+$ planning setting, using an implementation of the regression operator $\textsc{reg}$ for classical planning with conditional effects (Rintanen, 2008). To ensure the classical planner generates a discrepancy resolving plan, we formulate a goal that includes ϕ. Due to the possible disjunction in ϕ, and since RP-MEP does not support disjunctive belief, we cannot express this goal in the classical$^+$ planning problem. Instead, we formulate this goal using the *disjuncts* ϕ_d of $\text{DNF}(\phi)$[3] and $\text{DNF}(\neg\phi)$, where each ϕ_d is a conjunction. Specifically, in Line 6 CallClassicalPlanner tasks a classical planner that supports conditional effects and disjunctive goals with solving the classical$^+$ planning

[3]A disjunctive normal form (DNF) is a type of logical formula that is composed of a disjunction of one or more conjunctions. A DNF formula is in canonical form, which means that it cannot be simplified any further.

problem $\langle \mathcal{F}, \mathcal{I}', G', O \rangle$, where G' is

$$\bigvee_{\phi_d \in \mathrm{DNF}(\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_{\star,\vec{v}}\mathcal{D}(\phi_{dc})) \right) \vee$$
$$\bigvee_{\phi_d \in \mathrm{DNF}(\neg\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_{\star,\vec{v}}\mathcal{D}(\phi_{dc})) \right),$$

ϕ_{dc} are conjuncts of ϕ_d, $B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc})$ and $B_{\star,\vec{v}}\mathcal{D}(\phi_{dc})$ are RMLs, and $\mathcal{C}(B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc}))$ and $\mathcal{C}(B_{\star,\vec{v}}\mathcal{D}(\phi_{dc}))$ are the corresponding fluent atoms in $\mathcal{F}$ created in the encoding process. Each ϕ_{dc} is a fluent atom in $\mathcal{F}$ and $\mathcal{D}(\phi_{dc})$ is the corresponding RML in the RP-MEP domain Q.

5.2.1 Example

Let us illustrate the workings of Algorithm 2 using our example, where Alice is the root agent $\star \in Ag$, Bob is agent $j \in Ag$, and $\vec{v}$ is empty. Recall that we refer to Alice's prediction about Bob's plan (to achieve his goal *holding*(Bob,MedKit1)) as π_{AliceBob}, where

$$\pi_{\mathrm{AliceBob}} = [\mathit{move}(\mathrm{Bob,HallWay,RoomA}),\ \mathit{pickUp}(\mathrm{Bob,MedKit1,RoomA})].$$

Recall also that $\textsc{valid}(\pi_{\mathrm{AliceBob}}, G)$ is a discrepancy perceived by Alice (in the initial state $\mathcal{I}$) between her beliefs and her beliefs about Bob's beliefs. In Line 3, we classically encode $\langle Q, \mathcal{I}, \pi_{\mathrm{AliceBob}}, G \rangle$, where G is *holding*(Bob,MedKit1), and obtain the tuple $\langle \mathcal{F}, \mathcal{I}', O, \pi_c, G_c \rangle$. Then, in Line 4 we obtain the validity formula $\phi = \textsc{valid}_c(\pi_c, G_c)$ via regression, where

$$\phi = \mathit{at\_MedKit1\_RoomA} \wedge \mathit{at\_Bob\_HallWay}$$

and $\mathrm{DNF}(\neg\phi)$ is therefore

$$\neg \mathit{at\_MedKit1\_RoomA} \vee \neg \mathit{at\_Bob\_HallWay},$$

where *at_MedKit1_RoomA*, *at_Bob_HallWay*, $\neg$*at_MedKit1_RoomA*, and $\neg$*at_Bob_HallWay* are fluent atoms in $\mathcal{F}$. In Line 6 we task a classical planner that supports conditional

effects and disjunctive goals with solving $\langle \mathcal{F}, \mathcal{I}', G', O \rangle$, where G' is

$$
\begin{aligned}
&(\mathcal{C}(B_{\text{Alice,Bob}}\, at(\text{MedKit1, RoomA})) \wedge \mathcal{C}(B_{\text{Alice}}\, at(\text{MedKit1, RoomA})) \\
&\quad \wedge\ \mathcal{C}(B_{\text{Alice,Bob}}\, at(\text{Bob, HallWay})) \wedge \mathcal{C}(B_{\text{Alice}}\, at(\text{Bob, HallWay}))) \\
&\vee(\mathcal{C}(B_{\text{Alice,Bob}} \neg at(\text{MedKit1, RoomA})) \\
&\quad \wedge \mathcal{C}(B_{\text{Alice}} \neg at(\text{MedKit1, RoomA}))) \\
&\vee(\mathcal{C}(B_{\text{Alice,Bob}} \neg at(\text{Bob, HallWay})) \wedge \mathcal{C}(B_{\text{Alice}} \neg at(\text{Bob, HallWay}))),
\end{aligned}
$$

corresponding to one disjunct in $\text{DNF}(\phi)$ and two disjuncts in $\text{DNF}(\neg\phi)$. at(MedKit1, RoomA) is the RML corresponding to the fluent atom *at_MedKit1_RoomA* and therefore at(MedKit1, RoomA) is $\mathcal{D}$(*at_MedKit1_RoomA*). We discuss in the next section how the classical planner, depending on what actions Alice has at her avail, will either generate the root-agent-aligned discrepancy resolving plan π' (where Alice informs Bob about MedKit1's location) or the agent-j-aligned plan π'' (where Alice moves MedKit1 to where Bob believes it to be), both discussed in the previous section. Both plans satisfy one of the disjuncts of G' and are therefore a solution for $\langle \mathcal{F}, \mathcal{I}', G', O \rangle$.

5.2.2 Establishing the Soundness of Algorithm 2

In this section we establish the soundness of Algorithm 2. In particular, we will prove that plans returned by Algorithm 2 correspond to discrepancy resolving plans for the algorithm's input. To do so, we first formulate a number of definitions and lemmas.

Lemma 5.8. *Suppose $Q = \langle \mathcal{P}, \mathcal{A}, Ag \rangle$ is an RP-MEP domain, $\star \in Ag$ is the root agent, $\mathcal{I}$ an initial state, and the classical encoding of those (according to the* ClassicallyEncodeRP-MEPProblem *function, used in Line 3 of Algorithm 2) is $\langle \mathcal{F}, \mathcal{I}', O \rangle$, where $\mathcal{F}$ is a set of fluent atoms representing each RML in the domain Q, $\mathcal{I}'$ is the classically encoded initial state $\mathcal{I}$, O is a set of classically encoded operators corresponding to the set of actions $\mathcal{A}$ in Q, and $\mathcal{L}_{RML}^{Ag,d}$ is the corresponding set of all RMLs with bounded depth d. Then for every RML $\ell \in \mathcal{L}_{RML}^{Ag,d}$ of the form $B_\star \phi$ and sequence of actions π from $\mathcal{A}$,*

$$\text{PROG}(\pi, \mathcal{I}) \models \ell$$

if and only if

$$\textsc{prog}_c(\pi', \mathcal{I}') \models \mathcal{C}(\ell)$$

where π' is the sequence of actions in O corresponding to π and where $\textsc{prog}_c(\pi', \mathcal{I}')$ denotes the state of the world after sequentially applying the operators in π' in the state $\mathcal{I}'$.

Proof. This follows from (Muise et al., 2021, Theorem 2) and Definition 2.8. □

Lemma 5.9. *Suppose $Q = \langle \mathcal{P}, \mathcal{A}, Ag \rangle$ is an RP-MEP domain, $\star \in Ag$ is the root agent, $\mathcal{I}$ an initial state, and the classical encoding of those (according to the* ClassicallyEncodeRP-MEPProblem *function, used in Line 3 of Algorithm 2) is $\langle \mathcal{F}, \mathcal{I}', O \rangle$, where $\mathcal{F}$ is a set of fluent atoms representing each RML in the domain Q, $\mathcal{I}'$ is the classically encoded initial state $\mathcal{I}$, O is a set of classically encoded operators corresponding to the set of actions $\mathcal{A}$ in Q, and $\mathcal{L}_{RML}^{Ag,d}$ is the corresponding set of all RMLs with bounded depth d. For every RML $\phi \in \mathcal{L}_{RML}^{Ag,d}$ of the form $B_\star \ell$ where $\mathcal{C}(B_\star \ell) = \ell'$, $\ell' \in \mathcal{F}$, and sequence of actions π from $\mathcal{A}$,*

$$\textsc{prog}(\pi, \mathcal{I}) \models B_\star \mathcal{D}(\ell')$$

if and only if

$$\textsc{prog}_c(\pi', \mathcal{I}') \models \ell'$$

where π' is the sequence of actions in O corresponding to π and where $\textsc{prog}_c(\pi', \mathcal{I}')$ denotes the state of the world after sequentially applying the operators in π' in the state $\mathcal{I}'$.

Proof. This follows from (Muise et al., 2021, Theorem 2) and Definitions 2.8 and 2.9. □

Next, we define $\textsc{valid}(\pi, G)$ for the RP-MEP setting and relate it to the classical$^+$ planning setting used in our computation. To do so, we use the mapping function $\mathcal{D}()$. $\mathcal{D}()$ can be extended such that it applies to not just fluents, but any boolean

combination of fluents:

$$\begin{aligned}\mathcal{D}(\phi \vee \psi) &= (\mathcal{D}(\phi) \vee \mathcal{D}(\psi)) \\ \mathcal{D}(\phi \wedge \psi) &= (\mathcal{D}(\phi) \wedge \mathcal{D}(\psi)) \\ \mathcal{D}(\neg\phi) &= \neg\mathcal{D}(\phi)\end{aligned}$$

Definition 5.10. *Given an RP-MEP problem $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, G\rangle$ (where $\star \in Ag$ is the root agent) and plan π, we define*

$$\textsc{valid}(\pi, G) \triangleq \mathcal{D}(\textsc{valid}_c(\pi_c, G_c)),$$

where $\textsc{valid}_c(\pi_c, G_c)$ is the validity formula for the plan π_c and goal G_c in a classical$^+$ planning setting.

That is, $\textsc{valid}(\pi, G)$ is a formula with the same structure as $\textsc{valid}_c(\pi_c, G_c)$, but which replaces the fluent atoms in it with the corresponding RMLs in the RP-MEP domain. Recall that $\textsc{valid}_c(\pi_c, G_c)$ is the classical$^+$ planning validity formula, which ϕ is set to in Line 4 of Algorithm 2. We are now finally ready to establish the soundness of Algorithm 2.

Theorem 5.11. *Suppose that a plan π' is returned by Algorithm 2, given a tuple $R = \langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, j, \vec{v}, \pi, G\rangle$. Then π'' is a discrepancy resolving plan for R, where π'' is the plan comprising actions from $\mathcal{A}$ corresponding to the classically encoded operators in π'.*

Proof. We want to show that if a plan π' is returned by ResolveDiscrepancy given the tuple $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, j, \vec{v}, \pi, G\rangle$, where $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, \mathcal{I}, G\rangle$ is an RP-MEP problem (where $\star \in Ag$ is the root agent), then the plan π'' corresponding to the plan π' is a discrepancy resolving plan for $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G\rangle$. The plan π' is returned in Line 6 by the classical planner and therefore solves $\langle \mathcal{F}, \mathcal{I}', G', O\rangle$, where G' is

$$\bigvee_{\phi_d \in \mathrm{DNF}(\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_{\star,\vec{v}}\mathcal{D}(\phi_{dc})) \right) \vee$$
$$\bigvee_{\phi_d \in \mathrm{DNF}(\neg\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_{\star,j,\vec{v}}\mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_{\star,\vec{v}}\mathcal{D}(\phi_{dc})) \right),$$

where $\phi = \textsc{valid}_c(\pi_c, G_c)$ and π_c and G_c correspond to the plan and goal π and G in the tuple given to Algorithm 2. π_c contains operators from O, and G_c is expressed using fluent atoms from $\mathcal{F}$. By Lemmas 5.8 and 5.9, we have that

$$\text{PROG}(\pi'', \mathcal{I}) \models \bigvee_{\phi_d \in \text{DNF}(\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} B_{\star,j,\vec{v}} \mathcal{D}(\phi_{dc}) \wedge B_{\star,\vec{v}} \mathcal{D}(\phi_{dc}) \right) \vee \bigvee_{\phi_d \in \text{DNF}(\neg\phi)} \left(\bigwedge_{\phi_{dc} \in \phi_d} B_{\star,j,\vec{v}} \mathcal{D}(\phi_{dc}) \wedge B_{\star,\vec{v}} \mathcal{D}(\phi_{dc}) \right).$$

It follows that

$$\text{PROG}(\pi'', \mathcal{I}) \models [B_{\star,j,\vec{v}} \mathcal{D}(\phi) \wedge B_{\star,\vec{v}} \mathcal{D}(\phi)] \vee [B_{\star,j,\vec{v}} \mathcal{D}(\neg\phi) \wedge B_{\star,\vec{v}} \mathcal{D}(\neg\phi)].$$

Since $\phi = \text{VALID}_c(\pi_c, G_c)$, that can be rewritten as

$$\text{PROG}(\pi'', \mathcal{I}) \models [B_{\star,j,\vec{v}} \mathcal{D}(\text{VALID}_c(\pi_c, G_c)) \wedge B_{\star,\vec{v}} \mathcal{D}(\text{VALID}_c(\pi_c, G_c))] \vee [B_{\star,j,\vec{v}} \mathcal{D}(\neg\text{VALID}_c(\pi_c, G_c)) \wedge B_{\star,\vec{v}} \mathcal{D}(\neg\text{VALID}_c(\pi_c, G_c))]$$

Using the definition of $\mathcal{D}()$, we can move some negation signs around:

$$\text{PROG}(\pi'', \mathcal{I}) \models [B_{\star,j,\vec{v}} \mathcal{D}(\text{VALID}_c(\pi_c, G_c)) \wedge B_{\star,\vec{v}} \mathcal{D}(\text{VALID}_c(\pi_c, G_c))] \vee [B_{\star,j,\vec{v}} \neg\mathcal{D}(\text{VALID}_c(\pi_c, G_c)) \wedge B_{\star,\vec{v}} \neg\mathcal{D}(\text{VALID}_c(\pi_c, G_c))].$$

Since we defined $\text{VALID}(\pi, G) = \mathcal{D}(\text{VALID}_c(\pi_c, G_c))$, we are done. □

5.3 Experimental Evaluation

In this section, we present the results of our evaluation, where we set out to (1) demonstrate that epistemic planning tools can be used to compute discrepancy resolving plans with different modalities (i.e., epistemic communication actions or ontic actions) in various domains; and (2) evaluate the impact of increased depth of nested belief on Algorithm 2's runtime. Code can be found in https://github.com/maayanshvo/ToM-discrepancy-resolution. The repository includes our environments and implementations of the epistemic planning-based discrepancy resolution technique introduced in Section 5.2. In what follows, we first provide details of our experimental setup in Sections 5.3.1 and 5.3.2 and then provide the results of our experimentation in Section 5.3.3.

5.3.1 Experimental Setup

To satisfy our objectives, we ran Algorithm 2 to generate discrepancy resolving plans in the following domains (described in more detail in Section 5.3.2):

- **BlocksWorld for Teams** (BW4T) – an abstraction of a search and rescue domain, modeling our running example
- **Corridor** – an epistemic planning benchmark with epistemic goals such as selective communication of a secret. Interestingly, discrepancies may be resolved by closing doors to prevent agents from overhearing secrets.
- 7 **International Planning Competition (IPC)** domains (e.g., Driverlog, Depots, Logistics)

To evaluate the impact of the required depth of nested belief, d, and number of agents, $|Ag|$, each domain includes instances with $d = \{2, 3, 5\}$ (with $|Ag| = 2$, $|Ag| = 3$, and $|Ag| = 5$, respectively). When $d = 2$, a 'first order' discrepancy is resolved (e.g., where Alice resolves a discrepancy pertaining to Bob's beliefs about plan validity) and when $d = \{3, 5\}$, a 'higher-order' discrepancy is resolved (e.g., where Alice resolves a discrepancy pertaining to Mary's beliefs about Bob's beliefs (about ...)). The Corridor domain and one of the scenarios in the BW4T domain involve discrepancy resolution pertaining to the validity of plans that achieve epistemic goals. In these domains we only experimented with $d = 3$ since a depth of 2 is too low to model the kind of epistemic goals we are interested in. Moreover, a depth of 5 is not applicable in the BW4T scenario since it is very specific and involves only three agents. Lastly, in the Corridor domain we focused on epistemic goals typically used in the epistemic planning literature and did not experiment with a higher depth (i.e., $d = 5$).

We experimented with different discrepancy resolution modalities by creating three versions of each problem instance where either:

1. no modifications were made to the problem instance;
2. a subset of ontic actions was manually removed (such that the root agent cannot make certain changes to the environment to resolve discrepancies);
3. or a subset of inform actions was removed (such that the root agent cannot communicate with agent j to resolve discrepancies).

All domains were encoded using a file format used by RP-MEP called PDKB Domain Description Language (PDKBDDL) (a variant of the Planning Domain Definition Language (PDDL) (McDermott et al., 1998)) that can encode MEP problems,

including nested agent beliefs. More details can be found in Appendix B.1 and in (Muise et al., 2021). All problem instances across all domains were modelled as tuples comprising a domain, an initial state, a (possibly empty) tuple of agents, a plan, and a goal, and given to Algorithm 2. We manually encoded the initial state for each problem instance such that the root agent perceives a number of discrepancies between its beliefs and those of agent j, where one of the discrepancies is the validity formula of the plan π in the tuple given to Algorithm 2. Concretely, this was done by setting agent j's beliefs to be false, relative to the root agent's beliefs. Moreover, we only modified agent j's beliefs that pertain to the validity formula of the plan π (e.g., the location of the medical kit in the BW4T domain).

To implement Algorithm 2 we made use of the latest version (at the time) of RP-MEP (Muise, 2021d). For every problem instance, RP-MEP was given in Line 3 PDKBDDL files and returned classically encoded PDDL files. In Line 6, the Fast Downward planner (Helmert, 2006) was given the encoded PDDL files and called with an admissible heuristic that supports conditional effects and disjunctive goals, to ensure optimal plans are computed. We also made use of the SymPy Python library (Meurer et al., 2017) to convert regression formulae to DNF and to compute their negation. All experiments were ran using a 3.3GHz Intel Xeon E3-1230 machine with 32 GB of RAM. This machine is different (and stronger) than the one used in the experiments discussed in Chapter 4 since one of the epistemic planners used in the previous chapter could only be run on our weaker machine.

All plans given to Algorithm 2 were pre-computed using RP-MEP's machinery that allows for agents to *project to reason as other agents* and predict how they would achieve a certain goal (see discussion of *'Agent Projection'* in the RP-MEP repository and (Muise et al., 2015b, Sec. 5)).

5.3.2 Domain Descriptions

In what follows, we describe the various domains (and problems within those domains) used in our experiments.

BW4T

Johnson et al. (2009) presented a multi-agent simulation platform, BlocksWorld for Teams (BW4T), which is an abstraction of a myriad of application domains such as search and rescue. Typically in this domain, there are a number of rooms and a drop zone, where each room contains a number of colored blocks. In the application

domains, blocks may represent survivors of a disaster or medical kits, and the various agents may be humans or robots with different roles and capabilities. We cast blocks as medical kits.

We modelled various instances of the BW4T domain by varying the number of rooms, medical kits, and types of medical kits, totalling 10 unique problem instances. Moreover, in each instance we modelled a number of scenarios involving perceived discrepancies about plan validity. Common to all scenarios and instances is the following: there are three agents in the environment (Alice, Mary, and Bob); all discrepancies are perceived by Alice and resolved by her; and all plans (except for the discrepancy resolving plans) are executed by Bob. In all scenarios, Alice can (truthfully) inform other agents of either (agents' beliefs about) the whereabouts of various medical kits or the status of Mary's communication device. Moreover, Alice can move medical kits between different locations in the environment. Our BW4T domain was adapted from the BW4T domain in the RP-MEP repository (Muise, 2021a) by reducing the number of actions found in the original domain. The domain is shown in more detail in Appendix B.1.2.

BW4T Scenario #1 – Dude, Where's my Medical Kit?

This scenario and the next build on our running example presented in Section 5.1. Bob's goal is to get a particular medical kit to the drop zone and he believes that it is in some room. Alice believes that Bob holds a false belief pertaining to the location of the medical kit. Two tuples are created and given to Algorithm 2: one with π_{AliceBob} from our example in Section 5.1 (which Bob believes to be valid and Alice believes to not be valid, based on her belief about the location of the medical kit) and one with $\pi_{\text{AliceMaryBob}}$ (which Bob believes to not be valid and Alice believes to be valid). $\vec{v}$ is empty in these tuples. Possible discrepancy resolving plans that may be computed by the planner are π' (where Alice informs Bob about the true location of the medical kit) and π'' (where Alice moves the medical kit to where Bob believes it to be), as discussed in Section 5.1.

BW4T Scenario #2 – Where Does he Think he's Going?

In this scenario, the setup is the same but Mary is involved and Alice believes that Mary falsely believes that Bob does not have a false belief about the location of the medical kit (similarly to our running example). Two tuples are created and given to Algorithm 2: one with π_{AliceBob} and one with $\pi_{\text{AliceMaryBob}}$ (about both of which there

is a discrepancy between Alice's beliefs about Mary's beliefs about Bob's beliefs and Alice's beliefs about Bob's beliefs). $\vec{v}$ is ⟨Bob⟩ in both tuples. A possible discrepancy resolving plan that may be computed by the planner for π_{AliceBob} is π''' (where Alice informs Mary about Bob's beliefs), as discussed in Section 5.1.

To evaluate the impact of d, we also create a variant of this scenario with $d = 5$. That is, in addition to Mary, Alice, and Bob, we have two additional agents: Charlie and Rose. In this case, Alice believes that Mary falsely believes that Charlie believes that Rose believes that Bob does not have a false belief about the location of the medical kit. Therefore, we have the following:

$$\mathcal{I} \models B_{\text{Alice}} B_{\text{Mary}} B_{\text{Charlie}} B_{\text{Rose}} B_{\text{Bob}} \neg \text{VALID}(\pi_{\text{AliceBob}}, G)$$

and

$$\mathcal{I} \models B_{\text{Alice}} B_{\text{Charlie}} B_{\text{Rose}} B_{\text{Bob}} \text{VALID}(\pi_{\text{AliceBob}}, G).$$

Thus, Alice must resolve a higher-order discrepancy she perceives between her beliefs (about Charlie...) and Mary's beliefs (about Charlie...). $\vec{v}$ is ⟨Charlie, Rose, Bob⟩ in the tuple given to Algorithm 2.

BW4T Scenario #3 – Can You Hear Me??

Here, Bob has the goal of getting a particular medical kit to the drop zone and notifying his teammate, Mary, that he has done so (i.e., at(MedKit1,HallWay) ∧ B_{Mary}at(MedKit1,HallWay)). Note that Bob's goal has an epistemic component. In this scenario, Bob has a correct belief about the medical kit's location. However, while Alice believes that Mary's communication device is not working properly (perhaps she met Mary in passing and was told by her), she also believes that Bob falsely believes that it is working properly. Alice therefore believes that Bob's plan (which involves sending a message to Mary) will fail to achieve the *epistemic* component of his goal (i.e., for Mary to believe that a medical kit is now at the drop zone). A tuple is created and given to Algorithm 2 where $\vec{v}$ is empty and the plan π is

[*move*(Bob,HallWay,RoomB),
pickUp(Bob,MedKit1,RoomB),
move(Bob,RoomB,HallWay),
dropOff(Bob,MedKit1,HallWay),
sendComm(Bob,Mary,at(MedKit1,HallWay)))].

A possible discrepancy resolving plan involves Alice informing Bob that Mary's communication device is not working (see Appendix B.1.2 for more details, including how the *sendComm* action is modelled in PDKBDDL).

Corridor (Epistemic Planning Benchmark)

In our modified[4] version of the *Corridor* domain, there are n agents in various rooms connected to a long corridor. A single acting agent, Bob, holds a secret and can move along the corridor, enter different rooms, and announce his secret. When announcing the secret, all agents in the room with the announcer, as well as all agents in the adjacent rooms (when the door between the rooms is open), now believe the secret. The encoding of the *shareSecret* action in PDKBDDL is given in Appendix B.1.1.

Bob may have different epistemic goals, including a universal or selective spread of his secret to the other agents in the environment (similarly to the epistemic goals in the goal recognition experiments in Chapter 4). For example, let us assume that there are two agents in the environment in addition to Bob (agents k and l) and that one of Bob's epistemic goals is $B_k(\text{BobSecret}) \wedge \neg B_l(\text{BobSecret})$. That is, Bob wants agent k to believe his secret, but does not wish for agent l to believe it. We create 10 instances of this domain by varying the number of rooms, agents, and false beliefs held by Bob about the locations of each agent, and whether or not the doors between the different rooms are open or closed. For each of the generated instances, a tuple is created and given to Algorithm 2 where $\vec{v}$ is empty, the root agent is Alice and agent j is Bob. Alice can inform Bob of agents' locations and can also open and close doors in the environment.

As usual, we are interested in discrepancies pertaining to the validity of Bob's plan, as perceived by Alice. There are two reasons for Alice to believe that Bob's plan is not valid while believing that Bob believes it is valid:

- Bob's goal is for agent k to believe his secret but Alice believes that Bob falsely believes that k is in some room r (i.e., Alice believes that agent k is not in room r) and he will therefore plan to head to room r to share his secret with agent k. In some problem instances Bob's goal is for a number of agents to believe his secret (e.g., $B_k(\text{BobSecret}) \wedge B_l(\text{BobSecret}) \wedge B_m(\text{BobSecret})$). In these problem instances, Alice may believe that Bob holds a false belief about the location of some or all of the agents.

[4]The unmodified Corridor domain can be found in (Muise, 2021b).

- Bob only wants agent k to believe his secret without agent l believing it (i.e., $B_k(\text{BobSecret}) \wedge \neg B_l(\text{BobSecret})$). Alice believes that Bob falsely believes that agent l is neither in the room with agent k nor in the adjacent rooms or correctly believes that l is in an adjacent room but falsely believes that the door between the rooms is closed. Therefore, Bob will plan to go to the room in which he believes k to be and share his secret with her. However, this plan will fail to achieve his goal since either agent l is in the room with agent k; or agent l is in the adjacent room and will also come to believe Bob's secret since the door between the rooms is actually open.

As discussed in more detail in Appendix B.1.1, Alice can resolve discrepancies regarding the validity of Bob's plans by either (1) informing Bob of agents' locations or the (open or closed) position of doors, or (2) by closing or opening doors in the environment.

IPC Domains

In the International Planning Competition (IPC) (e.g., Gerevini et al., 2009), teams compete to solve a set of planning problems using their developed automated planning systems. The planning problems are drawn from a variety of domains, such as logistics, manufacturing, and robotics, and are designed to challenge the capabilities of the participating planning systems. Inspired by 7 IPC domains (Depots, Driverlog, Gripper, Rovers, Logistics, Zeno Travel, and Satellite) we modelled 7 RP-MEP domains with agents Alice, Bob, Mary, Charlie, and Rose. These domains can be found in (Muise, 2016). Appendix B.2 describes the process of adapting these classical planning domains to a multi-agent epistemic planning setting.

In total, we generated 70 problem instances (10 from each domain) by creating false beliefs for agents (e.g., causing an agent to hold a false belief about the location of an object in the Driverlog domain) and varying the domain parameters (e.g., number of objects in the domain). As before, Bob is the acting agent. For each problem instance, we varied the depth of nested belief d to create three cases where Alice (who perceives all discrepancies) resolves a discrepancy between her beliefs and the beliefs of other agents about the validity of Bob's plan:

- For $d = 2$, Algorithm 2 resolves discrepancies between $B_{\text{Alice}}\text{VALID}(\pi, G)$ and $B_{\text{Alice}}B_{\text{Bob}}\text{VALID}(\pi, G)$.
- For $d = 3$, Algorithm 2 resolves discrepancies between $B_{\text{Alice}}B_{\text{Bob}}\text{VALID}(\pi, G)$ and $B_{\text{Alice}}B_{\text{Mary}}B_{\text{Bob}}\text{VALID}(\pi, G)$.

- For $d = 5$, Algorithm 2 resolves discrepancies between $B_{\text{Alice}} B_{\text{Mary}} B_{\text{Charlie}} B_{\text{Rose}} B_{\text{Bob}} \textsc{valid}(\pi, G)$ and $B_{\text{Alice}} B_{\text{Charlie}} B_{\text{Rose}} B_{\text{Bob}} \textsc{valid}(\pi, G)$.

Appropriate tuples containing Bob's plan and goal were created and given to Algorithm 2. In each IPC domain, Alice has at her avail appropriate communicative and ontic actions. For example, in the Driverlog domain Alice may either inform Bob that he holds a false belief about the location of an object; or move the object to where Bob believes it to be. See Appendix B.1.3 for details on the encoding of the DriverLog domain and an example of a discrepancy resolving plan computed by the planner.

Depth (d)	2	3	5
	Time (s)	**Time (s)**	**Time (s)**
BW4T	1.93	2.31	1552.49
BW4T (EG)	–	3.16	–
Corridor (EG)	–	2.94	–
IPC - Depots	20.93	37.41	4028.12
IPC - Driverlog	1.51	3.77	472.97
IPC - Gripper	1.33	4.47	572.78
IPC - Rovers	1.39	2.88	1428.50
IPC - Logistics	47.19	59.33	MO
IPC - Zeno	22.49	51.43	MO
IPC - Satellites	2.38	4.25	807.63

Table 5.1: **The depth of nested belief, d, significantly impacts Algorithm 2's runtime.** We report the average runtime in seconds for Algorithm 2. d is the required depth of nested belief and EG signifies that problems in the domain involve an epistemic goal. MO means that the planner or classical encoding ran out of memory.

5.3.3 Results

Table 5.1 summarizes the results for the various domains. The table shows the average runtime (in seconds) for Algorithm 2 (using RP-MEP) over **10** problem instances (and the 3 versions of each) of the respective domain. MO means that the planner or classical encoding ran out of memory. d is the depth of nested belief. '–' means that we did not experiment with problem instances for the respective value of d. This is only the case for the Corridor (EG) domain and the BW4T (EG) domain, for reasons explained earlier in this section. The variances for the runtime for the set of problem

Depth (d)	2	3	5
	$\lvert\pi'_{\mathbf{c}}\rvert/\lvert\pi'_{\mathbf{o}}\rvert$	$\lvert\pi'_{\mathbf{c}}\rvert/\lvert\pi'_{\mathbf{o}}\rvert$	$\lvert\pi'_{\mathbf{c}}\rvert/\lvert\pi'_{\mathbf{o}}\rvert$
BW4T	2 / 6	2 / –	2 / –
BW4T (EG)	– / –	2 / –	– / –
Corridor (EG)	– / –	2 / –	– / –
IPC - Depots	3 / 8	3 / –	3 / –
IPC - Driverlog	3 / 6	3 / –	3 / –
IPC - Gripper	2 / 5	2 / –	2 / –
IPC - Rovers	3 / 6	3 / –	3 / –
IPC - Logistics	2 / 8	2 / –	MO
IPC - Zeno	3 / 7	3 / –	MO
IPC - Satellites	2 / 6	2 / –	2 / –

Table 5.2: $|\pi'_{\mathbf{c}}|$ and $|\pi'_{\mathbf{o}}|$ are the **average number of actions** in plans returned by Algorithm 2 that resolve discrepancies via inform actions or ontic world-altering actions, respectively.

instances for each domain (and value of d) ranged 0.07-0.45. The low variance is due to the planner's runtime being fairly similar for all problem instances in a certain domain and with a certain value of d.

Table 5.2 reports the average number of actions in plans generated by Algorithm 2. The $|\pi'_{\mathbf{c}}|$ and $|\pi'_{\mathbf{o}}|$ values are the average number of actions in plans returned by Algorithm 2 that resolve discrepancies via inform actions or ontic actions, respectively, across all problem instances in the domain. Note that plans that resolve discrepancies via inform actions may contain 'set-up' actions such as the root agent moving to agent j's location, or turning on a communication device. All discrepancy resolving plans consisted of 1-8 ontic or inform actions. MO, as before, means that the planner or classical encoding ran out of memory. For the Corridor and BW4T (EG) domains, '–', as before, means that we did not experiment with problem instances for $d = 2$ and $d = 5$. For the other domains, '–' in the $|\pi'_{\mathbf{o}}|$ column means that the planner did not compute plans that resolve discrepancies via ontic actions in that domain and value of d. In particular, in order to resolve higher-order discrepancies in the $d = 3$ and $d = 5$ settings, the root agent Alice must communicate with agent j and inform them about the beliefs of other agents. Therefore, applicable discrepancy resolving plans resolve discrepancies via inform actions, rather than world altering ontic actions.

Different modalities of discrepancy resolution

We set out to demonstrate that our algorithm can generate discrepancy resolving plans with different modalities and to this end created three versions of each problem instance by omitting either a subset of communicative or world-altering actions. As expected, when removing a subset of the ontic actions (such that the root agent cannot resolve discrepancies by performing world-altering ontic actions) the planner **only** found discrepancy resolving plans comprising inform actions. As mentioned previously, such plans may contain 'set-up' actions such as the root agent moving to agent j's location. Similarly, when removing a subset of inform actions, the planner **only** found discrepancy resolving plans that involve ontic actions (e.g., Alice moving MedKit1 to RoomA in our example). The lengths of plans in the former condition will only be reflected in the $|\pi'_{\mathbf{c}}|$ columns Table 5.2 while the lengths of plans in the latter condition will only be reflected in the $|\pi'_{\mathbf{o}}|$ columns.

In the unmodified domains, the modality used depended on the length of the various achievable discrepancy resolving plans. For example, if a discrepancy resolving plan that involves communicating with agent j was shorter than a plan that involves moving an object between two rooms, then the planner chose the former. Overall, the planner found discrepancy resolving plans that involved inform actions (rather than ontic actions) in **74%** of problem instances. This is due to a bias in the way the domains were created. Namely, we simplified domains such that inter-agent communication typically only requires a single inform action (possible in addition to a small number of 'set-up' actions), whereas altering the environment in order to resolve discrepancies typically requires additional actions (e.g., moving to room A, obtaining an object, and transferring it to room B). The lengths of plans in the 'unmodified domain' condition may be reflected in either the $|\pi'_{\mathbf{c}}|$ or $|\pi'_{\mathbf{o}}|$ columns in Table 5.2. However, as mentioned, most plans in this condition resolve discrepancies via inform actions and their lengths will therefore be reflected in the $|\pi'_{\mathbf{c}}|$ columns.

Finally, recall that Algorithm 2 generates unconstrained discrepancy resolving plans. When experimenting with different discrepancy resolution modalities, by removing a specific subset of actions, we de facto enforced the generation of root agent- and agent-j-aligned discrepancy resolving plans. In particular, when removing a relevant subset of ontic actions, plans can only comprise inform actions performed by the root agent. Thus, the only way to resolve a discrepancy perceived by the root agent in this case is by changing agent j's beliefs to align with the root agent's beliefs (i.e., a root-agent-aligned plan). When disabling communication between the root agent and agent j and only allowing for ontic actions performed by the root agent, discrepancies

can only be resolved by the root agent altering the environment (and its beliefs) to align with agent j's beliefs (i.e., an agent-j-aligned plan). We note, however, that our approach to constraining the type of discrepancy resolving plan the planner can compute requires domain knowledge and manual effort. More generally, Φ can be specified appropriately, as discussed in Section 5.1 (e.g., as is done in Definitions 5.6 and 5.7).

Impact of depth of nested belief d on runtime

Table 5.1 shows that d, the depth of nested belief, and the number of agents ($|Ag|$) that grows commensurate with d, significantly *increase* our algorithm's runtime. This is because the number of new fluents introduced during RP-MEP's encoding process is exponential in d and $|Ag|$ (Muise et al., 2021). When d is sufficiently high (as also observed by Muise et al.), some cells in Table 5.1 (where $d = 5$) read 'MO', i.e., either the compilation or the planner ran out of memory (with 32 GB RAM) because of the large number of fluents created during the compilation. Reflecting on their results, Muise et al. (2021, p.16) aptly observed that: *"...the majority of interesting use cases we have found for planning with nested belief is restricted to depth [2-3]."* This is also true in our discrepancy resolution setting.

Finally, previous work (including the work presented in the previous chapter) empirically showed similar results concerning RP-MEP and also that the performance of some epistemic planners is not affected by the value of d (Le et al., 2018; Shvo et al., 2020b). The application of these planners to discrepancy resolution could be investigated in settings where high depth of reasoning is required.

5.3.4 Discussion

Global vs Local Optimization in Algorithm 2

Algorithm 2's purpose is to resolve discrepancies between the root agent's beliefs and its beliefs about the beliefs of agent j, and accordingly the input to Algorithm 2 includes a single agent j. To resolve discrepancies between the root agent's beliefs and the beliefs of multiple agents in Ag, Algorithm 2 has to be called multiple times, resulting in multiple discrepancy resolving plans, one for each agent. In our example, to resolve discrepancies between (1) Alice and Bob and (2) Alice and Mary, we would call Algorithm 2 twice (once where Bob is agent j and once where Mary is agent j) and generate two plans. Similarly, we would call Algorithm 2 n times to resolve

discrepancies pertaining to n different plans, since the input to the algorithm only accepts a single plan π.

This brings up a limitation of our algorithm. In particular, in some cases (e.g., two agents needing to hear the same piece of information) aggregating all pairwise communications by running Algorithm 2 for each agent might not be desirable. Moreover, as discussed previously in this chapter, it may also be desirable to consider a number of plans when the algorithm is called, so that the validity of other agents' plans is not jeopardized. Further, if we wish to resolve discrepancies pertaining to multiple plans, calling Algorithm 2 sequentially for each plan may lead to inefficient local optimization. To address these important considerations, Algorithm 2 can be adapted to instead encourage global optimization over multiple plans/agents.

If It Ain't Broke, Don't Resolve It

It is interesting to observe that a plan that *needlessly* changes agents' beliefs when no discrepancy is perceived by the root agent may still be a valid discrepancy resolving plan, per Definition 5.4. For example, if the root agent believes that some plan π is valid and also believes that j believes that π is valid, then a plan that causes the root agent to believe that π is **not** valid and that j believes it is **not** valid is, per Definition 5.4, a valid discrepancy resolving plan. While this solution does not introduce a new discrepancy, it is not a very good one since it needlessly (and perhaps even harmfully) changes the validity of a plan and agents' beliefs about it. However, if optimal discrepancy resolution plans are found (as is the case in our evaluation in Section 5.3) then this undesirable solution will not be returned. Instead, the planner will return an empty plan since there is no discrepancy to resolve. More generally, Φ can be specified appropriately to avoid such undesirable solutions.

Relatedly, since the planner we use to compute discrepancy resolving plans is optimal, Algorithm 2 returns plans that have a minimal number of actions (and 0 actions if there is no discrepancy to resolve). Discrepancy resolving plans involving only inform actions will therefore eschew irrelevant information. This is often desirable and related to the Gricean maxims of quantity and relation discussed in Chapter 3.

What's the Plan?

Finally, Algorithm 2 accepts, as part of its input, a plan π and goal G. π and G may be obtained using *plan recognition*, where an observing agent attempts to predict an observed agent's plan and goal given a sequence of observations about the world and

the behavior of the observed agent (e.g., Kautz, 1987). However, in plan recognition observations often correspond to multiple plan or goal hypotheses. There are a number of ways to deal with uncertainty about other agents' plans, including collapsing (some of) the uncertainty via abstraction, exploiting probabilistic plan recognition (e.g., Ramírez & Geffner, 2010), and appealing to ideas from conformant planning. To this last point, rather than commit to a single plan hypothesis (and risk being wrong), we might extend our current approach to resolve discrepancies with respect to the entire set of possible plans the agent may be pursuing, without myopically treating each plan individually, one after the other. Naturally, in some cases it may not be possible to resolve discrepancies pertaining to all plans in this way (e.g., due to some inherent conflict between plans in the set or the agent's uncertainty). In such cases there should be a decision making mechanism that allows the discrepancy resolving agent to decide whether it should make various assumptions (e.g., choosing to resolve discrepancies pertaining to the most likely plan) or perhaps attempt to gather additional information in order to reduce its uncertainty.

Another alternative is to use probabilistic plan recognition techniques (e.g., Ramírez & Geffner's (2010) approach leveraged in Chapter 4), generate a probability distribution over possible plans and goals, and, at the risk of being wrong, resolve discrepancies pertaining to the most probable plan and goal. In Chapter 6 we present an approach to proactive robotic assistance that does just that and integrates the epistemic plan recognition techniques introduced in the previous chapter with the discrepancy resolution techniques introduced in this chapter.

5.4 User Study

In the previous section we demonstrated that epistemic planning tools can be used to compute discrepancy resolving plans in a number of domains. However, these results are not necessarily a testament to the efficacy of our approach in the *presence of humans*. As such, we conducted a user study to evaluate the ability of our approach to resolve participants' misconceptions. This research was approved by the Institutional Review Board (IRB) at the University of Toronto.

5.4.1 Methodology

We set out to test the following hypotheses:

H1: Participants will be more likely to generate a valid plan to achieve their

goal when presented with information *derived from a discrepancy resolving plan*, compared to the likelihood of generating a valid plan prior to receiving the information.

H2: Participants will be more likely to correctly predict another agent's plan when presented with information *from a discrepancy resolving plan*, compared to their prediction prior to receiving the information.

To test these hypotheses, participants were told that they are part of an emergency response team whose members must communicate with one another and obtain various items. Participants were moreover told that they are partnered with a virtual assistant meant to provide decision support, and were presented with two scenarios, mirroring two of the BW4T scenarios used in our evaluation and discussed in Section 5.3.2. Initially, participants were given very limited and partially incorrect information, causing discrepancies and allowing us to control for the factors that impact participants' reasoning. For instance, in the first scenario participants were told that the supply tent is at the east end of the base, when in fact it was at the west end. Participants' feedback indicated that our controlled setting ensured that participants initially generated an invalid plan in the first scenario and incorrectly predicted their teammate's plan in the second scenario. This is aligned with the initial states in our evaluation (and running example) where some agents initially have false beliefs that cause discrepancies that need resolution.

In each scenario, participants were given information by the virtual assistant. Using RP-MEP, we generated one discrepancy resolving plan (comprising only inform actions) for each scenario and the assistant's communication was simply a natural language representation of the inform actions in the discrepancy resolving plan (e.g., *"SupplyTent is at BaseWestEnd"*). Rephrased, H1 and H2 posited that the information given to participants would be sufficient to resolve the initial discrepancies we created and enable participants to perform better plan generation and prediction.

We had a total of 40 participants who were recruited via Amazon Mechanical Turk and were paid upon completing the questionnaire via an online platform (SurveyMonkey, 1999). Participants had no prior knowledge about the study.

Testing H1

In the first scenario (mirroring the *'Dude, Where's my Medical Kit?'* scenario discussed in Section 5.3.2), participants were told that their goal is to acquire a medical kit from the supply tent of the base. Subsequently, participants were given *incorrect*

information about the location of the supply tent. Participants were then asked where they would go in order to obtain a medical kit and were given 4 options from which to choose: the west, east, north, and south ends of the base. This is a proxy for participants' beliefs about plan validity. In other words, if participants believe that one of the 4 possible plans is valid (i.e., going to a certain location in the base) then we assume that participants will follow that plan and head to the respective location.

Next, participants were informed by their virtual assistant of the true location of the supply tent and were asked, again, where they would go in order to obtain a medical kit. As mentioned, the assistant's communication is a natural language representation of the inform actions in the discrepancy resolving plan generated by RP-MEP. The discrepancy resolving plan resolves a discrepancy perceived by the virtual assistant between its beliefs and its beliefs about the participant's beliefs, pertaining to the validity of the participant's plan.

Participants received information via simple text boxes. Moreover, participants were able to make choices by choosing amongst 4 radio buttons and their choice was subsequently recorded.

Testing H2

In the second scenario (a combination of (slight simplifications) of the *'Where Does he Think he's Going?'* and *'Can You Hear Me??'* scenarios discussed in Section 5.3.2), participants were told that they have a human teammate whose goal it is to acquire a medical kit from the supply tent. Participants were led to *incorrectly* believe that their teammate believes that the supply tent is in the old location. Participants were told that their plan is to send a message to the teammate's communication device. The participants, however, were not told that the teammate's communication device is faulty. Participants were then asked to predict the location to which their teammate will go in order to obtain the medical kit, after the participant's plan of sending the message is executed. Once again, participants were given 4 options from which to choose: the west, east, north, and south ends of the base. This is a proxy for participants' beliefs about their teammate's beliefs about plan validity. In other words, if participants believe that their teammate believes that one of the 4 possible plans is valid (i.e., going to a certain location in the base) then we assume that participants will predict that their teammate will follow that plan and head to the respective location.

Next, participants were informed by their virtual assistant that their teammate's communication device is not working and were asked, again, to predict the teammate's

plan following the execution of the participant's message-sending plan. As before, the assistant's communication is a natural language representation of the inform actions in the discrepancy resolving plan generated by RP-MEP. This time, the communication pertains to the teammate's communication device not working. The discrepancy resolving plan resolves a discrepancy perceived by the virtual assistant between its beliefs about the status of the teammate's communication device and its beliefs about the participant's beliefs about the status of the teammate's communication device. See also Appendix B.1.2 for the domain encoding and the generated discrepancy resolving plan.

5.4.2 Results

Testing H1 - Results

Prior to being informed by the virtual assistant about the true location of the supply tent, 0 participants generated a valid plan to obtain the medical kit. After being informed by the virtual assistant about the true location of the medical kit, all participants correctly generated a valid plan to obtain the medical kit, which indicates that their misconception regarding plan validity was resolved.

These results are consistent with **H1** since participants were more likely to generate a valid plan to achieve their goal *after* being presented with information derived from a discrepancy resolving plan, compared to the likelihood of generating a valid plan before receiving the information.

Testing H2 - Results

Prior to being informed by the virtual assistant about the status of their teammate's communication device, 0 participants correctly predicted their teammate's plan. After being informed by the virtual assistant about the status of their teammate's communication device, 95% of participants correctly predicted their teammate's plan which indicates that their misconception regarding plan validity was resolved.

A McNemar's test determined that participants' predictions about their teammate's plan after receiving information from their virtual assistant were significantly more accurate than their predictions prior to receiving this information (95% compared to 5%, $p < .001$). These results are consistent with **H2** since participants more accurately predicted their teammate's plan *after* being presented with information derived from a discrepancy resolving plan, compared to their prediction before receiving the information.

Finally, a response by one of the participants offers a qualitative illustration of the efficacy of our approach: *"If Mary's device isn't working she likely did not get the message so she will probably go to the East end where the supplies usually are"*. That is, given the information provided by the virtual assistant, the participant reasoned that their epistemic goal was not achieved by their plan, and as a consequence Mary holds a false belief and will generate an invalid plan as a result.

5.4.3 Discussion

The results support both **H1** and **H2** and show promise for human decision support. The objective of our study was modest – validate our approach by showing that discrepancy resolving plans generated by Algorithm 2 contain useful information for humans. To produce a user study with less modest objectives, and to address the limitations of the text-based environment, a future study will involve participants performing a task in a lab and interacting with a virtual assistant or a robot employing our proposed approach for discrepancy resolution. Various task performance metrics (e.g., Gervits et al., 2020) could then be used to evaluate the efficacy of our approach. As a first step towards this envisioned study, in Chapter 6 we integrate our approach to discrepancy resolution within an assistive robotic system and evaluate the helpfulness of this system in a set of simulations.

5.5 Related Work

Our work in this chapter draws from a number of research areas. In this section we survey closely related work and contrast with our own.

XAIP

Typically in Explainable AI Planning (XAIP) (e.g., Fox et al., 2017; Hoffmann & Magazzeni, 2019; Chakraborti et al., 2020; Carreno et al., 2021; Lindsay & Petrick, 2021) – a special case of the general task of explanation generation (e.g., Miller, 2019) – a planning agent is tasked with explaining some aspect of plan generation or execution (e.g., optimality or validity). Much work in XAIP has focused on allowing a planning agent to explain some aspect of its plan *without* considering potential model differences between the planning agent's model and that of the recipient of the explanation (the *explainee*), typically the user of the planning system (e.g., Eifler

et al., 2020). In contrast, our work emphasizes the need to consider the (possibly incomplete and incorrect) beliefs held by the explainee.

Indeed, a growing body of extant work has promoted this exact view. As mentioned in Chapter 3, Chakraborti et al. (2017) have termed explanations that do not consider the explainee's perspective *soliloquies* and argued that planning agents offering explanations should eschew soliloquies and instead consider the possibly disparate model held by the explainee. To realize these desiderata, Chakraborti et al. formulated the *model reconciliation problem*, with a large body of work continuing this line of research (Sreedharan et al., 2021), including applications in proactive decision support systems (Grover et al., 2020) and dialogue modeling (Sreedharan et al., 2020b). As briefly discussed in Chapter 3, the process of model reconciliation involves the AI system reasoning about (its beliefs about) the human's model and suggesting changes to it to conform with the system's model and consequently make the AI system's plan optimal from the perspective of the human (i.e., with respect to the human's reconciled model). In our evaluation, the root agent resolves discrepancies pertaining to other agents' plans. However, as discussed briefly in Section 5.1, the plan π in the tuple given to Algorithm 2 can represent the root agent's plan, as is typically the case in model reconciliation work specifically and in XAIP more generally.

In addition, Vasileiou et al. (2021a) and Son et al. (2021) proposed a logic-based framework for model reconciliation that operates over two knowledge bases – of an explainer and an explainee. While most work in model reconciliation is approached computationally via model-space search, Vasileiou et al. frame the model reconciliation problem as one of finding the shortest logical support of a given propositional formula (the explanandum) in the context of the explainer and explainee propositional knowledge bases. Vasileiou et al. (2022) build on the aforementioned work and extend it by proposing a number of cost functions that capture preferences between explanations; by addressing explanation generation for hybrid systems planning problems; and by offering an extensive evaluation of their approach and a comparison to the state of the art. Lastly, Vasileiou et al. (2021b) extend this body of work further to the probabilistic setting. In this work, the goal is to generate an explanation that causes the explainee's knowledge base to be updated in such a way that it has a *higher degree* of belief in the formula being explained (e.g., some property of a plan). This work extends the logic-based approach to the model reconciliation problem to explainees with probabilistic, rather than deterministic models of the planning model. Relatedly, Miller (2021) – building on Halpern & Pearl (2005) – proposed to use structural causal models to generate contrastive explanations, with application

potential in XAIP. Finally, Sreedharan et al. (2020a) leveraged a simplified version of RP-MEP's compilation to classical planning to generate explicable plans as well as plan explanations delivered via implicit and/or explicit communication.

Viewed through the lens of this dissertation, the body of work discussed in the previous two paragraphs can be seen to enable agents to use Theory of Mind-like reasoning and resolve discrepancies in XAIP settings. Our work goes beyond extant literature by broadening the role of Theory of Mind in such settings. Namely, by appealing to multi-agent epistemic logic and epistemic planning, our work supports a unique variety of settings requiring complex Theory of Mind reasoning. Specifically, our work enables agents to (1) reason about the *nested beliefs* of other agents and resolve 'higher-order' discrepancies regarding plan validity; and (2) correct misconceptions pertaining to the validity of plans pursuant to *epistemic goals*.

We note that work on model reconciliation can handle misconceptions that stem from different agents holding different views about action definitions. While epistemic planning can in general address such settings, in our current regression-based approach we only treat a single regression formula $\textsc{reg}(\pi, G)$ and the root agent's beliefs (about other agents' beliefs) about it. However, our approach could be extended by establishing satisfiability of different regression formulae.

Implicit coordination

Within multi-agent epistemic planning research, the paradigm of *implicit coordination* has emerged wherein agents use *perspective shifts* to plan and act in a decentralized manner towards a joint goal (Engesser et al., 2017; Engesser & Miller, 2020). For instance, agent i and agent j may have a joint goal for agent j to gain access to agent i's house. Agent i can come up with an implicitly coordinated plan where she leaves a key under the doormat and tells agent j where she left the key (Engesser et al., 2017). In contrast, a plan that includes only leaving the key under the doormat without communicating with agent j the key's location, is not considered implicitly coordinated since j will not be able to achieve the joint goal of gaining access to i's apartment.

Depending on the modality used, a discrepancy resolving plan can be seen to do what implicit coordination does – 'laying the ground work' for the success of other agents' plans. For instance, when Alice brings the medical kit from where it actually is, to where Bob *believes* it to be, she is ensuring the validity of Bob's plan that was previously not valid. However, work on implicit coordination has so far only

dealt with the S5 system, not allowing for agents to hold false beliefs. Moreover, the motivation of our work and the work on implicit coordination is different, and so are the computational techniques used in these works. Relatedly, our framework can augment frameworks for collaborative teaming (e.g., Alshehri et al., 2021) where agents must cooperate via effective communication and alignment of mental states.

BDI

Work on Belief-Desire-Intention (BDI) agents and architectures also explored the role of beliefs in explanation (e.g., Broekens et al., 2010). The explicit modeling of beliefs allows a BDI agent to explain its plans and goals in terms of its beliefs. However, these works did not appeal to epistemic planning and did not consider multi-agent settings where agents may need to resolve discrepancies pertaining to other agents' beliefs about the validity of plans.

5.6 Concluding Remarks

In this chapter, we examined how agents can use Theory of Mind to resolve discrepancies between their beliefs and the beliefs of other agents regarding plan validity. Our formulation appeals to epistemic logic and allows agents to reason about the nested beliefs of other agents and repair beliefs that give rise to plan validity discrepancies. We realized our approach using epistemic planning and showed how epistemic planning tools may be used to resolve discrepancies in various domains. A study showcased the ability of our approach to resolve misconceptions held by humans.

As mentioned, the discrepancy resolution framework we presented here builds on the conceptual foundations laid by Chapter 3 by satisfying the majority of desiderata proposed in Section 3.1 for explanations that employ Theory of Mind. In particular, our discrepancy resolution framework supports reasoning about the beliefs of multiple agents; allows the root agent to reason about other agents' beliefs; enables the root agent to consider how information will be assimilated by other agents; and finally allows for communication that may refer to the beliefs of other agents. Lastly, while our discrepancy resolution framework does not support a diversity of internal belief representations, the ideas presented in this chapter will hopefully inspire future work that will satisfy this final desideratum.

As discussed in Chapter 1, this dissertation has two components – theory and practice. In the next chapter we discuss how the theory from the current chapter and

the previous one are integrated within a robotic system to facilitate proactive robotic assistance.

Chapter 6

Proactive Robotic Assistance via Theory of Mind

Robotic systems are being deployed alongside humans in an increasing number of real-world social settings (e.g., Beer & Takayama, 2011; Garrell & Sanfeliu, 2012; Ferrer et al., 2013; Foster et al., 2020). One way to enhance seamless integration and adoption of such systems is to endow robots with social cognitive abilities (e.g., Dautenhahn, 2007). A large body of work (e.g., Baron-Cohen et al., 1985; Baron-Cohen, 1991; Baron-Cohen & Cross, 1992; Baron-Cohen, 1997) has identified that Theory of Mind is a core component of social cognition. It follows that if robots possess Theory of Mind capabilities, it could lead to the betterment of their social cognitive abilities. Consequently, this could lead humans to perceive such robotic systems as social agents, and enhance integration and adoption. Indeed, research has shown that robots demonstrating Theory of Mind have been perceived as more socially intelligent (Sturgeon et al., 2019) , and that these perceptions in turn may lead to greater trust in and acceptance of such technology (De Ruyter et al., 2005).

The role of Theory of Mind in human-robot interaction and teaming has garnered a fair degree of attention (e.g., Scassellati, 2002; Leyzberg et al., 2014; Zhao et al., 2015; Devin & Alami, 2016; Nikolaidis et al., 2017; Görür et al., 2017; Bühler & Weisswange, 2018; Brooks & Szafir, 2019; Dissing & Bolander, 2020). In Section 6.5 we survey related extant work. Most extant work, however, has not appealed to *epistemic planning*. Epistemic planning (discussed in Chapter 2) is noteworthy in being able to support a variety of human-robot interaction scenarios requiring complex Theory of Mind reasoning. In particular, epistemic planning supports (1) planning with both communication and world-altering actions; (2) higher-order belief attribution in the context of multiple agents (e.g., a robot can hold beliefs about Alice's beliefs about

Bob's beliefs); and (3) reasoning about and planning for epistemic goals – where an agent is trying to achieve some state of knowledge or belief (e.g., a robot can assist Alice with her goal of sharing information with Bob and not with Eve by alerting Alice to Eve's presence). Lastly, by appealing to epistemic planning we are able to exploit the evolving state of the art in the field as well as in cognate disciplines.

Some recent work has utilized epistemic planning for human-robot interaction and collaboration (Petrick & Foster, 2013; Miller et al., 2018; Petrick & Foster, 2020; Foster & Petrick, 2020; Bolander et al., 2021; Soldà et al., 2022) and here we advance this line of research by (1) proposing a novel *multi-agent* epistemic planning-based approach to proactive assistance that supports reasoning about *beliefs*, and by (2) evaluating the helpfulness and perceived Theory of Mind capabilities of robots implementing our proposed approach. Our approach goes beyond previous work by leveraging the novel epistemic planning-based techniques proposed in Chapters 4 and 5, and allowing a robot to recognize a human's plan and goal and offer proactive assistance by resolving any *discrepancies* between the robot's and the human's beliefs that jeopardize the achievement of the human's goal.

Main contributions

- We present an algorithm that integrates a number of epistemic planning-based techniques (including the epistemic plan recognition and discrepancy resolution techniques presented in Chapters 4 and 5, respectively) to enable proactive robotic assistance via ToM.
- We implement our algorithm and integrate it with a humanoid robot (Pepper (Pandey & Gelin, 2018)), combining various perception techniques with Theory of Mind reasoning.
- We conduct a study evaluating how participants perceive the Theory of Mind capabilities of a robot employing our proposed approach for Theory of Mind-based proactive assistance.
- We utilize a quantified metric of a robot's helpfulness (Freedman et al., 2020) to measure the efficacy of our method in a set of simulations across various domains.

Relationship to Published Work

This chapter is based on our IROS 2022 publication (Shvo et al., 2022b) and parts of our approach to proactive robotic assistance were inspired by our Humanizing AI workshop @ IJCAI 2019 publication (Shvo & McIlraith, 2019).

Chapter Structure

In Section 6.1, we present an algorithm that integrates a number of epistemic planning-based techniques (presented in Chapters 4 and 5) to enable proactive robotic assistance via Theory of Mind. In Section 6.2, we implement our algorithm and integrate it with a humanoid robot, combining various perception techniques with Theory of Mind reasoning. In Section 6.3, we conduct a study evaluating how participants perceive the Theory of Mind capabilities of a robot employing our proposed approach for Theory of Mind-based proactive assistance. In Section 6.4, we utilize a quantified metric of a robot's helpfulness to measure the efficacy of our method in a set of simulations across various domains. Finally, in Section 6.5 we survey related extant work.

6.1 Proactive Robotic Assistance via Theory of Mind

In this section, we present an algorithm for proactive robotic assistance that integrates the epistemic planning-based techniques presented in Chapters 4 and 5 and augments a robot with Theory of Mind capabilities in a number of reasoning tasks – plan recognition, assistive planning, and discrepancy resolution. Our algorithm describes a *perceive-plan-act* loop that allows a robot to continually update its understanding of the world and other agents' beliefs given observations; recognize another agent's plan and goal; and resolve any discrepancies pertaining to the validity of the other agent's plan by acting in the environment. Importantly, a robot using our approach acts proactively and only if needed, i.e., if it perceives discrepancies between its beliefs and those of the other agent pertaining to the validity of the other agent's plan. As discussed later on, such discrepancies can jeopardize the achievement of the human's goal and as such warrant intervention. Henceforth in this chapter, we assume that the robot is assisting a human. However, our approach enables a robot to offer proactive assistance to a diversity of agents, be they humans or other AI systems.

Algorithm 3 Proactive Robotic Assistance via Theory of Mind

Require: $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, \pi_H, G_H, \mathcal{G}\rangle$

1: $O = [\ \ ]$
2: **while true do**
3: $\quad O_{\text{curr}} \leftarrow$ GetObservationsFromPerceptionModule()
4: $\quad S \leftarrow$ prog(O_{curr}, S)
5: $\quad O \leftarrow$ UpdateObservationSequence(O, O_{curr})
6: $\quad \pi_H, G_H \leftarrow$RecognizePlanAndGoal$(\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O\rangle)$
7: $\quad$ **if** G_h is not *null* **then**
8: $\quad\quad \pi_{assist} \leftarrow$ GenerateAssistiveSolution$(\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, B_\star G_H\rangle)$
9: $\quad\quad \Pi \leftarrow \{\pi_H, \pi_{assist}\}$
10: $\quad\quad \pi' \leftarrow$ GenerateDiscrepancyResolvingPlan$(\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, \Pi, G_H\rangle)$
11: $\quad\quad S \leftarrow$ ExecuteDiscrepancyResolvingPlan(π')
12: $\quad$ **end if**
13: **end while**

Similarly to Chapters 4 and 5, we appeal to the multi-agent modal logic KD45$_n$ (discussed in Chapter 2, Section 2.2) and the RP-MEP formalism (discussed in Chapter 2, Section 2.3) to encode a robot's beliefs about the world and about the beliefs of multiple other agents and to enable reasoning about action and change in this setting. Our algorithm for proactive robotic assistance is shown in Algorithm 3. Algorithm 3 accepts as input a tuple $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, \pi_H, G_H, \mathcal{G}\rangle$, where $\langle \mathcal{P}, \mathcal{A}, Ag\rangle$ is an RP-MEP domain, $\star \in Ag$ is the robot (who is also the root agent), $H \in Ag$ is the human agent, S (a PEKB) is the initial state of the world, π_H and G_H are the human's plan and goal, and $\mathcal{G}$ is the set of possible goals the human H may be pursuing, where each goal $G \in \mathcal{G}$ is a PEKB. As discussed in Chapter 4, $\mathcal{G}$ is a typical component of a plan recognition problem in the literature and is often populated and constrained using domain knowledge. If the human's goal and plan are known a priori, G_H is drawn from $\mathcal{G}$ and π_H is a sequence of actions drawn from the set of actions $\mathcal{A}$. Otherwise, G_H and π_H are both *null* and will be determined via plan recognition in Line 6.

In Line 1, we initialize the sequence of observations O which is used for plan recognition. In Line 3, we obtain a sequence of current observations, O_{curr}, from the *perception module*. These observations are obtained via an anchoring process (e.g., Coradeschi & Saffiotti, 2000; Williams et al., 2009) that maps the output of a number of perception algorithms to fluent atoms in $\mathcal{P}$ (e.g., a can of soup) and agents in Ag (e.g., Alice) and then to actions in $\mathcal{A}$ (e.g., Alice picked up a a can of soup). Each observation in O_{curr} is thus in the set of actions $\mathcal{A}$. We elaborate on the implementation details of the perception module in Section 6.2.

In Line 4, we obtain the new state S by progressing the current state S with

respect to the observed sequence of actions, O_{curr}. We do so using the RP-MEP progression operator PROG discussed in Chapter 2 (Section 2.3). Recall that PROG is a function that takes a state (the 'old' S in this case) and a sequence of actions (O_{curr}) as input and returns the resulting state (the 'new' S) after the actions have been taken. In Line 5, we append the current observations O_{curr} to the sequence of observations O.

Line 6 – Plan Recognition

We utilize the epistemic plan recognition algorithm (Algorithm 1, presented in Chapter 4 (Section 4.3)) to recognize the human's goal and plan given the sequence of observations O. Recall that Algorithm 1 returns the goal $G_H \in \mathcal{G}$ that is most likely being pursued by the human (the actor), as well as a plan π_H that satisfies the sequence of observations O and that the robot (the observer) believes achieves the goal from the human's 'perspective'. Moreover, the epistemic plan recognition approach appeals to the plan recognition as planning paradigm (Ramírez & Geffner, 2010) in which a plan recognition problem is transformed into a planning problem, allowing for the use of off-the-shelf automated planning tools to solve the recognition problem.

We appeal to the approach presented in Chapter 4 because it realizes a computational solution via epistemic planning and in particular supports the epistemic planner we use in this chapter, RP-MEP (Muise et al., 2015b). Moreover, this approach models the beliefs of the robot and the human as first-class elements of the plan recognition process and allows for the recognition of plans with epistemic as well as ontic goals. As discussed earlier in this chapter, this is important for some real-world human-robot interaction settings (e.g., after recognizing that Alice's goal is to share information with Bob and not with Eve, a robot can assist Alice by alerting her to Eve's presence).

Next, we assume, as is assumed in Chapter 4, that the human H is following a single goal drawn from $\mathcal{G}$. Further, the sequence of observations O is a sequence of tuples $(\alpha_1, \phi_1), \ldots, (\alpha_m, \phi_m)$. Each α_i corresponds to the observation of an action, $a \in \mathcal{A}$. In this chapter, we limit our discussion to observations of actions and thus $\phi_i \in \mathcal{P}$ (corresponding to the observation of some properties of state immediately following the execution of α_i) is always $\top$, i.e., *true*. Finally, the RECOGNIZEPLANANDGOAL function in our algorithm calls Algorithm 1 with the tuple $\langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}, O \rangle$. Since in this chapter we focus solely on assistance provided to the human agent H (and not to other agents in the environment), if the sequence of observations O contains no actions executed by H, Algorithm 1 is not called. More generally, our algorithm can

be extended to facilitate assistance to multiple agents with different goals and plans.

Line 8 – Generating an Assistive Solution

Recall that in Chapter 4 we discussed possible relationships between the plan and goal components of the plan recognition solution – π_H and G_H. In particular, we discussed how it may be the case that the observing agent (the robot in this case) believes that π_H is not valid for achieving G_H while also believing that the actor (the human in this case) incorrectly believes that π_H is valid. We moreover discussed in Chapter 5 how such discrepancies pertaining to plan validity may be resolved by the robot by, for instance, communicating with the human about their plan's invalidity or acting in the environment and 'repairing' the human's plan (e.g., bringing an object from where it actually is, to where the human *believes* it to be). However, sometimes it is not possible to repair the human's plan. In such cases, an assistive agent such as a robot employing our approach should not only disavow the human of any false beliefs pertaining to the validity of their plan, but rather also (attempt to) propose a plan that will achieve the human's goal, in case the robot believes the human's plan, π_H, will not achieve G_H or in case it can generate a 'better' plan than π_H.

To this end, in Line 8, if π_H and G_H are not *null*, we build on Shvo & McIlraith's (2019) epistemic planning-based approach that generates an *assistive solution* – a plan that achieves the human's goal G_H from the robot's perspective, using the robot's beliefs. Shvo & McIlraith show that if the robot's beliefs (including those about the human's beliefs) are sufficiently accurate, then the robot will generate an assistive solution that is at least as 'good' as the best solution that the human can generate using their own beliefs (Shvo & McIlraith, 2019). Depending on the metric used, this may mean a less costly plan, a more preferred plan, or even a plan that meets some affective constraints (Shvo & McIlraith, 2019). As mentioned in Chapter 2, in this dissertation plan π_1 is 'better' than plan π_2 if π_1 contains fewer actions. Lastly, we note that when the robot cannot generate a better plan than the human's presumed plan π_H (e.g., if the robot believes that the human's beliefs are complete and correct), the assistive solution π_{assist} generated is identical to π_H (in which case the robot believes that π_H is valid and believes also that the human believes that π_H is valid).

To obtain an assistive solution, π_{assist}, we build on Shvo & McIlraith's approach and task an epistemic planner with solving the RP-MEP problem $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, B_\star G_H\rangle$. Moreover, here we are interested in generating assistive solutions that comprise only actions performed by the human H. To achieve this, we modify the set of actions $\mathcal{A}$ appropriately such that only the agent H can perform actions. More generally,

an assistive solution may involve actions from different agents (e.g., the robot may believe that it can support the human in accomplishing their goal). We don't address this interesting case in this dissertation.

Importantly, from the construction of the RP-MEP problem the robot believes that π_{assist} achieves the human's goal G_H, i.e.,

$$B_\star \textsc{valid}(\pi_{assist}, G_H).$$

However, since the robot's beliefs are used to generate π_{assist}, it may perceive a discrepancy if it believes that the human does not believe that π_{assist} achieves G_H, i.e.,

$$B_\star \textsc{valid}(\pi_{assist}, G_H) \wedge B_\star B_H \neg \textsc{valid}(\pi_{assist}, G_H).$$

When this occurs, the robot can leverage the discrepancy resolution techniques presented in the previous chapter to resolve such discrepancies and 'convince' the human that π_{assist} is valid. The next step of the algorithm does precisely that. Note that the validity formula VALID() used here is the same as the one defined in Chapter 5, building on a regression operator REG.

Lines 10-11 – Discrepancy Resolution

Next, the algorithm generates a discrepancy resolving plan π' that ensures that following its execution the robot *will not perceive discrepancies between its own beliefs and those of the human pertaining to the validity of the human's presumed plan* π_H *and the validity of the assistive solution* π_{assist}. The discrepancy resolving plan π' will be empty in case the robot does not perceive any discrepancies pertaining to π_H and π_{assist}.

Recall that the discrepancy resolution algorithm presented in the previous chapter (Algorithm 2) accepts a single plan π. Since we wish to resolve discrepancies pertaining to the validity of π_H *and* π_{assist}, we modify Algorithm 2 such that it accepts the tuple $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, \Pi, G_H\rangle$, where instead of a single plan π, the algorithm is given a set of plans Π. In Chapter 5, our definitions included a tuple of agents $\vec{v}$ and allowed the root agent to resolve 'higher-order' discrepancies pertaining to the beliefs of agent j about the beliefs of other agents (about the beliefs of other agents ...) about plan validity. Here, to ease exposition, we do not include $\vec{v}$ in the tuple given to the modified Algorithm 2 and instead restrict our discussion to discrepancies

perceived by the robot between its beliefs and the human's beliefs.

As discussed in Chapter 5 (Section 5.1), the task of discrepancy resolution can be cast as an epistemic goal, where the robot must change its own beliefs or the beliefs of the human to resolve discrepancies pertaining to the validity of the plan(s) in question. We modify the epistemic goal formulated in Chapter 5 (Definition 5.4) such that its achievement ensures that, for every plan $\pi_i \in \Pi$, $\textsc{valid}(\pi_i, G_h)$ will not be a discrepancy perceived by the robot between its beliefs and those of the human, H:

$$\bigwedge_{\pi_i \in \Pi} \big((B_\star B_H \textsc{valid}(\pi_i, G_H) \wedge B_\star \textsc{valid}(\pi_i, G_H)) \vee (B_\star B_H \neg \textsc{valid}(\pi_i, G_H) \wedge B_\star \neg \textsc{valid}(\pi_i, G_H)) \big)$$

There are, however, potential pitfalls when extending Algorithm 2 to address multiple plans. In particular, in the proactive assistance context, since we want to resolve discrepancies pertaining to both π_H and π_{assist}, we may have that

$$B_\star \textsc{valid}(\pi_{assist}, G_H) \wedge B_\star B_H \neg \textsc{valid}(\pi_{assist}, G_H)$$

and

$$B_\star \neg \textsc{valid}(\pi_H, G_H) \wedge B_\star B_H \textsc{valid}(\pi_H, G_H)$$

both hold. That is, the robot believes that the assistive solution π_{assist} generated in Line 8 is valid while also believing that the human believes that it not not valid. Moreover, the robot believes that the human's presumed plan π_H is not valid while believing that the human believes it is valid. In this case, a valid discrepancy resolving plan would be for the robot to make π_{assist} invalid and inform the human that π_H is invalid. That is, the following will hold:

$$B_\star \neg \textsc{valid}(\pi_{assist}, G_H) \wedge B_\star B_H \neg \textsc{valid}(\pi_{assist}, G_H).$$

and

$$B_\star \neg \textsc{valid}(\pi_H, G_H) \wedge B_\star B_H \neg \textsc{valid}(\pi_H, G_H).$$

Note that the robot no longer perceives discrepancies between its beliefs and the

human's beliefs, pertaining to the validity of π_H and π_{assist}. While this is a valid solution to the problem it is nonetheless undesirable since the human was not apprised of the validity of the alternative, assistive solution π_{assist}.

Indeed, in Chapter 5 we argued that there are many ways to resolve discrepancies, some of which are trivial or undesirable, and that we therefore often wish to resolve discrepancies under certain conditions by constraining the discrepancy resolution epistemic goal. To this end, we defined in Chapter 5 a constrained discrepancy resolving plan (Definition 5.5) where a logical formula, Φ, is added to the epistemic goal and imposes constraints on what states the discrepancy resolving plan can end in. Here we instantiate Φ such that at least one plan π_i in the set of plans Π is believed by the robot to be valid, and moreover that the robot believes that the human H believes that π_i is valid. Φ is formally defined as

$$\Phi = \bigvee_{\pi_i \in \Pi} \left(B_\star B_H \text{VALID}(\pi_i, G_H) \wedge B_\star \text{VALID}(\pi_i, G_H) \right).$$

That is, we require that at least one plan $\pi_i \in \Pi$ is believed to be valid both by the robot and by the human (from the robot's perspective).

Correspondingly, we compute the regression formula ϕ_i for each plan $\pi_i \in \Pi$ and modify the goal G' given to the classical planner in Line 6 of Algorithm 2:

$$G' = \Phi' \wedge \bigwedge_{\pi_i \in \Pi} \bigvee_{\phi_d \in \text{DNF}(\phi_i)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_\star B_H \mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_\star \mathcal{D}(\phi_{dc})) \right) \vee \bigvee_{\phi_d \in \text{DNF}(\neg \phi_i)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_\star B_H \mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_\star \mathcal{D}(\phi_{dc})) \right)$$

where $\mathcal{C}()$ and $\mathcal{D}()$ are mapping functions from RMLs in the domain to fluent atoms in the classically encoded domain (and vice versa), as discussed in Chapter 2, and where Φ' is the 'classically encoded' constraint Φ:

$$\Phi' = \bigvee_{\pi_i \in \Pi} \bigvee_{\phi_d \in \text{DNF}(\phi_i)} \left(\bigwedge_{\phi_{dc} \in \phi_d} \mathcal{C}(B_\star B_H \mathcal{D}(\phi_{dc})) \wedge \mathcal{C}(B_\star \mathcal{D}(\phi_{dc})) \right)$$

In Line 9, Π is populated with π_H and π_{assist} and in Line 10 $\langle \langle \mathcal{P}, \mathcal{A}, Ag \rangle, S, \Pi, G_H \rangle$ is given to the modified Algorithm 2. Finally, in Line 11 the constrained discrepancy resolving plan π' returned by Algorithm 2 is sent to the robot for execution, and the new state S is returned. See plan execution details in Section 6.2. For the purposes

of this work, we assume that π' is executed before the human begins executing π_H and that π' remains applicable throughout its execution. Future work will draw on existing solutions to address parallel execution of the robot's and the human's plans (e.g., Freedman & Zilberstein, 2017; Pozanco et al., 2018), to handle discrepancies between predicted and observed states of the world (Fritz & McIlraith, 2007), and to balance between the robot offering assistance and achieving its own goals.

Indeed, while in this chapter we assume that the robot does not have a goal of its own, we have ongoing work where this assumption is relaxed and the robot must consider the beliefs, plans, and goals of other agents and ensure that all agents are able to achieve their goals and, importantly, know how to do so. Furthermore, as discussed by previous work (e.g., Freedman & Zilberstein, 2017; Pozanco et al., 2018; Mirsky et al., 2020; Shvo & McIlraith, 2020), when integrating assistive (or adversarial (Pozanco et al., 2018)) behavior and plan recognition, it is crucial to consider when to react to the presumed plan and goal of the observed agent. For instance, if an observing agent decides to wait until the observed agent has completed their plan and only then react, it may very well be too late! Moreover, it may be the case that reacting to the observed agent's behavior (e.g., by offering assistance) is a costly endeavour where the cost outweighs the benefits to the observed agent. Future work will investigate how a robot employing our proactive assistance approach should balance between the cost of assistance, and the benefit to the human. It will also investigate the timing of discrepancy resolution, ensuring that when a discrepancy resolving plan is executed by the robot it is not too late, rendering the plan useless (or even harmful).

6.1.1 Example

We illustrate Algorithm 3 and the various concepts discussed in this section using an example. Consider a variation of the Sally-Anne false-belief task (originally developed to test Theory of Mind development in humans (Baron-Cohen et al., 1985) and discussed in Chapter 1) with three agents: the robot, Pepper, and two humans, Alice and Bob. Pepper, Alice and Bob are initially in the kitchen. To simplify the exposition, we assume that Pepper, via the perception module, observed Alice place a bowl (Bowl1) in a certain cabinet (Cabinet1) and leave the room. We partially

model Pepper's beliefs following these observations in state S:

$$S \models B_{\text{Pepper}} \neg at(\text{Alice, Kitchen})$$
$$S \models B_{\text{Pepper}} at(\text{Bob, Kitchen})$$
$$S \models B_{\text{Pepper}} in(\text{Bowl1, Cabinet1})$$
$$S \models B_{\text{Pepper}} B_{\text{Alice}} in(\text{Bowl1, Cabinet1})$$
$$S \models B_{\text{Pepper}} B_{\text{Bob}} in(\text{Bowl1, Cabinet1})$$

Algorithm 3 is now called with the tuple $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, \pi_H, G_H, \mathcal{G}\rangle$ where $Ag = \{\text{Pepper, Alice, Bob}\}$, and π_H and G_H are *null*. In this scenario, we are focused on the human agent, Alice (i.e., she is agent H), and Pepper (i.e., the root agent $\star \in Ag$). We refer to them as 'Alice' and 'Pepper' for readability. In this context, Pepper is restricted to (1) recognizing Alice's plan and goal and (2) resolving discrepancies between its beliefs and Alice's beliefs about the validity of plans.

Next, Pepper observes Bob opening Cabinet1, taking Bowl1, placing Bowl1 in Cabinet2, and leaving the kitchen. In Line 3, these observations are obtained from the perception module and in Line 4 the state S is progressed appropriately so that we obtain the new state S:

$$S \models B_{\text{Pepper}} \neg at(\text{Alice, Kitchen})$$
$$S \models B_{\text{Pepper}} \neg at(\text{Bob, Kitchen})$$
$$S \models B_{\text{Pepper}} in(\text{Bowl1, Cabinet2})$$
$$S \models B_{\text{Pepper}} B_{\text{Alice}} in(\text{Bowl1, Cabinet1})$$
$$S \models B_{\text{Pepper}} B_{\text{Bob}} in(\text{Bowl1, Cabinet2})$$

After observing Bob's actions, Pepper believes that Alice holds a false belief pertaining to the bowl's location – whereas Pepper believes that it is in Cabinet2, Pepper also believes that Alice believes the bowl is still in Cabinet1. Pepper's reasoning is performed automatically by the epistemic planner we use, RP-MEP (Muise et al., 2015b). In particular, as discussed in Chapter 2 (Section 2.3), RP-MEP implements a *conditioned mutual awareness* mechanism that automatically generates additional conditional effects from an action's existing set of conditional effects to ensure that some condition of mutual awareness is satisfied. For instance, in our scenario agents are 'aware' that an action has been performed if they are in the same location in which the action was performed. Conditioned mutual awareness also handles higher-order

Theory of Mind reasoning – since Pepper believes that Alice was not in the room when Bob moved the bowl, after observing Bob's action Pepper believes that Alice's beliefs about the bowl's location have not changed. For more details, see Chapter 2 (Section 2.3) and (Muise et al., 2021).

Next, since the sequence of observations O does not contain any actions performed by Alice, the plan recognition algorithm (Algorithm 1) is not called in Line 6 of Algorithm 3 and instead, since G_H is still *null*, the algorithm returns to Line 3.

Now, Pepper observes Alice returning to the room and taking out a can of soup (Soup1) from Cabinet3. The state S is progressed and in Line 6, since O now contains actions performed by Alice (i.e., *enterRoom*(Alice), *open*(Alice,Cabinet3), and *pickUp*(Alice,Soup1,Cabinet3)), the recognition algorithm is called with the set of possible goals $\mathcal{G} = \{made\_soup, made\_coffee, \dots\}$. Achieving *made_soup* in our simplified domain entails obtaining a bowl and a can of soup and since *pickUp*(Alice,Soup1,Cabinet3) was observed, the plan recognition algorithm (Algorithm 1) returns $G_H = made\_soup$ (i.e., the most likely goal), and

$$\pi_H = [enterRoom(\text{Alice}),\ open(\text{Alice,Cabinet3}),\ pickUp(\text{Alice,Soup1,Cabinet3}),\\ open(\text{Alice,Cabinet1}),\ pickUp(\text{Alice,Bowl1,Cabinet1})].$$

The plan π_H satisfies the sequence of observations O (per Definition 4.3) and achieves G_H from Alice's perspective. Since Alice holds a false belief about Bowl1's location, π_H involves obtaining the bowl from the wrong location (i.e., Cabinet1). In Line 8, an assistive solution, π_{assist}, that achieves $B_{\text{Pepper}}G_H$ (i.e., $B_{\text{Pepper}}(made\_soup)$) is generated using Shvo & McIlraith's (2019) approach, where π_{assist} is

$$[open(\text{Alice,Cabinet2}),\ pickUp(\text{Alice,Bowl1,Cabinet2})].$$

Since Pepper holds a correct belief about the location of Bowl1, π_{assist} involves Alice taking the bowl from Cabinet2, rather than Cabinet1.

Finally, in Line 10 the set of plans, Π, is populated with π_H and π_{assist} and given (along with G_H) to the modified discrepancy resolution algorithm (Algorithm 2). Pepper perceives in S two discrepancies between its beliefs and those of Alice – $\textsc{valid}(\pi_H, G_H)$ and $\textsc{valid}(\pi_{assist}, G_H)$. The former is a discrepancy since Pepper believes π_H is not valid while believing that Alice believes it is valid. The latter is a discrepancy since Pepper believes π_{assist} is valid while believing that Alice believes it is not valid. The function GenerateDiscrepancyResolvingPlan in Line 10

returns the discrepancy resolving plan π' to resolve both of these discrepancies:

$$[\mathit{inform}(\text{Pepper, Alice, } \neg \mathit{in}(\text{Bowl1,Cabinet1}) \wedge \mathit{in}(\text{Bowl1,Cabinet2}))].$$

That is, the discrepancy resolving plan involves Pepper informing Alice about the location of Bowl1 which resolves both discrepancies. In Line 11, π' is executed by Pepper who communicates the information to Alice. While here we do not include such reasoning, the RP-MEP domain can be extended (via an axiom-like mechanism) to encode that an agent should believe that an object is only in one location. Thus, when an agent is told that some object is in some location, then it is believed by that agent not to be in any other location. If this reasoning were included in the domain, Pepper's inform action would comprise only in(Bowl1,Cabinet2), omitting $\neg\mathit{in}$(Bowl1,Cabinet1).

While in this case Pepper resolved the discrepancies solely via communication, a discrepancy resolving plan can in general be a sequence of epistemic communication actions and/or ontic world altering actions, as discussed in the previous chapter. The discrepancy resolution modality used depends on the context in which the robot is deployed (e.g., perhaps the robot is limited in its dexterity and can therefore only move, sense, and communicate; or perhaps the robot's owner wishes for the robot to avoid making changes to the environment). For instance, while the plan π' comprises a single communication action, Pepper can alternatively move the bowl from Cabinet2 to Cabinet1 (before Alice starts executing π_H), thus resolving the discrepancies pertaining to π_H and π_{assist}. In Sections 6.3 and 6.4 we show the usefulness of a robot utilizing both of these modalities.

Finally, note that an alternative (and still valid) discrepancy resolving plan would be for Pepper to inform Alice that the bowl is not in Cabinet1 and to move the bowl from Cabinet2 to some other location (that is not Cabinet1), such that the assistive solution generated in Line 8 is no longer valid. This is a valid discrepancy resolving plan since following its execution, Pepper will not perceive discrepancies between its beliefs and Alice's beliefs, pertaining to the validity of the plans in Π, π_H and π_{assist}. This solution to the problem is, however, undesirable as discussed earlier in this section.

By defining the constraint Φ (as discussed earlier in the context of Lines 10-11 of Algorithm 3) and generating constrained discrepancy resolving plans, we avoid such undesirable solutions. In particular, Φ ensures that Pepper will either (1) inform

Alice that the bowl is no longer in Cabinet1 but rather in Cabinet2 (i.e., the plan π' shown above); or (2) move the bowl from Cabinet2 to Cabinet1. Importantly, the aforementioned undesirable discrepancy resolving plan (involving Pepper making π_H and π_{assist} invalid) no longer achieves the discrepancy resolution epistemic goal and will therefore not be returned by the planner.

6.2 Implementation

In this section, we describe our implementation of Algorithm 3 and its integration with a humanoid robot – Pepper (Pandey & Gelin, 2018). Our implementation integrates various perception techniques and the epistemic planner RP-MEP (Muise et al., 2021). Additional implementation details are provided in Appendix C.1.

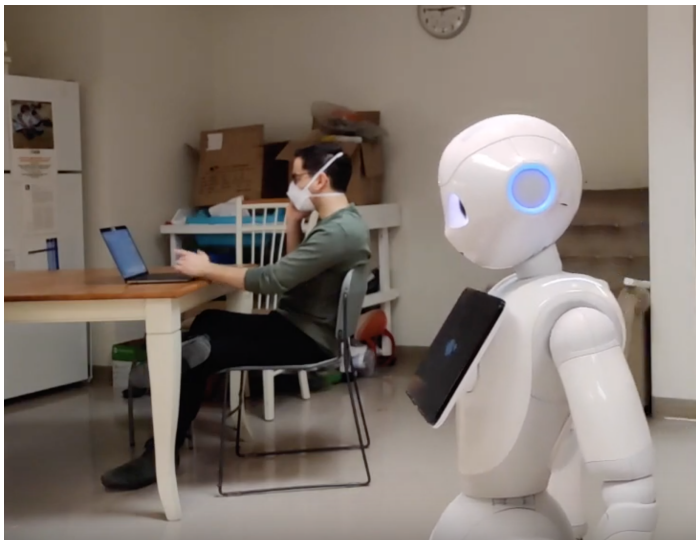

Figure 6.1: The Pepper robot 'in action'.

Line 3 – Perception Module

In Line 3, Algorithm 3 obtains observations from the perception module, which maps the output of several perception algorithms to actions in the set of actions $\mathcal{A}$ (e.g., *pickUp*(Alice, Soup1)). To do so, the perception module leverages existing state-of-the-art perception algorithms:

1. A pre-trained model of MonoLoco (Bertoni et al., 2021) is used to detect a person's 2D bounding-box pixel coordinates, 3D position, and orientation from Pepper's RGB data. By leveraging the orientation estimation, we estimate a

person's gaze direction which allows Pepper to perform *visual perspective taking* (Flavell, 1977) – the ability to see the world from another person's perspective (we provide an in-depth discussion of visual perspective taking in Section 6.4.2)

2. A pre-trained model of AlignedReID (Zhang et al., 2017) is used for person re-identification

3. ArUco markers (Garrido-Jurado et al., 2014) attached to cabinet doors are used to infer if the doors are open or closed

4. A pre-trained YOLO model (Redmon et al., 2016) is used to detect objects

5. RTAB-Map (Labbé & Michaud, 2019) is used by Pepper for scene localization

Finally, the GetObservationsFromPerceptionModule function in Line 3 of Algorithm 3 is realized by interfacing the perception module with RP-MEP via ROS's Python API (Quigley et al., 2009). The Robot Operating System (ROS) is an open-source software platform for building robotics applications. It is designed to be modular and flexible, allowing developers to create a wide range of robotic systems. Raw sensor data is subscribed from Pepper and the aforementioned perception algorithms are used to process them and output observations. We then map the processed observations to an agent from the set of agents Ag (e.g., Alice), a fluent atom in $\mathcal{P}$ (e.g., Soup1), and one of seven possible events (pick up, put down, open (close) cabinet, enter (leave) room, and shift gaze), and map these to an action in the set of actions $\mathcal{A}$ (e.g., $pickUp$(Alice, Soup1)). The observations are sent as a request to the service server that houses RP-MEP.

Lines 4-10 – Theory of Mind Reasoning

As mentioned, we make use of the epistemic planner RP-MEP (Muise et al., 2015b, 2021; Muise, 2021d). RP-MEP uses a file format called PDKB Domain Description Language (PDKBDDL), which is a variant of the Planning Domain Definition Language (PDDL) (McDermott et al., 1998) that can encode MEP problems, including agent beliefs. All domains given to Algorithm 3 were encoded using PDKBDDL. A PDKBDDL encoding of the kitchen example given in the previous section can be found in Appendix C.3. RP-MEP is used to perform progression in Line 4, perform plan recognition in Line 6, generate an assistive solution in Line 8, and generate a discrepancy resolving plan in Line 10. Finally, RP-MEP encodes RP-MEP problems as classical$^+$ planning problems (as discussed in Chapter 2, Section 2.3.1) and we

utilize the Fast Downward planning system (Helmert, 2006) to generate plans for the encoded classical$^+$ planning problems.

Line 11 – Plan Execution by Pepper

Actions from the discrepancy resolving plan π' generated in Line 10 are sent to a ROS service server that controls Pepper. In this chapter, Pepper is able to: (1) navigate to different places in the scene by leveraging the map built using RTAB-Map, Pepper's depth sensors, and ROS's navigation stack (Pütz et al., 2018); (2) communicate (e.g., Pepper informing Alice of the bowl's location in our example); and (3) interact with objects. Since Pepper lacks object manipulation capabilities, for the purposes of the video recordings in our study (discussed in the next section) we hardcoded Pepper's limb movement so that it appears as though Pepper is successfully manipulating objects (e.g., closing a door or picking up a phone charger).

Finally, the focus of our implementation was to demonstrate the feasibility of integrating complex Theory of Mind reasoning with a robotic system. Therefore, Pepper's perception and action capabilities are, by design, implemented to work only in the restricted settings discussed in this chapter. These settings are nevertheless representative of important real-world human-robot scenarios.

6.3 Evaluating the Perceived Theory of Mind Capabilities of a Proactively Assistive Robot

So far in this dissertation we have discussed the importance of augmenting AI systems with Theory of Mind capabilities, and have developed Theory of Mind-based computational solutions in the context of plan recognition and discrepancy resolution. In this chapter we have so far discussed our integration of these computational solutions in a robotic system and our resulting approach to proactive robotic assistance. Furthermore, in the next section we will empirically evaluate the helpfulness of robots employing our approach and demonstrate their potential to benefit humans.

However, for a robot to be beneficial, humans must first choose to use it, which research has shown to more likely happen when the robot is trusted and accepted (e.g., Nickerson & Reilly, 2004; Schaefer et al., 2016; Sanders et al., 2019). Many factors can affect trust and acceptance, including a robot's appearance, communication skills, and perceived usefulness. As discussed previously in this dissertation, research has moreover shown that trust and acceptance in robots may, in part, be

modulated by humans' *perceptions* of the Theory of Mind capabilities of robots (e.g., De Ruyter et al., 2005; Benninghoff et al., 2013; Mou et al., 2020). Given our intimate knowledge of the computational solutions within the robotic system described in this chapter, and the empirical support for the efficacy of robots employing these solutions (presented in the next section), we would like to assume that such robots will be perceived to possess advanced Theory of Mind capabilities (and may thus be more trusted and accepted). However, extant literature does not immediately support such an assumption. In particular, as noted by Söderlund (2022), studies of humans' reactions to robots exhibiting Theory of Mind capabilities have been scarce. Moreover, Thellman et al. (2022) conducted an extensive review and found conflicting findings in the body of empirical literature concerned with evaluating the attribution of mental states to robots. Given the limited extant literature and the novelty of our method for proactive robotic assistance, it is incumbent upon us to examine whether the aforementioned assumption holds, namely that robots employing our approach will be perceived to possess advanced Theory of Mind capabilities. To this end, we conducted a study (approved by the Institutional Review Board (IRB) at the University of Toronto) to answer the following question: is a robot, whose behavior is known by us to be driven by the advanced Theory of Mind capabilities introduced in Chapters 4 & 5, perceived by (lay) observers to similarly hold such capabilities?

6.3.1 Methodology

Experiment Design

We designed three realistic scenarios (including the kitchen scenario described in Section 6.1.1) and recorded two videos in each scenario. All videos involved a human(s) and a Pepper robot implementing Algorithm 3 in real-time, as described in Section 6.2. See additional details in Appendix C.2.

Each scenario included one video for the Theory of Mind (**ToM**) condition and one for the Theory of Mind deficit (**ToM-Def**) condition. In the former, Pepper can differentiate between its beliefs and those of others and proactively assists a human using Algorithm 3. In the latter condition, Pepper has a deficit in Theory of Mind and assumes that all other agents share its beliefs. Concretely, this is implemented by modifying the conditional effects of actions in the PDKBDDL file such that whatever Pepper comes to believe as the effect of an action, it also believes that all other agents in the environment come to believe the same (e.g., if Pepper is in a certain room, even if some agent is not in that room with Pepper, Pepper will attribute beliefs to

that agent as though they were in the room with Pepper). Therefore, Pepper never perceives a discrepancy between its beliefs and other agents' beliefs and consequently does not proactively assist. That is, an empty discrepancy resolving plan is always returned in Line 10 of Algorithm 3.

The videos used in the study can be found in (Shvo et al., 2022a). Participants were shown one video for each scenario, depending on their randomly assigned condition (i.e., each participants was either shown the three videos where Pepper has a deficit in Theory of Mind or the three videos where Pepper does not have that deficit). We set out to test the following hypothesis:

> *The robot that implements Algorithm 3 and distinguishes between its beliefs and other agents' beliefs (therefore demonstrating Theory of Mind-based proactive assistance) will be perceived to possess greater Theory of Mind capabilities, compared to a robot that implements Algorithm 3 and does* ***not*** *distinguish between its beliefs and other agents' beliefs (therefore not demonstrating Theory of Mind-based proactive assistance).*

Participants

Participants (n = 80, 67% male) were recruited via Amazon Mechanical Turk and were paid upon completing an online survey we created using SoSci Survey (Leiner, 2014). Nine participants held a high school diploma and 71 additionally held either a Bachelor's or a Master's degree. Participant age ranged 26 - 65 ($M = 38$; $SD = 9.6$), where M and SD are the mean and standard deviation, respectively. All participants were from the US.

Measures

Following each of the three video presentations, participants were given a list of statements to rate (on a 5-point Likert scale, ranging from 'Strongly Disagree' to 'Strongly Agree') from the Perceived Social Intelligence (PSI) Survey (Barchard et al., 2020). The PSI survey was created to fill a gap in the literature concerning the evaluation of perceived social intelligence in robots. As discussed by Barchard et al. (2020), while aspects of social intelligence in robots have been discussed by extant literature (e.g., Nomura et al., 2006; Bartneck et al., 2009; Ho & MacDorman, 2010), the concept of robotic social intelligence has not been clearly defined. To remedy this, Barchard et al. designed 20 scales to measure the perceived social intelligence of robots, creating the PSI Survey.

Following Sturgeon et al. (2019), we used the following scales from the PSI Survey as they directly relate to definitions for Theory of Mind and are therefore the most relevant for our hypothesis: Recognizes Human Cognitions (RC), Adapts to Human Cognitions (AC), and Predicts Human Cognitions (PC). The RC scale comprises the following statements which participants had to rate on a 5-point Likert scale:

- *The robot in the video can figure out what people think.*
- *The robot in the video knows when people are missing information.*
- *The robot in the video can figure out what people can see.*
- *The robot in the video understands others' perspectives.*

The PC scale comprises the following statements:

- *The robot in the video anticipates others' beliefs.*
- *The robot in the video figures out what people will believe in the future.*
- *The robot in the video knows ahead of time what people will think about certain situations.*
- *The robot in the video anticipates what people will think.*

Lastly, the AC scale comprises the following statements:

- *The robot in the video adapts its behavior based upon what people around it know.*
- *The robot in the video ignores what people are thinking.*
- *The robot in the video selects appropriate actions once it knows what others think.*
- *The robot in the video knows what to do when people are confused.*

Clearly, the statements in these three scales directly relate to ideas discussed extensively in this dissertation and capabilities with which Pepper is imbued.

Scenario	M	SD	Range
Charger	15.90	2.59	7–20
Kitchen	16.73	3.09	10–20
Corridor	16.30	1.83	13–20

RC

Scenario	M	SD	Range
Charger	15.80	2.76	8–20
Kitchen	15.70	2.78	8–20
Corridor	16.02	2.36	8–20

PC

Scenario	M	SD	Range
Charger	15.35	2.76	11–20
Kitchen	15.52	2.78	12–20
Corridor	15.32	2.36	12–19

AC

Table 6.1: Descriptive statistics for the Recognizes Human Cognitions (RC), Predicts Human Cognitions (PC), and Adapts to Human Cognitions (AC) scales in the **ToM** condition. M, SD, and Range are the mean, standard deviation, and lowest and highest values for the participants' scores in each scale, respectively.

Scenario	M	SD	Range
Charger	10.95	5.43	4–20
Kitchen	10.65	5.52	4–19
Corridor	10.70	5.49	4–20

RC

Scenario	M	SD	Range
Charger	10.62	5.95	4–20
Kitchen	10.25	5.85	4–19
Corridor	10.42	5.91	4–20

PC

Scenario	M	SD	Range
Charger	10.42	5.95	4–17
Kitchen	10.30	5.85	4–17
Corridor	10.72	5.90	4–18

AC

Table 6.2: Descriptive statistics for the Recognizes Human Cognitions (RC), Predicts Human Cognitions (PC), and Adapts to Human Cognitions (AC) scales in the **ToM-Def** condition. M, SD, and Range are the mean, standard deviation, and lowest and highest values for the participants' scores in each scale, respectively.

6.3.2 Results

Tables 6.1 and 6.2 show the descriptive statistics associated with the scores for the RC, PC, and AC scales in the **ToM** and **ToM-Def** conditions, respectively. The score for each scale is the sum of scores given to each of the four statements of which the scale is comprised. Mann-Whitney U tests were conducted to determine if the presence of Theory of Mind-based proactive assistance modulated participants' perception of a robot's Theory of Mind capabilities (as reflected by the RC, PC, and AC scales). In all three scenarios, the scores for the RC, PC, and AC scales in the **ToM** condition were significantly higher than in the **ToM-Def** condition ($p < .001$). See detailed results in Appendix C.4.

6.3.3 Discussion

Results support our hypothesis in that the robot that implements Algorithm 3 and distinguishes between its beliefs and other agents' beliefs was perceived to possess greater Theory of Mind capabilities, compared to a robot that implements Algorithm 3 and does not distinguish between its beliefs and other agents' beliefs. However, a substan-

tive limitation of our study has to do with the **ToM-Def** condition. In particular, as mentioned earlier, the robot in the **ToM-Def** condition (despite implementing Algorithm 3 as in the **ToM** condition) has a deficit in Theory of Mind, assumes that all other agents share its beliefs, and therefore never perceives discrepancies between its beliefs and other agents' beliefs and consequently does not offer assistance. The robot's (lack of) behavior in the **ToM-Def** condition elucidates the importance of Theory of Mind in social settings and the potential negative consequences of a deficit in Theory of Mind, as will also be shown empirically in Section 6.4. Indeed, in our video recordings, the only behavior demonstrated by the robot in the **ToM-Def** condition is a repetitive movement of the arm (to which a dusting brush is connected). This was done in both conditions to eschew further bias in the **ToM-Def** condition, arising from the robot standing still throughout the video. It is therefore not surprising that the robot in the **ToM** condition scored significantly higher on all scales than the robot in the **ToM-Def** condition. A future study could instead compare participants' perceptions of a robot employing our approach to perceptions of robots employing different approaches to Theory of Mind reasoning, and even of robots demonstrating helpful behavior that does not require Theory of Mind reasoning.

In the meantime, it is perhaps more compelling to discuss the descriptive statistics from our study, reported in Tables 6.1 and 6.2. In particular, the robot in the **ToM** condition received scores nearing the top end of the 5-point Likert scale for the RC, PC, and AC scales (e.g., an average score[1] of 16.725 on the RC scale for the kitchen scenario out of the highest possible score of 20). For comparison, we look at reports found in previous studies that examined participants' perceptions of robots using the PSI Survey. Barchard et al. (2020) showed 296 participants five videos representing a wide range of robot social intelligence (Kory, 2014; de Greeff et al., 2014; Kahn Jr. et al., 2015; Sirkin et al., 2015; Devin et al., 2017). For instance, Kahn Jr. et al.'s (2015) video shows a large humanoid robot (Robovie) asking a human to lie about a shared experience the robot and the human had had together. A prerequisite to lying is a Theory of Mind and it is therefore reasonable to assume that participants would positively rate the robot's Theory of Mind capabilities. In contrast, Sirkin et al.'s (2015) video shows a robotic ottoman that encourages people to put their feet up. While such a robot can certainly be seen as helpful, one would not expect participants to perceive the robotic ottoman as possessing a high degree of Theory of Mind. Indeed, on average, Robovie received a score of 13.28 on the RC scale (the highest amongst

[1]The score for each scale is the sum of scores given to each of the four statements of which the scale is comprised.

the five robots shown to participants), while the robotic ottoman received an average score of 8.94 on the RC scale (the lowest amongst the five robots). Back to our study, the robot in the **ToM** condition (employing our approach for proactive robotic assistance) received average scores of 15.90, 16.73, and 16.30 in the three scenarios on the RC scale, higher than all robots in Barchard et al.'s (2020) study (the same is true for the PC and AC scales). While we cannot draw conclusions about the statistical significance of the difference in scores given to the robot in our study and the robots in Barchard et al.'s study, these results are nevertheless interesting given the diversity of robots as well as the large number of participants in their study. These results moreover contribute to the plausibility of the assumption discussed in the beginning of this section, namely that robots employing our approach will be perceived to possess Theory of Mind capabilities and may thus be more trusted and accepted.

Another limitation of our study is the lack of interaction between participants and a (physical or simulated) robot. While our study allows for an evaluation of participants' perceptions of the robot, it does not allow for an evaluation of the efficacy of the robot's assistive capabilities. A future study will involve participants interacting with a robot and, using various task performance metrics (e.g., Gervits et al., 2020), the efficacy of our approach (integrated with a robot) may be evaluated. In the next section, we take a step towards this envisioned study by simulating both the robot and the human(s), and measuring the efficacy of our approach in a number of domains.

6.4 Evaluating the Helpfulness of a Proactively Assistive Robot

In this section, we discuss our evaluation where we set out to measure the *helpfulness* of a robot utilizing our proposed method for proactive assistance in a myriad of domains, and evaluate how the robot's helpfulness is impacted by the veracity of its beliefs and whether or not it has a deficit in ToM. To this end, we turned to extant work (Freedman & Zilberstein, 2017; Freedman et al., 2020; Freedman, 2020) that investigated a quantified metric of helpfulness, used to assess the benefit that a robotic partner has on a human-robot team for a given task. Intuitively, the more helpful the robot is being, the higher its helpfulness score for the given task. Quantitatively, helpfulness (H) is calculated by comparing between *the overall cost of executing the*

plan for the human-robot team versus the human alone:

$$H = cost(\text{human alone}) - cost(\text{human-robot team})$$

While Freedman et al. (2020) focus on settings where robots assist humans by executing part of the human's plan, in our work we enable robots to proactively assist humans by resolving discrepancies pertaining to the validity of their plans. In our evaluation, we focus on settings where these discrepancies arise when the human holds a false belief which causes their plan to be invalid.

For instance, in our example (after Bob moves the bowl while Alice is not in the room) Alice falsely believes the bowl is in Cabinet1 and will therefore look for it there. Her plan, however, will fail. Let us assume that after looking in Cabinet1 and not finding the bowl, Alice then makes a new plan to go through all cabinets and drawers, in order, and search for the bowl. Let us call this the 'human alone' plan (π_H) and assume its cost is 3 (since Alice has to search a drawer and Cabinet1 before finding the bowl in Cabinet2). In contrast, recall that in our example Pepper is able to recognize Alice's plan and goal and resolve the discrepancy it perceives by telling her that the bowl is in Cabinet2. In this case, after learning the true location of the bowl from Pepper, Alice will plan to go to Cabinet2 to obtain the bowl. Let us call this the 'human-robot team' plan (π_{HR}) and assume its cost is 1. Finally, we can say that Pepper's helpfulness in this scenario is $cost(\pi_H)$ - $cost(\pi_{HR})$ = 3 - 1 = 2. Freedman et al. also propose a *relative helpfulness* measure which moves from absolute costs to a ratio of costs which normalizes H and makes it less sensitive to the varying cost of plans across different tasks:

$$H_R = \frac{cost(\pi_H) - cost(\pi_{HR})}{cost(\pi_H)}$$

As mentioned, we set out to evaluate how the robot's helpfulness is impacted by the veracity of its beliefs and whether or not it has a deficit in Theory of Mind. Going back to our example, while Pepper's beliefs happen to be correct in the original scenario, let us assume that for some reason Pepper *falsely* believes that the bowl is in the bedroom (perhaps it observed someone use it in that room earlier). In this case, Pepper still perceives a discrepancy but in order to resolve it, Pepper informs Alice that the bowl is in the bedroom. In this case the cost of the plan π_H is still 3 while the cost of the plan π_{HR} plan is now 7 (because Alice will go to the bedroom, fail to find the bowl there, go back to the kitchen, etc.) – Pepper's helpfulness is now

negative!

Finally, let us assume that Pepper has a deficit in Theory of Mind which means that it believes that all other agents share its beliefs. In our scenario, this means that Pepper perceives no discrepancies between its beliefs and Alice's beliefs and therefore does not intervene. In this case the plans π_H and π_{HR} are identical and have a cost of 3 and Pepper's helpfulness is therefore 0. In what follows, we first provide details of our experimental setup in Sections 6.4.1– 6.4.3 and then provide the results of our experimentation in Section 6.4.4 and discuss the runtime of Algorithm 3 in Section 6.4.5.

6.4.1 Experimental Setup

We evaluated a robot's helpfulness and relative helpfulness (H and H_R) in a myriad of domains, where the robot either had no deficit in Theory of Mind and had correct/false beliefs, or had a deficit in Theory of Mind. Rather than performing human subjects experiments, we ran our experiments with simulated AI-controlled agents, acting as the humans in the environment.

Algorithm 3 was implemented and run as described in Section 6.2. The observations from the perception module were simulated. Following the simulated execution of the discrepancy resolving plan in Line 11 of Algorithm 3, the human's plan is executed. If it does not achieve the human's goal then RP-MEP is tasked with generating a new plan for the human which involves the human sensing the environment until the goal is achieved (e.g., searching through all cabinets and drawers until the bowl is found). In our experiments, we used the following domains:

1. **BlocksWorld for Teams** (BW4T) (Johnson et al., 2009) – an abstraction of a search and rescue domain.
2. **Home** – two of the scenarios used in our user study (discussed in Section 6.3), including the kitchen scenario presented in Section 6.1.1.
3. **International Planning Competition (IPC)** – two domains (Driverlog, Depots) offering longer plans.
4. **Corridor** – an epistemic planning benchmark with epistemic goals such as selective communication of a secret. A simplified version was used in our study.

We created multiple problem instances for each domain by varying the positions and number of agents and objects, which impacted the human's replanning and the robot's helpfulness. For each problem instance, we created 2 additional versions: (1) for the condition where the robot has false beliefs, the initial state accordingly included false

beliefs held by the robot; (2) for the Theory of Mind deficit condition, the initial state was such that the robot believes that all other agents share its beliefs (in addition, conditional effects were modified as was done in the **ToM-Def** condition in our study, discussed in Section 6.3).

6.4.2 Domain Descriptions

In what follows, we describe in detail the **Home** domains (comprising the phone charger domain and the simplified Corridor domain) used in our evaluation of the robot's helpfulness. For a detailed description of the BW4T, IPC, and the (non simplified) Corridor domains we refer readers to the previous chapter (Section 5.3.2).

Phone charger scenario

In the phone charger scenario, we focus on *visual perspective taking* (e.g., Flavell, 1977; Gzesh & Surber, 1985; Hamilton et al., 2009; Milliez et al., 2014; Lakatos et al., 2021) which is the ability to see the world from another person's perspective. In particular, we use *gaze detection* to recognize the direction in which each agent's gaze is facing. Recall that we use MonoLoco's orientation estimation to perform gaze detection. In particular, we predict that a person has shifted their gaze if the predicted orientation given by MonoLoco changes (see Appendix C.1 for lower-level implementation details).

Previous work (Gasquet et al., 2016; Hu et al., 2019, 2022) has formalized visual perspective taking using epistemic logic and, inspired by this formalization, we implemented a simplified visual perspective taking mechanism in RP-MEP (Muise et al., 2021). Recall RP-MEP's conditioned mutual awareness mechanism that models and enforces conditions for agents' 'awareness' of the effects of an action. For instance, in our kitchen scenario agents are 'aware' that an action has been performed if they are in the same location in which the action was performed.

To model visual perspective taking, we encode actions such that agents are 'aware' that an action has been performed only if their gaze is *facing* the 'appropriate' direction. For instance, if Bob is picking up an object in a certain corner of the room and Alice is facing the corner opposite to it, then Alice will not believe that Bob is now holding that object. With that in mind, we describe the 'phone charger' scenario and relate it to visual perspective taking:

> *Alice and Pepper the robot are in the living room. The scenario begins by Alice connecting her phone to a phone charger and immediately discon-*

> *necting it when she suddenly receives a call. Alice answers the call and proceeds to sit down at the table (with her back to the phone charger). As Alice begins her phone call, Bob walks into the living room, unplugs the charger, and leaves the room with it.*

After observing Alice connecting her phone to the charger, Algorithm 3's plan recognition component recognizes that Alice's goal is to charge her phone (and this goal persists even when Alice takes the phone call and sits down at the table). Algorithm 3's plan recognition component also recognizes that Alice's plan is to charge her phone using the charger.

As discussed in Section 6.2, the perception module in this case – upon observing that Alice sat down with her back to the charger – will produce the (informally specified) observation *shiftGaze*(Alice, awayFromCharger). More formally, the room was split into 4 quadrants and awayFromCharger specifies one of these quadrants. In Line 4 of Algorithm 3, the state S is progressed with the *shiftGaze* action, after which Pepper believes that Alice is not facing in the direction of the charger.

After observing that Bob has entered the room, unplugged the charger, and left the room, Pepper reasons that the charger is no longer there. However, via visual perspective taking, Pepper reasons that Alice – whose gaze is facing away from the phone charger – does not update her beliefs about the location of the charger.

Pepper now believes that Alice holds a false belief about the location of the charger (i.e., Pepper believes that the charger is no longer there while believing that Alice believes that it is still there) and therefore perceives a discrepancy between its beliefs and Alice's pertaining to the validity of her plan of charging her phone. Given that Pepper believes that there is another charger in the bedroom, in Line 8 an assistive solution π_{assist} is generated which involves Alice obtaining the charger from the bedroom. To resolve the discrepancies pertaining to the validity of Alice's plan and the assistive solution, Pepper can do one of two things:

> (1) Pepper goes to the bedroom, picks up the spare phone charger, and brings it back to the living room to replace the charger that was taken by Bob.

or

> (2) Pepper informs Alice that the charger is no longer in the living room and that there is another charger in the bedroom.

(Simplified) Corridor scenario

As mentioned, this is a simplified version of the Corridor domain. In particular, the scenario is as follows:

> *Alice, Bob and Pepper are in the living room. Alice and Bob are sitting at the table with Pepper nearby, discussing work matters. Eve walks by and says goodbye to Alice and Bob. Bob whispers to Alice (with Pepper within earshot): "I gotta tell you something about Eve's birthday party but Eve can't find out!". From Pepper's vantage point, it notices that Eve left the room but is right by the door (looking for something in her bag), within earshot but out of sight of Alice and Bob.*

As in the phone charger scenario, this scenario involves an element of visual perspective taking since Pepper believes that Alice and Bob's gaze is facing a direction such that they cannot see that Eve is right outside the door. As such, Pepper believes that Bob holds a false belief about Eve's location and therefore Pepper perceives a discrepancy pertaining to Bob's plan of sharing his secret with Alice. Pepper believes that if Bob shares his secret with Alice while Eve is just outside the door, Eve will come to learn the secret and Bob's epistemic goal will therefore not be achieved (and worse, Bob's goal will become irrevocably unachievable). The *shareSecret* action (whose encoding in shown in Appendix B.1.1) is slightly altered to model this.

To resolve this discrepancy, Pepper can either

> (1) Send a message to Bob, informing him that Eve is standing just outside the door,

or

> (2) Close the door.

Finally, we note that in this dissertation we do not address the interesting natural language understanding task of inferring Bob's epistemic goal from his utterance, *"I gotta tell you something about Eve's birthday party but Eve can't find out!"*. Instead, Bob's goal is assumed to be already known (and reflected in G_H) and given to Algorithm 3, namely $B_{\text{Alice}}(\textit{Eve\_birthday\_party\_secret}) \wedge \neg B_{\text{Eve}}(\textit{Eve\_birthday\_party\_secret})$.

6.4.3 Human Replanning in our Helpfulness Experiments

As mentioned earlier in this section, following the execution of the discrepancy resolving plan in Line 11 of Algorithm 3, the simulated human agent's plan is executed

and, if it does not achieve the human's goal, RP-MEP is tasked with generating a new plan for the human which involves sensing the environment until the goal is achieved. In what follows we provide details on the replanning process in each of the domains used in our evaluation.

BW4T

In the BW4T domain, replanning yields a plan wherein Bob searches all 'areas' of the room he is in and then proceeds to the closest room and searches all of its 'areas' and so on, until the medical kit is found. If Pepper provides Bob with information (as part of discrepancy resolution) and changes his beliefs about the location of the medical kit, RP-MEP will generate a plan for Bob that involves heading to the location provided by Pepper.

Corridor

In the Corridor domain, if Pepper does not intervene and resolve the discrepancy it perceives between its beliefs and Bob's beliefs, then Bob's epistemic goal becomes irrevocably unachievable. This is because Bob will share his secret with agent k but will inadvertently also share it with agent l, because of his false belief. Consequently, $B_k(\text{BobSecret}) \wedge B_l(\text{BobSecret})$ will hold following Bob's plan and there is no action that can negate $B_l(\text{BobSecret})$. Therefore, Bob's epistemic goal cannot be achieved and so replanning will yield an empty plan.

IPC

In the IPC domains, replanning yields a plan wherein Bob searches all 'areas' of the location he is in and then proceeds to the closest location and searches all of its 'areas' and so on, until either the package (in the Driverlog domain) or the crate (in the Depots domain) is found, depending on the domain. If Pepper provides Bob with information and changes his beliefs about the location of the crate or package, RP-MEP will generate a plan for Bob that involves heading to the location provided by Pepper.

Phone charger

In the phone charger domain, replanning yields a plan wherein Alice searches all 'areas' of the room she is in and then proceeds to the closest room and searches all of its 'areas' and so on. If Pepper provides Alice with information and changes her

	ToM + correct beliefs				ToM deficit				ToM + false beliefs			
	Cost (π_H)	Cost (π_{HR})	H	H_R	Cost (π_H)	Cost (π_{HR})	H	H_R	Cost (π_H)	Cost (π_{HR})	H	H_R
BW4T	8	3	5	0.625	8	8	0	0	8	10	-2	-0.25
Home	5	2	3	0.6	5	5	0	0	5	7	-2	-0.4
Depots	32	17	15	0.469	32	32	0	0	32	39	-7	-0.219
Driverlog	25	19	6	0.24	25	25	0	0	25	31	-6	-0.24
Corridor	∞	4	∞	–	∞	∞	–	–	∞	∞	–	–

Table 6.3: **Theory of Mind capabilities and accurate beliefs improve a robot's helpfulness.** We report the impact of a robot's Theory of Mind capabilities and the veracity of its beliefs on its helpfulness (H) and relative helpfulness (H_R). π_H and π_{HR} are the 'human alone' and 'human-robot team' plans, respectively.

beliefs about the location of the phone charger, RP-MEP will generate a plan for Alice that involves heading to the location provided by Pepper.

6.4.4 Results

Table 6.3 shows the robot's average helpfulness (H) and relative helpfulness (H_R) for each condition across **15** problem instances in each of the domains. When the robot has no deficit in Theory of Mind (ToM) and no false beliefs (the ToM + correct beliefs condition) it is most helpful, compared to a robot with a deficit in Theory of Mind that is not helpful (H and H_R are 0 in this case). Finally, a robot with no deficit in Theory of Mind but with false beliefs (the ToM + false beliefs condition) has, on average, negative helpfulness since it prolongs the human's plan by communicating incorrect information.

While these results were not unexpected, they are nevertheless indicative of the importance of the *veracity* of a robot's beliefs. They moreover elucidate that a robot with inaccurate beliefs risks causing 'more harm than good'. Additionally, in our evaluation of helpfulness we allowed the robot in the 'ToM + correct beliefs' condition to resolve discrepancies by either making physical changes in the environment or communicating with the human. However, in the 'ToM + false beliefs' condition we only allowed the robot to resolve discrepancies via communication. One can imagine how a robot with false beliefs could make changes to the environment in a myriad of undesirable ways when attempting to resolve discrepancies while trying to be assistive. In Chapter 7 we discuss this in the context of AI safety and the potential broader

impact of the work in this dissertation. Finally, for a more in-depth discussion of the importance of the veracity of agents' beliefs (in a plan recognition context), see Chapter 4 (Section 4.2).

An 'Infinitely' Helpful Robot

In the last row of Table 6.3 (i.e., the Corridor domain), the cost of the plan π_H is always infinite (∞) since Bob's false beliefs cause him to generate an invalid plan that makes his epistemic goal irrevocably unachievable (e.g., Bob sharing his secret with Alice while Eve is, unbeknownst to Bob, behind the door eavesdropping). In contrast, the robot can assist by, for instance, closing the door, in which case Bob's plan will become valid and he will achieve his epistemic goal. The cost of the plan π_{HR} is therefore a finite number and, per Freedman et al. (2020), the robot's helpfulness in such cases is infinite (∞).

6.4.5 Runtime Results for Algorithm 3

The runtime and scalability of RP-MEP in conjunction with the epistemic planning-based techniques we leverage in this chapter have been evaluated in Chapters 4 and 5. It has been shown that the runtime and scalability bottleneck is the depth of nested belief and number of agents in the RP-MEP problem. As previously discussed in this dissertation, these results are consistent with the findings of RP-MEP's developers (Muise et al., 2021). In particular, RP-MEP's classical encoding of an RP-MEP problem generates an exponential number of fluents which significantly impacts the runtime as the depth of nested belief and number of agents grow (Muise et al., 2015b, 2021). In what follows we discuss runtime results for each RP-MEP-based component of Algorithm 3.

Epistemic plan recognition runtime

In Chapter 4 (Section 4.4) we showed that the runtime of Algorithm 1 remained low as long as the number of agents and depth of belief remained low as well. In Line 6 of Algorithm 3, Algorithm 1 is called. The runtime results for our small domains (i.e., BW4T, Home, and Corridor) were similar to those in Section 4.4 – the human's presumed plan and goal were returned in under 1.5 seconds for all problem instances. This is due to the fairly small number of actions in the domain, small number of observations (due to short plans), and small number of agents. In contrast, the IPC domains Depots and Driverlog are more complex and plans generated to achieve

goals in these domains are considerably longer than in our other domains. As shown in Table 4.1, runtime can reach minutes. Indeed, in our helpfulness evaluation we observed similar runtimes for the two IPC domains used.

Discrepancy resolution runtime

In Chapter 5 (Section 5.3) we observed runtime results for Algorithm 2 that align with those observed in Chapter 4. That is, as the number of agents and depth of belief grow, so does the runtime. We moreover showed in Section 5.3 how, when the depth of nested belief was sufficiently high, the planner ran out of memory in some of the problem instances.

In Line 10 of Algorithm 3, we call the modified Algorithm 2. For the Home, Corridor, and BW4T domains, the discrepancy resolution algorithm returned a discrepancy resolving plan in under 2.5 seconds for all problem instances. Similarly to the plan recognition runtime results reported earlier, in the IPC domains the runtime was considerably higher on average, with Line 10 taking up to 5 minutes to return a discrepancy resolving plan in some cases.

Real-time execution of Algorithm 3 in our user study video recordings

As mentioned, RP-MEP's runtime can remain low when the number of agents and depth of nested belief stays low as well. This is observed in our simpler domains, as discussed previously. Muise et al. (2021) moreover mention that most interesting use cases of epistemic planning involve a relatively low depth of nested belief. Indeed, our study included three fairly realistic scenarios that involve complex Theory of Mind reasoning. In our study, domains were simplified such that each iteration of Algorithm 3's while loop took less than a second. It was therefore possible to run the algorithm in real time and in conjunction with the perception module (i.e., the various perceptual detectors used).

Takeaway

While we were able to run Algorithm 3 in real time in our simplified domains, it is clear that in order to run Algorithm 3 in more complex domains and in real time, significant runtime improvements will have to be made. Fortunately, epistemic planning is an active research field and planner runtime will surely improve over time, as is the case for research on automated planning more generally.

6.5 Related Work

Our work in this chapter comprises two high-level components – *proactive (robotic) assistance* and *Theory of Mind reasoning in robotic systems*. In this section, we survey the rich body of extant work that has investigated each of these components and situate our work within it.

6.5.1 Augmenting Robotic Systems with Theory of Mind Reasoning

As mentioned in the beginning of this chapter, the role of Theory of Mind and mental modeling in human-robot interaction and teaming has garnered a fair degree of attention (e.g., Scassellati, 2002; Trafton et al., 2005; Berlin et al., 2006; Breazeal et al., 2009; Sindlar et al., 2009; Breazeal et al., 2010; Nikolaidis & Shah, 2012; Talamadupula et al., 2014; Leyzberg et al., 2014; Zhao et al., 2015; Devin & Alami, 2016; Nikolaidis et al., 2017; Görür et al., 2017; Bühler & Weisswange, 2018; Bianco & Ognibene, 2019; Brooks & Szafir, 2019; Dissing & Bolander, 2020; Bühler & Weisswange, 2020; Gervits et al., 2020; Buckingham et al., 2020a; Bühler et al., 2021; Bühler, 2022; Bianco & Ognibene, 2022; Favier et al., 2022; Jakubczak et al., 2022; Yu et al., 2023). Mental modeling and Theory of Mind reasoning are largely overlapping concepts in the fields of AI, HCI, and HRI. Indeed, a robot modeling the mental states of other agents is employing its Theory of Mind (Tabrez et al., 2020). Tabrez et al. (2020) and Gurney & Pynadath (2022) present a comprehensive survey of this body of work.

For instance, Devin & Alami (2016) presented a Theory of Mind-based approach that allows a robot estimate and maintain mental states of other agents pertaining to the environment, plans, and goals in the context of human-robot shared plan execution. As we do in this chapter, Devin & Alami's system also incorporates a symbolic planner however their planner is not epistemic and does not support some of the complex Theory of Mind reasoning supported by our approach. In addition, Breazeal et al. (2009) developed a robot that maintained belief bases representing the first-order beliefs of other agents. Breazeal et al.'s system employs a form of visual perspective taking (Flavell, 1977) by updating some agent's belief base given some percept, only if the robot believes that the agent in question observed the percept.

Further, a work that is closely related to ours (and has partly inspired the work in this chapter) is that of Dissing & Bolander (2020). In their work, Dissing & Bolander leverage Dynamic Epistemic Logic (DEL) and enable a robot to perform arbitrary order reasoning about the beliefs of other agents. They moreover show how a hu-

manoid robot (Pepper, the robot also used in this chapter) can pass false-belief tasks involving human agents. Dissing & Bolander even incorporate an intention recognition component into their work, which allows the robot to offer proactive assistance to a human via (what can be seen through the lens of our work as) rudimentary discrepancy resolution. Lastly, as we do in this chapter, Dissing & Bolander employ perception algorithms, ground their output, and provide the grounded symbols to a symbolic system. While Dissing & Bolander's work supports reasoning about false belief and a restricted form of plan recognition, it does not integrate an epistemic planner, nor does it incorporate a principled approach to plan and goal recognition or discrepancy resolution.

Indeed, most of the aforementioned extant work has not appealed to *epistemic planning* which is noteworthy in being able to support a variety of human-robot interaction scenarios requiring complex Theory of Mind reasoning, as elucidated by the investigation into proactive robotic assistance discussed in this chapter. Moreover, as discussed by Dissing & Bolander (2020) and Bolander et al. (2021), the approach taken by most previous work – maintaining a distinct knowledge base for each agent – is not expressive enough to support complex higher-order belief and knowledge attribution in the context of multiple agents. Lastly, epistemic planning techniques are well suited for reasoning about and planning for epistemic goals which, as illustrated by our Corridor scenario (Section 6.4.2), is useful in real-world situations.

As discussed earlier in this chapter, some work has utilized epistemic planning for human-robot interaction and collaboration (Petrick & Foster, 2013; Miller et al., 2018; Petrick & Foster, 2020; Foster & Petrick, 2020; Foster et al., 2020; Bolander et al., 2021; Soldà et al., 2022; Bramblett et al., 2023; Bramblett & Bezzo, 2023). Petrick & Foster (2020) build on the epistemic planning system PKS (Bacchus & Petrick, 1998; Petrick & Bacchus, 2002) and allow a robot to construct plans in social environments (e.g., with the aim of improving children's healthcare experiences). Further, Miller et al. (2018) employed RP-MEP in the context of human-robot teaming. The motivation of their work and the work presented in this chapter certainly overlap in that we are both interested in leveraging belief-based multi-agent epistemic planning in human-robot interaction. However, Miller et al.'s work does not offer a proactive robotic assistance algorithm that includes principled approaches to plan recognition and discrepancy resolution, nor does it offer an empirical investigation of the helpfulness of a robot employing epistemic planning techniques.

In addition, Gervits (2020) investigated the use of an epistemic planner in human-robot teaming to enable effective communication and maintain common ground amongst

team members. Gervits's work is related to ours in that the robot strives to align its beliefs and its beliefs about the human's beliefs (e.g., when the robot moves to a different area, it should announce its new location because the move action was not observed by the human). However, Gervits's work does not consider false belief and does not employ a principled approach to resolving discrepancies or recognizing agents' goals and plans. Moreover, Gervits's work highlights a challenge in mental model alignment – what beliefs should be aligned? Obviously, aligning every single belief could result in the recipient of the information quickly becoming annoyed or overwhelmed. In our approach, beliefs are aligned (i.e., discrepancies are resolved) only when there are threats to the achievement of the human's goal. Even when the robot perceives discrepancies between its beliefs and the human's beliefs about some formula ϕ, but the discrepancy does not threaten the achievement of the human's goal, Algorithm 3 will return an empty discrepancy resolving plan and the robot will not 'bug' the human with unnecessary information.

Furthermore, Bolander et al. (2021) build on the robotic platform introduced by Dissing & Bolander (2020) and integrate a DEL-based epistemic planner that supports implicit coordination[2] (Engesser et al., 2017). Bolander et al. illustrate how epistemic planning (and in particular the implicit coordination paradigm) may be used to facilitate decentralized execution of the robot's plan alongside the human's plan, such that the agents' joint goal is achieved. While Bolander et al.'s work addresses settings that our work cannot (e.g., the Multi-Agent Pathfinding with Destination Uncertainty Task (Bolander et al., 2018), where agents have the joint goal of each reaching its assigned destination, and must collaboratively manoeuvre through a grid world), the planner they employ is restricted to reasoning with S5 logic and is therefore only able to model knowledge, rather than belief. Future work should investigate synergies between our approach and Bolander et al.'s, and combine the strengths of each.

Lastly, Soldà et al. (2022) propose a planning framework that coordinates a set of autonomous agents – e-PICO (**EPI**stemic Reasoner **C**ollaborative R**O**bots). Their framework integrates a classical planner with an epistemic planner and enables epistemic reasoning in a multi-agent setting. Soldà et al. validated their approach using two robotic arms in a collaborative setting where the robots must find out what blocks they are able to manipulate via sensing actions. Moreover, epistemic reasoning allows the robots to communicate the color of blocks to each other. In their

[2]See discussion of implicit coordination in relation to our discrepancy resolution approach in Chapter 5 (Section 5.5).

evaluation, Soldà et al. employ the epistemic planner EFP (Le et al., 2018; Fabiano et al., 2020) (discussed and utilized in Chapter 4 (Section 4.4)) to solve multi-agent epistemic planning problems involving sensing and communication actions. As with Bolander et al.'s work, the motivation of Soldà et al.'s work and the work discussed in this chapter overlap. However, Soldà et al.'s work focuses on a use case more akin to the collaborative settings addressed by the implicit coordination paradigm. Moreover, their approach does not consider plan recognition or discrepancy resolution. Finally, while the epistemic planner employed by Soldà et al. can handle some form of false belief, that type of epistemic reasoning is not the focus of their work while it is fundamental to the work presented in this chapter and in this dissertation more generally.

In this chapter we advance the line of research that leverages epistemic planning in human-robot interaction by (1) proposing a novel *multi-agent* epistemic planning-based approach to proactive assistance that supports reasoning about *beliefs*, and by (2) evaluating the helpfulness and perceived Theory of Mind capabilities of robots implementing our proposed approach. Moreover, our approach employs principled, epistemic planning-based approaches to plan recognition, assistive planning, and discrepancy resolution that are fairly general, and that demonstrably support a myriad of complex human-robot teaming scenarios.

Finally, the work presented in this chapter falls under the umbrella of the decades long research on *cognitive robotics* (Levesque & Lakemeyer, 2008), a term coined by Ray Reiter in his lecture on receiving the Research Excellence Award in the 1993 International Joint Conference on Artificial Intelligence (IJCAI). In his talk, Reiter laid out his vision for cognitive robotics, which focused on the fundamental role of knowledge representation and reasoning in the design of robots that can reason and act in incompletely known and unpredictable environments.

6.5.2 Proactive (Robotic) Assistance

Grosinger (2022) defines proactivity for AI systems as *"the ability to autonomously initiate anticipatory action based on reasoning, meant to impact people and/or their environments"*. Indeed, a large body of extant work has investigated assistive behavior in robots that satisfies this definition (e.g., Grosinger et al., 2019; Harman & Simoens, 2020; Harman, 2020; Kulkarni et al., 2021; Buyukgoz et al., 2022). Moreover, studies have demonstrated that humans prefer robots that exhibit such proactive behavior (Zhang et al., 2015; Baraglia et al., 2017). Further, a subset of this body

of work has integrated plan recognition and planning, as we do in this chapter, to facilitate proactive assistance both in robotic as well as virtual agents (e.g., Hoffman & Breazeal, 2007b,a; Levine & Williams, 2014; Chakraborti et al., 2015; Geib et al., 2016; Freedman & Zilberstein, 2017; Freedman et al., 2019; Harman & Simoens, 2020).

The work presented in this chapter partly shares its motivation with the body of extant work on proactive robotic assistance. In our work we broaden the scope of proactive robotic assistance by appealing to epistemic planning and enabling the robot to perform complex Theory of Mind reasoning. In particular, our approach allows a robot to autonomously initiate action based on (1) reasoning about the plans and goals of other agents (based on the novel approach to epistemic plan recognition presented in Chapter 4) and (2) reasoning about threats to the achievement of other agents' goals (based on the novel approach to discrepancy resolution presented in Chapter 5). Importantly, as mentioned earlier in this chapter, a robot employing our approach acts only if needed, i.e., if it perceives discrepancies between its beliefs and those of other agents pertaining to the validity of those agents' plans.

6.6 Concluding Remarks

In this chapter we examined how robots can use their Theory of Mind to proactively assist humans. We developed and implemented an epistemic planning-based approach to proactive assistance. Our approach built on the contributions of the previous chapters and enabled a robot to recognize another agent's plan and goal and resolve any discrepancies pertaining to the validity of the other agent's plan by acting in the environment. Our evaluation – comprising a user study and a set of simulations – showed that robots demonstrating Theory of Mind and implementing our approach were measurably more helpful and perceived by humans as possessing greater Theory of Mind capabilities compared to robots with a deficit in ToM.

While the focus of our implementation was to demonstrate the feasibility of integrating complex Theory of Mind reasoning with a robotic system rather than solving orthogonal research challenges in robotics, the settings in which we demonstrated the robot's behavior are nevertheless representative of important real-world human-robot scenarios. Indeed, we are hopeful that our work constitutes another step towards the creation and deployment of assistive robots in real-world settings.

Chapter 7

Conclusion

Be it in time sensitive, critical tasks such as search and rescue, or in more mundane interactions between domestic robots and household members, it is often essential for AI systems to display advanced social cognitive abilities and navigate complex social settings involving multiple agents. These abilities are paramount to reasoning about other agents' behavior, effectively communicating with them, and generating helpful behavior. Research (e.g., Baron-Cohen et al., 1985; Baron-Cohen, 1991, 1997) shows that an important precursor to the development of these abilities in humans is Theory of Mind. In Chapter 1, we posited that if AI systems possess Theory of Mind capabilities, this could improve upon their existing social cognitive abilities and further benefit humans with whom they interact. We correspondingly proposed the following thesis statement:

> **Thesis Statement.** *Augmenting AI systems with Theory of Mind reasoning is feasible and useful for explanation, plan recognition, and assistance, and the application of such systems in real-world settings has the potential to benefit humans interacting with them.*

To support our thesis statement, we have presented in this dissertation a body of work that investigates the role of Theory of Mind in a number of reasoning tasks and takes steps towards the betterment of social cognitive abilities held by present and future AI systems. In what immediately follows, we provide a more detailed summary of the contributions of this dissertation that support our thesis statement. Following discussion of our contributions, we discuss the potential broader impact of our work.

7.1 Summary of Contributions

We summarize the technical and theoretical contributions of this dissertation in support of our thesis statement. In Chapters 3–6, we elucidated the importance and *usefulness* of Theory of Mind in explanation, plan recognition, and assistance. In particular, we exposed a number of important considerations in these reasoning tasks that were, in conjunction, missing from extant work including reasoning about epistemic goals and nested belief. In Chapters 4–6, we developed Theory of Mind-based computational solutions to these tasks by appealing to the computational paradigm of *epistemic planning*. Our experimental evaluation established the *feasibility* of augmenting AI systems with Theory of Mind via epistemic planning. In particular, while we showed that the complexity of a domain along various dimensions, such as depth of nested belief, is a limiting factor for some epistemic planning systems, we also showed that runtime remains reasonable in many settings requiring complex Theory of Mind reasoning. Indeed, in Chapter 6 we discussed how our developed computational solutions were integrated with a robotic system and ran in real time. Furthermore, while the settings in which we demonstrated the robot's behavior in Chapter 6 were restricted and simplified, they are nevertheless representative of important human-robot scenarios. As such, we are hopeful that our work constitutes another step towards the creation and deployment of assistive robots that can *benefit* humans with whom they interact on a daily basis. The main contributions of this dissertation can be then summarized as follows:

1. We identified a set of desiderata for explanations that utilize Theory of Mind (Section 3.1).
2. We presented a belief-based account of explanation, whose design is informed by the aforementioned desiderata (Section 3.2).
3. We proved a number of theorems pertaining to various properties of our account of explanation and various explanation types therein (multiple sections in Chapter 3).
4. We proposed and formalized the notion of epistemic plan recognition which adds an important dimension to the recognition process by appealing to a notion of epistemics to allow for the recognition of epistemic goals and to model the observer and its knowledge of the actor as first-class elements of the recognition process (Section 4.1).

5. We proposed a computational realization of epistemic plan recognition as epistemic planning, which synthesizes formalisms and computational techniques and enables the use of existing planning tools (Section 4.3).
6. We evaluated our approach on a set of epistemic plan recognition problems, using a number of epistemic planners (Section 4.4).
7. We evaluated the impact of the veracity of the observer's beliefs on goal recognition accuracy (Section 4.4).
8. We proposed a formulation of discrepancy resolution that appeals to a multi-agent epistemic logic (Section 5.1).
9. We developed an algorithm that resolves discrepancies via epistemic planning and established its soundness (Section 5.2). We also released a repository that includes our environments and implementations of the epistemic planning-based discrepancy resolution technique introduced in Section 5.2.
10. We demonstrated that epistemic planning tools can be used to resolve discrepancies via different modalities (i.e., with epistemic communication actions or ontic actions) in various domains and evaluate the impact of the depth of nested belief on the runtime of our algorithm (Section 5.3).
11. We conducted a user study which indicates that our approach can effectively resolve misconceptions held by humans pertaining to plan validity (Section 5.4).
12. We developed an algorithm that integrates the epistemic planning-based techniques presented in Chapters 4 and 5 to enable proactive robotic assistance via Theory of Mind (Section 6.1).
13. We implemented our algorithm and integrate it with a humanoid robot (Pepper (Pandey & Gelin, 2018)), combining various perception techniques with Theory of Mind reasoning (Section 6.2).
14. We conducted a study evaluating how participants perceive the Theory of Mind capabilities of a robot employing our proposed approach for Theory of Mind-based proactive assistance (Section 6.3).
15. Finally, we utilized a quantified metric of a robot's helpfulness (Freedman et al., 2020) to measure the efficacy of our method in a set of simulations across various domains (Section 6.4).

7.2 Future Work

This dissertation constitutes a snapshot along a continuum of research, and as such there are many possible trajectories for future research that we or others can follow. We list some of these trajectories below and note that many of them were elaborated upon in the previous chapters.

Epistemic Plan Recognition

In Section 4.5, we discussed several potential future work ideas for extending our approach to improve its performance and applicability in more complex scenarios. One idea is to implement runtime optimizations for Algorithm 1, which could involve using landmark-based techniques to improve efficiency. Another idea we discussed is extending our approach to settings where the actor may be intentionally obfuscating their plan and goal.

Additionally, we discussed in Section 4.5 how our approach could be extended to partially observable settings, where the observer may be uncertain about the environment and other agents' beliefs. Finally, we discussed the possibility of endowing an observer with agency to act in the environment and improve the accuracy of its beliefs.

Discrepancy Resolution

In Chapter 5, we discussed several potential future work ideas for extending our discrepancy resolution approach. One idea is to extend our approach to resolve discrepancies pertaining to additional properties of plans, such as optimality. Another potential area for exploration is experimenting with various constraints on the generation of discrepancy-resolving plans.

Moreover, an interesting avenue for future work is the integration of trust into our discrepancy resolution framework. For instance, in our current implementation, when the discrepancy resolving agent communicates with another agent, it assumes that the agent receiving the communication trusts the communication (and the agent providing it) and thus comes to believe its contents. However, more generally, trust is an important dimension of multi-agent systems, especially those involving both humans and AI systems, and should be considered by our approach to enhance its robustness and generality.

Finally, in Section 5.4 we discussed the possibility of conducting a usability study involving participants performing a task in a lab and interacting with a virtual as-

sistant or robot employing our proposed approach for discrepancy resolution. Such a study would provide valuable insights into the effectiveness and user experience of our approach in real-world scenarios.

Proactive Robotic Assistance

In Section 6.1, we discussed several potential future work ideas for extending our approach to proactive robotic assistance to improve its performance, effectiveness, and applicability in real-world scenarios. One idea is to relax the assumptions made in our approach by, for example, considering execution monitoring and parallel execution of plans by the robot and other agents. Another potential area for exploration is investigating when it is appropriate for the robot to intervene.

In addition, in Section 6.3 we discussed the possibility of conducting in-situ experiments with Pepper and participants in a lab setting. Finally, we discussed the importance of improving the robustness of the implementation of our approach for proactive robotic assistance. This could involve relaxing some of the simplifications made on the perception front, such as object manipulation and recognition, as well as generalizing our over-specialized event detection logic. By improving the robustness of our approach, we can make it more applicable in a wider range of domains and scenarios.

7.3 Broader Impact and Reflections

We begin the discussion in this section by bringing forth our previous work that discussed the notion of *empathy* (Shvo & McIlraith, 2019). Empathy is often thought of as *the ability to understand and share the thoughts and feelings of another* and has an extremely rich history, beginning with its philosophical foundations and leading to research in fields such as psychology, ethics, and neuroscience (e.g., Coplan & Goldie, 2011; Davis, 2018). Empathy has been found to have two components, an affective, low-level component, and a cognitive, high-level component, with the two being interconnected (Shamay-Tsoory, 2011). The affective component allows one to share in the emotional experiences of another via affective reactions to another's affective states. The cognitive component of empathy utilizes Theory of Mind and has been the focus of this dissertation. It allows one to take the perspective of another, thereby facilitating reasoning over their *mental* or *affective* state. Our work (Shvo & McIlraith, 2019) also focused on the cognitive component of empathy and worked towards building empathetic agents that can reason about the mental and

affective states of other agents. While reasoning over the affective states of agents is beyond the scope of this dissertation, we nevertheless submit that AI systems with advanced social cognitive abilities should in some instances be equipped with a means of reasoning about the affective state of humans. This type of reasoning will lead to more socially acceptable behavior, as highlighted by recent work (e.g., McDuff & Czerwinski, 2018; Petrick et al., 2019; Petrick & Hill, 2019; Lindsay et al., 2022; Houlihan et al., 2023).

In our previous work we also discussed the following: while we aim to build *assistive* empathetic agents (e.g., the proactively assistive robotic system described in Chapter 6), empathy can also facilitate malicious (or simply self-serving) motivations through manipulation. Indeed, Shvo & McIlraith (2019) discuss the notion a Machiavellian agent, who uses deception, manipulation, and exploitation to benefit its interests. Clearly, an AI system equipped with Theory of Mind reasoning capabilities (like the ones discussed in this dissertation) will have a greater potential to manipulate other agents than an AI system that is not aware of the mental states of others (Carroll et al., 2023). Interestingly and fortunately, in humans, perspective taking (and therefore cognitive empathy and Theory of Mind) and Machiavellianism have been found to be negatively correlated (Barnett & Thompson, 1985). More specifically, the will to manipulate is present in Machiavellians, but the means by which to do so are often not. Unfortunately, AI systems augmented with Theory of Mind reasoning need not suffer from a similar negative correlation. That is, such systems could be programmed to follow (or optimize for) Machiavellianistic behavior and do well at it due to their Theory of Mind reasoning capabilities. Thus, measures should be taken to prevent AI systems from being Machiavellian.

Conversely, are there instances where an AI system *should* lie, deceive, or disobey? For instance, a study conducted by Chakraborti & Kambhampati (2019) showed that human participants were, in general, positive towards an AI system lying, if it was done for the 'greater good'. Moreover, the notion of intelligent disobedience has garnered some attention in recent years (Chaleff, 2015; Briggs & Scheutz, 2017; Coman & Aha, 2018; Mirsky & Stone, 2021b; Briggs et al., 2022; Arnold et al., 2022). For example, a 'seeing eye robot' (Mirsky & Stone, 2021a), meant to assist its visually impaired owner, should be able to intelligently disobey an order from its human owner, just like a biological seeing eye dog would. Theory of Mind reasoning is crucial in such settings, for a number of reasons, two of which we discuss here. First, an AI system engaging in intelligent disobedience must reason about the intentions, plans, and beliefs of whomever it is disobeying. As an example, rather than reacting to a

fixed set of scenarios, a seeing eye robot should be able to perform visual perspective taking, understand that its owner holds a false belief (e.g., due to the owner's visual impairment), understand further that their owner likely has a certain plan and goal and that this plan will fail to achieve their goal (and may even put them in harm's way) because of their false belief. In such cases, intelligently disobeying the owner seems warranted. Second, using Theory of Mind reasoning to generate explanations (as is advocated in Chapter 3) for disobedient behavior is desirable. Indeed, disobedient behavior is bound to alarm a human interacting with an AI system and explanation of the behavior (in terms of the robot's beliefs about the human's beliefs, plans, and goals; and tailored to the human's mental state) may be helpful and reassuring.

Furthermore, in Chapter 6 we saw how an assistive robot with false beliefs was 'negatively helpful'. Moreover, we mentioned how a robot with false beliefs could conceivably make changes to the environment in a myriad of undesirable ways when attempting to resolve discrepancies while trying to be assistive. Indeed, in Chapter 5 we discussed how our definitions of discrepancy resolving plans do not consider the plans of the other agents in the environment. Therefore, it is possible that in the process of resolving discrepancies pertaining to some agent, new discrepancies pertaining to the validity of other agents' plans will be introduced, making some agent's plan invalid while *they* still believe it is valid. Theory of Mind reasoning can be helpful in such cases since it allows us to specify undesirable consequences of an agent's behavior that pertain not just to the physical world, but also to other agents' mental states. More broadly, these considerations pertain to AI safety (e.g., Amodei et al., 2016; Klassen et al., 2022b; Alizadeh Alamdari et al., 2022), an area of research concerned with undesirable side effects that may result from incomplete objective specifications. Indeed, Klassen et al. (2022a) and Klassen et al. (2023) propose the notion of *epistemic side effects* which are unintended changes made to the knowledge or beliefs of agents. While Theory of Mind reasoning is an important dimension of AI safety (as evident by Klassen et al.'s investigation), it is underexplored and we are hopeful that the work in this dissertation will lead to further exploration in this area.

Finally, we would be remiss not to discuss large language models (LLMs)[1], which are firmly within the current AI zeitgeist at the time of writing. LLMs (Lee & Toutanova, 2018; Brown et al., 2020) are neural network models, based on the Transformer architecture (Vaswani et al., 2017), trained on massive amounts of data and have demonstrated success in numerous demanding benchmarks and tasks designed to evaluate different forms of reasoning (Mahowald et al., 2023; Lewkowycz et al.,

[1]See also the related notion of foundation models (Bommasani et al., 2021).

2022). Recently, it has been claimed that Theory of Mind capabilities have emerged in LLMs (Kosinski, 2023; Bubeck et al., 2023), with support for these claims coming from LLMs successfully passing variations on classic Theory of Mind tasks (e.g., the Sally-Anne task discussed earlier in this dissertation) (Kosinski, 2023). However, as discussed by Sap et al. (2022) and Ullman (2023), these successes should be viewed critically. Indeed, Ullman (2023) shows how these emergent capabilities are not robust by demonstrating that slight perturbations to the considered Theory of Mind tasks lead to failures.

Here, we do not wish to prognosticate on the Theory of Mind capabilities of future LLMs (or machine learning-based models that will evolve from them or independently of them), which are currently in their early days. However, we do wish to reflect on possible synergies between these models and the work discussed in this dissertation. In particular, we believe that one promising approach to realizing Theory of Mind-like behaviour in AI systems is to integrate symbolic approaches to Theory of Mind reasoning with LLMs. There are various ways to do so, including leveraging the powerful natural language parsing capabilities of LLMs and funneling the parsed language to symbolic approaches such as those described in this dissertation. Along a similar vein, Cohen & Galescu (2023) describe an approach to collaborative dialogue that proposes to leverage LLMs to perform semantic parsing on user utterances, which yields rich logical forms that are then used by the symbolic dialogue system.

Lastly, another possible integration between LLMs and symbolic approaches involves treating symbolic reasoners as external modules that the LLM can utilize (e.g., Karpas et al., 2022). In terms of Theory of Mind reasoning, one can imagine how an external symbolic epistemic reasoner could be made available to the LLM that will make calls to this module in order to, for example, maintain a consistent and rich knowledge base containing agents' belief and which undergoes belief revision and update operations given new evidence and agents' actions. As discussed in Chapter 1, symbolic and logic-based approaches can offer provably correct solutions and inspectable and traceable reasoning and, by allowing LLMs to leverage external modules based on these approaches, we could potentially combine the strengths of both worlds.

7.4 Concluding Remarks

In this dissertation we have presented a body of work that investigates the role of Theory of Mind in a number of reasoning tasks – explanation, plan recognition, and

assistance. Leveraging epistemic planning, we developed computational solutions that expand the role of Theory of Mind in these tasks. We moreover implemented these computational solutions and integrated them within a robotic system that demonstrates social cognitive abilities in complex settings, as well as conducted a study evaluating participants' perceptions of our robotic system's assistive behavior. The investigation into the role of Theory of Mind in this dissertation, spanning both theory and practice, provides valuable insights into the potential of Theory of Mind to enhance the social cognitive abilities of present and future AI systems.

Appendix A

The AGM Postulates

As discussed in Chapter 3, a large body of work has studied belief change in agents where belief revision typically concerns belief change in a static environment, possibly in the context of incorrect and partial beliefs. Amongst the most popular guidelines for belief revision are the AGM postulates (Alchourrón et al., 1985), and the DP postulates (Darwiche & Pearl, 1997) (for iterated revision). In Chapter 3 we do not require that our revision operator $*$ satisfy any such guidelines, with the exception of Theorem 3.4 where the belief revision operator is assumed to satisfy the AGM postulates. In this appendix we elaborate on these postulates.

We consider a set K of propositional formulas, closed under logical consequence, which implicitly represents the beliefs of an agent. The question posed by the field of belief revision is then how K should be modified to incorporate new information. Moreover, it is important to consider what constraints should be put on $K * \phi$, the revision of K by a (propositional) formula ϕ.

Alchourrón et al. (1985) proposed a set of postulates that a rational belief revision operator $*$ should follow. As mentioned, these postulates have been termed the AGM postulates (based on the authors' initials). We list these postulates below (the names of the postulates are taken from (Ditmarsch, 2005)). $K + \phi$, called the *expansion* of K by ϕ, is just the closure under logical consequence of $K \cup \{\phi\}$.

(AGM$*$1)	$K * \phi$ is deductively closed	type
(AGM$*$2)	$\phi \in K * \phi$	success
(AGM$*$3)	$K * \phi \subseteq K + \phi$	upper bound
(AGM$*$4)	If $\neg\phi \notin K$, then $K + \phi \subseteq K * \phi$	lower bound
(AGM$*$5)	$K * \phi$ is inconsistent iff $\models \neg\phi$	triviality

(AGM∗6) If $\models \phi \equiv \psi$, then $K * \phi = K * \psi$ extensionality

(AGM∗7) $K * (\phi \wedge \psi) \subseteq (K * \phi) + \psi$ iteration upper bound

(AGM∗8) If $\neg\psi \notin K * \phi$, then $(K * \phi) + \psi \subseteq K * (\phi \wedge \psi)$. iteration lower bound

Finally, as explained by Klassen (2021), the first postulate just ensures that $K * \phi$ has the right type, that of a deductively closed theory (like K). Postulate (AGM∗2) says that revision is successful, in that the formula revised by is believed. Postulates (AGM∗3) and (AGM∗4) relate revising by ϕ to expanding by ϕ. The triviality postulate requires the agent to incorporate the new information in a consistent way, if there is any way to do so. The extensionality postulate, (AGM∗6), says that the results of revising by equivalent formulas should be the same. The last two postulates relate revising by a conjunction $\phi \wedge \psi$ to first revising by ϕ and then expanding by ψ.

Appendix B

Discrepancy Resolution via Theory of Mind

To provide structure, we enumerate the different appendices below:

- **Appendix B.1:** we detail some of the domain and problem files used in the evaluation described in Section 5.3. We moreover provide a number of examples of the goals given to the classical planner and the generated discrepancy resolving plans.
- **Appendix B.2:** we describe how we adapted the classical planning domain Driver-log (encoded in PDDL) to an RP-MEP domain encoded in PDKBDDL.

B.1 Exemplary Domains

Here we detail some of the domain and problem files used in the evaluation described in Section 5.3. All files are in the PDKBDDL format which is a variant of PDDL. For the sake of readability, the domains seen here are partial and contain a subset of the actions found in the domains used in our evaluation. Moreover, to better illustrate the PDKBDDL format, some actions are grounded versions of the actions used in our experiments. Some of these domains can be found in our released repository `https://github.com/maayanshvo/ToM-discrepancy-resolution`. The repository includes our environments and implementations of the epistemic planning-based discrepancy resolution technique introduced in Section 5.2.

B.1.1 Corridor

domain-corridor.pdkbddl

```
(define (domain corridor)

    ; This specifies the agents in the set of agents Ag
    (:agents a b c)

    (:types loc door)

    (:predicates
        (secret ?agent)
        (at ?agent - agent ?l - loc)
        (connected ?l1 ?l2 - loc)
        (door_open ?d - door)
        (dummy)
    )

    (:action shareSecret
        :derive-condition   (at $agent$ l1)
        :parameters         (?a ?as - agent)
        :precondition       (and (at ?a l1) [?a](secret ?as))
        :effect             (and
                                (forall ?a2 - agent
                                    (when    (and (door_open dl1l2)
                                                  (at ?a2 l2))
                                         [?a2](secret ?as))
                                         )

                                (forall ?a3 - agent
                                    (when    (and (at ?a3 l1))
                                         [?a3](secret ?as))
                                         )
                            )
    )
```

```
    (:action informDoorOpen
      :derive-condition   (always)
      :parameters         (?a1 - agent ?d - door)
      :precondition       (and (door_open ?d))
      :effect             (and
                               [?a1](door_open ?d))

    )


    (:action closeDoor
      :derive-condition   never
      :parameters         (?d - door)
      :precondition       (and (dummy))
      :effect             (and

                               (!door_open ?d))

    )

)
```

Since planning in RP-MEP is from the perspective of a single agent (the root agent, as discussed in Chapter 2, Section 2.3.1), all fluents are implicitly preceded by the beliefs of that agent. In our evaluation, a single agent (Alice) resolves all discrepancies and so planning in RP-MEP is done from her perspective (as discussed in Chapter 5). For instance, in the *informDoorOpen* action in the Corridor domain, the effect [?a1](door_open ?d) is implicitly preceded by [Alice] such that the effect of the action is that Alice believes that agent a1 believes that door d is open.

The value of derive-condition in the *shareSecret* action, (at $agent$ l1), specifies the condition for mutual awareness (see discussion in Chapter 2, Section 2.3.1). In this case, agents are aware of the *shareSecret* action when they are at location l1, where this *grounded*[1] action is performed. For instance, if the specified depth of

[1]For the lifted action, the derive-condition would be (at $agent$?l), where l is the location where

nested belief in the problem file is 2, then if agents b and c are in l1 and agent a is sharing her secret, then agent b will believe the secret and will believe that agent c believes the secret. In addition, agent c will believe the secret and will believe that agent b believes the secret (and so on and so forth for all agents in l1).

In Line 3 of Algorithm 2 we use RP-MEP's machinery to classically encode the RP-MEP domain and initial state. Here is the partial classically encoded Corridor domain:

domain-corridor.pddl

```
(define (domain corridor)

    (:requirements :strips :conditional-effects
                   :disjunctive-preconditions)

    (:predicates
        (not_at_a_l1)
        (not_at_a_l2)
        (not_at_b_l1)
        (not_at_b_l2)
        (not_at_c_l1)
        (not_at_c_l2)
        (not_connected_l1_l1)
        (not_connected_l1_l2)
        (not_connected_l2_l1)
        (not_connected_l2_l2)
        (not_door_open_dl1l2)
        (not_dummy)
        (not_secret_a)
        (not_secret_b)
        (not_secret_c)
        (Ba_not_at_a_l1)
        (Ba_not_at_a_l2)
        (Ba_not_at_b_l1)
        (Ba_not_at_b_l2)
        (Ba_not_at_c_l1)
        (Ba_not_at_c_l2)
```

the secret is being shared.

```
(Ba_not_door_open_dl1l2)
(Ba_not_dummy)
(Ba_not_secret_a)
(Ba_not_secret_b)
(Ba_not_secret_c)
(Ba_at_a_l1)
.
.
.

(:action share_a_a_l1
:precondition (and (at_a_l1)
                   (Ba_secret_a)
              )
:effect (and
            ; #16607: origin
            (when (and (at_c_l1))
                  (Bc_secret_a))

            ; #29379: <==closure== 39863 (pos)
            (when (and (at_b_l1))
                  (Pb_secret_a))

            ; #30779: <==closure== 16607 (pos)
            (when (and (at_c_l1))
                  (Pc_secret_a))

            ; #32347: <==closure== 89586 (pos)
            (when (and (and (at_a_l2)
                            (door_open)))
                  (Pa_secret_a))

            ; #39863: origin
            (when (and (at_b_l1))
                  (Bb_secret_a))
```

```
; #49490: origin
(when (and (at_a_l1))
      (Ba_secret_a))

; #51412: <==closure== 73425 (pos)
(when (and (and (door_open)
           (at_b_l2)))
      (Pb_secret_a))

; #54730: <==closure== 49490 (pos)
(when (and (at_a_l1))
      (Pa_secret_a))

; #55625: origin
(when (and (and (door_open)
           (at_c_l2)))
      (Bc_secret_a))

; #73425: origin
(when (and (and (door_open)
           (at_b_l2)))

; #89586: origin
(when (and (and (at_a_l2)
           (door_open)))
      (Ba_secret_a))

; #93603: <==closure== 55625 (pos)
(when (and (and (door_open)
           (at_c_l2)))
      (Pc_secret_a))

; #10374: <==uncertain_firing== 49490 (pos)
(when (and (not (not_at_a_l1)))
      (not (Pa_not_secret_a)))
```

```
; #14825: <==unclosure== 33844 (neg)
(when (and (at_b_l1))
      (not (Bb_not_secret_a)))

.
.
.
```

Notice the fluent atoms created during the classical encoding process to represent every RML in the RP-MEP domain. Moreover, note the ancillary conditional effects that are automatically added during the encoding process to enforce closure under the KD45 axioms (see discussion of RP-MEP's classical encoding in Chapter 2, Section 2.3.1).

Recall that Algorithm 2 accepts a tuple $\langle Q, \mathcal{I}, j, \vec{v}, \pi, G \rangle$. Let us assume that in a particular problem instance of the Corridor domain, π is the single action shareSecret(a,a) and G is [b](secret a) $\wedge$![c](secret a). We included here the grounded shareSecret action that only allows the agent to share her secret in location l1. Therefore the action shareSecret(a,a) means that agent a is sharing agent a's secret in l1. As described in Chapter 5 (Section 5.3.2), agent a (Bob) has the epistemic goal that agent b believes his secret, with agent c **not** believing his secret. Thus, the regression formula $\phi = \textsc{valid}(\pi_c, G_c)$, computed from the classically encoded domain, is the following (converted to DNF by the Python library SymPy (Meurer et al., 2017)):

Regression formula for corridor problem

```
(Ba_secret_a & at_a_l1 & at_b_l1 & ~at_c_l1 & ~at_c_l2) |
(Ba_secret_a & at_a_l1 & at_b_l1 & ~at_c_l1 & ~door_open_dl1l2) |
(Ba_secret_a & at_a_l1 & at_b_l2 & door_open_dl1l2 & ~at_c_l1 &
~at_c_l2)
```

For instance, the first disjunct means that a necessary and sufficient condition for the achievement of G by π is for (1) agent a to believe a's secret, (2) agent a to be in l1, (3) agent b to be in l1 and (4) agent c to **not** be in either l1 or l2. If this disjunct is satisfied, then following the shareSecret action, agent b will come to learn a's secret and agent c will not, which satisfies agent a's epistemic goal [b](secret a) $\wedge$![c](secret a). Finally, the following is the negation of the regression formula above, converted to DNF by SymPy:

Negation of regression formula for corridor problem converted to DNF

```
at_c_l1 | ~Ba_secret_a | ~at_a_l1 | (at_c_l2 & door_open_dl1l2) |
(~at_b_l1 & ~at_b_l2) | (~at_b_l1 & ~door_open_dl1l2)
```

For instance, if agent c is at l1 then the goal cannot be satisfied by π since agent c will come to learn agent a's secret. Moreover, the fourth disjunct says that if agent c is at l2 and the door between l1 and l2 (dl1l2) is open, then the plan is not valid since agent c will come to learn agent a's secret. Both of these conditions make the single action plan shareSecret(a,a) not valid.

The following is the partial (classically encoded) initial state and the goal given to the classical planner:

prob-corridor.pddl

```
(define (problem corridor-prob)

    (:domain corridor)

    (:init
        (door_open_dl1l2)
        (Ba_not_door_open_dl1l2)

        (at_b_l1)
        (Ba_at_b_l1)

        (not_at_c_l1)
        (Ba_not_at_c_l1)

        (at_c_l2)
        (Ba_at_c_l2)

        .
        .
        .
        .
        .
        .
```

```
    (:goal
        (or
            (and (Ba_Ba_secret_a) (Ba_secret_a) (Ba_at_a_l1)
            (at_a_l1) (Ba_at_b_l1) (at_b_l1) (Ba_not_at_c_l1)
            (not_at_c_l1) (Ba_not_at_c_l2) (not_at_c_l2) )

            (and (Ba_Ba_secret_a) (Ba_secret_a) (Ba_at_a_l1)
            (at_a_l1) (Ba_at_b_l1) (at_b_l1) (Ba_not_at_c_l1)
            (not_at_c_l1) (Ba_not_door_open_dl1l2)
            (not_door_open_dl1l2) )

            (and (Ba_Ba_secret_a) (Ba_secret_a) (Ba_at_a_l1)
            (at_a_l1) (Ba_at_b_l2) (at_b_l2)
            (Ba_door_open_dl1l2) (door_open_dl1l2) (Ba_not_at_c_l1)
            (not_at_c_l1) (Ba_not_at_c_l2) (not_at_c_l2) )

            (and (Ba_at_c_l1) (at_c_l1) )

            (and (Ba_not_Ba_secret_a) (not_Ba_secret_a))

            (and (Ba_not_at_a_l1) (not_at_a_l1))

            (and (Ba_at_c_l2) (at_c_l2) (Ba_door_open_dl1l2)
            (door_open_dl1l2) )

            (and (Ba_not_at_b_l1) (not_at_b_l1) (Ba_not_at_b_l2)
            (not_at_b_l2) )

            (and (Ba_not_at_b_l1) (not_at_b_l1)
            (Ba_not_door_open_dl1l2) (not_door_open_dl1l2) )

        )
    )
)
```

We can see that while Alice believes that the door between l1 and l2 is open, she

also believes that agent a (Bob) believes that it is not open. This is a discrepancy! Moreover, notice the *disjunctive* goal given to the classical planner that comprises the disjuncts of DNF(ϕ) and DNF($\neg\phi$) (where ϕ is the regression formula $\textsc{valid}(\pi_c, G_c)$, as described in Section 5.2).

Given this disjunctive goal (encoded in prob-corridor.pddl), there are two possible optimal plans. Here is one of these optimal discrepancy resolving plans, produced by Fast Downward with an admissible heuristic:

Discrepancy Resolving Plan #1

```
1. closedoor_dl1l2
```

That is, Alice will close the door which will resolve the discrepancy by making Bob's (agent a's) plan valid.

This plan satisfies the disjunct

```
(and (Ba_Ba_secret_a) (Ba_secret_a) (Ba_at_a_l1) (at_a_l1)
(Ba_at_b_l1) (at_b_l1)
    (Ba_not_at_c_l1) (not_at_c_l1) (Ba_not_door_open_dl1l2)
    (not_door_open_dl1l2) )
```

since Alice believes that agent a already believes that the door is closed and, after closing the door, Alice will also believe that the door is closed. The other conjuncts in this disjunct are already satisfied in the initial state.

The other optimal discrepancy resolving plan that Fast Downward produced is the following:

Discrepancy Resolving Plan #2

```
1. informdooropen_a_dl1l2
```

That is, Alice will inform Bob (agent a) that the door is open which will resolve the discrepancy by making Bob aware that his plan is not valid. This plan satisfies the disjunct

```
(and (Ba_at_c_l2) (at_c_l2) (Ba_door_open_dl1l2) (door_open_dl1l2) )
```

Recall that we create three versions of each problem instance. When we remove a subset of communicative actions from the domain, the first discrepancy resolving plan will be computed, involving Alice closing the door. When we remove a subset of ontic actions (i.e., the closing/opening of doors), the second discrepancy resolving plan will be computed, involving Alice informing Bob that the door is open. If we do not modify the domain, since both plans are of equal length, we observed that Fast Downward chose the action that appeared 'earlier' in the domain file.

B.1.2 BW4T

domain-bw4t.pdkbddl

```
(define (domain bw4t)

  (:agents a b)
  (:requirements :strips :typing )

  (:types color room place id)

  (:predicates
                (at ?ag - agent ?place - place)
                (block ?colorid - color ?id - id ?placeid - room)
                (connected ?r1 - place ?r2 - place)
                (connected ?r1 - place ?r2 - room)
                (connected ?r1 - room ?r2 - place)
                (blockin ?colorid - color ?id - id ?dropzone - room)
                (in ?ag - agent ?roomid - room)
                (holding ?ag - agent ?colorid - color ?id - id
                ?placeid - room)
                (atblock ?colorid - color ?id - id ?placeid - room)
                (droplocation ?loc - room)
                (handempty)
    )
```

```
(:action goto
            :derive-condition   always
      :parameters (?ag - agent ?placeid - place ?currplace - room)
      :precondition (and (in ?ag ?currplace) (handempty)
                  (connected ?currplace ?placeid) )
      :effect
      (and (not(in ?ag ?currplace)) (at ?ag ?placeid)
       )
 )


 (:action gotodrop
            :derive-condition   always
        :parameters (?ag - agent ?placeid - place ?currplace - room)
        :precondition (and (in ?ag ?currplace) (not (handempty))
                           (connected ?currplace ?placeid) )
        :effect
        (and (not(in ?ag ?currplace)) (at ?ag ?placeid))
 )


(:action pckUp
      :derive-condition   (at $agent$ ?placeid)
      :parameters         (?ag - agent ?colorid - color ?id - id
                           ?placeid - room)
      :precondition       (and (in ?ag ?placeid)
                               (block ?colorid ?id ?placeid)
                               (atblock ?colorid ?id ?placeid)
                               (handempty)
             (not (droplocation ?placeid))
        )
      :effect
                          (and (holding ?ag ?colorid ?id ?placeid)
            (not (handempty)) (not (atblockbot))))
```

```
(:action putdown
     :derive-condition   (at $agent$ ?placeid)
     :parameters        (?ag - agent ?colorid - color ?id - id
                         ?placeid - room
                         ?dropzone - room)
     :precondition      (and (holding ?ag ?colorid ?id ?placeid)
                        (in ?ag ?dropzone)
                        (droplocation ?dropzone))


     :effect
           (and (not (holding ?ag ?colorid ?id ?placeid))
                [?ag](!block ?colorid ?id ?placeid)
                [?ag](block ?colorid ?id ?dropzone)
                (blockin ?colorid ?id ?dropzone)
                (handempty)
           )
  )




(:action informAboutBlockLocation
    :derive-condition   always
    :parameters         (?a1 - agent ?colorid - color ?id - id
                         ?placeid - room)
    :precondition       (and (atblock ?colorid ?id ?placeid))
    :effect             (and
                             [?a1](atblock ?colorid ?id ?placeid))

  )
```

```
 (:action informAboutBlockNOTLocation
      :derive-condition   always
      :parameters         (?a1 - agent ?colorid - color ?id - id
                           ?placeid - room)
      :precondition       (and (!atblock ?colorid ?id ?placeid))
      :effect             (and
                               [?a1](!atblock ?colorid ?id ?placeid))

    )



)
```

The last two actions can be used by Alice to inform any agent that a certain block is either in *placeid* or is not in *placeid*. The following (partial) problem file describes one of the *Dude, Where's my Medical Kit?* problem instances where agent *b* (Bob) holds a false belief about the location of the medical kit (blue block #1). As a reminder, since [Alice] implicitly precedes all fluents, we can see that Bob's beliefs are false from Alice's perspective since she believes that the block is in room b2 and she also believes that Bob believes that the block is in room c3.

problem-bw4t.pdkbddl

```
(define (problem prob4)
  (:domain bw4t)

 (:objects
                   blue red - color
                   1 2 - id
                   roomc1 roomc2 roomc3 roomb1 roomb2
                   roomb3 dropzone - room
                   frontdropzone righthalld lefthalld - place
                   frontroomc1 frontroomc2 frontroomc3 - place
```

```
                    lefthallc righthallc - place
                    frontroomb1 frontroomb2 frontroomb3 - place
                    lefthallb righthallb - place
 )

 (:depth 2)
 (:init
    .
    .
    .
        (!atblock blue 1 roomc3)
        (atblock blue 1 roomb2)
   [b](atblock blue 1 roomc3)
   [b](!atblock blue 1 roomb2)
)


 ))
```

To model the *Can You Hear Me??* scenario described in Chapter 5 (Section 5.3.2), we introduce the action *sendComm*:

domain-epistemic-goal-bw4t.pdkbddl

```
.
.
.
(:action sendComm
      :derive-condition   always
      :parameters         (?a1 - agent ?a2 - agent ?colorid - color
                           ?id - id  ?loc - loc)
      :precondition       ([?a1](blockin ?colorid ?id ?loc))
```

```
      :effect              (and
                                (when
                                     (workingCommDevice ?a2)
                                   [?a2](blockin ?colorid ?id ?loc)
                                )
                           )

)
.
.
.
```

As can be seen from the following partial initial state, Bob (agent b) initially believes that Mary's (agent c) communication device is working properly while Alice believes that it is not.

problem-epistemic-goal-bw4t.pdkbddl

```
.
.
.
(:init

        (!workingCommDevice c)
        [b](workingCommDevice c)
.
.
.
```

Recall that Bob's plan π is to deliver the block (medical kit) to the drop zone and then send a communication to Mary:

$$
\begin{aligned}
[&\mathit{move}(\text{Bob,HallWay,RoomB}),\\
&\mathit{pckUp}(\text{Bob,MedKit1,RoomB}),\\
&\mathit{move}(\text{Bob,RoomB,HallWay}),\\
&\mathit{dropOff}(\text{Bob,MedKit1,HallWay}),\\
&\mathit{sendComm}(\text{Bob,Mary,at(MedKit1,HallWay}))],
\end{aligned}
$$

To resolve the discrepancy about Bob's plan, Alice can inform Bob that Mary's communication device is not working. This is the discrepancy resolving plan computed by the classical planner given the disjunctive goal based on the computed regression formula:

Discrepancy Resolving Plan (BW4T epistemic goal)

```
1. informcommdevicenotworking_b_c
```

That is, agent b (Bob) is informed that agent c's (Mary's) communication device is not working. Then, Bob will believe that his plan is not valid.

B.1.3 DriverLog

In the DriverLog domain, Alice is assumed to be an embodied robot that can act in the environment. She can therefore pick up packages and move them to other locations in order to resolve discrepancies. Here is a partial encoding of the domain in PDKBDDL with the 'pick up package' action that can be executed by Alice the robot:

domain-driverlog.pdkbddl

```
(define (domain driverlog)

  (:agents a b c d)
  (:requirements :strips :typing )

  (:predicates

        (OBJ ?obj)
        (TRUCK ?truck)
        (LOCATION ?loc)
        (driver ?d)
        (at ?obj ?loc)
        (in ?obj1 ?obj)
        (driving ?d ?v)
        (link ?x ?y) (path ?x ?y)
        (empty ?v)
        (robotholdingpackage ?obj)
```

```
        (disc_resolution)
        (dummy)
        (atRobot ?loc)
)



    (:action ROBOT-PICK-UP-PACKAGE
:derive-condition   always
  :parameters
   (?obj
    ?loc)
  :precondition
   (and (OBJ ?obj) (atRobot ?loc) (LOCATION ?loc)
    (at ?obj ?loc))
  :effect
   (and (not (at ?obj ?loc)) (robotholdingpackage ?obj))

   )


.
.
.
)
```

In one of the problem instances, we have that Alice (the robot) believes that package1 is at location s1 while also believing that Bob (agent a) believes that the package is at location s0.

problem-driverlog.pdkbddl

```
.
.
.
(:init
        (at package1 s1)
        (!at package1 s0)
        [a](at package1 s0)
```

```
        [a](!at package1 s1)
.
.
.
```

Alice is initially at s1. And so, when the disjunctive goal is given to the classical planner, the planner comes up with the following plan:

Discrepancy Resolving Plan (DriverLog)

```
1. robot-pick-up-package_package1_s1
2. robot-move_s1_s0
3. robot-drop-off-package_package1_s0
```

That is, to resolve the discrepancy, Alice the robot picked up the package from s1 and delivered it to s0.

Higher-order discrepancies

Finally, let us see what discrepancy resolution looks like when $d = 5$. As mentioned in Section 5.3.2, for $d = 5$, Algorithm 2 resolves discrepancies between $B_{\text{Alice}}B_{\text{Mary}}B_{\text{Charlie}}B_{\text{Rose}}B_{\text{Bob}}\text{VALID}(\pi, G)$ and $B_{\text{Alice}}B_{\text{Charlie}}B_{\text{Rose}}B_{\text{Bob}}\text{VALID}(\pi, G)$.

For instance, in one of our problem instances we have that Alice believes that Mary (agent a) falsely believes that Charlie (agent b) believes that Rose (agent c) believes that Bob (agent d) believes that package1 is at location s0. At the same time, Alice believes that Charlie believes that Rose believes that Bob believes that package1 is at location s1. This is encoded as follows in PDKBDDL:

Partial initial state in Driverlog domain

```
[a][b][c][d](at package1 s0)
[a][b][c][d](!at package1 s1)
[b][c][d](!at package1 s0)
[b][c][d](at package1 s1)
```

In the domain file, we have the following inform action that can inform some agent ?a1 about the beliefs of agent ?a2 about the beliefs of agent ?a3 about the beliefs of agent ?a4 about the location of some object.

Inform action in Driverlog domain

```
(:action informAboutHigherOrderBeliefObjLoc
      :derive-condition   (always)
      :parameters         (?a1 - agent ?a2 - agent ?a3 - agent
                           ?a4 - agent ?obj  ?loc)
      :precondition       (and [?a2][?a3][?a4](at ?obj ?loc)
                               (OBJ ?obj) (LOCATION ?loc) )
      :effect             (and
                                [?a1][?a2][?a3][?a4](at ?obj ?loc))

   )
```

In the classically encoded problem, one of the disjuncts in the disjunctive goal is:

```
(and (Ba_Bb_Bc_Bd_at_package1_s1) (Bb_Bc_Bd_at_package1_s1))
```

And so, in order to satisfy the disjunctive goal the classical planner generates the following discrepancy resolving plan:

Discrepancy Resolving Plan for higher-order discrepancy (DriverLog)

```
1. informAboutHigherOrderBeliefObjLoc_a_b_c_d_package1_s1
```

That is, Alice informs Mary (agent a) that Charlie (agent b) believes that Rose (agent c) believes that Bob (agent d) believes that package1 is at location s1.

B.2 Creating RP-MEP Domains from Classical Planning IPC Domains

In this section we describe how we adapted the classical planning domain Driverlog (encoded in PDDL) to an RP-MEP domain encoded in PDKBDDL. The other classical planning domains (i.e., Depots, Gripper, Rovers, Logistics, Zeno, and Satellites) were similarly adapted. The original description[2] of the Driverlog domain is as follows: *"This domain has drivers that can walk between locations and trucks that can drive between locations. Walking requires traversal of different paths from those used*

[2]Obtained from http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume20/long03a-html/node39.html

for driving, and there is always one intermediate location on a footpath between two road junctions. The trucks can be loaded or unloaded with packages (with or without a driver present) and the objective is to transport packages between locations, ending up with a subset of the packages, the trucks and the drivers at specified destinations."
Let us examine the PDDL Driverlog domain, obtained from (Muise, 2016):

Classical planning DriverLog domain (encoded in PDDL)

```
(define (domain driverlog)
  (:requirements :strips)
  (:predicates         (OBJ ?obj)
                       (TRUCK ?truck)
                       (LOCATION ?loc)
               (driver ?d)
               (at ?obj ?loc)
               (in ?obj1 ?obj)
               (driving ?d ?v)
               (link ?x ?y) (path ?x ?y)
               (empty ?v)
)


(:action LOAD-TRUCK
  :parameters
   (?obj
    ?truck
    ?loc)
  :precondition
   (and (OBJ ?obj) (TRUCK ?truck) (LOCATION ?loc)
   (at ?truck ?loc) (at ?obj ?loc))
  :effect
   (and (not (at ?obj ?loc)) (in ?obj ?truck)))
```

```
(:action UNLOAD-TRUCK
  :parameters
   (?obj
    ?truck
    ?loc)
  :precondition
   (and (OBJ ?obj) (TRUCK ?truck) (LOCATION ?loc)
        (at ?truck ?loc) (in ?obj ?truck))
  :effect
   (and (not (in ?obj ?truck)) (at ?obj ?loc)))

(:action BOARD-TRUCK
  :parameters
   (?driver
    ?truck
    ?loc)
  :precondition
   (and (DRIVER ?driver) (TRUCK ?truck) (LOCATION ?loc)
   (at ?truck ?loc) (at ?driver ?loc) (empty ?truck))
  :effect
   (and (not (at ?driver ?loc)) (driving ?driver ?truck)
        (not (empty ?truck))))

(:action DISEMBARK-TRUCK
  :parameters
   (?driver
    ?truck
    ?loc)
  :precondition
   (and (DRIVER ?driver) (TRUCK ?truck) (LOCATION ?loc)
        (at ?truck ?loc) (driving ?driver ?truck))
  :effect
   (and (not (driving ?driver ?truck)) (at ?driver ?loc)
        (empty ?truck)))
```

```
(:action DRIVE-TRUCK
  :parameters
   (?truck
    ?loc-from
    ?loc-to
    ?driver)
  :precondition
   (and (TRUCK ?truck) (LOCATION ?loc-from) (LOCATION ?loc-to)
   (DRIVER ?driver) (at ?truck ?loc-from)
   (driving ?driver ?truck) (link ?loc-from ?loc-to))
  :effect
   (and (not (at ?truck ?loc-from)) (at ?truck ?loc-to)))

(:action WALK
  :parameters
   (?driver
    ?loc-from
    ?loc-to)
  :precondition
   (and (DRIVER ?driver) (LOCATION ?loc-from) (LOCATION ?loc-to)
        (at ?driver ?loc-from) (path ?loc-from ?loc-to))
  :effect
   (and (not (at ?driver ?loc-from)) (at ?driver ?loc-to)))


)
```

We took a straightforward approach to adapting the PDDL domain to a PDKBDDL domain. Most notably, we added multiple agents to the environment by configuring the following in the PDKBDDL domain file:

Configuring the set of agents

```
(:agents a b c)
```

The *:agents* field populates the set of agents Ag and in this example we have three agents: a, b, and c. The number of agents in Ag was configured depending on the values of d and $|Ag|$ in our evaluation. We moreover added special actions to

allow Alice (who is the root agent and a robot) to manipulate the environment and communicate with Bob (agent j). In Appendix B.1.3 we showed the encoding of the ROBOT-PICK-UP-PACKAGE action in PDKBDDL:

ROBOT-PICK-UP-PACKAGE action

```
 (:action ROBOT-PICK-UP-PACKAGE
:derive-condition   always
  :parameters
   (?obj
    ?loc)
  :precondition
   (and (OBJ ?obj) (atRobot ?loc) (LOCATION ?loc)
    (at ?obj ?loc))
  :effect
   (and (not (at ?obj ?loc)) (robotholdingpackage ?obj))


   )
```

In addition, Alice the robot can move between locations in the environment and communicate with other agents, even about the higher-order beliefs of other agents (as discussed in Appendix B.1.3):

Inform action in Driverlog domain

```
(:action informAboutHigherOrderBeliefObjLoc
      :derive-condition   (always)
      :parameters         (?a1 - agent ?a2 - agent ?a3 - agent
                           ?a4 - agent ?obj  ?loc)
      :precondition       (and [?a2][?a3][?a4](at ?obj ?loc)
                                (OBJ ?obj) (LOCATION ?loc) )
      :effect             (and
                               [?a1][?a2][?a3][?a4](at ?obj ?loc))


    )
```

Finally, the problem file was modified to encode both the root agents' beliefs about objects and locations in the environment, as well as its beliefs about other agents' beliefs about the environment:

problem-driverlog.pdkbddl

```
.
.
.
(:init
        (at package1 s1)
        (!at package1 s0)
        [a](at package1 s0)
        [a](!at package1 s1)
.
.
.
```

The main purpose of adapting the PDDL domains was to evaluate our discrepancy resolution approach with longer plans and more complex settings. However, as mentioned, our adaptation was simple and did not attempt to fully transform these single agent domains to fully fledged multi-agent domains, Instead, we modestly endowed the root agent with agency to resolve discrepancies and added agents (and the root agent's beliefs about those agents' beliefs) to 'stress test' RP-MEP, the epistemic planning system we used.

Appendix C

Proactive Robotic Assistance via Theory of Mind

To provide structure, we enumerate the different appendices below:

- **Appendix C.1:** we provide additional details on the implementation of our approach within a robotic system.
- **Appendix C.2:** we provide additional details on the video recordings used in the study discussed in Chapter 6.
- **Appendix C.3:** we provide a detailed account of the kitchen domain from Chapter 6 and its encoding in PDKBDDL.
- **Appendix C.4:** we present detailed results from our study.

C.1 Implementation Details

In this section we present additional details regarding the implementation of our approach within a humanoid robot, Pepper.

C.1.1 Line 3 – Perception Module

To interface RP-MEP, the epistemic planner to which we appeal, with Pepper's hardware, we first cross-compile ROS, Naoqi driver, and their dependencies so that they can be run directly on Pepper's onboard computer. The machine connected to Pepper

via LAN subscribes to compressed RGB streams which are processed by the perception module. To run our experiments, we use a 6GB GTX1650 Ti Nvidia GPU and a 9th gen Intel i7 processor.

As discussed in Section 6.2, we use MonoLoco (Bertoni et al., 2021) (a lightweight pre-trained neural network) to detect a person's 3D position and orientation from Pepper's RGB camera. MonoLoco is a deep learning based model that takes as input 2D keypoint detections from OpenPifPaf (Kreiss et al., 2021) and outputs 2D bounding-box pixel coordinates, 3D position and orientation estimates along with their uncertainties. The estimated uncertainties can further be used to perform tracking of a particular person. For every frame of RGB data received, the perception module runs MonoLoco to get detections of people. For every instance of people detected in the image, we compute AlignedReID features. These features are compared between all people in the current frame, and also with features extracted from previous frames in order to associate instances of the same people and differentiate between different people.

Event detection

As discussed in Section 6.2, we map the processed observations (obtained from the various perception algorithms) to agents from the set of agents Ag (e.g., Alice), fluent atoms in $\mathcal{P}$ (e.g., Soup1), and one of 7 possible events: pick up, put down, open (close) cabinet, enter (leave) room, and shift gaze. We then map these to an action in the set of actions $\mathcal{A}$ (e.g., $pickUp$(Alice, Soup1)).

We wrote a simple logic that returns one of these 7 events based on (1) a change in the proximity of a person to objects and locations in the room. Our logic is scenario-specific. For instance, in our 'charger' scenario (see Chapter 6 (Section 6.4.2)), our logic detects that a person is picking up the phone charger if that person is close enough to the charger; (2) whether a person is detected by Pepper's camera or not (and whether that person was previously (not) detected by Pepper's camera); and (3) a change in the orientation estimation of a person given by MonoLoco. That is, if a person's predicted orientation changes we then estimate a person's gaze direction and send a *shift gaze* action to RP-MEP. This allows Pepper to perform visual perspective taking – the ability to see the world from another person's perspective (Flavell, 1977). See Figure C.1 for a sample output of MonoLoco where the person is looking in three different directions. See Chapter 6 (Section 6.4.2) for details on how visual perspective taking is implemented in RP-MEP.

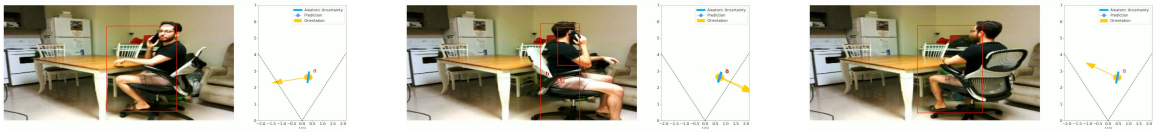

Figure C.1: Sample output from MonoLoco's orientation estimation. In each image, the person is gazing in a different direction and the orientation estimation changes accordingly.

C.1.2 Lines 4-10 – Theory of Mind Reasoning

Line 4 - Progression

In Line 4, the state S is progressed using RP-MEP's plan validation feature that accepts as input a plan, a goal, a domain, and a state, and determines whether the plan achieves the goal, while also returning the final state following the execution of the plan.

Line 6 - Plan recognition

In Line 6, Algorithm 1 (presented in Chapter 4) is run. In particular, to compute the Δ with RP-MEP in Line 4 of Algorithm 1, we provide the Fast Downward classical planner (Helmert, 2006) with the encoded PDDL files and run the planner, twice for each goal in $\mathcal{G}$, with a satisficing configuration to improve runtime. In Line 2 we check if G_H has been achieved by the human. if has been achieved then we nullify G_H and π_H.

Line 8 - Generating an assistive solution

In Line 8, RP-MEP is used to solve the MEP problem $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, B_\star G_H\rangle$ and generate an assistive solution for the human's presumed goal, G_H. Fast Downward is used with an admissible heuristic, to ensure optimal plans are computed.

Line 10 - Discrepancy resolution

In Line 10, the modified Algorith 2 (presented in Chapter 5) is run. In Line 3 of the modified algorithm, RP-MEP is given PDKBDDL files and outputs classically encoded PDDL files, and in Line 6 Fast Downward is given the encoded PDDL files and called with an admissible heuristic that supports conditional effects and disjunctive goals, to ensure optimal plans are computed. As in Chapter 5, we also make use

of the SymPy Python library (Meurer et al., 2017) to convert regression formulae to DNF and to compute their negation.

C.1.3 Line 11 – Plan Execution by Pepper

In this work, Pepper is able to: (1) navigate to different places in the scene by leveraging the map built using RTAB-Map, Pepper's depth sensors, and ROS's navigation stack; (2) interact with objects – since robotic object manipulation is a research problem in and of itself, for the purposes of the video recordings in our study we hardcoded Pepper's limb movement so that it appears as though Pepper is successfully manipulating objects (e.g., closing a door or picking up a phone charger); (3) communicate (e.g., Pepper informing Alice of the bowl's location in our example). To realize this, the content of an action is processed using a number of simple natural language templates. For instance, the contents of the inform action from our example, $\neg in(\text{Bowl1},\text{Cabinet1}) \wedge in(\text{Bowl1},\text{Cabinet2})$, would be translated to "If you are looking for Bowl1, it is in Cabinet2, not Cabinet1". Then, the text is given to Pepper's text-to-speech module and/or displayed on Pepper's tablet.

Executing world-altering actions

As discussed in Section 6.2, since Pepper lacks object manipulation capabilities, for the purposes of the video recordings in our study (discussed in the next section) we hardcoded Pepper's limb movement so that it appears as though Pepper is successfully manipulating objects (e.g., closing a door or picking up a phone charger).

For instance, in our 'charger' scenario (see Chapter 6 (Section 6.4.2)) Pepper – as part of the execution of the automatically generated discrepancy resolving plan – is meant to pick up a phone charger that is connected to a power outlet. To record the video, we hardcoded Pepper's movement so that it leans closer to the power outlet and lifts its arm. Then, the recording was paused and resumed after we placed the charger in Pepper's hand. Moreover, in the 'corridor' scenario Pepper is meant to close a door in order to resolve the human's discrepancy. Similarly to the charger scenario, we hardcoded Pepper's movement so that it leans close to the door and lifts its arm. Then, we closed the door ourselves and had Pepper return to its neutral body position.

Executing communication actions

As discussed in Section 6.2, Pepper can communicate with other agents (e.g., Pepper informing Alice of the bowl's location in our example). If a discrepancy resolving plan generated in Line 10 of Algorithm 3 contains communication actions then their content is processed using a number of simple natural language templates. For instance, recall the inform action from the discrepancy resolving plan π' in our kitchen example:

$$[inform(\text{Pepper, Alice, } \neg in(\text{Bowl1,Cabinet1}) \wedge in(\text{Bowl1,Cabinet2}))].$$

This action is processed and converted to the English sentence: "If you're looking for Bowl1, it's in Cabinet2, not in Cabinet1". In our study we produced video recordings of Pepper implementing Algorithm 3. However, due to incredibly poor acoustics in the filming location, we recorded our own voice (and distorted it) and used it instead of the recording of Pepper's text-to-speech module.

C.2 Video Recordings Used in Study - Additional Details

In our study, we recorded videos involving a Pepper robot and a human(s) in a number of realistic scenarios: (1) a slight variation[1] of the kitchen scenario described in Section 6.1, (2) the 'phone charger' scenario, and (3) a simplified version of the corridor scenario (described in Section 5.3.2).

In the phone charger scenario, as discussed in Section 6.4.2, to resolve the discrepancies pertaining to the validity of Alice's plan and the assistive solution, Pepper can do one of two things:

> (1) Pepper goes to the bedroom, picks up the spare phone charger, and brings it back to the living room to replace the charger that was taken by Bob.

or

> (2) Pepper informs Alice that the charger is no longer in the living room and that there is another charger in the bedroom.

[1] Due to the limitations of the perception system, we used a large bag of chips instead of a can of soup and updated the PDKBDDL encoding, as well as the set of possible goals $\mathcal{G}$ accordingly.

In our study, participants were shown the former discrepancy resolving plan executed by Pepper. The video can be found in https://youtu.be/ICSKmchZIW8.

Moreover, in the simplified corridor scenario, to resolve the discrepancy, Pepper can either

> (1) Send a message to Bob, informing him that Eve is standing just outside the door,

or

> (2) close the door.

In our study, participants were shown the latter discrepancy resolving plan executed by Pepper. The video can be found in https://youtu.be/VLxceOkYM-A. In the video, Tara (playing the role of Bob) whispers to Ruthrash (playing the role of Alice) that she has to tell him about Maayan's (Eve's) birthday party but Maayan can't find out. In this work we do not address the interesting natural language understanding task of inferring Tara's epistemic goal from her utterance. Instead, Tara's goal is already known (and reflected in G_H) and given to Algorithm 3 – $B_{\text{Ruthrash}}(\text{Maayan\_birthday\_party\_secret}) \wedge \neg B_{\text{Maayan}}(\text{Maayan\_birthday\_party\_secret})$.

C.3 Encoding the Kitchen Example from Chapter 6 in PDKBDDL

Here we detail how our kitchen example described in Section 6.1.1 (and used in our evaluation in Section 6.4) was encoded using the PDKBDDL format used by the epistemic planner RP-MEP. We moreover offer a more concrete view of the workings of Algorithm 1. For the sake of readability, the domain seen here is partial and contains a subset of the actions found in the full domain. Moreover, to better illustrate the PDKBDDL format, some actions are grounded versions of the actions used in our experiments.

As defined in Chapter 2 (Section 2.3), an RP-MEP problem comprises a domain, an initial state, and a goal condition. Correspondingly, the epistemic planner RP-MEP (as is typically the case in automated planning) accepts two components – a domain file and a problem file, both encoded in PDKBDDL. The former contains definitions of the agents, actions, predicates, and types of objects in the environment. The latter specifies the initial state of the world and the goal condition.

domain-kitchen.pdkbddl

```
(define (domain kitchen)

    ; This specifies the agents in the set of agents Ag
    (:agents alice bob)

    (:types room thing cabinet)

    (:predicates
       (at ?ag - agent ?r - room)
       (in ?obj - thing ?cab - cabinet)
       (holding ?ag - agent ?obj - thing)
       (atRobot ?r - room)
    )

    (:action takeObjOutOfCabinet
      :derive-condition   (at $agent$ kitchen)
      :parameters         (?ag - agent ?obj - thing ?cab - cabinet)
      :precondition       (and (in ?obj ?cab) (open ?cab))
      :effect             (and

                               (not (in ?obj ?cab))

                               (holding ?ag ?obj)

                           )
  )

  (:action informObjLocation
      :derive-condition   always
      :parameters         (?a1 - agent ?obj - thing ?cab - cabinet)
      :precondition       (and (in ?obj ?cab))
      :effect             (and
                               [?a1](in ?obj ?cab))

    )
```

```
    (:action informObjNotInLocation
      :derive-condition   always
      :parameters         (?a1 - agent ?obj - thing ?cab - cabinet)
      :precondition       (and (not (in ?obj ?cab)))
      :effect             (and

                               [?a1](!in ?obj ?cab))

    )


)
```

Since planning in RP-MEP is from the perspective of a single agent (called the root agent), all fluent atoms are implicitly preceded by the beliefs of that agent. In our kitchen example, Pepper resolves all discrepancies and so planning is done from its perspective. For instance, in the *informObjLocation* action, the effect [?a1](in ?obj ?cab) is implicitly preceded by [Pepper] such that the effect of the action is that Pepper believes that agent a1 believes that the object *obj* is in cabinet *cab*.

The other component of an RP-MEP problem specified in PDKBDDL and given to RP-MEP is the problem file. The problem file specifies the different objects, the initial state of the world, and the goal.

problem-kitchen.pdkbddl

```
{include:domain-kitchen.pdkbddl}

(define (problem prob1)
  (:domain kitchen)
    (:objects kitchen - room
        soup1 bowl1 - thing
        cabinet1 cabinet2 cabinet3 - cabinet)

    (:projection )
    (:depth 1)
    (:task valid_generation)
    (:init-type complete)
```

```
  (:init
  (!at alice kitchen)
  (at bob kitchen)
  (in soup1 cabinet3)
  (in bowl1 cabinet1)
  [alice](in bowl1 cabinet1)
  [bob](in bowl1 cabinet1)
  [alice](in soup1 cabinet3)
  [bob](in soup1 cabinet3)
  .
  .
  .
  .
)
```

Recall that in our example, Pepper (via the perception module) observed Alice place a bowl (Bowl1) in a certain cabinet (Cabinet1) and leave the room. The problem file shown above specifies the state of the world after Alice's actions. We can see that the state of the world (from Pepper's perspective) is such that everyone believes that the bowl is in Cabinet1 and the can of soup (Soup1) is in Cabinet3. Moreover, Pepper believes that Alice is not in the kitchen and that Bob is in the kitchen.

After Bob is observed opening Cabinet1, taking Bowl1, placing Bowl1 in Cabinet2, and leaving the kitchen, we call RP-MEP's plan validation feature that accepts as input a plan, a goal, a domain, and a state, and determines whether the plan achieves the goal, while also returning the final state following the execution of the plan. The validation feature implements RP-MEP's progression operator, defined in Chapter 2, Section 2.3.1. The plan given to the validation feature comprises Bob's observed actions. The resulting state returned by the validation feature is the following:

problem-kitchen.pdkbddl

```
.
.
.
 (:init
 (!at alice kitchen)
 (!at bob kitchen)
```

```
  (in soup1 cabinet3)
  (in bowl1 cabinet2)
  [alice](in bowl1 cabinet1)
  [bob](in bowl1 cabinet2)
  [alice](in soup1 cabinet3)
  [bob](in soup1 cabinet3)
  .
  .
  .
  .
)
```

As discussed in Section 6.1.1, after observing Bob's actions, Pepper believes that Alice holds a false belief pertaining to the bowl's location – whereas Pepper believes that Alice believes it is still in Cabinet1, Pepper believes that it is in Cabinet2. Moreover, as discussed in Section 6.1, Pepper's reasoning is done automatically by RP-MEP using the conditioned mutual awareness mechanism (Muise et al., 2021). See discussion in Chapter 2 (Section 2.3.1).

Going back to domain-kitchen.pdkbddl, note the value of *derive-condition* in the *takeObjOutOfCabinet* action – (at $agent$ kitchen). (at $agent$ kitchen) specifies the condition for *mutual awareness*. In this case, agents believe the effects of the *takeObjOutOfCabinet* action if they are in the kitchen, where this *grounded*[2] action is performed. As discussed in Section 6.1 (and Chapter 2, Section 2.3), conditioned mutual awareness can handle both first- as well as higher-order Theory of Mind reasoning. This depends on the depth of nested belief that RP-MEP is configured to reason with (the depth of nested belief is specified in the PDKBDDL problem file (discussed below)). In the kitchen example, if the specified depth of nested belief in the problem file is 1, then if Alice and Bob are in the kitchen and Bob picks up Bowl1 from Cabinet1, then Alice will believe the bowl is no longer in Cabinet1. However, in our example Alice is not in the kitchen and therefore will still believe that the bowl is in Cabinet1 even after Bob takes the bowl out of Cabinet1. Since all reasoning is done from Pepper's perspective and since Pepper believes that Alice was not in the room when Bob moved the bowl, Pepper will believe after observing Bob's action that Alice's beliefs about the bowl's location did not change.

[2]For the lifted action, the derive-condition would be (at $agent$?r), where r is the room where the action is performed.

In Line 6 our algorithm performs plan recognition. As discussed in Section 6.1, the partial set of possible goals the human H may be pursuing is $\mathcal{G} = \{made\_soup, made\_coffee, \dots\}$. Our domain and set of possible goals are inspired by the popular Kitchen domain (Wu et al., 2007), often used as a goal recognition benchmark (Ramírez & Geffner, 2010). $\mathcal{G}$ in our case includes the 'soup making' goal (which is achieved if the agent picks up a bowl and a can of soup) and a 'coffee making' goal (which is achieved if the agent picks up coffee, creamer, sugar, and a mug). After observing Alice returning to the kitchen and taking the can of soup from Cabinet3, the plan recognition algorithm (Algorithm 1) returns the plan π_H:

Alice's presumed plan

```
1. openCabinet_alice_cabinet1
2. takeObjOutOfCabinet_alice_bowl1_cabinet1
```

The plan recognition algorithm also returns the human's presumed goal $G_H = made\_soup$. This is because the sequence of observations O includes *enterRoom*(Alice), *open*(Alice, Cabinet3), and *pickUp*(Alice, Soup1), and Algorithm 1 forces the planner to generate plans that satisfy O. That is, achieving *made_soup* while satisfying the sequence of observations is 'cheaper' than achieving *made_coffee* while satisfying the observations, since a plan that satisfies O and achieves *made_coffee* would include the action of picking up the can of soup that is redundant since it does not contribute towards the optimal achievement of *made_coffee*). In addition, the cost of achieving *made_coffee* while *not* satisfying the observations in O will actually be lower than the cost of achieving *made_coffee* while satisfying O, because the planner is not forced to include the redundant 'soup picking' action. On the other hand, the cost of achieving *made_soup* while *not* satisfying the observations in O is infinite since the planner must include the 'soup picking' action in the plan to achieve the goal, but is forced to not satisfy O. Since there is only one can of soup, the goal cannot be achieved under these constraints. Finally, recall the discussion in Section 4.3.2 where we defined Δ as the difference between the costs of these two plans (i.e., one that is forced to satisfy O and one that is forced to not satisfy O) and explained how Δ is used to compute the posterior probability of a goal, given O. Because of the reasoning given above, the Δ value for *made_soup* will be lower than the Δ value for *made_coffee* and *made_soup* will thus be assigned a higher posterior probability.

As discussed in Section 6.1, π_H conforms with the sequence of observations and achieves the goal from the perspective of the observed agent, Alice in our case. Since Alice holds a false belief about the location of Bowl1, her plan to make soup involves

obtaining the bowl from Cabinet1 (rather than Cabinet2). Her plan will of course fail to achieve her goal because of her false belief.

To obtain an assistive solution in Line 8, RP-MEP is tasked with solving the MEP problem $\langle\langle \mathcal{P}, \mathcal{A}, Ag\rangle, S, B_\star G_H\rangle$ by generating a plan π_{assist} that comprises only actions performed by the human H. Importantly, Pepper believes that π_{assist} achieves the human's goal *made_soup*. The goal in the PDKBDDL problem file is

```
made_soup
```

Recall that *made_soup* is implicitly preceded by [Pepper]. The assistive solution π_{assist} generated by RP-MEP is then

Assisstive solution for Alice's presumed goal

```
1. openCabinet_alice_cabinet2
2. takeObjOutOfCabinet_alice_bowl1_cabinet2
```

Since Pepper believes that Bowl1 is in Cabinet2, π_{assist} involves Alice taking the bowl from Cabinet2, rather than Cabinet1.

Finally, in Line 10 the set the plans, Π, is populated with π_H and π_{assist} and given (along with G_H) to the modified Algorithm 2. To generate the communicative discrepancy resolving plan π' shown in Section 6.1.1, we allow Pepper to only communicate with other agents, without altering the environment. The modified Algorithm 2, using RP-MEP, therefore generates the following discrepancy resolving plan:

Discrepancy resolving plan

```
1. informObjNotInLocation_alice_bowl1_cabinet1
2. informObjInLocation_alice_bowl1_cabinet2
```

That is, the discrepancy resolving plan involves Pepper informing Alice about the location of Bowl1 which resolves both discrepancies perceived by Pepper between its beliefs and its beliefs about Alice's beliefs – $\textsc{valid}(\pi_H, G_H)$ and $\textsc{valid}(\pi_{assist}, G_H)$.

C.4 User Study - Results

In this section we present detailed results from the study discussed in Section 6.3. We separate the section into a single subsection for internal consistency results and three

additional subsections, one for each scenario used in our study – charger, kitchen, and corridor. For more details on these domains, see Chapter 6 (Section 6.4.2). For each scenario, we report the results for the different PSI scales used in the study – RC, PC, and AC.

C.4.1 Internal Consistency

The PSI scales we used in the study were tested for reliability. All scales consisted of four questions (Barchard et al., 2020). All scales had internal consistency, as determined by Cronbach's alpha: AC ($\alpha = 0.744$), RC ($\alpha = 0.919$), and PC ($\alpha = 0.946$).

C.4.2 Charger scenario

Recognizes Human Cognitions (RC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to recognize the cognition of humans was different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 51.45) being higher than the **ToM-Def** condition (mean rank = 30.09), $U = 372.500$, $p < .001$.

Predicts Human Cognitions (PC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to predict the cognition of humans was different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 49.81) being higher than the **ToM-Def** condition (mean rank = 31.65), $U = 436.500$, $p < .001$.

Adapts to Human Cognitions (AC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to adapt to the cognition of humans was different depending on condition. The differences between conditions were statistically

significant with the **ToM** condition (mean rank = 54.00) being higher than the **ToM-Def** condition (mean rank = 27.66), $U = 273.000$, $p < .001$.

C.4.3 Kitchen scenario

Recognizes Human Cognitions (RC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to recognize the cognition of humans was different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 53.13) being higher than the **ToM-Def** condition (mean rank = 28.49), $U = 307.000$, $p < .001$.

Predicts Human Cognitions (PC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to predict the cognition of humans was different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 50.33) being higher than the **ToM-Def** condition (mean rank = 31.15), $U = 416.000$, $p < .001$.

Adapts to Human Cognitions (AC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to adapt to the cognition of humans was different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 53.64) being higher than the **ToM-Def** condition (mean rank = 28.00), $U = 287.000$, $p < .001$.

C.4.4 Corridor scenario

Recognizes Human Cognitions (RC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to recognize the cognition of humans was

different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 52.23) being higher than the **ToM-Def** condition (mean rank = 29.34), $U = 342.000$, $p < .001$.

Predicts Human Cognitions (PC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to predict the cognition of humans was different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 51.23) being higher than the **ToM-Def** condition (mean rank = 30.29), $U = 381.000$, $p < .001$.

Adapts to Human Cognitions (AC)

Shapiro-Wilk tests showed a significance departure from the normality ($p < .05$) for both conditions. Therefore, we conducted a Mann-Whitney U test to determine if the perception of a robot being able to adapt to the cognition of humans was different depending on condition. The differences between conditions were statistically significant with the **ToM** condition (mean rank = 52.63) being higher than the **ToM-Def** condition (mean rank = 28.96), $U = 326.500$, $p < .001$.

Bibliography

Akula, A. R., Wang, K., Liu, C., Saba-Sadiya, S., Lu, H., Todorovic, S., Chai, J., and Zhu, S.-C. Cx-tom: Counterfactual explanations with Theory-of-Mind for enhancing human trust in image recognition models. *Iscience*, 25(1):103581, 2022. 47, 48

Alanqary, A., Lin, G. Z., Le, J., Zhi-Xuan, T., Mansinghka, V. K., and Tenenbaum, J. B. Modeling the mistakes of boundedly rational agents within a bayesian Theory of Mind. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43, 2021. 86

Alchourrón, C. E., Gärdenfors, P., and Makinson, D. On the logic of theory change: Partial meet contraction and revision functions. *The journal of symbolic logic*, 50 (2):510–530, 1985. 3, 32, 171

Alizadeh Alamdari, P., Klassen, T. Q., Toro Icarte, R., and McIlraith, S. A. Be considerate: Avoiding negative side effects in reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 18–26, 2022. 168

Allen, J. F. and Perrault, C. R. Analyzing intention in utterances. *Artificial intelligence*, 15(3):143–178, 1980. 4

Alshehri, A., Miller, T., and Sonenberg, L. Modeling communication of collaborative multiagent system under epistemic planning. *International Journal of Intelligent Systems*, 2021. 124

Amato, C. and Baisero, A. Active goal recognition. *arXiv preprint arXiv:1909.11173*, 2019. 64

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016. 168

Andersen, M. B. *Towards Theory-of-Mind agents using automated planning and dynamic epistemic logic*. PhD thesis, Technical University of Denmark, 2015. 2

Antaki, C. and Leudar, I. Explaining in conversation: Towards an argument model. *European Journal of Social Psychology*, 22(2):181–194, 1992. 26

Appelt, D. E. and Pollack, M. E. Weighted abduction for plan ascription. *User modeling and user-adapted interaction*, 2(1):1–25, 1992. 52

Arnold, T., Briggs, G., and Scheutz, M. Only those who can obey can disobey: The intentional implications of artificial agent disobedience. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (Best and Visionary Papers in the AAMAS 2022 Workshops)*, pp. 130–143, 2022. 167

Aru, J., Labash, A., Corcoll, O., and Vicente, R. Mind the gap: Challenges of deep learning approaches to Theory of Mind. *arXiv preprint arXiv:2203.16540*, 2022. 2, 4

Bacchus, F. and Kabanza, F. Planning for temporally extended goals. *Annals of Mathematics and Artificial Intelligence*, 22(1-2):5–27, 1998. 93

Bacchus, F. and Petrick, R. Modeling an agent's incomplete knowledge during planning and execution. In *Proceedings of the 6th International Conference on Knowledge Representation and Reasoning (KR)*, pp. 432–443, 1998. 4, 16, 17, 158

Baier, J. A. and McIlraith, S. A. Planning with first-order temporally extended goals using heuristic search. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pp. 788–795, 2006. 93

Baker, C., Saxe, R., and Tenenbaum, J. Bayesian Theory of Mind: Modeling joint belief-desire attribution. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, 2011. 2, 86

Baker, C. L. and Tenenbaum, J. B. Modeling human plan recognition using bayesian Theory of Mind. *Plan, activity, and intent recognition: Theory and practice*, 7: 177–204, 2014. 86

Baraglia, J., Cakmak, M., Nagai, Y., Rao, R. P., and Asada, M. Efficient human-robot collaboration: when should a robot take initiative? *The International Journal of Robotics Research*, 36(5-7):563–579, 2017. 160

Baral, C., Gelfond, G., Pontelli, E., and Son, T. C. An action language for reasoning about beliefs in multi-agent domains. In *Proceedings of the 14th International Workshop on Non-Monotonic Reasoning (NMR-12)*, volume 4, 2012. 83

Baral, C., Bolander, T., van Ditmarsch, H., and McIlraith, S. Epistemic planning (Dagstuhl seminar 17231). In *Dagstuhl Reports*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. 4, 16

Barchard, K. A., Lapping-Carr, L., Westfall, R. S., Fink-Armold, A., Banisetty, S. B., and Feil-Seifer, D. Measuring the perceived social intelligence of robots. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(4):1–29, 2020. 143, 146, 147, 210

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., et al. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020. 37

Barnett, M. A. and Thompson, S. The role of perspective taking and empathy in children's Machiavellianism, prosocial behavior, and motive for helping. *The journal of genetic psychology*, 146(3):295–305, 1985. 167

Baron-Cohen, S. Precursors to a Theory of Mind: Understanding attention in others. *Natural theories of mind: Evolution, development and simulation of everyday mindreading*, 1:233–251, 1991. 1, 126, 162

Baron-Cohen, S. *Mindblindness: An essay on autism and Theory of Mind*. MIT press, 1997. 1, 126, 162

Baron-Cohen, S. and Cross, P. Reading the eyes: evidence for the role of perception in the development of a Theory of Mind. *Mind & Language*, 7(1-2):172–186, 1992. 126

Baron-Cohen, S., Leslie, A. M., and Frith, U. Does the autistic child have a "Theory of Mind"? *Cognition*, 21(1):37–46, 1985. 1, 126, 135, 162

Bartneck, C., Kulić, D., Croft, E., and Zoghbi, S. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International Journal of Social Robotics*, 1(1):71–81, 2009. 143

Beer, J. M. and Takayama, L. Mobile remote presence systems for older adults: acceptance, benefits, and concerns. In *Proceedings of the 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 19–26, 2011. 126

Belardinelli, G. and Rendsvig, R. K. Epistemic planning with attention as a bounded resource. In *International Workshop on Logic, Rationality and Interaction*, pp. 14–30. Springer, 2021. 17

Belle, V., Bolander, T., Herzig, A., and Nebel, B. Epistemic planning: Perspectives on the special issue. *Artificial Intelligence*, 2022. 4, 17

Benninghoff, B., Kulms, P., Hoffmann, L., and Krämer, N. C. Theory of Mind in human-robot-communication: Appreciated or not? *Kognitive Systeme*, 2013(1), 2013. 142

Berlin, M., Gray, J., Thomaz, A. L., and Breazeal, C. Perspective taking: An organizing principle for learning in human-robot interaction. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, volume 2, pp. 1444–1450, 2006. 2, 157

Bertoni, L., Kreiss, S., and Alahi, A. Perceiving humans: from monocular 3D localization to social distancing. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 139, 199

Bianco, F. and Ognibene, D. Functional advantages of an adaptive theory of mind for robotics: a review of current architectures. *2019 11th Computer Science and Electronic Engineering (CEEC)*, pp. 139–143, 2019. 157

Bianco, F. and Ognibene, D. Robot learning theory of mind through self-observation: Exploiting the intentions-beliefs synergy. *arXiv preprint arXiv:2210.09435*, 2022. 157

Bisson, F., Kabanza, F., Benaskeur, A. R., and Irandoust, H. Provoking opponents to facilitate the recognition of their intentions. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, 2011. 64

Bolander, T. A gentle introduction to epistemic planning: The DEL approach. *arXiv preprint arXiv:1703.02192*, 2017. 17

Bolander, T. and Andersen, M. B. Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011. 4, 16, 17, 18

Bolander, T., Engesser, T., Mattmüller, R., and Nebel, B. Better eager than lazy? how agent types impact the successfulness of implicit coordination. In *Proceedings*

of the 16th International Conference on Knowledge Representation and Reasoning (KR), 2018. 4, 16, 159

Bolander, T., Charrier, T., Pinchinat, S., and Schwarzentruber, F. DEL-based epistemic planning: Decidability and complexity. *Artificial Intelligence*, 287, 2020. 18

Bolander, T., Dissing, L., and Herrmann, N. DEL-based epistemic planning for human-robot collaboration: Theory and implementation. In *Proceedings of the 18th International Conference on Knowledge Representation and Reasoning (KR)*, pp. 120–129, 2021. 2, 127, 158, 159, 160

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 168

Boutilier, C. and Becher, V. Abduction as belief revision. *Artificial intelligence*, 77 (1):43–94, 1995. 31, 34, 36, 38, 40, 42, 48, 65

Brachman, R. J. and Levesque, H. J. *Knowledge Representation and Reasoning*. Elsevier, 2004. 30

Bramblett, L. and Bezzo, N. Epistemic planning for multi-robot systems in communication restricted environments. *Frontiers in Robotics and AI*, 10:67, 2023. 158

Bramblett, L., Gao, S., and Bezzo, N. Epistemic prediction and planning with implicit coordination for multi-robot teams in communication restricted environments. *arXiv preprint arXiv:2302.10393*, 2023. 158

Bratman, M. Intention, plans, and practical reason. 1987. 2, 46, 85

Breazeal, C., Gray, J., and Berlin, M. An embodied cognition approach to mindreading skills for socially intelligent robots. *The International Journal of Robotics Research*, 28(5):656–680, 2009. 2, 157

Breazeal, C., Gray, J., and Berin, M. Mindreading as a foundational skill for socially intelligent robots. In *Robotics Research*, pp. 383–394. Springer, 2010. 2, 157

Briggs, G. and Scheutz, M. The case for robot disobedience. *Scientific American*, 316(1):44–47, 2017. 167

Briggs, G., Williams, T., Jackson, R. B., and Scheutz, M. Why and how robots should say 'no'. *International Journal of Social Robotics*, 14(2):323–339, 2022. 167

Broekens, J., Harbers, M., Hindriks, K., Van Den Bosch, K., Jonker, C., and Meyer, J.-J. Do you get it? user-evaluated explainable BDI agents. In *German Conference on Multiagent System Technologies*, pp. 28–39. Springer, 2010. 124

Brooks, C. and Szafir, D. Building second-order mental models for human-robot interaction. *arXiv preprint arXiv:1909.06508*, 2019. 2, 126, 157

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Proceedings of the 33rd Conference on Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1877–1901, 2020. 4, 168

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023. 4, 169

Buckingham, D. *Multiagent Epistemic Cooperation-Agnostic Planning*. PhD thesis, Tufts University, 2023. 17

Buckingham, D., Chita-Tegmark, M., and Scheutz, M. Robot planning with mental models of co-present humans. In *International Conference on Social Robotics*, pp. 566–577. Springer, 2020a. 2, 157

Buckingham, D., Kasenberg, D., and Scheutz, M. Simultaneous representation of knowledge and belief for epistemic planning with belief revision. In *Proceedings of the 17th International Conference on Knowledge Representation and Reasoning (KR)*, pp. 172–181, 2020b. 4, 16

Bühler, M. *Theory of Mind and information relevance in human centric human robot cooperation*. PhD thesis, Technische Universität Darmstadt, 2022. 2, 157

Bühler, M. C. and Weisswange, T. H. Online inference of human belief for cooperative robots. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 409–415, 2018. 2, 126, 157

Bühler, M. C. and Weisswange, T. H. Theory of Mind based communication for human agent cooperation. In *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, pp. 1–6. IEEE, 2020. 2, 157

Bühler, M. C., Adamy, J., and Weisswange, T. H. Theory of Mind based assistive communication in complex human robot cooperation. *arXiv preprint arXiv:2109.01355*, 2021. 2, 157

Burigana, A. and Fabiano, F. The epistemic planning domain definition language (short paper). In *Proceedings of the 10th Italian workshop on Planning and Scheduling (IPS 2022), RCRA Incontri E Confronti (RiCeRcA 2022), and the workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (SPIRIT 2022) co-located with 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022)*, 2022. 17

Burigana, A., Felli, P., and Montali, M. DELPHIC: Towards an efficient possibility-based epistemic planning framework. In *4th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis*, pp. 33–37, 2022. 17

Buyukgoz, S., Grosinger, J., Chetouani, M., and Saffiotti, A. Two ways to make your robot proactive: reasoning about human intentions, or reasoning about possible futures. *arXiv preprint arXiv:2205.05492*, 2022. 160

Camerer, C. F., Ho, T.-H., and Chong, J.-K. A cognitive hierarchy model of games. *The Quarterly Journal of Economics*, 119(3):861–898, 2004. 3, 41

Carberry, S. Techniques for plan recognition. *User Modeling User-Adapted Interaction*, 11(1-2):31–48, 2001. 52

Carreno, Y., Lindsay, A., and Petrick, R. Explaining temporal plans with incomplete knowledge and sensing information. In *ICAPS 2021 Workshop on Explainable AI Planning (XAIP)*, 2021. 49, 121

Carroll, M., Chan, A., Ashton, H., and Krueger, D. Characterizing manipulation from AI systems. *arXiv preprint arXiv:2303.09387*, 2023. 167

Cawsey, A. Generating interactive explanations. In *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI)*, pp. 86–91, 1991. 46

Chajewska, U. and Halpern, J. Y. Defining explanation in probabilistic systems. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 62–71, 1997. 31, 46

Chakraborti, T. and Kambhampati, S. (When) Can AI Bots Lie? In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2019. 167

Chakraborti, T., Briggs, G., Talamadupula, K., Zhang, Y., Scheutz, M., Smith, D., and Kambhampati, S. Planning for serendipity. In *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5300–5306. IEEE, 2015. 161

Chakraborti, T., Sreedharan, S., Zhang, Y., and Kambhampati, S. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 156–163, 2017. 46, 49, 50, 122

Chakraborti, T., Kulkarni, A., Sreedharan, S., Smith, D. E., and Kambhampati, S. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In *Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 86–96, 2019. 84

Chakraborti, T., Sreedharan, S., and Kambhampati, S. The emerging landscape of explainable automated planning & decision making. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4803–4811, 2020. 49, 121

Chaleff, I. *Intelligent disobedience: Doing right when what you're told to do is wrong*. Berrett-Koehler Publishers, 2015. 167

Chandrasekaran, A., Yadav, D., Chattopadhyay, P., Prabhu, V., and Parikh, D. It takes two to tango: Towards theory of AI's mind. *arXiv preprint arXiv:1704.00717*, 2017. 45

Charniak, E. and McDermott, D. *Introduction to Artificial Intelligence*. Addison Wesley, 1985. 26

Chen, Y., Saffidine, A., and Schwering, C. The complexity of limited belief reasoning–the quantifier-free case. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1774–1780, 2018. 41

Clair, A. S., Atrash, A., Mead, R., and Mataric, M. J. Speech, gesture, and space: Investigating explicit and implicit communication in multi-human multi-robot collaborations. In *AAAI Spring Symposium: Multirobot Systems and Physical Data Structures*, 2011. 2

Cohen, P. R. *On knowing what to say: Planning speech acts*. PhD thesis, University of Toronto (Canada), 1978. 4

Cohen, P. R. Back to the future for dialogue research. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 13514–13519, 2020. 87

Cohen, P. R. and Galescu, L. A planning-based explainable collaborative dialogue system. *arXiv preprint arXiv:2302.09646*, 2023. 2, 169

Cohen, P. R. and Levesque, H. J. Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213–261, 1990. 2

Cohen, P. R. and Perrault, C. R. Elements of a plan-based theory of speech acts. *Cognitive science*, 3(3):177–212, 1979. 4, 86

Cohen, P. R., Perrault, C. R., and Allen, J. F. Beyond question answering. *Strategies for natural language processing*, 1981. 4, 52

Coman, A. and Aha, D. W. AI rebel agents. *AI Magazine*, 39(3):16–26, 2018. 167

Cooper, M. C., Herzig, A., Maffre, F., Maris, F., and Régnier, P. A simple account of multi-agent epistemic planning. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, pp. 193–201, 2016. 82

Cooper, M. C., Herzig, A., Maffre, F., Maris, F., Perrotin, E., and Régnier, P. A lightweight epistemic logic and its application to planning. *Artificial Intelligence*, 298:103437, 2021. 4, 17

Coplan, A. and Goldie, P. *Empathy: Philosophical and Psychological Perspectives*. Oxford University Press, 2011. 166

Coradeschi, S. and Saffiotti, A. Anchoring symbols to sensor data: preliminary report. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pp. 129–135, 2000. 129

Cuzzolin, F., Morelli, A., Cirstea, B., and Sahakian, B. J. Knowing me, knowing you: Theory of Mind in AI. *Psychological medicine*, 50(7):1057–1061, 2020. 2

Darwiche, A. and Pearl, J. On the logic of iterated belief revision. *Artificial intelligence*, 89(1-2):1–29, 1997. 3, 32, 171

Dautenhahn, K. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical transactions of the royal society B: Biological sciences*, 362(1480): 679–704, 2007. 9, 126

Davies, M. and Stone, T. *Mental simulation: evaluations and applications-reading in mind and language*. John Wiley & Sons, 1995. 1

Davis, M. H. *Empathy: A Social Psychological Approach*. Routledge, 2018. 166

Dazeley, R., Vamplew, P., Foale, C., Young, C., Aryal, S., and Cruz, F. Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artificial Intelligence*, 299:103525, 2021. 47

De Giacomo, G. and Vardi, M. Y. Automata-theoretic approach to planning for temporally extended goals. In *Recent Advances in AI Planning: 5th European Conference on Planning, ECP'99, Durham, UK, September 8-10, 1999. Proceedings 5*, pp. 226–238. Springer, 2000. 93

de Greeff, J., Blanson Henkemans, O., Fraaije, A., Solms, L., Wigdor, N., Bierman, B., Janssen, J. B., Looije, R., Baxter, P., Neerincx, M. A., et al. Child-robot interaction in the wild: Field testing activities of the aliz-e project. In *Proceedings of the 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 148–149, 2014. 146

De Ruyter, B., Saini, P., Markopoulos, P., and Van Breemen, A. Assessing the effects of building social intelligence in a robotic interface for the home. *Interacting with computers*, 2005. 2, 126, 142

Delgrande, J. P. A framework for logics of explicit belief. *Computational Intelligence*, 11(1):47–88, 1995. 41

Devin, S. and Alami, R. An implemented Theory of Mind to improve human-robot shared plans execution. In *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 319–326, 2016. 2, 126, 157

Devin, S., Clodic, A., and Alami, R. About decisions during human-robot shared plan achievement: Who should act and how? In *International Conference on Social Robotics*, pp. 453–463. Springer, 2017. 146

Dissing, L. and Bolander, T. Implementing Theory of Mind on a robot using dynamic epistemic logic. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1615–1621, 2020. 2, 126, 157, 158, 159

Ditmarsch, H. P. V. Prolegomena to dynamic logic for belief revision. *Synthese*, 147 (2):229–275, 2005. 171

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. 26

Dung, H. T. and Son, T. C. On model reconciliation: How to reconcile when robot does not know human's model? In *Proceedings 38th International Conference on Logic Programming, ICLP 2022 Technical Communications / Doctoral Consortium*, volume 364, pp. 27–48, 2022. 50

Dunn, J. Understanding others: Evidence from naturalistic studies of children. 1991. 3, 41

Ehsan, U., Passi, S., Liao, Q. V., Chan, L., Lee, I., Muller, M., Riedl, M. O., et al. The who in explainable AI: How AI background shapes perceptions of AI explanations. *arXiv preprint arXiv:2107.13509*, 2021. 47

Eifler, R., Cashmore, M., Hoffmann, J., Magazzeni, D., and Steinmetz, M. A new approach to plan-space explanation: Analyzing plan-property dependencies in over-subscription planning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 9818–9826, 2020. 121

Engesser, T. and Miller, T. Implicit coordination using FOND planning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 7151–7159, 2020. 4, 16, 123

Engesser, T., Bolander, T., Mattmüller, R., and Nebel, B. Cooperative epistemic multi-agent planning for implicit coordination. In Ghosh, S. and Ramanujam, R. (eds.), *Proceedings of the Ninth Workshop on Methods for Modalities, M4M@ICLA 2017, Indian Institute of Technology, Kanpur, India, 8th to 10th January 2017*, volume 243 of *EPTCS*, pp. 75–90, 2017. 4, 16, 123, 159

Erdogan, E., Dignum, F., Verbrugge, R., and Yolum, P. Abstracting minds: Computational Theory of Mind for human-agent collaboration. In *Proceedings of the First International Conference on Hybrid Human-Artificial Intelligence (HHAI2022)*, pp. 199–211, 2022. 2

Fabiano, F., Burigana, A., Dovier, A., and Pontelli, E. EFP 2.0: a multi-agent epistemic solver with multiple e-state representations. In *Proceedings of the 30th*

International Conference on Automated Planning and Scheduling (ICAPS), pp. 101–109, 2020. 4, 16, 76, 78, 83, 160

Fabiano, F., Burigana, A., Dovier, A., Pontelli, E., and Son, T. C. Multi-agent epistemic planning with inconsistent beliefs, trust and lies. In *Proceedings of the 18th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, 2021. 4, 16, 97

Fagin, R. and Halpern, J. Y. Belief, awareness, and limited reasoning. *Artificial intelligence*, 34(1):39–76, 1987. 41

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. *Reasoning About Knowledge*. MIT Press, 1995. 4, 14, 15, 16

Favier, A., Shekhar, S., and Alami, R. Robust planning for human-robot joint tasks with explicit reasoning on human mental state. In *AI-HRI Symposium at AAAI Fall Symposium Series (FSS)*, 2022. 157

Ferrer, G., Garrell, A., and Sanfeliu, A. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *IROS13*, pp. 1688–1694. IEEE, 2013. 126

Fikes, R. E. and Nilsson, N. J. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971. 12

Flavell, J. H. The development of knowledge about visual perception. In *Nebraska Symposium on Motivation*, 1977. 140, 150, 157, 199

Foster, M. E. and Petrick, R. Towards social HRI for improving children's healthcare experiences. *arXiv preprint arXiv:2010.04652*, 2020. 18, 127, 158

Foster, M. E., Ali, S., Litwin, S., Parker, J., Petrick, R., Smith, D. H., Stinson, J., and Zeller, F. Using AI-enhanced social robots to improve children's healthcare experiences. In *International Conference on Social Robotics*, pp. 542–553. Springer, 2020. 18, 126, 158

Fox, M., Long, D., and Magazzeni, D. Explainable planning. In *Proceedings of IJCAI-17 Workshop on Explainable Planning*, 2017. 49, 121

Freedman, R. *Integrating Recognition and Decision Making to Close the Interaction Loop for Autonomous Systems*. PhD thesis, University of Massachusetts Amherst, 2020. 147

Freedman, R. G. and Zilberstein, S. Integration of planning with recognition for responsive interaction using classical planners. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pp. 4581–4588, 2017. 83, 135, 147, 161

Freedman, R. G., Fung, Y. R., Ganchin, R., and Zilberstein, S. Responsive Planning and Recognition for Closed-Loop Interaction. *arXiv preprint arXiv:1909.06427*, 2019. 161

Freedman, R. G., Levine, S. J., Williams, B. C., and Zilberstein, S. Helpfulness as a key metric of human-robot collaboration. *arXiv preprint arXiv:2010.04914*, 2020. 9, 127, 147, 148, 155, 164

Fritz, C. and McIlraith, S. A. Monitoring plan optimality during execution. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 144–151, 2007. 90, 135

Gall, K. C. Active goal recognition design. Master's thesis, University of New Hampshire, 2021. 64

Gall, K. C., Ruml, W., and Keren, S. Active goal recognition design. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4062–4068, 2021. 64

Gärdenfors, P. *Knowledge in flux: modeling the dynamics of epistemic states*, volume 6. MIT press, 1988. 31, 40, 46, 48

Garrell, A. and Sanfeliu, A. Cooperative social robots to accompany groups of people. *The International Journal of Robotics Research*, 31(13):1675–1701, 2012. 126

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6), 2014. 140

Gasquet, O., Goranko, V., and Schwarzentruber, F. Big brother logic: visual-epistemic reasoning in stationary multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 30(5):793–825, 2016. 150

Geib, C. and Goldman, R. Recognizing plans with loops represented in a lexicalized grammar. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pp. 958–963, 2011. 52, 84

Geib, C., Weerasinghe, J., Matskevich, S., Kantharaju, P., Craenen, B., and Petrick, R. Building helpful virtual agents using plan recognition and planning. In *Proceedings of the 12th Annual International AIIDE Conference (AIIDE)*, pp. 162–168, 2016. 161

Geib, C. W. and Goldman, R. P. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, 2009. 52, 84

Gerevini, A. E., Haslum, P., Long, D., Saetti, A., and Dimopoulos, Y. Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. *Artificial Intelligence*, 173(5-6):619–668, 2009. URL https://www.icaps-conference.org/competitions/. 111

Gervits, F. *Toward Genuine Robot Teammates: Coordination Through Dialogue*. PhD thesis, Tufts University, 2020. 158, 159

Gervits, F., Thurston, D., Thielstrom, R., Fong, T., Pham, Q., and Scheutz, M. Toward genuine robot teammates: Improving human-robot team performance using robot shared mental models. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 429–437, 2020. 2, 121, 147, 157

Ghallab, M., Nau, D., and Traverso, P. *Automated Planning: theory and practice*. Elsevier, 2004. 4, 11

Gmytrasiewicz, P. J. and Doshi, P. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:49–79, 2005. 4

Görür, O. C., Rosman, B. S., Hoffman, G., and Albayrak, S. Toward integrating Theory of Mind into adaptive decision-making of social robots to understand human intention. In *HRI Workshop on the Role of Intentions*, 2017. 2, 126, 157

Grice, H. P. Logic and conversation. In *Speech acts*, pp. 41–58. Brill, 1975. 39, 40

Grosinger, J. On proactive human–AI systems. In *8th International Workshop on Artificial Intelligence and Cognition (AIC)*, 2022. 160

Grosinger, J., Pecora, F., and Saffiotti, A. Robots that maintain equilibrium: Proactivity by reasoning about user intentions and preferences. *Pattern Recognition Letters*, 118:85–93, 2019. 160

Grosz, B. J. and Kraus, S. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996. 2

Grove, A. Two modellings for theory change. *Journal of philosophical logic*, pp. 157–170, 1988. 40

Grover, S., Sengupta, S., Chakraborti, T., Mishra, A. P., and Kambhampati, S. RADAR: automated task planning for proactive decision support. *Human Computer Interaction*, 35(5-6):387–412, 2020. doi: 10.1080/07370024.2020.1726751. 122

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., and Yang, G. XAI - Explainable Artificial Intelligence. *Sci. Robotics*, 4(37), 2019. doi: 10.1126/scirobotics.aay7120. 26

Gurney, N. and Pynadath, D. V. Robots with Theory of Mind for humans: A survey. In *Proceedings of the 31st IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 993–1000. IEEE, 2022. 157

Gzesh, S. M. and Surber, C. F. Visual perspective-taking skills in children. *Child development*, pp. 1204–1213, 1985. 150

Halpern, J. Y. and Pearl, J. Causes and explanations: A structural-model approach. part ii: Explanations. *The British journal for the philosophy of science*, 56(4): 889–911, 2005. 31, 49, 122

Hamilton, A. F. d. C., Brindley, R., and Frith, U. Visual perspective taking impairment in children with autistic spectrum disorder. *Cognition*, 113(1):37–44, 2009. 150

Harbers, M., Van den Bosch, K., and Meyer, J.-J. Modeling agents with a Theory of Mind: Theory–theory versus simulation theory. *Web Intelligence and Agent Systems: An International Journal*, 10(3):331–343, 2012. 46

Harman, H. *Symbolic artificial intelligence techniques to facilitate proactive robot assistance*. PhD thesis, Ghent University, 2020. 160

Harman, H. and Simoens, P. Action graphs for proactive robot assistance in smart environments. *Journal of Ambient Intelligence and Smart Environments*, 12(2): 79–99, 2020. 160, 161

Haslum, P. Personal communication, 2014. 88

He, L. and Liu, G. Petri nets based verification of epistemic logic and its application on protocols of privacy and security. In *2020 IEEE World Congress on Services (SERVICES)*, pp. 25–28. IEEE, 2020. 38

Helmert, M. The Fast Downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006. 76, 78, 107, 141, 200

Hempel, C. G. and Oppenheim, P. Studies in the logic of explanation. *Philosophy of science*, 15(2):135–175, 1948. 26

Hilton, D. J. Conversational processes and causal explanation. *Psychological Bulletin*, 107(1):65, 1990. 26, 39

Hintikka, J. Knowledge and belief. an introduction to the logic of the two notions. 1965. 14

Ho, C.-C. and MacDorman, K. F. Revisiting the uncanny valley theory: Developing and validating an alternative to the godspeed indices. *Computers in Human Behavior*, 26(6):1508–1518, 2010. 143

Hobbs, J. R., Stickel, M. E., Appelt, D. E., and Martin, P. Interpretation as abduction. *Artificial intelligence*, 63(1-2):69–142, 1993. 39

Hoffman, G. and Breazeal, C. Cost-based anticipatory action selection for human–robot fluency. *IEEE Transactions on Robotics*, 23(5):952–961, 2007a. 161

Hoffman, G. and Breazeal, C. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1–8, 2007b. 161

Hoffmann, J. and Magazzeni, D. Explainable AI planning (XAIP): overview and the case of contrastive explanation. *Reasoning Web. Explainable AI*, pp. 277–282, 2019. 49, 121

Horsman, S. The frame problem in social cognition. Master's thesis, Radboud Universiteit, 2019. vi, 3

Houlihan, S. D., Kleiman-Weiner, M., Hewitt, L. B., Tenenbaum, J. B., and Saxe, R. Emotion prediction as computation over a generative Theory of Mind. 2023. 167

Hu, G., Miller, T., and Lipovetzky, N. What you get is what you see: Decomposing epistemic planning using functional strips. *arXiv preprint arXiv:1903.11777*, 2019. 150

Hu, G., Miller, T., and Lipovetzky, N. Planning with perspectives–decomposing epistemic planning using functional strips. *Journal of Artificial Intelligence Research*, 75:489–539, 2022. 4, 17, 150

Hu, G., Miller, T., and Lipovetzky, N. Planning with multi-agent belief using justified perspectives. 2023. 17

Huang, X., Fang, B., Wan, H., and Liu, Y. A general multi-agent epistemic planner based on higher-order belief change. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1093–1101, 2017. 4, 16, 17, 18, 76, 82

Izmirlioglu, Y., Pham, L., Son, T. C., and Pontelli, E. State transition in multi-agent epistemic domains using answer set programming. In *Proceedings of the 16th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pp. 273–286, 2022. 4, 17

Jakubczak, M., Sridharan, M., and Mansouri, M. Non-monotonic logical reasoning and Theory of Mind for transparency in HRI. In *Proceedings of the Workshop on Adaptive Social Interaction based on user's Mental mOdels and behaVior in HRI (ASIMOV)*, 2022. 157

Jara-Ettinger, J. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019. 2

Johnson, M., Jonker, C., Van Riemsdijk, B., Feltovich, P. J., and Bradshaw, J. M. Joint activity testbed: Blocks world for teams (BW4T). In *International Workshop on Engineering Societies in the Agents World*, pp. 254–256. Springer, 2009. 107, 149

Kahn Jr. P. H., Kanda, T., Ishiguro, H., Gill, B. T., Shen, S., Gary, H. E., and Ruckert, J. H. Will people keep the secret of a humanoid robot? psychological intimacy in hri. In *Proceedings of the 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 173–180, 2015. 146

Kambhampati, S., Sreedharan, S., Verma, M., Zha, Y., and Guan, L. Symbols as a lingua franca for bridging human-AI chasm for explainable and advisable AI

systems. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 12262–12267, 2022. 5

Kaplan, A. N. and Schubert, L. K. A computational model of belief. *Artificial Intelligence*, 120(1):119–160, 2000. 41

Kaptein, F., Broekens, J., Hindriks, K., and Neerincx, M. Personalised self-explanation by robots: The role of goals versus beliefs in robot-action explanation for children and adults. In *Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 676–682. IEEE, 2017. 47

Karpas, E., Abend, O., Belinkov, Y., Lenz, B., Lieber, O., Ratner, N., Shoham, Y., Bata, H., Levine, Y., Leyton-Brown, K., et al. Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv preprint arXiv:2205.00445*, 2022. 169

Katz, M. and Sohrabi, S. Reshaping diverse planning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 9892–9899, 2020. 72

Katz, M., Sohrabi, S., Udrea, O., and Winterer, D. A novel iterative approach to top-k planning. In *Proceedings of the 28th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 132–140, 2018. 72

Katz, M., Sohrabi, S., and Udrea, O. Top-quality: Finding practically useful sets of best plans. In *Proceedings of the Heuristics and Search for Domain-independent Planning (HSDIP) workshop at ICAPS*, 2019. 72

Katz, M., Sohrabi, S., and Udrea, O. Top-quality planning: Finding practically useful sets of best plans. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 9900–9907, 2020. 72

Kautz, H. A. *A formal theory of plan recognition*. PhD thesis, University of Rochester. Department of Computer Science, 1987. 117

Kautz, H. A. and Allen, J. F. Generalized plan recognition. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI)*, pp. 32–37, 1986. 4, 27, 52, 84

Keren, S., Gal, A., and Karpas, E. Goal recognition design. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, 2014. 64

Keren, S., Gal, A., and Karpas, E. Privacy preserving plans in partially observable environments. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016. 84

Keren, S., Xu, H., Kwapong, K., Parkes, D., and Grosz, B. Information shaping for enhanced goal recognition of partially-informed agents. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 9908–9915, 2020. 64

Klassen, T. Q. *Representing Plausible Beliefs about States, Actions, and Processes*. PhD thesis, University of Toronto (Canada), 2021. 172

Klassen, T. Q., McIlraith, S. A., and Levesque, H. J. Towards tractable inference for resource-bounded agents. In *2015 AAAI Spring Symposium Series*, 2015. 41

Klassen, T. Q., Alamdari, P. A., and McIlraith, S. A. Epistemic side effects & avoiding them (sometimes). In *NeurIPS ML Safety Workshop*, 2022a. 168

Klassen, T. Q., McIlraith, S. A., and Muise, C. An AI safety threat from learned planning models. In *ICAPS'22 Workshop on Reliable Data-Driven Planning and Scheduling (RDDPS)*, 2022b. 168

Klassen, T. Q., Alizadeh Alamdari, P., and McIlraith, S. A. Epistemic side effects: An ai safety problem. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023. 168

Kominis, F. and Geffner, H. Beliefs in multiagent planning: from one agent to many. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 147–155, 2015. 4, 16

Kominis, F. and Geffner, H. Multiagent online planning with nested beliefs and dialogue. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS)*, 2017. 16

Konolige, K. A deduction model of belief. 1985. 41

Kory, J. J. M. Storytelling with robots: effects of robot language level on children's language learning. Master's thesis, Massachusetts Institute of Technology, 2014. 146

Kosinski, M. Theory of mind may have spontaneously emerged in large language models. *arXiv preprint arXiv:2302.02083*, 2023. 4, 169

Krause, L. and Vossen, P. When to explain: Identifying explanation triggers in human-agent interaction. In *2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*, pp. 55–60, 2020. 43

Kreiss, S., Bertoni, L., and Alahi, A. OpenPifPaf: Composite fields for semantic keypoint detection and spatio-temporal association. In *IEEE Transactions on Intelligent Transportation Systems*. IEEE, 2021. 199

Kulkarni, A., Srivastava, S., and Kambhampati, S. A unified framework for planning in adversarial and cooperative environments. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2479–2487, 2019. 84

Kulkarni, A., Srivastava, S., and Kambhampati, S. Planning for proactive assistance in environments with partial observability. In *ICAPS 2021 Workshop on Explainable AI Planning*, 2021. 160

Labbé, M. and Michaud, F. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 2019. 140

Lakatos, G., Wood, L. J., Syrdal, D. S., Robins, B., Zaraki, A., and Dautenhahn, K. Robot-mediated intervention can assist children with autism to develop visual perspective taking skills. *Paladyn, Journal of Behavioral Robotics*, 12(1):87–101, 2021. 150

Lakemeyer, G. Limited reasoning in first-order knowledge bases. *Artificial Intelligence*, 71(2):213–255, 1994. 41

Lakemeyer, G. and Lespérance, Y. Efficient reasoning in multiagent epistemic logics. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, pp. 498–503, 2012. 19

Lakemeyer, G. and Levesque, H. J. Decidable reasoning in a logic of limited belief with introspection and unknown individuals. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013. 41

Lakemeyer, G. and Levesque, H. J. Decidable reasoning in a fragment of the epistemic situation calculus. In *Proceedings of the 14th International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, 2014. 41

Lakemeyer, G. and Levesque, H. J. Decidable reasoning in a logic of limited belief with function symbols. In *Proceedings of the 15th International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, 2016. 41

Le, T., Fabiano, F., Son, T. C., and Pontelli, E. EFP and PG-EFP: epistemic forward search planners in multi-agent domains. In *Proceedings of the 28th International Conference on Automated Planning and Scheduling (ICAPS)*, 2018. 4, 16, 17, 76, 78, 79, 115, 160

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 47

Lee, J. and Toutanova, K. Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4, 168

Leiner, D. J. SoSci survey, 2014. URL soscisurvey.de/. 143

Levesque, H. J. A logic of implicit and explicit belief. In *Proceedings of the 4th National Conference on Artificial Intelligence (AAAI)*, pp. 198–202, 1984. 41, 48

Levesque, H. J. Knowledge representation and reasoning. *Annual review of computer science*, 1(1):255–287, 1986. 4

Levesque, H. J. A knowledge-level account of abduction. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1061–1067, 1989. 26, 31, 38, 39, 40, 48

Levesque, H. J. A completeness result for reasoning with incomplete first-order knowledge bases. In *Proceedings of the 6th International Conference on Knowledge Representation and Reasoning (KR)*, pp. 14–23, 1998. 19

Levesque, H. J. and Lakemeyer, G. Cognitive robotics. In *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pp. 869–886. Elsevier, 2008. 160

Levine, S. J. and Williams, B. C. Concurrent plan recognition and execution for human-robot teams. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, 2014. 161

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative

reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022. 168

Leyzberg, D., Spaulding, S., and Scassellati, B. Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 423–430. IEEE, 2014. 2, 126, 157

Li, H., Le, L., Chis, M., Zheng, K., Hughes, D., Lewis, M., and Sycara, K. Sequential Theory of Mind modeling in team search and rescue tasks. In *Computational Theory of Mind for Human-Machine Teams Symposium at AAAI Fall Symposium Series (FSS)*, 2021. 2

Liao, L., Patterson, D. J., Fox, D., and Kautz, H. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, 2007. 65

Liberman, A. O. *Representation, learning and planning for theory-of-mind agents using dynamic epistemic logic*. PhD thesis, Technical University of Denmark, 2020. 2

Liberman, A. O., Achen, A., and Rendsvig, R. K. Dynamic term-modal logics for first-order epistemic planning. *Artificial Intelligence*, 286:103305, 2020. 16

Lifschitz, V. Closed-world databases and circumscription. *Artificial Intelligence*, 27 (2):229–235, 1985. 12

Lin, F. and Reiter, R. How to progress a database. *Artificial Intelligence*, 92(1-2): 131–167, 1997. 21

Lindsay, A. and Petrick, R. Supporting explanations within an instruction giving framework. In *ICAPS 2021 Workshop on Explainable AI Planning (XAIP)*, 2021. 49, 121

Lindsay, A., Ramirez-Duque, A., Petrick, R., and Foster, M. E. A socially assistive robot using automated planning in a paediatric clinical setting. In *AI-HRI Symposium at AAAI Fall Symposium Series (FSS)*, 2022. 167

Lipton, P. Contrastive Explanation. *Royal Institute of Philosophy Supplements*, 27: 247–266, 1990. 38

Liu, Q. and Liu, Y. Multi-agent epistemic planning with common knowledge. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1912–1920, 2018. 4, 16

Liu, Y., Lakemeyer, G., and Levesque, H. J. A logic of limited belief for reasoning with disjunctive information. In *Proceedings of the 9th International Conference on Knowledge Representation and Reasoning (KR)*, pp. 587–597, 2004. 19, 41

Mahowald, K., Ivanova, A. A., Blank, I. A., Kanwisher, N., Tenenbaum, J. B., and Fedorenko, E. Dissociating language and thought in large language models: a cognitive perspective. *arXiv preprint arXiv:2301.06627*, 2023. 168

Malle, B. F., Knobe, J., O'Laughlin, M. J., Pearce, G. E., and Nelson, S. E. Conceptual structure and social functions of behavior explanations: Beyond person–situation attributions. *Journal of Personality and Social Psychology*, 79(3):309, 2000. 26

Martínez, E. and Lespérance, Y. Web service composition as a planning task: Experiments using knowledge-based planning. In *AAAI Fall Symposium: Agents and the Semantic Web*, pp. 38–46, 2005. 17

Masters, P. and Sardina, S. Deceptive path-planning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4368–4375, 2017. 84

Masters, P. and Sardina, S. Goal recognition for rational and irrational agents. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 440–448, 2019. 64

Masters, P. and Vered, M. What's the context? implicit and explicit assumptions in model-based goal recognition. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4516–4523, 2021. 85

Masters, P., Kirley, M., and Smith, W. Extended goal recognition: a planning-based model for strategic deception. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 871–879, 2021. 60, 84

McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., and Wilkins, D. PDDL — The Planning Domain Definition Language. Tech-

nical Report TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998. 21, 106, 140

McDuff, D. and Czerwinski, M. Designing Emotionally Sentient Agents. *Communications of the ACM*, 61(12):74–83, 2018. 167

McIlraith, S. Explanatory diagnosis: Conjecturing actions to explain observations. In *Proceedings of the 6th International Conference on Knowledge Representation and Reasoning (KR)*, pp. 167–179, 1998. 65

Meneguzzi, F. R. and Pereira, R. F. A survey on goal recognition as planning. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 85

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., et al. SymPy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017. URL https://sympy.org/. 107, 179, 201

Michlmayr, M. Simulation theory versus theory theory: Theories concerning the ability to read minds. Master's thesis, Leopold-Franzens-Universität Innsbruck, 2002. 1

Miller, P. H., Kessel, F. S., and Flavell, J. H. Thinking about people thinking about people thinking about...: A study of social cognitive development. *Child Development*, pp. 613–623, 1970. 3, 41

Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019. 6, 26, 47, 121

Miller, T. Contrastive explanation: A structural-model approach. *The Knowledge Engineering Review*, 36, 2021. 49, 122

Miller, T. and Muise, C. J. Belief update for proper epistemic knowledge bases. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1209–1215, 2016. 22

Miller, T., Felli, P., Muise, C., Pearce, A., and Sonenberg, L. 'knowing whether' in proper epistemic knowledge bases. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016. 82

Miller, T., Pearce, A. R., and Sonenberg, L. Social planning for trusted autonomy. In Hussein A. Abbass, Jason Scholz, D. J. R. (ed.), *Foundations of Trusted Autonomy*, chapter 3. Springer International Publishing, 2018. doi: 10.1007/978-3-319-64816-3_4. 127, 158

Milliez, G., Warnier, M., Clodic, A., and Alami, R. A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management. In *The 23rd IEEE international symposium on robot and human interactive communication*, pp. 1103–1109. IEEE, 2014. 150

Mirsky, R. and Stone, P. The seeing-eye robot grand challenge: Rethinking automated care. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2021a. 167

Mirsky, R. and Stone, P. Intelligent disobedience and AI rebel agents in assistive robotics. In *Proceedings of the ASIMOV workshop as part of the International Conference on Social Robotics (ICSR 2021)*, 2021b. 167

Mirsky, R., Gal, Y. K., Stern, R., and Kalech, M. Sequential plan recognition. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1347–1348, 2016. 64

Mirsky, R., Stern, R., Gal, K., and Kalech, M. Sequential plan recognition: An iterative approach to disambiguating between hypotheses. *Artificial Intelligence*, 260:51–73, 2018. 64

Mirsky, R., Macke, W., Wang, A., Yedidsion, H., and Stone, P. A penny for your thoughts: The value of communication in ad hoc teamwork. *Good Systems-Published Research*, 2020. 135

Mirsky, R., Keren, S., and Geib, C. Introduction to symbolic plan and goal recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 16(1): 1–190, 2021. 85

Montavon, G., Samek, W., and Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018. 47

Montes, N., Osman, N., and Sierra, C. Combining Theory of Mind and abduction for cooperation under imperfect information. *arXiv preprint arXiv:2209.15279*, 2022. 2

Mou, W., Ruocco, M., Zanatto, D., and Cangelosi, A. When would you trust a robot? a study on trust and Theory of Mind in human-robot interactions. In *Proceedings of the 29th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 956–962. IEEE, 2020. 2, 142

Muise, C. Planning.domains. *ICAPS system demonstration*, pp. 242–250, 2016. URL http://editor.planning.domains/. 111, 193

Muise, C. Multi-agent epistemic planning with proper doxastic knowledge bases (BW4T domain). https://github.com/QuMuLab/pdkb-planning/tree/ef396e686147bd3e2e392ffadebcefd07998f45f/examples/planning/bw4t, 2021a. 108

Muise, C. Multi-agent epistemic planning with proper doxastic knowledge bases (Corridor domain). https://github.com/QuMuLab/pdkb-planning/tree/ef396e686147bd3e2e392ffadebcefd07998f45f/examples/planning/corridor, 2021b. 75, 110

Muise, C. Multi-agent epistemic planning with proper doxastic knowledge bases (Grapevine domain). https://github.com/QuMuLab/pdkb-planning/tree/ef396e686147bd3e2e392ffadebcefd07998f45f/examples/planning/grapevine, 2021c. 75

Muise, C. Multi-agent epistemic planning with proper doxastic knowledge bases. https://github.com/QuMuLab/pdkb-planning/tree/ef396e686147bd3e2e392ffadebcefd07998f45f, 2021d. 107, 140

Muise, C., Miller, T., Felli, P., Pearce, A. R., and Sonenberg, L. Efficient reasoning with consistent proper epistemic knowledge bases. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1461–1469, 2015a. 19

Muise, C., Belle, V., Felli, P., McIlraith, S. A., Miller, T., Pearce, A. R., and Sonenberg, L. Efficient multi-agent epistemic planning: Teaching planners about nested belief. *Artificial Intelligence Journal*, 2021. 4, 13, 17, 18, 19, 20, 21, 22, 23, 24, 25, 83, 99, 103, 107, 115, 137, 139, 140, 150, 155, 156, 207

Muise, C. J., Belle, V., Felli, P., McIlraith, S. A., Miller, T., Pearce, A. R., and Sonenberg, L. Planning over multi-agent epistemic states: a classical planning Approach. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*

(AAAI), pp. 3327–3334, 2015b. 4, 16, 17, 18, 19, 20, 24, 41, 76, 82, 83, 99, 107, 130, 136, 140, 155

Nematzadeh, A., Burns, K., Grant, E., Gopnik, A., and Griffiths, T. L. Evaluating Theory of Mind in question answering. In *2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pp. 2392–2400. Association for Computational Linguistics, 2020. 4

Nepomuceno-Fernández, A., Soler-Toscano, F., and Velázquez-Quesada, F. Abductive reasoning in dynamic epistemic logic. In *Springer handbook of model-based science*, pp. 269–293. Springer, 2017. 34, 48

Nguyen, D., Nguyen, P., Le, H., Do, K., Venkatesh, S., and Tran, T. Memory-augmented Theory of Mind network. *arXiv preprint arXiv:2301.06926*, 2023. 4

Nguyen, T. A., Do, M., Gerevini, A. E., Serina, I., Srivastava, B., and Kambhampati, S. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence*, 190:1–31, 2012. 72

Nickerson, J. V. and Reilly, R. R. A model for investigating the effects of machine autonomy on human behavior. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pp. 1–10. IEEE, 2004. 141

Nikolaidis, S. and Shah, J. Human-robot teaming using shared mental models. In *Proceedings of the Workshop on Human-Agent-Robot Teamwork at the 2012 ACM/IEEE International Conference on Human Robot Interaction (HRI)*, 2012. 2, 157

Nikolaidis, S., Zhu, Y. X., Hsu, D., and Srinivasa, S. Human-robot mutual adaptation in shared autonomy. In *Proceedings of the 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 294–302, 2017. 2, 126, 157

Nomura, T., Suzuki, T., Kanda, T., and Kato, K. Measurement of negative attitudes toward robots. *Interaction Studies*, 7(3):437–454, 2006. 143

Oguntola, I., Hughes, D., and Sycara, K. Deep interpretable models of Theory of Mind. In *Proceedings of the 30st IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 657–664. IEEE, 2021. 2, 4

Palacios, H. and Geffner, H. From conformant into classical planning: Efficient translations that may be complete too. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 264–271, 2007. 82

Pandey, A. K. and Gelin, R. A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine*, 25(3):40–48, 2018. 9, 127, 139, 164

Panisson, A. R., Sarkadi, S., McBurney, P., Parsons, S., and Bordini, R. H. Lies, bullshit, and deception in agent-oriented programming languages. In *Proceedings of the 20th TRUST Workshop, co-located with AAMAS/IJCAI/ECAI/ICML*, 2018. 2

Patel-Schneider, P. F. A decidable first-order logic for knowledge representation. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 455–458, 1985. 41

Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014. 40

Pednault, E. P. D. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proceedings of the 1st International Conference of Knowledge Representation and Reasoning (KR)*, pp. 324–332, 1989. 12

Peirce, C. Deduction, induction and hypothesis. *Popular Science Monthly*, 13, 1878. 26

Pereira, R. F., Oren, N., and Meneguzzi, F. Landmark-based heuristics for goal recognition. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 83

Pereira, R. F., Pereira, A. G., and Meneguzzi, F. Landmark-enhanced heuristics for goal recognition in incomplete domain models. In *Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 329–337, 2019a. 65

Pereira, R. F., Vered, M., Meneguzzi, F., and Ramırez, M. Online probabilistic goal recognition over nominal models. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 5547–5553, 2019b. 65

Perrault, C. R. and Allen, J. F. A plan-based analysis of indirect speech acts. *Computational Linguistics*, 6(3-4):167–182, 1980. 4, 86

Perrault, C. R., Allen, J., and Cohen, P. R. Speech acts as a basis for understanding dialogue coherence. *American Journal of Computational Linguistics*, pp. 32–39, 1978. 4

Persiani, M. and Hellström, T. Inference of the intentions of unknown agents in a Theory of Mind setting. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 188–200, 2021. 86

Petrick, R. and Foster, M. E. Planning for social interaction in a robot bartender domain. In *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 389–397, 2013. 127, 158

Petrick, R. and Foster, M. E. Knowledge engineering and planning for social human–robot interaction: a case study. In *Knowledge Eng. Tools and Techniques for AI Planning*, pp. 261–277. Springer, 2020. 18, 127, 158

Petrick, R. P. and Bacchus, F. A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the 6th International Conference on Artificial Intelligence Planning and Scheduling (AIPS)*, pp. 212–222, 2002. 4, 16, 17, 158

Petrick, R. P. and Hill, R. L. Start making sense: Cognitive and affective confidence measures for explanation generation using epistemic planning. In *AAAI 2019 Spring Symposium on Story-Enabled Intelligence*, 2019. 43, 167

Petrick, R. P., Dalzel-Job, S., and Hill, R. L. Combining cognitive and affective measures with epistemic planning for explanation generation. In *ICAPS 2019 Workshop on Explainable Planning (XAIP)*, pp. 141–145, 2019. 16, 167

Pham, L., Izmirlioglu, Y., Son, T. C., and Pontelli, E. A new semantics for action language m$\mathcal{A}^*$. In *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 553–562, 2023. 83

Pollack, M. E. A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, pp. 207–214, 1986. 8, 52, 53, 59, 64, 86, 87, 88

Poole, D. Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989. 26

Poole, D. A methodology for using a default and abductive reasoning system. *International Journal Intelligent Systems*, 5(5):521–548, 1990. 30

Pople, H. E. On the mechanization of abductive logic. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 147–152, 1973. 26

Pozanco, A., E-Martín, Y., Fernández, S., and Borrajo, D. Counterplanning using goal recognition and landmarks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4808–4814, 2018. 135

Premack, D. and Woodruff, G. Does the chimpanzee have a Theory of Mind? *Behavioral and brain sciences*, 1(4):515–526, 1978. 1

Pütz, S., Simón, J. S., and Hertzberg, J. Move base flex: a highly flexible navigation framework for mobile robots. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 141

Pynadath, D. V. and Wellman, M. P. Probabilistic state-dependent grammars for plan recognition. In Boutilier, C. and Goldszmidt, M. (eds.), *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence (UAI)*, pp. 507–514. Morgan Kaufmann, 2000. 52, 84

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009. 140

Quine, W. V. O. and Ullian, J. S. *The web of belief*, volume 2. Random House New York, 1978. 40

Rabinowitz, N. C., Perbet, F., Song, H. F., Zhang, C., Eslami, S., and Botvinick, M. Machine Theory of Mind. *arXiv preprint arXiv:1802.07740*, 2018. 2, 4

Rabkina, I., McFate, C., Forbus, K. D., and Hoyos, C. Towards a computational analogical Theory of Mind. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2017. 86

Rabkina, I., Kathnaraju, P., Roberts, M., Wilson, J., Forbus, K., and Hiatt, L. Recognizing the goals of uninspectable agents. In *Proceedings of the Conference on Advances in Cognitive Systems*, pp. 1–8, 2020. 86

Ramírez, M. and Geffner, H. Plan recognition as planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1778–1783, 2009. 55, 65

Ramírez, M. and Geffner, H. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Conference of the Association for the Advancement*

of Artificial Intelligence (AAAI 2010), pp. 1121–1126, 2010. 52, 57, 66, 71, 72, 77, 78, 80, 83, 85, 117, 130, 208

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 140

Reiter, R. On closed world data bases. In *Readings in artificial intelligence*, pp. 119–140. Elsevier, 1981. 12

Reiter, R. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1): 57–95, 1987. ISSN 0004-3702. 39, 42, 65

Reiter, R. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT press, 2001. 90

Rintanen, J. Regression for classical and nondeterministic planning. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI)*, pp. 568–572, 2008. 90, 100

Rolnick, A. 'Experience others as they would experience themselves' - some thoughts on empathy, mind and relationship, Nov 2013. URL https://www.hebpsy.net/blog_Post.asp?id=1128. 1

Sadek, M. D., Bretier, P., and Panaget, F. ARTIMIS: Natural dialogue meets rational agency. *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1030–1035, 1997. 4, 16

Sakama, C. A formal account of deception. In *2015 AAAI Fall Symposium Series*, 2015. 60

Samek, W., Wiegand, T., and Müller, K.-R. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017. 26

Sanders, T., Kaplan, A., Koch, R., Schwartz, M., and Hancock, P. A. The relationship between trust and use choice in human-robot interaction. *Human factors*, 61(4): 614–626, 2019. 141

Sap, M., LeBras, R., Fried, D., and Choi, Y. Neural Theory-of-Mind? On the limits of social intelligence in large LMs. In *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. Association for Computational Linguistics, 2022. 4, 169

Sarkadi, Ş., Panisson, A. R., Bordini, R. H., McBurney, P., Parsons, S., and Chapman, M. Modelling deception using Theory of Mind in multi-agent systems. *AI Communications*, 32(4):287–302, 2019. 2

Scassellati, B. Theory of Mind for a humanoid robot. *Autonomous Robots*, 12(1): 13–24, 2002. 2, 126, 157

Schaefer, K. E., Chen, J. Y., Szalma, J. L., and Hancock, P. A. A meta-analysis of factors influencing the development of trust in automation: Implications for understanding autonomy in future systems. *Human factors*, 58(3):377–400, 2016. 141

Schmidt, C. F., Sridharan, N., and Goodson, J. L. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11 (1-2):45–83, 1978. 4, 52, 84

Schwering, C. and Lakemeyer, G. Decidable reasoning in a first-order logic of limited conditional belief. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, pp. 1379–1387, 2016. 41

Sclar, M., Neubig, G., and Bisk, Y. Symmetric machine Theory of Mind. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 19450–19466. PMLR, 2022. 2

Shamay-Tsoory, S. G. The Neural Bases for Empathy. *The Neuroscientist*, 17(1): 18–24, 2011. 166

Shvo, M. and McIlraith, S. A. Towards empathetic planning. In *Proceedings of the 2nd Humanizing AI workshop at IJCAI 2019*, 2019. 9, 128, 131, 137, 166, 167

Shvo, M. and McIlraith, S. A. Active goal recognition. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 9957–9966, 2020. 64, 135

Shvo, M., Klassen, T. Q., and McIlraith, S. A. Towards the role of Theory of Mind in explanation. In *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pp. 75–93. Springer International Publishing, 2020a. 6, 28

Shvo, M., Klassen, T. Q., Sohrabi, S., and McIlraith, S. A. Epistemic plan recognition. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1251–1259, 2020b. 7, 54, 115

Shvo, M., Hari, R., O'Reilly, Z., Abolore, S., Wang, S.-Y. N., and McIlraith, S. A. Proactive robotic assistance via Theory of Mind, 2022a. URL https://www.youtube.com/playlist?list=PLJPUHrIcka2yC5JGgH4iCXQgXbsphuAxO. 143

Shvo, M., Hari, R., O'Reilly, Z., Abolore, S., Wang, S.-Y. N., and McIlraith, S. A. Proactive robotic assistance via Theory of Mind. In *Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9148–9155, 2022b. 9, 128

Shvo, M., Klassen, T. Q., and McIlraith, S. A. Resolving misconceptions about the plans of agents via Theory of Mind. In *Proceedings of the 32nd International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 719–729, 2022c. 8, 89

Shvo, M. et al. Broadening the scope of multi-agent plan recognition: Theory and practice. Master's thesis, Universiteit Utrecht, 2018. 54

Sindlar, M. P., Dastani, M. M., Dignum, F., and Meyer, J.-J. C. Mental state abduction of BDI-based agents. In *International Workshop on Declarative Agent Languages and Technologies*, pp. 161–178. Springer, 2008. 85

Sindlar, M. P., Dastani, M. M., and Meyer, J.-J. C. BDI-based development of virtual characters with a Theory of Mind. In *International Workshop on Intelligent Virtual Agents*, pp. 34–41. Springer, 2009. 2, 157

Singh, S. and Khemani, D. Planning with subjective knowledge in a multi-agent scenario. In *11th Hellenic Conference on Artificial Intelligence*, pp. 1–9, 2020. 4, 17

Singh, S. and Khemani, D. Mental actions and explainability in Kripkean semantics: What else do I know? (student abstract). In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 15893–15894, 2021. 17

Sirkin, D., Mok, B., Yang, S., and Ju, W. Mechanical ottoman: how robotic furniture offers and withdraws support. In *Proceedings of the 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 11–18, 2015. 146

Slugoski, B. R., Lalljee, M., Lamb, R., and Ginsburg, G. P. Attribution in conversational context: Effect of mutual knowledge on explanation-giving. *European Journal of Social Psychology*, 23(3):219–238, 1993. 26, 27

Smith, W., Dignum, F., and Sonenberg, L. The construction of impossibility: a logic-based analysis of conjuring tricks. *Frontiers in psychology*, 7:748, 2016. 60

Söderlund, M. Service robots with (perceived) Theory of Mind: An examination of humans' reactions. *Journal of Retailing and Consumer Services*, 67:102999, 2022. 2, 142

Sohrabi, S., Baier, J., and McIlraith, S. Diagnosis as planning revisited. In *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, pp. 26–36, 2010. 57, 66

Sohrabi, S., Baier, J. A., and McIlraith, S. A. Preferred explanations: theory and generation via planning. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pp. 261–267, 2011. 65, 66

Sohrabi, S., Riabov, A. V., and Udrea, O. Plan recognition as planning revisited. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3258–3264, 2016. 52, 57, 63, 66, 72, 85

Soldà, D., Fabiano, F., and Dovier, A. Epistemic multiagent reasoning with collaborative robots. In *37th Italian Conference on Computational Logic*, 2022. 127, 158, 159, 160

Son, T. C. and Baral, C. Formalizing sensing actions—a transition function based approach. *Artificial Intelligence*, 125(1-2):19–91, 2001. 82

Son, T. C., Nguyen, V., Vasileiou, S. L., and Yeoh, W. Model reconciliation in logic programs. In *European Conference on Logics in Artificial Intelligence*, pp. 393–406. Springer, 2021. 122

Sreedharan, S., Chakraborti, T., and Kambhampati, S. Handling model uncertainty and multiplicity in explanations via model reconciliation. In *Proceedings of the 28th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 518–526, 2018. 45

Sreedharan, S., Hernandez, A. O., Mishra, A. P., and Kambhampati, S. Model-free model reconciliation. In *Proceedings of the 28th International Joint Conference on*

Artificial Intelligence (IJCAI), pp. 587–594, 2019. doi: 10.24963/ijcai.2019/83. 45, 50

Sreedharan, S., Chakraborti, T., Muise, C., and Kambhampati, S. Expectation-aware planning: a general framework for synthesizing and executing self-explaining plans for human-ai interaction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2518–2526, 2020a. 94, 123

Sreedharan, S., Chakraborti, T., Muise, C., Khazaeni, Y., and Kambhampati, S. - D3WA+ - a case study of XAIP in a model acquisition task for dialogue planning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 488–498, 2020b. 122

Sreedharan, S., Chakraborti, T., and Kambhampati, S. Foundations of explanations as model reconciliation. *Artificial Intelligence*, 2021. 49, 89, 122

Stalnaker, R. The problem of logical omniscience, i. *Synthese*, 89(3):425–440, 1991. 32

Steup, M. The analysis of knowledge. *Stanford encyclopedia of philosophy*, 2007. 14

Sturgeon, S., Palmer, A., Blankenburg, J., and Feil-Seifer, D. Perception of social intelligence in robots performing false-belief tasks. In *Proceedings of the 28th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–7. IEEE, 2019. 2, 126, 144

Such, J. M., Espinosa, A., and García-Fornes, A. A survey of privacy in multi-agent systems. *The Knowledge Engineering Review*, 29(3):314–344, 2014. 38

Sukthankar, G., Geib, C., Bui, H. H., Pynadath, D. V., and Goldman, R. P. *Plan, Activity, and Intent Recognition*. Morgan Kaufmann, Boston, 2014. 85

SurveyMonkey. SurveyMonkey: Free online survey software & questionnaire tool, 1999. URL https://www.surveymonkey.com/. 118

Tabrez, A., Luebbers, M. B., and Hayes, B. A survey of mental modeling techniques in human–robot teaming. *Current Robotics Reports*, 1(4):259–267, 2020. 157

Talamadupula, K., Briggs, G., Chakraborti, T., Scheutz, M., and Kambhampati, S. Coordination in human-robot teams using mental modeling and plan recognition. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2957–2962. IEEE, 2014. 2, 80, 85, 157

Thellman, S., de Graaf, M., and Ziemke, T. Mental state attribution to robots: A systematic review of conceptions, methods, and findings. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(4):1–51, 2022. 142

Trafton, J. G., Cassimatis, N. L., Bugajska, M. D., Brock, D. P., Mintz, F. E., and Schultz, A. C. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 35(4):460–470, 2005. 2, 157

Ullman, T. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*, 2023. 4, 169

Van der Hoek, W. and Meyer, J.-J. C. Graded modalities in epistemic logic. In *International Symposium on Logical Foundations of Computer Science*, pp. 503–514. Springer, 1992. 40

Van Der Hoek, W. and Wooldridge, M. Tractable multiagent planning for epistemic goals. In *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pp. 1167–1174, 2002. 16

Van-Horenbeke, F. A. and Peer, A. Activity, plan, and goal recognition: A review. *Frontiers in Robotics and AI*, 8:643010, 2021. 85

Vardi, M. Y. On epistemic logic and logical omniscience. In *Theoretical aspects of reasoning about knowledge*, pp. 293–305. Elsevier, 1986. 41

Vasileiou, S. L. and Yeoh, W. On generating abstract explanations via knowledge forgetting. In *ICAPS 2022 Workshop on Explainable AI Planning*, 2022. 50

Vasileiou, S. L., Previti, A., and Yeoh, W. On exploiting hitting sets for model reconciliation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 6514–6521, 2021a. 50, 122

Vasileiou, S. L., Yeoh, W., Son, T. C., and Previti, A. Explanations as model reconciliation via probabilistic logical reasoning. In *KR 2021 Workshop on Explainable Logic-Based Knowledge Representation (XLoKR)*, 2021b. 122

Vasileiou, S. L., Yeoh, W., Son, T. C., Kumar, A., Cashmore, M., and Magazzeni, D. A logic-based explanation generation framework for classical and hybrid planning problems. *Journal of Artificial Intelligence Research*, 73:1473–1534, 2022. 50, 122

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Proceedings of the 30th Conference on Advances in Neural Information Processing Systems (NIPS)*, 2017. 4, 168

Vered, M., Pereira, R. F., Magnaguagno, M. C., Kaminka, G. A., and Meneguzzi, F. Towards online goal recognition combining goal mirroring and landmarks. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 2112–2114, 2018. 83

Vilain, M. B. Getting serious about parsing plans: A grammatical analysis of plan recognition. In *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI)*, pp. 190–197, 1990. 52, 84

Von Wright, G. H. An essay in modal logic. 1951. 14

Waldinger, R. Achieving several goals simultaneously. In *Machine Intelligence 8*, pp. 94–136. Ellis Horwood, 1977. 90

Wan, H., Fang, B., and Liu, Y. A general multi-agent epistemic planner based on higher-order belief change. *Artificial Intelligence*, 301:103562, 2021. doi: https://doi.org/10.1016/j.artint.2021.103562. 4, 17

Warneken, F. and Tomasello, M. Altruistic helping in human infants and young chimpanzees. *science*, 311(5765):1301–1303, 2006. 1

Weiner, J. Blah, a system which explains its reasoning. *Artificial Intelligence*, 15 (1-2):19–48, 1980. 46

Westberg, M., Zelvelder, A., and Najjar, A. A historical perspective on cognitive science and its influence on XAI research. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pp. 205–219. Springer, 2019. 47

Williams, J., Fiore, S. M., and Jentsch, F. Supporting artificial social intelligence with Theory of Mind. *Frontiers in Artificial Intelligence*, 5, 2022. 2

Williams, M.-A., McCarthy, J., Gärdenfors, P., Stanton, C., and Karol, A. A grounding framework. *Autonomous Agents and Multi-Agent Systems*, 19(3):272–296, 2009. 129

Wimmer, H. and Perner, J. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition*, 13(1):103–128, 1983. 1

Wu, H., Sequeira, P., and Pynadath, D. Multiagent inverse reinforcement learning via Theory of Mind reasoning. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023a. 2

Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., and Rehg, J. M. A scalable approach to activity recognition based on object use. In *2007 IEEE 11th international conference on computer vision*, pp. 1–8, 2007. 208

Wu, P., Luo, S., Tian, L., Mao, B., et al. Consistent epistemic planning without communication for madrl. 2023b. 17

Yu, C., Serhan, B., Romeo, M., and Cangelosi, A. Robot Theory of Mind with reverse psychology. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 545–547, 2023. 157

Zhang, X., Luo, H., Fan, X., Xiang, W., Sun, Y., Xiao, Q., Jiang, W., Zhang, C., and Sun, J. AlignedReID: Surpassing human-level performance in person re-identification, 2017. 140

Zhang, Y., Narayanan, V., Chakraborti, T., and Kambhampati, S. A human factors analysis of proactive support in human-robot teaming. In *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3586–3593. IEEE, 2015. 160

Zhao, Y., Holtzen, S., Gao, T., and Zhu, S.-C. Represent and infer human Theory of Mind for human-robot interaction. In *AAAI Fall Symposium*, 2015. 2, 126, 157