Multiagent Epistemic Cooperation-Agnostic Planning

A dissertation

submitted by

David Buckingham

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

TUFTS UNIVERSITY

February 2023

ADVISOR: Prof. Matthias Scheutz

In Epistemic Planning, an automated system decides what goal-oriented actions to perform in a task-environment where the mental perspectives (beliefs and knowledge) of multiple agents are of prime importance. This work presents three contributions to Epistemic Planning. First, three action languages improve upon prior languages to expand the class of planning domains that can be expressed with a natural-language-like syntax. Second, a new planning algorithm generalizes prior techniques by relaxing assumptions about how states and actions are represented. This approach also incorporates the predicted behavior of independent agents (beyond the planner's control), allowing the flexibility to plan for cooperative, competitive, or neutral (self-interested) multi-agent domains. Finally, we present and evaluate the Multi-Agent Cooperation-Agnostic Planner, which implements these methods.

For Ester, who never gave up.

Acknowledgments

This work was in part funded by AFOSR grant number FA9550-18-1-0465 and NASA grant number C17-2D00-TU. Matthias Scheutz has been a great advisor, combining helpful direction with support for exploration. Son Tran has collaborated in this work by suggesting and developing core concepts and by providing invaluable encouragement. Many current and former members of the HRI Lab have contributed helpful suggestions, discussion, and support. In particular, Daniel Kasenberg and Meia Chita-Tegmark have co-authored papers that are foundational to this dissertation, and Vasanth Sarathy has helped shape this work through fruitful discussions. The feedback of many anonymous reviewers has been very helpful. Sasha Scott has proofread all of my writing to improve its clarity and effectiveness.

DAVID BUCKINGHAM

TUFTS UNIVERSITY

February 2023

Contents

List of Tables

List of Figures

Chapter 1

Preface

This dissertation presents new planning techniques for robots acting in social settings involving humans and other agents. When a robot deliberates about what actions it should perform in order to achieve some goal, it is doing Automated Planning. This work is especially interested in Automated Planning that also involves other agents not controlled by the planner, and where belief, knowledge, and uncertainty, instead of just objective material facts, are of prime importance. Consider the following scenario. A robot working in a grocery store detects a dangerous slippery spill in aisle one, while a distracted shopper, unaware of the spill, walks towards the hazard. The robot knows there is a spill, but it also knows that the human thinks it is safe to proceed! The robot is programmed with goals including that it must not "through inaction, allow a human being to come to harm" (Asimov, 1950), but it cannot simply tell the human about the danger: its vocal subprocessor is damaged, and the human is wearing headphones. The robot must act. Disregarding its own well-being, it intentionally slips on the spill and falls. The human, seeing the robot fall, becomes aware of the spill, and avoids the hazard.

This scenario involves a robot modeling the mental state of a human: the robot knows about the spill, and it also knows that the human believes that there is no spill. It also involves false beliefs and *belief revision*: the human believes (wrongly) that it is safe to proceed, and then revises that belief in light of new information. Finally, it involves planning the actions of one agent (the robot) while considering also the expected behavior of another agent (the human) that the planner does not control, i.e. predictive behavior modeling. This

is an Epistemic Planning (EP) problem, a sub-field of Automated Planning that considers not only the objective state of the task environment (e.g. there is a spill so I should clean it) but also the beliefs and knowledge of multiple agents (e.g. someone doesn't know about the spill, so I should inform them). Note that while "epistemic" refers to knowledge and "doxastic" refers to belief, this work follows (Bolander et al., 2020) to use Epistemic Planning to refer generally to multi-agent planning involving epistemic and/or doxastic states, actions, and goals.

Existing EP techniques are insufficient to solve the grocery store task, either because they do not incorporate the behavior of agents acting independently of the planner, or because they do not represent false beliefs. Parallel to the task of developing more powerful EP strategies is the development of formal languages for describing epistemic planning problems. *Epistemic action languages* are used to define actions for EP, for example, to specify that if the robot slips on a spill then anyone observing the robot will learn about the spill. However, existing languages suffer from limitations that exclude important classes of problems, such as those involving *second-order* false beliefs, i.e. incorrect beliefs about someone else's beliefs.

This work offers three contributions which help develop EP as a tool for artificial agents, especially robots, embedded in complex social scenarios such as those involving Human-Robot Interaction (HRI). First, three action languages improve upon prior languages to expand the class of planning domains that can be expressed with a natural-language-like syntax. These action languages make it easier to define planning domains and open the possibility to use automated action learning techniques with EP. These languages introduce state and action representations each with their own strengths and limitations. Second, in order not to restrict EP to a single formalism for describing states or actions, a new epistemic planning algorithm generalizes prior techniques by relaxing assumptions about how states and actions should be represented. Furthermore, this algorithm incorporates predicted behavior of independent agents (beyond the planner's control) by querying oracular behavior prediction models without assuming cooperative, antagonistic, or other inter-agent relationships. These features allow a planner to adapt to the state or action representations used by an existing robotic system, to be modularly updated with domain-specific represen-

tations, and to make plans involving agents who act independently of the planner. Finally, we present the Multi-Agent Cooperation-Agnostic Planner (MECAP), which implements these methods to solve planning problems involving multiple agents, diverse inter-agent relationships, knowledge and uncertainty, beliefs, and belief revision.

Three published works form the foundation of this dissertation. First, Buckingham and Scheutz, 2017 proposed the main premise of MECAP: a robot's goals may differ from those of other agents in a task environment, and the robot may need to plan to rely upon, and even influence, the actions of those agents without assuming intentional cooperation. Second, Buckingham, Chita-Tegmark, and Scheutz, 2020 reported an experimental study that evaluated this approach with human participants; a modified version appears in Chapter 7. Third, Buckingham, Kasenberg, and Scheutz, 2020 presented an early version of the action language m$\mathcal{A}$-revise (Chapter 4), which was also foundational to the development of m$\mathcal{A}$-local (Chapter 3) and m$\mathcal{B}$ (Chapter 5).

Chapter 2

Background and Related Work

2.1 Epistemic Planning

Epistemic Planning has emerged mainly from the intersection of three fields: Dynamic Epistemic Logic (DEL), Knowledge Representation and Reasoning (KRR), and Automated Planning (Baral, Bolander, et al., 2017). DEL (introduced in Baltag, Moss, and Solecki (1998) and Gerbrandy (1999)) extends epistemic modal logic with *event models* (also called "action models") that give a semantics for events that cause epistemic change. Section 2.6 gives a concise overview of DEL, while Bolander (2017) provides an accessible introduction, and Ditmarsch and Kooi (2008) and Van Ditmarsch, Der Hoek, and Kooi (2007) give a thorough treatment. KRR is a broad discipline that studies theories of knowledge, action, and change as a foundation for artificial intelligence (see M. Gelfond and Kahl (2014) for an introduction). Automated Planning studies computational techniques for finding plans. Ghallab, Nau, and Traverso (2004) gives a comprehensive study of the field, while Ghallab, Nau, and Traverso (2016) updates and extends that work to incorporate plan execution (acting). Epistemic Planning is a type of *planning under uncertainty*, or planning with incomplete information (Son and Baral, 2001), but extended to domains involving multiple agents. While *multi-agent planning* has often assumed shared knowledge among agents, in EP agents may have divergent, overlapping, or contradictory beliefs and knowledge, including nested beliefs and knowledge about each others' mental states.

Epistemic Planning generally falls under two main approaches: the *semantic ap-*

proach and the *syntactic approach* (Bolander et al., 2020). Both approaches involve *epistemic states* (which we sometimes simply call *states*). An epistemic state is a formal representation of the condition of the task environment, including the beliefs and knowledge of the agents involved, that is used by a planner to reason about how actions can lead to various possible futures in the pursuit of goals. In the semantic approach an epistemic state is a "possible worlds" relational model (a Kripke structure), a directed graph whose nodes are *worlds* and whose edges express agents' uncertainty or false beliefs about which world corresponds to reality. For example, an agent who is uncertain whether a coin lies heads-up (h) or tails-up ($\neg h$) might consider two worlds to be possible: one where h holds and one where $\neg h$ holds. These models compactly represent an infinite number of statements about nested belief and knowledge of multiple agents, and can be combined with action representations consisting of multiple "possible events" by means of a "product update" state transition system. This work follows the semantic approach to EP in order to exploit the representational power of relational models.

The syntactic approach represents an epistemic state as a finite set of formulas, exchanging some of the expressive power of relational models for computational efficiency and increased planning speed. For example, Muise, Belle, Felli, McIlraith, Miller, A. R. Pearce, et al. (2022) gives a system to compile epistemic planning problems into classical planning problems that can be solved (usually very quickly) by off-the-shelf classical planners. This system requires a pre-determined bound on the nesting depth of epistemic formulas (the size and time of the compilation grows exponentially with this depth limit), and does not allow disjunctive formulas (although Miller, Felli, et al. (2016) shows how to extend this technique to incorporate so-called "knowing whether" formulas). Another syntactic technique, Wan, Fang, and Liu (2021) transforms formulas to a special form (alternating cover disjunctive formulas) for use with the PrAO (To, Son, and Pontelli, 2011) search algorithm. That approach allows disjunctive formulas, but not common knowledge.

More recently, a compact semantics for epistemic logic has been proposed using syntactic belief bases (Lorini, 2020). It was then shown how that system can handle multi-agent belief revision and can be used for EP (Lorini and Schwarzentruber, 2021), and it has been extended to incorporate common belief (Lorini and Rapion, 2022). Belief revision in

this system is based on retaining maximally consistent sets of beliefs when new information is learned, and is significantly faster and simpler than belief revision with possible-worlds models. However, this approach has yet to incorporate semi-private forms of belief revision, where a single event imparts different information to different agents. Examples of such events can be found in the Second-Order Sally-Anne problem discussed in Chapter 3, the Bicycle problem in Chapter 4, and the Second-Order Coin problem in Chapter 5.

2.2 Planning Without Assuming Cooperation

Multi-agent planning usually involves centralized planning for a team of agents with a joint task, or involves decentralized planning, with multiple planners coordinating their efforts to accomplish some shared goal (Torreño et al., 2017). However, a planner for an automated agent immersed in a social context should accommodate other agents who act independently. In such planning domains, agents whose actions will be determined by the generated plan are called *system agents*, and those that act independently of the planner are called *environment agents* (Bowling, R. Jensen, and Veloso, 2006). These considerations apply to any task environment involving actors that the planner doesn't control, (e.g. animals, independent robots), but the present work is especially interested in Human-Robot Interaction (HRI), where the system agents are robots and the environment agents are humans. HRI planning research often assumes cooperative human-robot relationships, where a joint plan is computed to achieve shared goals (Chakraborti, Briggs, et al., 2015; Chakraborti and Kambhampati, 2018; Chakraborti, Kambhampati, et al., 2017; Chen et al., 2018; Devin and Alami, 2016; Görür et al., 2018; Milliez et al., 2016). Another approach is adversarial planning (Bowling, R. Jensen, and Veloso, 2006; Yang, Lancaster, and Smith, 2018), where system agents and environment agents have antagonistic goals. There has also been work on planners to handle both adversarial and cooperative environments (Kulkarni, Srivastava, and Kambhampati, 2018). However, many real-world interactions occur between *self-interested* agents, who are neither teammates nor opponents, but who hold individual goals and consider other actors only to the extent that they are relevant to those goals. The behavior of such bystanders may be highly relevant to a system agent: consider a robot that "catches a ride"

with a human (environment agent) on an elevator that the robot cannot operate.

As humans can do many things that robots cannot, a robot's ability to accomplish its goals often relies, in part, on human actions. Most approaches to incorporating human agents into a robot's plans assume human willingness to collaborate, and achieves this coordination by means of explicit communication (Nikolaidis, Kwon, et al., 2018); the robot can simply ask for help (Rosenthal and Veloso, 2012). For example, in Vega Magro et al. (2019) a robot asks humans blocking its way to move. An important question is whether humans will help robots as they tend to help other humans. This will likely depend on multiple factors, such as how busy a person is and how often the robot asks for help (Booth et al., 2017; Rosenthal, Veloso, and Dey, 2011a,b). On the other hand, a robot may be able to persuade a human to help: Sandoval, Brandstetter, and Bartneck (2016) explore the use of bribery by a robot to elicit desired behavior from a human. A robot's ability to influence human behavior depends on many factors, including trust (Chen et al., 2018; Nikolaidis, Hsu, and Srinivasa, 2017; Nikolaidis, Zhu, et al., 2017), the robot's perceived gender (Siegel, C. Breazeal, and Norton, 2009), level of authority (Saunderson and Nejat, 2021), and the use of nonverbal cues such as tone, gaze, gestures, and proximity (Chidambaram, Chiang, and Mutlu, 2012). A non-cooperative or hostile human may be completely unwilling to perform requested actions. Furthermore, a robot may be unable to request help for any number of reasons (the robot can't speak, the human can't hear the robot, the human and the robot don't speak the same language, etc.). To the extent that a robot can estimate human goals and propensities (e.g. because it has a mental model) and to the extent that it might be able to influence human actions, it may be able to generate plans that utilize humans (possibly unbeknownst to them) to achieve its task or to improve its task performance, including when the robot's goal is to help the humans (e.g. Chakraborti, Briggs, et al., 2015; Talamadupula et al., 2014).

By treating environment agents as self-interested, a plan can involve indirect and implicit communication by means of actions that sense and alter a shared task environment, e.g. opening the door for someone without explicitly talking about it. This allows the robot to plan for goals that require actions that it cannot perform, but that a human could, even without using explicit communication. In such cases, it may still be possible for the robot

to cause the human to perform the desired actions (Buckingham and Scheutz, 2017). For example, in Gray and Cynthia Breazeal (2014), a robot in a competitive context uses actions that change the task environment to alter the human's mental state. Chapter 7 will examine how robots can utilize the behavior of environment agents to accomplish their goals without explicit communication or intentional helping. Unlike motion planning for predicted human actions, (Dragan, 2017; Fisac et al., 2018; Sadigh et al., 2016), which relies on statistical methods, we are interested in symbolic task planning with predictions about higher-level human behavior.

Weerdt and Clement (2009) review some work involving planning for self-interested multi-agent contexts, such as Buzing et al. (2006), where agents share a joint problem but are unwilling to revise their individual plans. Bowling, R. Jensen, and Veloso (2006) emphasize that in multi-agent domains a robot's plans are contingent partly upon the goals of other agents, and that those goals will depend upon whether those agents are teammates, adversaries, or have overlapping goals. Brafman and Domshlak (2008) formalize the concept of coupling between agents in a multi-agent system, and Nissim, Brafman, and Domshlak (2010) develop a planning algorithm based on that work for agents that can decide to enter into coordinated, collaborative plans. In Brafman, Domshlak, et al. (2009), self-interested agents can form coalitions if they decide it is beneficial to do so, and Ben Larbi, Konieczny, and Marquis (2007) also consider situations where agents with separate goals would decide to cooperate, especially when their relationship constitutes a Nash equilibrium. Bowling, R. Jensen, and Veloso (2003) study the interactive dynamics of independently-planning and acting agents in several example scenarios, and Prunera (2017) offers a discussion of self-interested planning within a game-theoretic framework.

This work makes no assumptions about whether environment agents will be co-operative, adversarial, or otherwise. In fact, a single scenario could include environment agents having diverse relations to the system agents. Thus, we rely on access to predictor models, maps from states to sets of actions predicting what an environment agent will do. Given the diversity of human behavior, planning for the effects of all possible human actions in every considered state will be intractable for any non-trivial problem, especially with multiple humans. Predictive models, on the other hand, give a set of *likely* environment agent

actions. Thus, the model provides a planning robot with a Theory of Mind (Premack and Woodruff, 1978), serving as a proxy for the decision-making of others. MECAP anticipates that environment agents' actions will be among those predicted by the models, reducing the number of reachable states for the planner to consider.

Chakraborti, Kambhampati, et al. (2017) has emphasized the importance of mental modeling for deliberative processes in teaming tasks. Chakraborti and Kambhampati (2018) discuss the use of a mental model by a robot to manipulate or exploit humans in the interest of "the greater good." Görür et al. (2018) proposes a system where a robot anticipates a human's mental state and acts in a joint plan to help a human only if the human's intentions are relevant to the joint goal and the human actually wants the robot's assistance. Devin and Alami (2016) develop a theory of mind of the human and use it to improve plan execution (not generation), assuming that the human will perform actions specified by the robot's joint plan. Cirillo, Karlsson, and Saffiotti (2010) develop a planner that uses predictions of future human actions to constrain the robot's plan. Unlike that work, however, in which the human's plan is taken as immutable, MECAP allows the possibility for the robot's plan to influence the human's behavior.

2.3 Epistemic Action Languages

The input to an automated planner must include a formal description of the actions available to whoever will execute the generated plan. This usually includes a description of the action's preconditions (under what conditions can it occur) and effects (what does it "do"), but may involve other information such as duration (how long does it take) or cost (e.g. energy expended) Ghallab, Nau, and Traverso (2016). An action language provides a high-level natural-language-like syntax for such descriptions. Compared to a more low-level approach, this provides a succinct way to represent planning domains: the increased expressivity of an action language means that fewer actions need to be defined. For example, the Action Description Language $\mathcal{A}$ (Pednault, 1987) augments STRIPS planning (Fikes and Nilsson, 1971) with conditional effects. The languages $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are concisely described in M. Gelfond and Lifschitz (1998). While these early languages are for single-agent domains,

EP extends planning to domains involving multiple agents by accounting for the diverging, overlapping, and nested beliefs and knowledge of multiple agents.

Consider a *peek* action whereby an agent observes a concealed coin to determine whether it is heads-up (h) or tails-up ($\neg h$). In a single-agent domain, *peek* might have the effect that the agent learns whether h. With two agents, A and B, many more effects may be relevant: B knows that A knows whether h, A knows that B knows that A knows whether h, etc. Different agents might have different perspectives of an action. For A, *peek* might reveal whether h. However, if B cannot actually see the coin, but only knows that A can see it, then B may remain uncertain about whether h. Most work in the semantic approach to EP (but not all, e.g. Baral, G. Gelfond, Pontelli, et al. (2010), Buckingham, Kasenberg, and Scheutz (2020), and Fabiano et al. (2020)) address this diversity of perspectives on actions by representing actions with *event models* (defined in Section 2.6), whose structure is analogous to that of possible-worlds epistemic states: multiple "possible events" and relations between events express agents' uncertainty or false beliefs about what occurs. For example, the representation for the *peek* action could contain two events, one where it is revealed that h, and the other where it is revealed that $\neg h$: A knows which event actually occurs, but B does not.

There are several challenges and limitations with using event models for epistemic planning. First is their construction. Given an intuitive understanding of what some action should "do", how should an appropriate event model be constructed? Second is their interpretation: given an event model representation of an action, how can a user, or an automated system, determine (and communicate in plain language) what the action does? Relatedly, given an action representation and an actual attempt by an embodied agent to execute the action: how should an automated system evaluate whether the action achieved its intended effects? A third limitation of using event models to represent actions is that they can blur the distinction between action types and action tokens (Baral, Bolander, et al., 2017). Action types are abstract and repeatable, while action tokens (or occurrences) are concrete unique instances of action types (Herzig, Lorini, et al., 2010). An event model is an action type (e.g. the action of peeking at a coin), but also encodes information belonging to an action token (agent A peeks at a coin while B observes her doing so and C is unaware of

the action). Epistemic action languages with model-based semantics address these issues by defining actions with natural-language-like syntax. This makes it explicit what an action is intended to do and creates the potential to bring existing techniques for automated action learning, such as with verbal instruction (Scheutz, Krause, et al., 2017), to EP. Furthermore, an action language can define action types, which can then be instantiated into action tokens. Augmenting event models with *edge conditions* (Bolander, 2018) also achieves this distinction between types and tokens, but does not provide the other advantages of an action language.

Baral, Son, and Pontelli (2010), Son, Pontelli, and Sakama (2009), and Son and Sakama (2010) investigated extending existing action languages ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$) to multi-agent domains. Having found those languages to be inadequate for multi-agent domains with nested knowledge, those authors proceeded to develop the epistemic action language m$\mathcal{A}^*$ (Baral, G. Gelfond, Pontelli, et al., 2022). An early version focused on multi-agent knowledge representation (Baral, G. Gelfond, Son, et al., 2010). The next version included static observability specifications (Baral, G. Gelfond, Pontelli, et al., 2010). While these earlier versions defined state transitions directly from action definitions, Baral and G. Gelfond (2011) showed how to compile action definitions (types) into event models (tokens). Then Baral, G. Gelfond, Pontelli, et al. (2015) added support for belief revision. Thus, m$\mathcal{A}^*$ is the first comprehensive action language for multi-agent epistemic planning (Fabiano et al., 2020). An extension to STRIPS called MA-STRIPS (Brafman and Domshlak, 2008) focused on developing plans for multiple agents to coordinate on achieving a goal. However, it does not consider nested knowledge and beliefs between agents. Another system, GDL-III (Engesser, Mattmüller, et al., 2018; Thielscher, 2017) differs form m$\mathcal{A}^*$ in that it requires the addition of extra variables to define nested knowledge. Furthermore, in GDL-III agents receive updates after each action with true information about the state of the system, while in m$\mathcal{A}^*$ perceptions are determined by observability statements. Thus GDL-III cannot describe actions causing an agent to have false beliefs. Unlike m$\mathcal{A}^*$, the epistemic action language DER (Rajaratnam and Thielscher, 2021) is complete with respect to event models (every model can be described with DER). It also features a simpler syntax that unites ontic and epistemic effects, at the cost of increased complexity of the underlying machinery.

However, DER is strictly epistemic: it does not handle (possibly false) belief. The language m$\mathcal{A}^{\rho}$ (Fabiano et al., 2020) maintains the syntax of m$\mathcal{A}^{*}$, but changes the underlying state representation from Kripke structures to "possibilities", non-well-founded sets that give a bisimulation-minimal representation of Kripke structures. Finally, Buckingham, Kasenberg, and Scheutz (2020) gives an action-language-based system for dynamic action observability and belief revision that employs *ad hoc* state transformation rules instead of event model semantics.

The action languages presented in Chapters 3, 4, and 5 offer improvements upon this prior work. While m$\mathcal{A}^{*}$ models action observability as well as uncertainty and false beliefs, it does not handle uncertainty and false beliefs *about* action observability. Chapter 3 presents m$\mathcal{A}$-local, which augments m$\mathcal{A}^{*}$ to allow, for example, situations where one agent does not know whether another agent observes some action. When an agent has a false belief and then receives new information correcting that belief (belief revision), m$\mathcal{A}^{*}$ causes the agent to correct even unrelated uncertainties or false beliefs. Chapter 4 presents m$\mathcal{A}$-revise, which employs a less drastic belief revision technique. Chapter 5 presents m$\mathcal{B}$, which uses an improved semantics (plausibility preorders) to support belief revision.

2.4 Belief Revision

Belief revision is the process whereby an agent's beliefs change because she learns something that contradicts her prior beliefs. The standard approach is AGM theory (Alchourrón, Gärdenfors, and Makinson, 1985), which gives postulates specifying how the agent should incorporate the new information while retaining as many of her prior beliefs as possible without being inconsistent. While AGM assumes that facts are static while beliefs change, KM theory (Katsuno and Mendelzon, 1992) distinguishes belief revision from belief update, where belief change happens in a dynamic setting. While these works provide general principles about what should be the outcome of belief revision (or update), developments in modal logic have attempted to apply these concepts to formal epistemic models. Benthem and Smets (2015) gives a historical overview of the development of several modal logics to handle belief change. For example, Propositional Dynamic Logic (Fischer and Ladner,

1979) gives modal operators to express that some formula will be true after an action takes place. While AGM does not consider higher-order belief change (beliefs about beliefs), DEL does. Public Announcement Logic (Ditmarsch, Hoek, and Kooi, 2004) relates DEL to AGM with modal operators to express public announcements, i.e. new information available to all agents. Quantitative belief revision systems, e.g. where agents have varying degrees of belief (Ditmarsch and Labuschagne, 2003) or where epistemic states are Spohn ranking models (Aucher, 2003; Spohn, 2012), have lead to the development of a purely quantitative approach employing plausibility preorders (Baltag and Smets, 2008; Benthem and Smets, 2015; Ditmarsch, 2005), which will be discussed in Chapter 5. In contrast, some syntactic approaches to epistemic planning handle belief revision by updating epistemic theories to incorporate new information (X. Huang et al., 2017; Miller and Muise, 2016).

2.5 Epistemic Logic

Assume a finite set of atomic propositional variables $\mathcal{P}$ called *propositions*. A *propositional literal* is either p or $\neg p$, where $p \in \mathcal{P}$. $\mathcal{L}^{\mathcal{P}}$ is the language of logical formulas over $\mathcal{P}$ built with the usual Boolean connectives, defined with the BNF:

$$\varphi \coloneqq p|\neg\varphi|(\varphi \wedge \varphi)$$

where $p \in \mathcal{P}$. A *propositional formula* is a formula in $\mathcal{L}^{\mathcal{P}}$.

Given a finite set of names $\mathcal{G}$ called *agents*, the language $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$ augments $\mathcal{L}^{\mathcal{P}}$ with a modal operator $\Box_i$ for each agent $i \in \mathcal{G}$, and is defined with the BNF:

$$\varphi \coloneqq p|\neg\varphi|(\varphi \wedge \varphi)|\Box_i\varphi|\mathbf{C}_g\varphi$$

where $p \in \mathcal{P}$, $i \in \mathcal{G}$, and $g \subseteq \mathcal{G}$. The modal operator $\Box_i$ indicates i's belief or knowledge, e.g. $\Box_i p$ means that "agent i knows (or believes) that p". The modal operator $\mathbf{C}_g$ indicates common knowledge among agents g.

The language $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$ uses two modal operators for each agent, $\mathbf{B}_i$ and $\mathbf{K}_i$, to express

belief and knowledge, respectively. It is defined with the following BNF:

$$\varphi \coloneqq p|\neg\varphi|(\varphi \wedge \varphi)|\mathbf{B}_i\varphi|\mathbf{K}_i\varphi|\mathbf{C}_g\varphi$$

where $p \in \mathcal{P}$, $i \in \mathcal{G}$, and $g \subseteq \mathcal{G}$. The modal operator $\mathbf{B}_i$ indicates i's belief, and $\mathbf{K}_i$ indicates i's knowledge, e.g. $\mathbf{B}_i(\mathbf{K}_j p)$ means that "agent i knows that agent j believes that p". The subscript of operators $\Box_i$, $\mathbf{B}_i$, and $\mathbf{K}_i$ is omitted when $|\mathcal{G}| = 1$, and the subscript of C_g is omitted when $g = \mathcal{G}$. As usual, let $\varphi \vee \psi \coloneqq \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \to \psi \coloneqq \neg\varphi \vee \psi$, $\bot \coloneqq p \wedge \neg p$, and $\top \coloneqq \neg\bot$.

2.6 Review of Dynamic Epistemic Logic

Dynamic Epistemic Logic (DEL) is the foundational technique for the semantic approach to EP. It defines transitions between states represented as Kripke structures by means of event models (also called "update models"). Kripke structures represent both the material condition of the task environment and agents' beliefs, including their uncertainties and false beliefs (Fagin et al., 1995). Similarly, event models capture beliefs, knowledge, and uncertainty about events that occur, and model the interactions between agents' mental states when events occur. Here we review the elements of DEL necessary for the present work, adopting much of the notation of Baral, G. Gelfond, Pontelli, et al. (2022).

A Kripke structure is a tuple $\langle W, V, R_1, \ldots, R_n\rangle$, where W is a set of *worlds*, $V : W \to 2^{\mathcal{P}}$ assigns a *valuation* over $\mathcal{P}$ to each world, and for $1 \leq i \leq n$, $R_i \subseteq W \times W$ is a binary relation over W. A *state* is a pair $\langle\langle W, V, R_1, \ldots, R_n\rangle, d\rangle$ where $d \in W$ is the *designated world*, whose valuation gives the "actual" value of each proposition. A state $\langle M, s\rangle$, where $M = \langle W, V, R_1, \ldots R_n\rangle$ and $s \in W$, models formulas in $\mathcal{L}^{\mathcal{P}}$ or $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$ as follows:

$\langle M, u\rangle \models p$ iff $p \in V(u)$

$\langle M, u\rangle \models \neg\varphi$ iff $\langle M, u\rangle \not\models \varphi$

$\langle M, u\rangle \models \varphi \wedge \psi$ iff $\langle M, u\rangle \models \varphi$ and $\langle M, u\rangle \models \psi$

$\langle M, u\rangle \models \Box_i\varphi$ iff for all $v \in W$, if $\langle u, v\rangle \in R_i$ then $\langle M, v\rangle \models \varphi$

$\langle M, u\rangle \models \mathbf{C}_g\varphi$ iff for all $v \in W$, if $u(\cup_{i\in g} R_i)^* v$ then $\langle M, v\rangle \models \varphi$

where $p \in \mathcal{P}$, $i \in \mathcal{G}$, $g \subseteq \mathcal{G}$, $\varphi, \psi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$, and $(R)^*$ is the transitive closure of R. If $\varphi \in \mathcal{L}^{\mathcal{P}}$ then we may also write $u \models \varphi$ iff $\langle M, u\rangle \models \varphi$ for some Kripke structure M (since the semantics involves only the valuation at u).

An $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$*-substitution* is a set $\{p_1 \leftarrow \varphi_1, \ldots, p_k \leftarrow \varphi_k\}$, where each p_i is a distinct proposition in $\mathcal{P}$ and each φ_i is in $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$. $SUB_{\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}}$ denotes the set of all $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$-substitutions. An *event model* is a tuple $\Sigma = \langle E, R^{\Sigma}_1, \ldots, R^{\Sigma}_n, pre, sub\rangle$, where E is a set of *events*, for $1 \leq i \leq n$, $R^{\Sigma}_i \subseteq E \times E$ is a binary relation over E, $pre : E \to \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$ is a function mapping each event to a formula (the event's preconditions), and $sub : E \to SUB_{\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}}$ is a function mapping each event to a substitution (ontic effects).

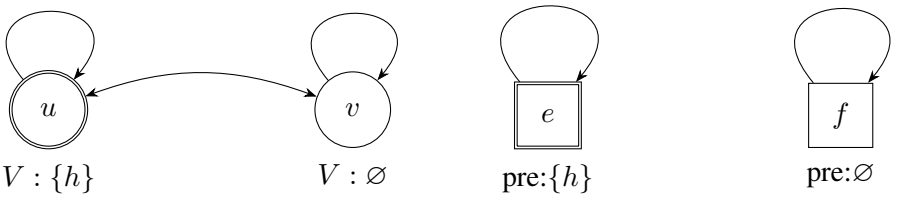
Given Kripke structure $M = \langle W, V, R_1, \ldots, R_n\rangle$ and event model $\Sigma = \langle E, R^{\Sigma}_1, \ldots, R^{\Sigma}_n, pre, sub\rangle$, the *product update* induced by Σ in M defines a new Kripke structure $M \times \Sigma \coloneqq \langle W', V', R'_1, \ldots, R'_n\rangle$ where

$$W' = \{\langle u, e\rangle \mid u \in W, e \in E, \langle M, u\rangle \models pre(e)\}, \tag{2.1}$$

$$\langle\langle u, e\rangle, \langle v, f\rangle\rangle \in R'_i \text{ iff } \langle u, e\rangle, \langle v, f\rangle \in W', \langle u, v\rangle \in R_i, \text{ and } \langle e, f\rangle \in R^{\Sigma}_i, \tag{2.2}$$

and for all $\langle u, e\rangle \in W'$ and $p \in \mathcal{P}$, $p \in V'(\langle u, e\rangle)$ iff

$$(p \in V(u) \text{ and for every } p' \leftarrow \varphi \in sub(e), p \neq p') \text{ or}$$
$$(p \leftarrow \varphi \in sub(e) \text{ and } \langle M, u\rangle \models \varphi). \tag{2.3}$$



(a) The example starting state $\langle M, u\rangle$.

(b) Example event template for *peek* $(\Sigma, \{e\})$.

(c) The state update $\langle M, u\rangle \times (\Sigma, \{e\})$

Figure 2.1: DEL example.

An *event template* is a pair $\langle \Sigma, \Gamma\rangle$, where $\Sigma = \langle E, R_1^\Sigma, \ldots, R_n^\Sigma, pre, sub\rangle$ is an event model and $\Gamma \subseteq E$ are the *designated events*. An event template $\langle \Sigma, \Gamma\rangle$ applied in a state $\langle M, d\rangle$ causes a *state update* resulting in a set of new states, $\langle M, d\rangle \times \langle \Sigma, \Gamma\rangle \coloneqq \{(M \times \Sigma, (d, e)) \mid e \in \Gamma, \langle M, d\rangle \models pre(e)\}$. Intuitively, $\langle M, d\rangle \times \langle \Sigma, \Gamma\rangle$ is the result of executing (Σ, Γ) in $\langle M, d\rangle$. For deterministic actions, this set of new states will be a singleton because there will only be one event in Γ whose precondition is satisfied at d.

Consider the single-agent example shown in Figure 2.1. Subfigure (a) shows a state, (b) shows an action, and (c) shows the new state resulting from the action occurring. The worlds of epistemic states are shown as circles. The events of the event model are squares. Double borders indicate designated worlds and events. Directed arrows indicate the epistemic relation, e.g. an arrow connecting world u to world v indicates that R contains (u, v). Each world and event is labeled with its name. Each world is also labeled with the valuation function V, listing atoms that are true in that world. Each event is labeled with its preconditions. For example, $\langle M, u\rangle = \langle\langle\{u, v\}, \{\langle u, \{h\}\rangle, \langle v, \varnothing\rangle\}, \{\langle u, u\rangle, \langle u, v\rangle, \langle v, u\rangle, \langle v, v\rangle\}\rangle, u\rangle$. In this starting state, a coin lies heads up (h), but the agent doesn't know whether it is heads up or tails up. Thus, $\langle M, u\rangle \models h \wedge \neg\Box h \wedge \neg\Box\neg h$. The event template $\langle \Sigma, \{e\}\rangle$ represents a *peek* action whereby the agent observes the coin. Applying the action results in a state where the agent knows how the coin lies: $\langle M, u\rangle \times \langle \Sigma, \{e\}\rangle \models h \wedge \Box h$.

2.7 Review of m$\mathcal{A}^*$

The m$\mathcal{A}^*$action language (Baral, G. Gelfond, Pontelli, et al., 2022) defines actions in natural-language-like syntax and gives semantics to compile action definitions into event models. Each action defined by m$\mathcal{A}^*$ is either an ontic action, a sensing action, or an announcement action. A strength of m$\mathcal{A}^*$is that it defines agents' observability of actions dynamically, i.e. as conditioned by formulas. Three tiers of observers are supported: full observers who know an action occurs, oblivious agents, who know nothing about the action, and an intermediate class, partial observers, applicable only for sensing and announcement actions. Partial observers know *that* full observers learn the values of sensed and announced propositions, but do not themselves learn those values.

Given a finite set of action names A and a finite set of agent names $\mathcal{G}$, The m$\mathcal{A}^*$ language consists of statements of the following forms:

1. "executable a if ψ"
2. "a causes l if ψ"
3. "a determines φ"
4. "a announces φ"
5. "i observes a if φ"
6. "i aware_of a if φ"

where $a \in A$, $\psi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$, $\varphi \in \mathcal{L}^{\mathcal{P}}$, l is a propositional literal, and $i \in \mathcal{G}$. A *theory* $\mathcal{T}$ is a finite collection of such statements that defines the actions of a planning domain. The first form means that a can occur only if ψ is true. When $\psi = \top$ the statement will be omitted. The second form means that if ψ is true, then a causes fluent p to become true if $l = p$, or false if $l = \neg p$. When $\psi = \top$ the statement is written without the trivial condition. The third form means that a is a sensing action, causing full observers to learn the value of φ. The fourth form means that a is an announcement action, causing full observers to learn that φ (false announcements such as lies are not allowed). Statements of the fifth form specify that i is a

full observer of a if φ is true. Statements of the sixth form specify that i is a partial observer of a if φ is true. An agent that is neither a full observer nor a partial observer is oblivious.

It is assumed that an action theory $\mathcal{T}$ is consistent with respect to every Kripke structure $M = \langle W, V, R_1, \ldots, R_n \rangle$, i.e. contains no two statements specifying contradictory effects or observability. Thus, for every world $u \in W$ and every pair of statements "a causes p if ψ" and "a causes $\neg p$ if ψ'" in $\mathcal{T}$, $\langle M, u \rangle \not\models \psi \wedge \psi'$, and for every pair of statements "i observes a if φ" and "i aware_of a if φ'" in $\mathcal{T}$, $\langle M, u \rangle \not\models \varphi \wedge \varphi'$. It is further assumed that each action is precisely either an ontic action, a sensing action, or an announcement action. Thus, for any a, $\mathcal{T}$ contains exclusively either "a causes l if ψ" (a is an ontic action) or "a determines φ" (a is a sensing action) or "a announces φ" (a is an announcement action). It is assumed that every action is associated with at most one statement of type 1, "executable a if ψ", and we will say that the *precondition* of a is ψ (or $\top$ if there is no such statement).

An event-model-based semantics takes an m$\mathcal{A}^*$ action theory, an action name, and a state and builds an event template which, when applied in that state, produces a new state expressing the effects of the action. Given state $\langle M, s \rangle$ and action a with precondition ψ defined in action theory $\mathcal{T}$, let

$$F = \{x \in \mathcal{G} \mid \text{"}x \text{ observes } a \text{ if } \varphi\text{"} \in \mathcal{T} \text{ such that } (M, s) \models \varphi\}, \tag{2.4}$$

$$P = \{x \in \mathcal{G} \mid \text{"}x \text{ aware\_of } a \text{ if } \varphi\text{"} \in \mathcal{T} \text{ such that } (M, s) \models \varphi\}, \tag{2.5}$$

$$O = \mathcal{G} \setminus (F_D(a, M, s) \cup P_D(a, M, s)). \tag{2.6}$$

Thus F are full observes of the action, P are partial observers, and O are oblivious agents. If a is an ontic action with precondition ψ then the induced update template is $\langle \langle \Sigma, R_1, \ldots, R_n, pre, sub \rangle, \{\theta\} \rangle$ where

- $\Sigma = \{\theta, \epsilon\}$;
- $R_i = \{\langle \theta, \theta \rangle, \langle \epsilon, \epsilon \rangle\}$ for $i \in F$ and $R_i = \{\langle \theta, \epsilon \rangle, \langle \epsilon, \epsilon \rangle\}$ for $i \in O$;
- $pre(\theta) = \psi$ and $pre(\epsilon) = \top$; and
- $sub(\epsilon) = \{p \leftarrow p \mid p \in \mathcal{P}\}$ and $sub(\theta) = \{p \leftarrow (\Psi^+(p, a) \vee (p \wedge \neg \Psi^-(p, a))) \mid p \in$

$\mathcal{P}\}$, where

$\Psi^{+}(p,a) \coloneqq \bigvee\{\varphi \mid$ "a causes p if $\varphi'' \in \mathcal{T}\}$

and

$\Psi^{-}(p,a) \coloneqq \bigvee\{\varphi \mid$ "a causes $\neg p$ if $\varphi'' \in \mathcal{T}\}$.

Thus for ontic actions m$\mathcal{A}^*$ constructs two events, θ and ϵ. The θ event corresponds to the perspective of observers, with a substitution function that implements the action's ontic effects. The ϵ event corresponds to the oblivious perspective, and has a trivial update function.

Sensing and announcement actions differ only in the designated events. If a is a sensing action then the update template is $\langle\langle \Sigma, R_1, \ldots, R_n, pre, sub\rangle, \{\theta, \tau\}\langle$ and if it is an announcement action the update template is $(\langle \Sigma, R_1, \ldots, R_n, pre, sub\rangle, \{\theta\})$ where

- $\Sigma = \{\theta, \tau, \epsilon\}$;
- $R_i = \{(\theta,\theta), (\tau,\tau), (\epsilon,\epsilon)\}$ for $i \in F$ and
- $R_i = \{(\theta,\theta), (\tau,\tau), (\epsilon,\epsilon), (\theta,\tau), (\tau,\theta)\}$ for $i \in P$ and
- $R_i = \{(\theta,\epsilon), (\tau,\epsilon), (\epsilon,\epsilon)\}$ for $i \in O$;
- $pre(\theta) = \psi \wedge \varphi$ and $pre(\tau) = \psi \wedge \neg\varphi$ and $pre(\epsilon) = \top$; and
- $sub(x) = \varnothing$ for each $x \in \Sigma$.

For sensing and announcement actions there are three events θ, τ, and ϵ. The θ and ϵ events operate as for ontic actions, representing the actual occurrence and the oblivious perspective, respectively. The τ event expresses the uncertainty of partial observers. For sensing actions that determine the value of φ, θ represents the case that φ is true, while τ represents the case that φ is false. Similarly, for announcement actions that announce proposition φ, τ represents the case that φ is false.

2.7.1 Belief Revision

Since m$\mathcal{A}^*$ involves beliefs (e.g. being oblivious to an action can cause an agent to have a false belief), it uses the symbol **B** for the modal operator instead of $\Box$. Suppose that

in some state $\langle M, u\rangle$ it is the case that φ and that agent i (wrongly) believes $\neg\varphi$, i.e. $\langle M, u\rangle \models \varphi \wedge \mathbf{B}_i\neg\varphi$. Also suppose that action a is either an announcement action that announces φ, or a sensing action that senses whether φ. If a occurs in $\langle M, u\rangle$ and i is a full observer, m$\mathcal{A}^*$generates an event template $\langle \Sigma, \Gamma\rangle$ such that $\langle M, u\rangle \times \langle \Sigma, \Gamma\rangle \models \mathbf{B}_i\bot$. That is, in the resulting state agent i's relation has no edges from the designated world u to any world, including to u, modeling that i trivially believes *any* formulas, even contradictions. To avoid this vacuous mental state and allow belief revision, m$\mathcal{A}^*$ utilizes a technique proposed by Eijck (2017) to define a state transition function Φ that maps from a state $\langle M, u\rangle$ and an event template $\langle \Sigma, \Gamma\rangle$ (generated from action a in $\langle M, u\rangle$) to a new state as follows. If a is a sensing or announcement action that announces or senses φ and $\langle M, u\rangle \models \varphi \wedge \mathbf{B}_i\neg\varphi$, then $\Phi(\langle M, u\rangle, \langle \Sigma, \Gamma\rangle) \coloneqq \langle M', u\rangle \times \langle \Sigma, \Gamma\rangle$, otherwise $\Phi(\langle M, u\rangle, \langle \Sigma, \Gamma\rangle) \coloneqq \langle M, u\rangle \times \langle \Sigma, \Gamma\rangle$. Model $\langle M', u\rangle$ is the same as $\langle M, u\rangle$ except that every edge $\langle u, v\rangle$ is removed from i's relation, and the reflexive edge $\langle u, u\rangle$ is added to i's relation. This removes all of i's false beliefs and uncertainties before the product update is computed.

2.7.2 Finitary $S5$ Theories

In addition to defining actions, m$\mathcal{A}^*$ includes a system (described in Son, Pontelli, Baral, et al. (2014)) to construct a Kripke structure from a restricted set of formulas called a *finitary $S5$ theory*. This is used to build the planning start state. A finitary $S5$ theory consists of statements of the following forms, where φ is a fluent formula: "**initially** φ" means that φ is true; "**initially C**φ" means that it is common knowledge that φ is true; "**initially C(B**$_i\varphi$**)**" means that agent i believes φ; "**initially C(B**$_i\varphi \vee \mathbf{B}_i\neg\varphi$**)**" means that agent i knows whether φ is true; "**initially C(**$\neg\mathbf{B}_i\varphi \wedge \neg\mathbf{B}_i\neg\varphi$**)**" means that agent i does not know whether φ is true or false. In the general case, a set of formulas (theory) can be satisfied by infinitely many models. A finitary $S5$ theory can be satisfied by only a finite number of models modulo bisimulation.

Chapter 3

m$\mathcal{A}$-local

This chapter presents our first action language, m$\mathcal{A}$-local, which gives an improved semantics to m$\mathcal{A}^*$. A central feature of m$\mathcal{A}^*$ is that action observability is dynamic: whether an agent observes an action is determined dynamically at execution time (encoded in the action tokens, not types), e.g. according to a fluent value that determines whether an agent is "watching". However, the observability status of each agent is computed globally, according to the actual state of affairs (i.e. at the designated world), and not locally at every world. Thus, it is only tracked whether an agent is an observer, not what an agent knows, does not know, believes, or does not believe another agent's observer status to be. As the authors explain, "...the current m$\mathcal{A}^*$ does not consider beliefs and knowledge of agents in the specification of observability statements..."(Baral, G. Gelfond, Pontelli, et al., 2022). For example, m$\mathcal{A}^*$ cannot capture a situation where agent A has an incorrect belief about whether agent B observes an action occurring, and where agent A thus develops a *second-order false belief* about agent B's beliefs.

The language m$\mathcal{A}$-local retains the syntax of m$\mathcal{A}^*$ but gives new semantics that represent agents' beliefs about each agent's ability to observe an action by moving the computation of observability to the level of individual worlds. With this improvement, it is possible to model second-order false-belief tasks and second-order uncertainty tasks where one agent does not know the truth about another agent's observations and resultant beliefs.

Consider the Second-Order Sally-Anne (SOSA) task (Braüner, Blackburn, and Polyanskaya, 2016). Sally and Anne are in a room containing a box and a basket. Sally

places a marble in a basket and leaves the room, but secretly watches the room without Anne knowing. Anne then takes the marble from the basket and places it in a box. When Sally returns, where does Anne expect Sally to look for the marble? Sally observed Anne moving the marble, and, therefore, knows that the marble is in the box. However, because Anne incorrectly believes that Sally did not observe her moving the marble, Anne now has the second-order false belief that Sally believes that the marble is in the basket. Thus Anne expects Sally to look in the basket.

As with many other second-order false-belief problems, such as the Second-Order Chocolate task (Arslan, Taatgen, and Verbrugge, 2013; Bolander, 2014; Flobbe et al., 2008), here the source of second-order false belief is false belief about whether an agent observes an event. Braüner, Blackburn, and Polyanskaya (2020) offers a deeper study of several second-order false belief tasks. We will present several more examples and apply m$\mathcal{A}$-local to them in Section 3.2.

3.1 Local Observability

m$\mathcal{A}$-local maintains the syntax of m$\mathcal{A}^*$, but extends its event model semantics by creating more events when there is uncertainty or false knowledge about agents' observation status. As with m$\mathcal{A}^*$ here we interpret the modal operator as belief and use **B** instead of $\Box$. Like m$\mathcal{A}^*$, we create a single ϵ-event. However, we create possibly multiple θ-events and τ-events corresponding to various ways of dividing agents into full observers, partial observers, and non-observers.

Consider action a defined by theory $\mathcal{T}$ applied in state $\langle M, s\rangle$ where $M = \langle W, V, R_1, \ldots, R_n\rangle$. Let F, P, and N map each world $u \in W$ to a subset of $\mathcal{G}$ (contrast this with equation 2.4).

$$\begin{aligned} F(u) :=& \{i \mid \exists \text{ “}i \text{ observes } a \text{ if } \varphi\text{”} \in \mathcal{T}.u \models \varphi\} \\ P(u) :=& \{i \mid \exists \text{ “}i \text{ aware\_of } a \text{ if } \varphi\text{”} \in \mathcal{T}.u \models \varphi\} \\ N(u) :=& \mathcal{G} \setminus F(u) \cup P(u) \end{aligned} \tag{3.1}$$

Note that $F(u) \cup P(u) \cup N(u) = \mathcal{G}$ and $F(u) \cap P(u) = F(u) \cap N(u) = P(u) \cap N(u) = \varnothing$. Let π partition $\mathcal{G}$ into subsets of fully observant, partially observant, and non-observant agents.

$$\pi(u) \coloneqq \langle F(u), P(u), N(u) \rangle$$

As with m$\mathcal{A}^*$, a partial observer to a sensing or announcement action knows *that* something has been sensed or announced, but not *what* was sensed or announced. We use the term "non-observant" to refer to agents who do not observe an action, while m$\mathcal{A}^*$ calls them "oblivious".

Let Ω map a partition of $\mathcal{G}$ to a formula that holds only in worlds where π produces that partition. The parameters to Ω are three sets of agents, called F, P, and N, where $F \cup P \cup N = \mathcal{G}$ and $F \cap P = F \cap N = P \cap N = \varnothing$.

$$\begin{aligned}\Omega(F,P,N) \coloneqq &\Big(\bigwedge_{i \in F} \bigvee_{\text{“}i \text{ observes } a \text{ if } \varphi\text{”} \in \mathcal{T}} \varphi \Big) \wedge \\ &\Big(\bigwedge_{i \in P} \bigvee_{\text{“}i \text{ aware\_of } a \text{ if } \varphi\text{”} \in \mathcal{T}} \varphi \Big) \wedge \\ &\Big(\bigwedge_{i \in N} \bigvee_{\text{“}i \text{ observes } a \text{ if } \varphi\text{”} \in \mathcal{T} \cup \text{“}i \text{ aware\_of } a \text{ if } \varphi\text{”} \in \mathcal{T}} \neg\varphi \Big)\end{aligned}$$

We have that $\Omega(F, P, N)$ holds in world u iff π generates $\langle F(u), P(u), N(u) \rangle$:

Lemma 3.1.1.

$$\forall u \in W. u \models \Omega(F, P, O) \textit{ iff } \pi(u) = \langle F, P, N \rangle.$$

The proof is in the Appendix (Section 9.1).

Let $F^i(M, u)$, $P^i(M, u)$, and $N^i(M, u)$ be the set of agents that agent i believes to

be full observers, partial observers, and non-observers, respectively, in state $\langle M, u\rangle$.

$$F^i(M,u) \coloneqq \{j \mid \exists \text{“}j \text{ observes } a \text{ if } \varphi\text{”} \in \mathcal{T}.\langle M,u\rangle \models \mathbf{B}_i\varphi\}$$
$$P^i(M,u) \coloneqq \{j \mid \exists \text{“}j \text{ aware\_of } a \text{ if } \varphi\text{”} \in \mathcal{T}.\langle M,u\rangle \models \mathbf{B}_i\varphi\}$$
$$N^i(M,u) \coloneqq \{j \in \mathcal{G} \mid \forall \text{“}j \text{ observes } a \text{ if } \varphi\text{”} \in \mathcal{T}.\langle M,u\rangle \models \mathbf{B}_i\neg\varphi,$$
$$\forall \text{“}j \text{ aware\_of } a \text{ if } \varphi\text{”} \in \mathcal{T}.\langle M,u\rangle \models \mathbf{B}_i\neg\varphi\}$$

Note that some agents might not be in any of these categories: $F^i(M,u) \cup P^i(M,u) \cup N^i(M,u) \subseteq \mathcal{G}$. However, if u is serial, i.e. $\exists v \in W.\langle u,v\rangle \in R_i$, then because of the requirement against contradictory observability statements in $\mathcal{T}$, these sets are non-intersecting: $F^i(M,u) \cap P^i(M,u) = F^i(M,u) \cap N^i(M,u) = P^i(M,u) \cap N^i(M,u) = \varnothing$.

We now define how an action applied in some state induces an event template. We will use partitions of agents as subscripts in event names to indicate that an event corresponds to a given partition into full, partial, and non-observers. For example, $e_{\pi(u)}$ names an event occurring where $F(u)$ are full observers, $P(u)$ are partial observers, and $N(u)$ are non-observers, where observability is defined at world u. For compactness, we omit the brackets around an agent partition when it subscripts an event name.

3.1.1 Sensing and Announcement Actions

Given theory $\mathcal{T}$, action a is a sensing action if $\mathcal{T}$ contains a statement of the form “a determines φ”, and is an announcement action if $\mathcal{T}$ contains a statement of the form “a announces φ”. Let a be a sensing or announcement action with precondition ψ. Following m$\mathcal{A}^*$ we assume that all announcements are truthful. Suppose that a is applied in state $\langle M, s\rangle$ where $M = \langle W, V, R_1, \ldots, R_n\rangle$, inducing event template $\langle \Sigma, \Gamma\rangle$ where $\Sigma = \langle E, R_1^\Sigma, \ldots, R_n^\Sigma, pre, sub\rangle$. The set of events is

$$E = \{\theta_{\pi(u)} \mid u \in W\} \cup \{\tau_{\pi(u)} \mid u \in W\} \cup \{\epsilon\}.$$

For sensing actions, the designated events are

$$\Gamma = \{\theta_{\pi(s)}, \tau_{\pi(s)}\}.$$

For announcement actions, the designated event is

$$\Gamma = \{\theta_{\pi(s)}\}.$$

Intuitively, each θ-event corresponds to the action occurring with a specific observability partition, each τ-event expresses partial observers' uncertainty or false beliefs about an observed or announced formula given a specific observability partition, and the ϵ-event represents the action not occurring.

The relations give full observers access from θ-events only to θ-events, and from τ-events only to τ-events. Partial observers also have access between θ-events and τ-events, expressing their uncertainty about what has been observed or announced. Non-observant agents access only the ϵ event. For each i in $\mathcal{G}$,

$$\begin{aligned}
R_i^{\Sigma} =& \{\langle \theta_{F,P,N}, \theta_{F,P,N} \rangle \mid i \in F \cup P\} \cup \\
& \{\langle \tau_{F,P,N}, \tau_{F,P,N} \rangle \mid i \in F \cup P\} \cup \\
& \{\langle \theta_{F,P,N}, \theta_{F',P',N'} \rangle \mid i \in F \cup P\} \cup \\
& \{\langle \tau_{F,P,N}, \tau_{F',P',N'} \rangle \mid i \in F \cup P\} \cup \\
& \{\langle \theta_{F,P,N}, \tau_{F,P,N} \rangle \mid i \in P\} \cup \\
& \{\langle \tau_{F,P,N}, \theta_{F,P,N} \rangle \mid i \in P\} \cup \\
& \{\langle \theta_{F,P,N}, \epsilon \rangle \mid i \in N\} \cup \\
& \{\langle \tau_{F,P,N}, \epsilon \rangle \mid i \in N\} \cup \\
& \{\langle \epsilon, \epsilon \rangle\}. \qquad (3.2)
\end{aligned}$$

The event preconditions bind θ-events to worlds where the sensed or announced propositional formula φ holds, τ-events to worlds where φ does not hold, and both θ- and τ- events to worlds where precondition ψ holds and that conform to their respective agent

observability partitions. For every θ-event $\theta_{F,P,N} \in E$,

$$pre(\theta_{F,P,N}) = \psi \wedge \varphi \wedge \Omega(F, P, N),$$

for every τ-event $\tau_{F,P,N} \in E$,

$$pre(\tau_{F,P,N}) = \psi \wedge \neg\varphi \wedge \Omega(F, P, N),$$

and

$$pre(\epsilon) = \top.$$

Since sensing and announcement actions have no ontic effects, for every event e in E,

$$sub(e) = \varnothing.$$

Proposition 3.1.2. *Given action theory $\mathcal{T}$, if $\mathcal{T}$ contains the statement "a determines φ" (a is a sensing action) and a is applied in state $\langle M, s\rangle$ resulting in state $\langle M', s'\rangle$, then it holds that*

1. *Full observers learn true sensed formulae: $\forall i \in F(s)$. if $s \models \varphi$ then $s' \models \boldsymbol{B}_i\varphi$;*
2. *Full observers learn false sensed formulae: $\forall i \in F(s)$. if $\langle M, s\rangle \models \neg\varphi$ then $\langle M', s'\rangle \models \boldsymbol{B}_i\neg\varphi$;*
3. *Non-observant agents are unaffected by sensing events: $\forall i \in N(s), \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}.\langle M', s'\rangle \models \boldsymbol{B}_i\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\omega$;*
4. *Full and partial observers learn that full observers learn sensed formulae: $\forall i \in F(s) \cup P(s), j \in F^i\langle M, s\rangle.\langle M', s'\rangle \models \boldsymbol{B}_i\big((\varphi \wedge \boldsymbol{B}_j\varphi) \vee (\neg\varphi \wedge \boldsymbol{B}_j\neg\varphi)\big)$;*
5. *Full and partial observers know that non-observant agents are unaffected: $\forall i \in F(s) \cup P(s), j \in N^i(M, s), \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}.\langle M', s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$.*

The proof is in the Appendix (Section 9.1).

Thus we have that full observers learn sensed information as specified by $\mathcal{T}$, that partial observers learn *that* full observers learn, but not *what* they learn, and that non-observant agents are oblivious to the sensed information and to changes in the beliefs of other agents.

Proposition 3.1.3. *If $\mathcal{T}$ contains the statement "a announces φ"(a is an announcement action), then it holds that*

1. *Full observers learn announcements: $\forall i \in F(s).\langle M', s'\rangle \models \boldsymbol{B}_i\varphi$,*
2. *Non-observant agents are unaffected by announcements: $\forall i \in N(s), \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}.\langle M', s'\rangle \models \boldsymbol{B}_i\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\omega$,*
3. *Full and partial observers learn that full observers learn announcements: $\forall i \in F(s) \cup P(s), j \in F^i\langle M, s\rangle.\langle M', s'\rangle \models \boldsymbol{B}_i\big((\varphi \wedge \boldsymbol{B}_j\varphi) \vee (\neg\varphi \wedge \boldsymbol{B}_j\neg\varphi)\big)$,*
4. *Full and partial observers know that non-observant agents are unaffected: $\forall i \in F(s) \cup P(s), j \in N^i\langle M, s\rangle, \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}.\langle M', s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$.*

The proof is in the Appendix (Section 9.1).

Thus we have that full observers learn sensed and announced information, partial observers learn *that* full observers learn, but not *what* they learn, and non-observant agents are oblivious to the announced information and to changes in the mental states of others.

3.1.2 Ontic Actions

Given theory $\mathcal{T}$, action a is an ontic action if $\mathcal{T}$ contains a statement of the form "a causes l if φ". Let a be an ontic action with precondition ψ. Suppose that a is applied in state $\langle M, s\rangle$ where $M = \langle W, V, R_1, \ldots, R_n\rangle$, inducing event template $\langle \Sigma, \Gamma\rangle$ where $\Sigma = \langle E, R_1^{\Sigma}, \ldots, R_n^{\Sigma}, pre, sub\rangle$. The set of events is

$$E = \{\theta_{\pi(u)} \mid u \in W\} \cup \{\epsilon\},$$

and the designated event is $\Gamma = \{\theta_{\pi(s)}\}$. Intuitively, each θ-event corresponds to the occurrence of the action given a specific observability partition, and the ϵ-event represents

the action not occurring. The event relations give observers access only to the θ-events, and give non-observant agents access only to the ϵ-event.

$$\begin{aligned} R_i^\Sigma = &\{\langle \theta_{F,P,N}, \theta_{F,P,N} \rangle \mid i \in F\} \cup \\ &\{\langle \theta_{F,P,N}, \theta_{F',P',N'} \rangle \mid i \in F\} \cup \\ &\{\langle \theta_{F,P,N}, \epsilon \rangle \mid i \in N\} \cup \\ &\{\langle \epsilon, \epsilon \rangle\} \end{aligned} \tag{3.3}$$

Event preconditions constrain events to occur only in worlds conforming to their observability partition. For every $\theta_{F,P,N} \in E$,

$$pre(\theta_{F,P,N}) = \psi \wedge \Omega(F,P,N)$$

and

$$pre(\epsilon) = \top.$$

The substitution function alters fluents according to ontic effects. For every $\theta_{F,P,N} \in E$,

$$sub(\theta_{F,P,N}) = \{p \leftarrow \Psi^+(p,a) \vee (p \wedge \neg\Psi^-(p,a)) \mid p \in \mathcal{F}\},$$

where

$$\Psi^+(p,a) = \bigvee_{\text{“}a \text{ causes } p \text{ if } \varphi\text{”} \in \mathcal{T}} \varphi,$$

and

$$\Psi^-(p,a) = \bigvee_{\text{“}a \text{ causes } \neg p \text{ if } \varphi\text{”} \in \mathcal{T}} \varphi.$$

For the ϵ event,

$$sub(\epsilon) = \varnothing.$$

That is, p will be true if an ontic effect makes it true, or if it was already true and no ontic effect makes it false.

Lemma 3.1.4. *The* sub *function for θ-events for ontic actions 1) causes proposition p to*

become true where formula φ holds if "a causes p if φ"$\in \mathcal{T}$, 2) causes p to become false where φ holds if "a causes $\neg p$ if φ"$\in \mathcal{T}$, and 3) preserves the value of p otherwise.

The proof is in the Appendix (Section 9.1).

Lemma 3.1.5. *For ontic actions, the application of the ϵ-event preserves the valuation of formulas. For every $\varphi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$ and $u \in S$, $\langle M, u\rangle \models \varphi$ iff $\langle M', \langle u, \epsilon\rangle\rangle \models \varphi$.*

The proof is in the Appendix (Section 9.1).

Proposition 3.1.6. *If $\mathcal{T}$ contains the statement "a causes l if φ" (a is an ontic action) and a is applied in state $\langle M, s\rangle$ resulting in state $\langle M', s'\rangle$, then it holds that*

1. *Ontic effects alter the state if their conditions are met: if $\langle M, s\rangle \models \varphi$ then $\langle M', s'\rangle \models l$,*
2. *Observers learn about effects they believe occur: for very $i \in F(s)$, if $\langle M, s\rangle \models \boldsymbol{B}_i\varphi$ then $\langle M', s'\rangle \models \boldsymbol{B}_i l$,*
3. *Non-observant agents are unaffected by ontic events: for every $i \in N(s)$ and $\omega \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$, $\langle M', s'\rangle \models \boldsymbol{B}_i\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\omega$,*
4. *Observers know that observers learn ontic effects: for every $i \in F(s)$ and $j \in F^i(M, s)$, if $\langle M, s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\varphi$ then $\langle M', s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j l$, and*
5. *Observers know that non-observant agents are unaffected: for every $i \in F(s)$ and $j \in N^i(M, s)$ and $\omega \in \mathcal{L}_{\mathcal{GB}}^{\mathcal{P}}$, $\langle M', s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$.*

The proof is in the Appendix (Section 9.1).

Thus we have that ontic effects conditionally alter the task environment as specified by $\mathcal{T}$, that observers learn about the action's effects upon the task environment and upon the mental states of other agents, and that non-observant agents do not.

3.2 Examples

This section applies m$\mathcal{A}$-local to four second-order false-belief and second-order uncertainty tasks. The scenarios we will present are the Second-Order Sally Anne problem (SOSA),

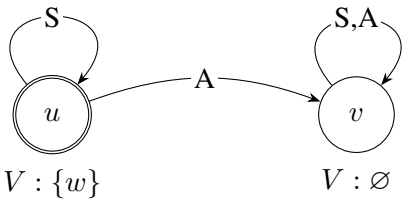


Figure 3.1: The Second-Order Sally-Anne start state.

the Second-Order Coin problem (Coin), the Loud Phonecall problem (Phone), and the Secret Distract problem (Distract). These problems are selected to vary several properties. First, *belief type*: whether an agent has false beliefs arising from false beliefs about another agent's observability status, or has uncertainty arising from uncertainty about another agent's observability status. Second, *actual status*: the second agent can be either a full observer, a partial observer, or non-observer. Third, *perceived status*: the first agent can wrongly believe that (or be uncertain about whether) the second agent is a full observer, a partial observer, or non-observant. Finally, we consider three *action types*: ontic actions alter the objective environment, sensing actions cause agents to learn truths through observation, and announcement actions convey information. Table 3.1 summarizes how our example tasks vary with respect to these properties.

task	belief type	actual status	perceived status	action type
SOSA	false-belief	full	non-observant	ontic
Coin	uncertainty	partial	non-observant	sensing
Phone	false-belief	full	partial	announcement
Distract	false-belief	non-observant	full	ontic

Table 3.1: Properties of example problems.

3.2.1 Second-Order Sally-Anne

Our demonstration of the the Second-Order Sally-Anne problem begins after Sally has left the room. Sally is secretly watching (w), and the marble is not yet in the box ($\neg x$). This start state is depicted in Figure 3.1. Each circle represents a world, the double border indicates the designated world, and arrows show the relation for each agent. World names are shown within the worlds, and each world is labeled with the propositions that hold according to the valuation function V. The marble is not in the box ($\neg x$) and Sally is watching (w). However,

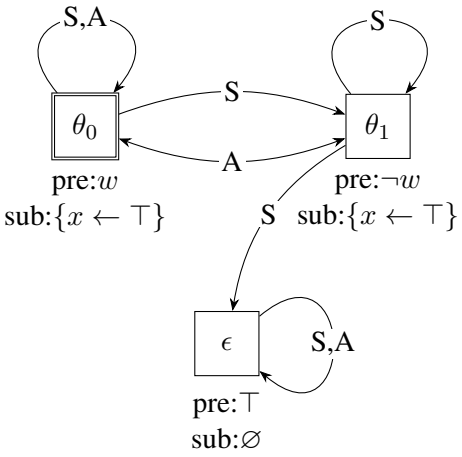


Figure 3.2: The event model induced by the application of the *move-marble* action. For compactness, the event subscripts are replaced with numerals as follows: $0 = \{S, A\}, \varnothing, \varnothing$; $1 = \{A\}, \varnothing, \{S\}$.

Anne believes that Sally is not watching ($\mathbf{B}_A\mathbf{B}_S\neg w$).

The *move-marble* action, whereby Anne moves the marble to the box, is defined in the following action theory:

- *move-marble* causes x
- A observes *move-marble*
- S observes *move-marble* if w

Figure 3.2 depicts the event model induced by *move-marble* in the start state. Each square represents an event, double borders indicate designated events, and arrows represent the event relation for each agent. Event names are shown within the events, and each event is labeled with values of the pre and sub functions.

Figure 3.3 shows the state resulting from the application of *move-marble*. The marble is in the box, and Sally and Anne both know it, but Anne believes that Sally believes that the marble is not in the box.

3.2.2 Second-Order Coin

In the Second-Order Coin problem, agents A and B know that a box contains a coin. Agent A performs the *peek* action, looking into the box and observing that the coin lies heads up.

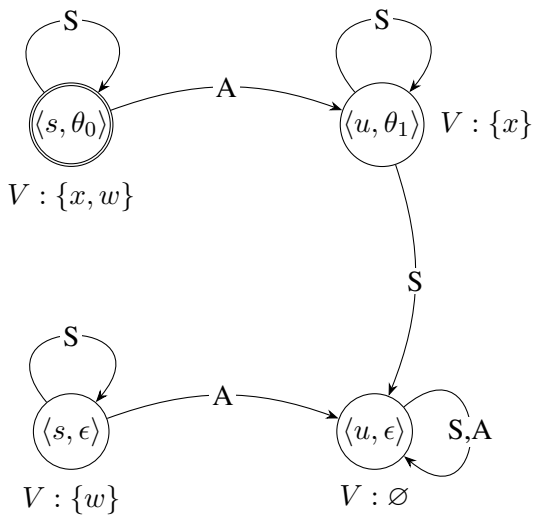


Figure 3.3: The state resulting from the *move-marble* action. For compactness, the event subscripts are replaced with numerals as follows: $0 = \{S, A\}, \varnothing, \varnothing$; $1 = \{A\}, \varnothing, \{S\}$.

Agent A does not know that B is watching A peek. Thus, B now knows that A knows how the coin lies, but B does not herself know how the coin lies. Agent A has second-order uncertainty: she does not know whether B was observing the *peek* action and therefore A does not know whether B knows that A knows how the coin lies.

Figure 3.4 shows the start state. A box contains a coin that lies heads up (h). Agent B is watching (w), but agent A does not know whether agent B is watching.

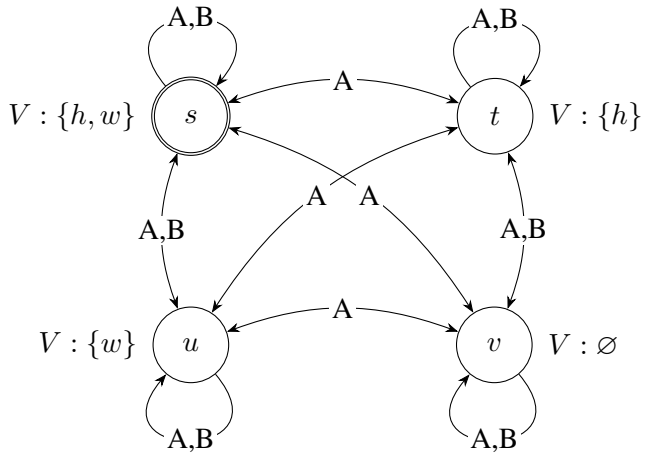


Figure 3.4: The start state for the Second-Order Coin problem.

The *peek* sensing action is defined with the following action theory:

- *peek* determines h

- A observes *peek*
- B aware_of *peek* if w

Figure 3.5 depicts the event model induced by the application of the *peek* sensing action in the Second-Order Coin problem start state. Agent B is a partial observer, but B doesn't know whether agent A is a partial observer or is non-observant.

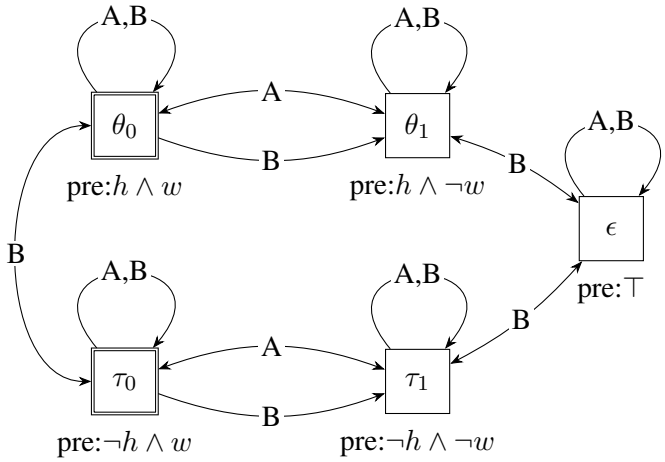


Figure 3.5: The event model induced by the *peek* sensing action in the Second-Order Coin start state. For compactness, the event subscripts are replaced with numerals as follows: $0 = \{A\}, \{B\}, \varnothing; 1 = \{A\}, \varnothing, \{B\}$.

Figure 3.6 depicts the state resulting from the *peek* action. Agent A knows that the coin lies heads up. Agent B does not know how the coin lies, but knows that A does. However, A does not know whether B knows that A knows how the coin lies.

3.2.3 Loud Phonecall

In the Loud Phonecall problem, Agent A hears proposition p announced on the telephone. Agent B is in the same room, and agent A believes that agent B cannot hear the phone, i.e. that B is a partial observer. However, unbeknownst to agent A, the phone is loud enough that agent B can hear the announcement. Thus, while agent A believes that agent B is a partial observer, agent B is, in fact, a full observer, and comes to know that p.

Figure 3.7 shows the start state. Proposition φ is true, but neither A nor B knows it. Unbeknownst to A, the phone is loud enough for B to hear (l).

The *announce* announcement action is defined with the following action theory:

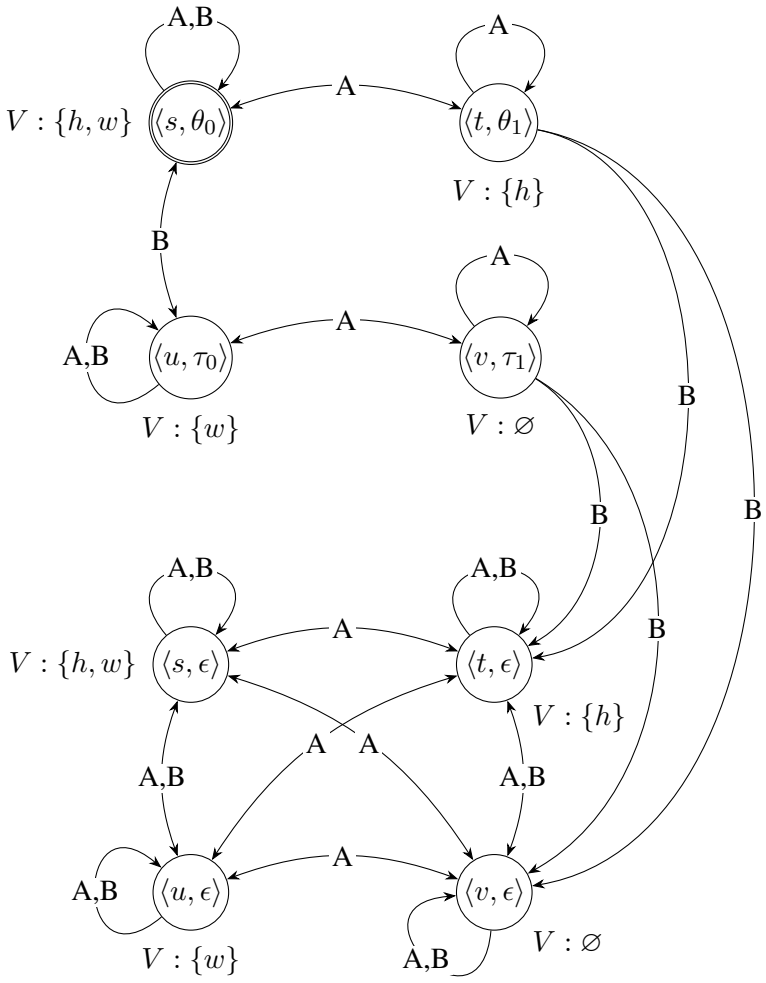


Figure 3.6: The final state of the Second-Order Coin problem.

- *announce* announces φ
- A observes *announce*
- B observes *announce* if l
- B aware_of *announce* if $\neg l$

Figure 3.8 depicts the event model induced by the application of the *announce* action in the Loud Phonecall problem start state. Agent B is a full observer, but A believes that B is a partial observer.

Figure 3.9 shows the state resulting from the *announce* action. Agents A and B both know φ, but A believes that B does not know whether φ.

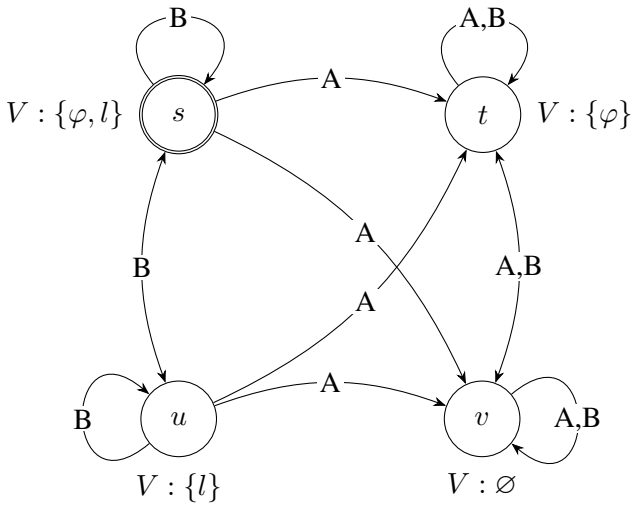


Figure 3.7: The Loud Phonecall problem start state.

3.2.4 Secret Distract

The Secret Distract problem involves three agents and two actions. Agents A, B, and C are in a room with a box. Agent A distracts agent C, causing C to look away from the box. Agent B does not observe this distraction, and continues to believe that agent C is watching. Then agent A opens the box. In the resulting state, agent B believes that agent C believes that the box is open, even though C actually believes that the box is closed. A similar problem was presented in Baral, G. Gelfond, Pontelli, et al. (2022) to illustrate the inability of m$\mathcal{A}^*$ to effectively handle second-order false belief tasks. Using the semantics of m$\mathcal{A}^*$, in the final state agent B believes that agent C believes that the box is not open. Because agent C's non-observance of the action is represented globally, agent B's false belief that agent C observes the action is not expressed. m$\mathcal{A}$-local resolves this limitation. The start state is depicted in Figure 3.10. It is common knowledge that the box is not open ($\neg o$) and that C is watching (w). For simplicity we do not model A and B watching.

The action theory which describes both the *distract* ontic action and the *open* ontic action is:

- *open* causes o
- A observes *open*
- B observes *open*

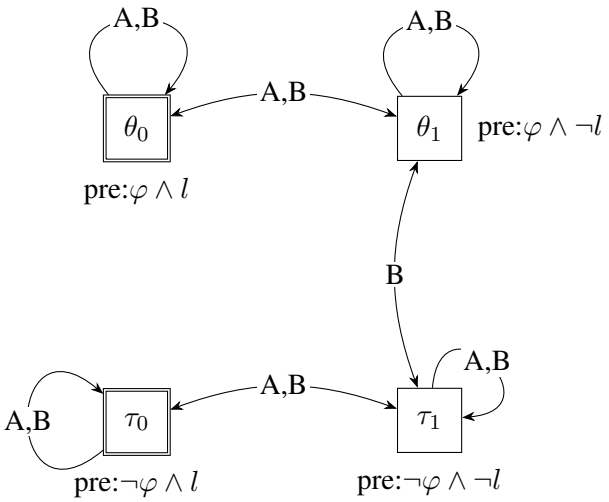


Figure 3.8: The event model induced by the *announce* action in the *phone* start state. For compactness, the event subscripts are replaced with numerals as follows: $0 = \{A, B\}, \varnothing, \varnothing$; $1 = \{A\}, \{B\}, \varnothing$. The ϵ event is omitted because there are no non-observant agents in any world.

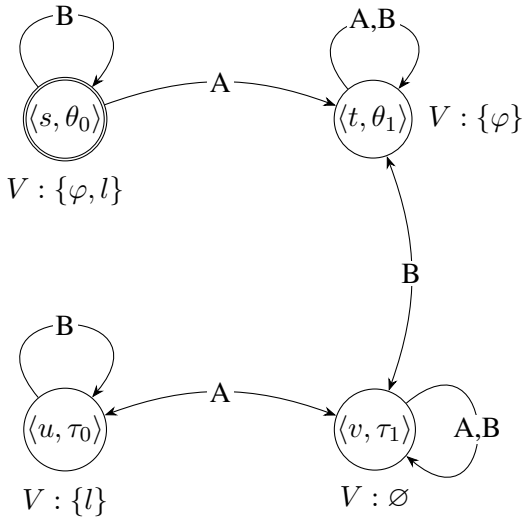


Figure 3.9: The Loud Phone problem final state. The disconnected ϵ worlds are omitted.

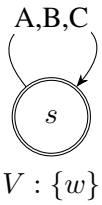


Figure 3.10: The start state for the Secret Distract problem.

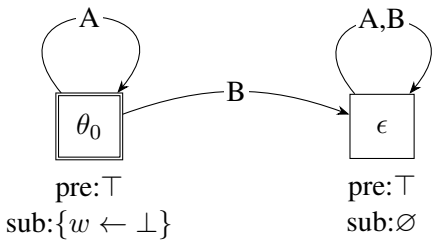


Figure 3.11: The event model induced by the application of the *distract* ontic action in the Secret Distract start state. For compactness, the event subscript $\{A, C\}, \varnothing, \{B\}$ is replaced with the numeral 0.

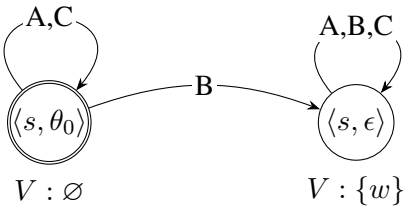


Figure 3.12: The intermediate state for the Secret Distract problem after C has been distracted. For compactness, we substitute 0 for the event subscript $\{A, C\}, \varnothing, \{B\}$.

- C observes *open* if w
- *distract* causes $\neg w$
- A observes *distract*
- C observes *distract*

Figure 3.11 shows the event model induced by the application of the *distract* action in the *secret distract* start state. Agent A distracts C without B knowing it. Figure 3.12 shows the intermediate state resulting from the *distract*(C)$\langle$A$\rangle$ action. Agent B wrongly believes that C is not looking ($\neg l$). Figure 3.13 shows the event model for the *open*$\langle$A$\rangle$ action. Agent C is non-observant, but B believes that agent C is a full observer. Figure 3.14 shows the final state of the *secret distract* problem. Agent B believes that agent C believes that the box is open, even though Agent C believes that the box is not open.

3.3 Discussion

We have focused on second-order false-belief and uncertainty arising from false-beliefs and uncertainty about the observability of actions. However there are other ways that second-

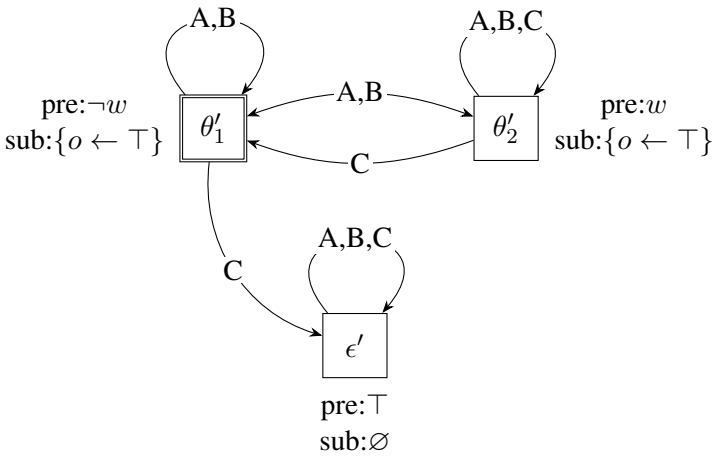


Figure 3.13: The event model induced by the application of the *open* action. For compactness, the event subscripts are replaced with numerals as follows: $1 = \{A, B\}, \varnothing, \{C\}$; $2 = \{A, B, C\}, \varnothing, \varnothing$.

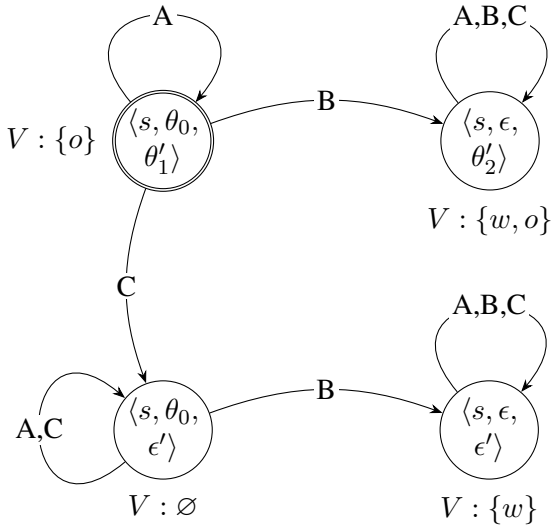


Figure 3.14: The final state of the *secret distract* problem. For compactness, the event subscripts are replaced with numerals as follows: $0 = \{A, C\}, \varnothing, \{B\}$; $1 = \{A, B\}, \varnothing, \{C\}$; $2 = \{A, B, C\}, \varnothing, \varnothing$.

order false-beliefs and uncertainty can occur. Consider an agent A who believes correctly that agent B believes proposition φ. Then A fails to observe an event that causes agent B to come to believe $\neg\varphi$. Now agent A has the second-order false-belief that agent B believes φ. Because it does not involve false belief or uncertainty about observability, m$\mathcal{A}^*$ can represent this scenario.

The state-transition system presented here is not guaranteed to preserve any frame

properties of epistemic states, such as $S5_n$ or $KD45_n$. For example, if an event would cause an agent to learn information that contradicts her prior false-belief, it could lead to a state where the agent vacuously believes all propositions (because the world relation loses seriality). To address such scenarios, it is necessary to incorporate some capacity for belief revision, as m$\mathcal{A}^*$does (Section 2.7.1) or as discussed in Chapters 4 and 5.

The event models constructed by m$\mathcal{A}^*$contain exactly two (for ontic actions) or three (for sensing and announcement actions) events. However, m$\mathcal{A}$-local potentially creates many more: two events for each way to partition agents' observability plus the ϵ-event. In the worst case, that can be $2|W| + 1$ events in a state with worlds W. Given that, in the general case, each event in an event model can result in a new world for each world in W, there may seem to be a risk that state size will explode. However, each of the θ-events and τ-events is preconditioned by a formula that is satisfied only where one of the observability partitions holds. Thus each world in W combines with at most two events (either one θ-event or one τ-event, and the ϵ-event) to create new worlds, and the size of the resulting successor state is therefore not significantly larger than with the three-event semantics of m$\mathcal{A}^*$.

The strength of m$\mathcal{A}$-local is that it makes a focused, incremental improvement to m$\mathcal{A}^*$. It retains the syntax of m$\mathcal{A}^*$, and its semantics follow the same style and overall strategy. It is straightforward to update an existing implementation of m$\mathcal{A}^*$ to employ m$\mathcal{A}$-local "under the hood", without altering parsing, product updates, or other infrastructure. The main limitation of m$\mathcal{A}$-local with respect to other languages considered in this work is that it does not offer any improvements to belief revision.

Chapter 4

m$\mathcal{A}$-revise

In DEL, when an agent observes some fact φ to be true, the agent's beliefs are updated by removing from the set of worlds which the agent considers possible all worlds where φ is false. If the agent had wrongly believed $\neg\varphi$, then no possible worlds remain, and the agent is left "ignorant", believing all propositions to be trivially true. To address this, when an agent's false belief is corrected by a sensing or announcement action m$\mathcal{A}^*$ removes *all* of an agent's uncertainty, even about things unrelated to the belief being revised. At least one planner (Le et al., 2018) implements this method.

This chapter presents m$\mathcal{A}$-revise, a language based on m$\mathcal{A}^*$ which supports belief revision by modeling knowledge and belief simultaneously. Following Hintikka (1962) and Kraus and Lehmann (1988) and others, we employ Kripke structures that have two binary relations on worlds for each agent: one relation to represent knowledge and one relation to represent belief. Thus we maintain two tiers of information, a lower (belief) tier, and a strictly more general higher (knowledge) tier, for each agent. We assume that agents have access to a source of information about the environment that can reliably inform knowledge (e.g. non-noisy sensors), as well as less reliable sources that can only inform belief (e.g. possibly-false announcements), also called "hard facts" and "soft facts" respectively (Van Benthem, 2007). An agent accepts the most recent information received into the appropriate tier, except that lower tier information does not override higher tier information, e.g. an agent ignores an announcement that she knows to be untrue. Belief revision draws on an agent's knowledge, adding to the set of worlds which the agent *believes* possible those worlds

not previously *known* by the agent to be impossible. Thus we use an agent's necessarily-true knowledge to support revision of her possibly-false beliefs, and refer to this process as belief *reset*. This approach can be used by multi-agent epistemic planning systems to explore problems involving false announcements (including lies), false beliefs, and recovery from false belief. The usefulness of being able to handle untruthful announcements in negotiations has been discussed in Son, Pontelli, Nguyen, et al. (2014). Unlike m$\mathcal{A}^*$ and m$\mathcal{A}$-local, m$\mathcal{A}$-revise does not build event models. Instead, the action semantics are given directly as transformations of epistemic states. Like m$\mathcal{A}$-local, m$\mathcal{A}$-revise represents action observability at the level of worlds, so can handle the second-order false belief problems discussed in Chapter 3.

Consider the Bicycle Story (Sullivan, Zaitchik, and Tager-Flusberg, 1994) in which it is Timmy's birthday and Timmy's mom intends to give him a bicycle which she has hidden in the basement. Wishing to surprise Timmy, she tells him that she does *not* intend to give him a bicycle. Unbeknownst to his mother, Timmy looks in the basement, sees the bicycle, and infers her intention. The result is that Timmy's mom wrongly believes that Timmy believes that she does not intend to give him a bicycle. An agent capable of effectively navigating such a domain will need to represent false announcements, false beliefs thereby induced, belief revision, whereby an agent corrects false beliefs, and second-order false beliefs.

4.1 Epistemic-Doxastic States

Let $\mathcal{G}$ be a set of n agents, and $\mathcal{P}$ a set of atomic propositions. We define a bimodal Kripke model (within this chapter simply a *model*) as a Kripke structure with two accessibility relations for each agent, i.e. a tuple $\langle W, V, \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{K}_1, \ldots, \mathcal{K}_n \rangle$, where

- W is a non-empty finite set of *worlds*,
- $V : \mathcal{P} \times W \to \{\top, \bot\}$ is a valuation function mapping each proposition at each world to *true* or *false*,
- $\mathcal{B}_i \subseteq W \times W$ is a relation representing agent i's belief, and
- $\mathcal{K}_i \subseteq W \times W$ is a relation representing agent i's knowledge.

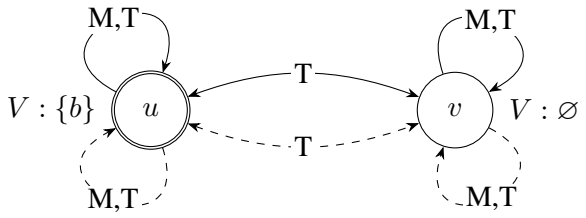


Figure 4.1: State s_1, the Bicycle initial state.

As shorthand, we write $M = \langle W, V, K_{i \in A}, B_{i \in A} \rangle$, and use square brackets to refer to components of M ($M[W]$, $M[V]$, $M[K_i]$, $M[B_i]$). Following Kraus and Lehmann, 1988, we assume that $\mathcal{B}_i$ satisfies the $KD45$ properties (is serial, transitive, and Euclidean), that $\mathcal{K}_i$ satisfies the $S5$ properties (is reflexive, transitive, and symmetric), and that $\mathcal{B}_i$ and $\mathcal{K}_i$ satisfy the $KB1$ ($\mathcal{B}_i \subseteq \mathcal{K}_i$) and $KB2$ ($\langle u, v \rangle \in K_i, \langle v, w \rangle \in B_i \rightarrow \langle u, w \rangle \in B_i$) (Hintikka, 1962) properties.

An epistemic-doxastic state (henceforth *state*) is a pair $\langle \mathcal{M}, d \rangle$, where $\mathcal{M}$ is a model and $d \in M[W]$ is the *designated world*, representing the actual state of the environment. A state $\langle \mathcal{M}, d \rangle$ models modal formulas in $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$ as follows (the semantics for non-modal formulas is the same as for $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$):

$$\langle \mathcal{M}, u \rangle \models \mathbf{B}_i \varphi \qquad \text{iff for all } v \in W, \text{ if } (u, v) \in \mathcal{B}_i \text{ then } (\mathcal{M}, v) \models \varphi$$

$$\langle \mathcal{M}, u \rangle \models \mathbf{K}_i \varphi \qquad \text{iff for all } v \in W, \text{ if } (u, v) \in \mathcal{K}_i \text{ then } (\mathcal{M}, v) \models \varphi$$

$$\langle \mathcal{M}, u \rangle \models \mathbf{C}_g \varphi \qquad \text{iff for all } v \in W, \text{if } u(\cup_{i \in g} \mathcal{K}_i)^* v \text{ then } \langle \mathcal{M}, v \rangle \models \varphi$$

where $i \in \mathcal{G}$, $g \subseteq \mathcal{G}$, $\varphi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, and $(K)^*$ is the transitive closure of K.

At the start of Bicycle, Timmy's mom ("M") has bought Timmy ("T") a bicycle ("*b*"), but T doesn't know it, and M has not yet said anything. We model these initial conditions with state s_1, presented in Figure 4.1, where solid arrows indicate knowledge relations and dashed arrows indicate belief relations. In this state b is *true*, and M believes it, but T neither believes nor disbelieves it: $s_1 \models b \wedge \mathbf{B}_M b \wedge \neg \mathbf{B}_T b \wedge \neg \mathbf{B}_T \neg b$.

4.2 Action Language

Unlike m$\mathcal{A}$-local, which separately defines ontic actions that cause changes to the environment, sensing actions that update agents' knowledge through observations, and announcement actions that represent communication, m$\mathcal{A}$-revise provides a unified system that allows a single action to incorporate two or all three of these qualities. In general, with the exception of ontic changes themselves, information learned by virtue of an action occurrence pertains to the state *before* the action occurs. Each agent may be fully observant, partially observant (e.g. knowing that announcement was made, but not what was announced), or non-observant of an action's execution.

Let A be a finite set of action names. The action language m$\mathcal{A}$-revise consists of all statements of the following forms:

1. "if φ then i observes a"
2. "if φ then i is aware of a"
3. "if φ then a is executable"
4. "if φ then a causes p to become l"
5. "a determines whether φ"
6. "a announces ψ"

where $a \in A$ is the name of an action, $i \in \mathcal{G}$ is an agent, $\varphi \in \mathcal{L}^{\mathcal{P}}$, $\psi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, and $l \in \{true, false\}$.

An *action theory* $\mathcal{T} \subseteq$ m$\mathcal{A}$-revise defines preconditions, ontic effects, sensing and announcement properties, and dynamic observability rules for actions A. Statements of form 1 specify conditional full observability of an action: if φ, then agent i will be a full observer of a. Statements of form 2 specify conditional partial observability: if φ, then i will be a partial observer (or will be aware) of a. An agent who is neither fully observant nor partially observant is non-observant. Statements of form 3 specify action preconditions: the action be executed only if φ. Statements of form 4 specify conditional ontic effects: if φ, then

the action will cause the value of proposition p to become either *true* or *false*. Statements of form 5 specify sensing, or observation: the action will cause full observers to come to know whether φ. Statements of form 6 specify announcements: The action will cause full observers to come to believe ψ, unless they reject the announcement. An agent will reject an announcement if it contradicts her prior knowledge, if it contradicts new knowledge she gains from the action, or if the action includes multiple contradictory announcements. Note that ψ is in $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, so announcements can be about beliefs and knowledge, and that ψ need not be true. Whenever a statement begins with "if $\top$ then" we will omit this trivial condition. Instead of restricting action theories to contain a single executability statement per action, let

$$pre(a) \coloneqq \bigwedge_{\text{“if } \varphi \text{ then } a \text{ is executable”} \in \mathcal{T}} \varphi$$

be the precondition of a.

As is common in announcement logics, m$\mathcal{A}$-revise does not describe *who* performs an action (such information could be encoded within an action's name), and action observability is independent of agency. In order that updates may be well defined, we require that no two ontic effects assign opposite values to the same proposition. Thus for every state $\langle \mathcal{M}, u \rangle$,

$$\forall \text{ “if } \varphi \text{ then } a \text{ causes } p\text{”, “if } \varphi' \text{ then } a \text{ causes } \neg p\text{”} \in \mathcal{T}. \langle \mathcal{M}, u \rangle \not\models \varphi \wedge \varphi'. \tag{4.1}$$

In order that observability definitions be unambiguous, we require that no agent be both a full observer and a partial observer: Thus for every state $\langle \mathcal{M}, u \rangle$,

$$\forall \text{ “if } \varphi \text{ then } i \text{ observes of } a\text{” “if } \varphi' \text{ then } i \text{ is aware of } a\text{”} \in \mathcal{T}. \langle \mathcal{M}, u \rangle \not\models \varphi \wedge \varphi' \tag{4.2}$$

The actions used in the Bicycle Story are *mom_tell* and *tim_look*. They are defined with the following action theory:

- *mom_tell* announces b
- M observes *mom_tell*

- T observes *mom_tell*
- *tim_look* determines whether b
- T observes *tim_look*

4.3 Intermediate Transition Function

Given an action theory $\mathcal{T}$, an action $a \in A$, and a state $\langle \mathcal{M}, u \rangle$, we would like to define a transition function, $\times$, that describes how the application of a in $\langle \mathcal{M}, d \rangle$ results in a new state $\langle \mathcal{M}, u \rangle \times a = \langle M', u' \rangle$. As a first step, we will define an intermediate operation, $\bar{\times}$, that takes a model M (ignoring a state's designated world) and an action α, and applies the action to the model. We call the result of this operation a *submodel*, and refer to it as M^α: $M \bar{\times} \alpha = M^\alpha = (W^\alpha, V^\alpha, B^\alpha_{i \in A}, K^\alpha_{i \in A})$. This operation describes how an action affects ontic properties, as well as the belief and knowledge of full and partial observers, but does not represent the uncertainties and false beliefs that α induces in non-observant agents. We will present $\bar{\times}$ over the next several subsections and then define it in Equation 4.23. As we will show in Section 4.4, our state transition function $\langle M, d \rangle \times a$ invokes $M \bar{\times} \alpha$ as a subroutine, at least one time with $a = \alpha$, and possibly multiple times with $a \neq \alpha$ to account for uncertainty about which action occurs.

4.3.1 Observability

Assuming an action theory $\mathcal{T}$ and a model M, and given an action α described by $\mathcal{T}$ and a world $u \in M[W]$, we define functions that classify agents as fully observant (F), partially observant (P), or non-observant (N) to the action's occurrence:

$$F(\alpha, u) := \{i \in A \mid \exists \text{ “if } \varphi \text{ then } i \text{ observes } \alpha\text{”} \in \mathcal{T}, u \models \varphi\} \tag{4.3}$$

$$P(\alpha, u) := \{i \in A \mid \exists \text{ “if } \varphi \text{ then } i \text{ is aware of } \alpha\text{”} \in \mathcal{T}, u \models \varphi\} \tag{4.4}$$

$$N(\alpha, u) := A \setminus (F(\alpha, u) \cup P(\alpha, u)) \tag{4.5}$$

The requirement for consistent observability definitions (Equation 4.2) ensures that

$$\forall \alpha \in A, u \in M[W].F(\alpha, u) \cap P(\alpha, u) = \varnothing$$

4.3.2 Acquired Knowledge

We define the propositional formula $\mathcal{K}^{\alpha}_{iu}$ for each agent $i \in \mathcal{G}$ and world $u \in M[W]$ to represent facts about the state before action α occurs that are gained as knowledge by i when α occurs in u. First, propositional formula $\overset{det}{\mathcal{K}}{}^{\alpha}_{u}$ represents facts an agent learns because she observes them.

$$\overset{det}{\mathcal{K}}{}^{\alpha}_{u} := \bigwedge_{\varphi|u\models\varphi \text{ and } ``\alpha \text{ determines whether } \varphi"\in\mathcal{T}} \quad \wedge \quad \bigwedge_{\neg\varphi|u\models\neg\varphi \text{ and } ``\alpha \text{ determines whether } \varphi"\in\mathcal{T}} \tag{4.6}$$

Second, propositional formula $\overset{eff}{\mathcal{K}}{}^{\alpha}_{u}$ represents the value, before the action occurs, of propositions that may (depending on their value before the action and on effect conditions) be altered by ontic effects. Since someone who observes an action learns about its ontic effects (e.g. sees when a door has been opened), it follows that they should observe what would be changed regardless of whether the ontic effect actually changed it. For example, even if an *open* action doesn't open a door, either because the door is locked or because it was already open, full and partial observers can determine whether the door had been open or closed before the action occurred.

$$\overset{eff}{\mathcal{K}}{}^{\alpha}_{u} := \bigwedge_{p|u\models p \text{ and } ``\text{if } \varphi \text{ then } \alpha \text{ causes } p \text{ to become } l"\in\mathcal{T}} \quad \wedge \quad \bigwedge_{\neg p|u\models\neg p \text{ and } ``\text{if } \varphi \text{ then } \alpha \text{ causes } p \text{ to become } l"\in\mathcal{T}} \tag{4.7}$$

Third, propositional formula $\overset{con}{\mathcal{K}}{}^{\alpha}_{u}$ represents facts about action effect conditions that an agent learns by observing an action. For example, consider the action "open_door" with a precondition defined by "if $\neg is\_open$ then $open\_door$ is executable" and with a conditional effect defined by "if $\neg door\_locked$ then $open\_door$ causes is_open to become *true*": an

observing agent will learn that the door was not locked just in the case that "is_open" becomes *true*, and will otherwise learn that the door was locked. To help with defining $\overset{con}{\mathcal{K}}{}^{\alpha}_{u}$, let $\phi_{u\alpha p}$ be the disjunction of the conditions of effects that would alter the value of p if they occurred in u.

$$\phi_{u\alpha p} = \bigvee_{\varphi | \text{“if } \varphi \text{ then } \alpha \text{ causes } p \text{ to become } l\text{”} \in \mathcal{T} \text{ and } V(p,u) \neq l} \varphi \tag{4.8}$$

Thus if multiple effect conditions can cause the same change in a proposition, the knowledge gained is the disjunction of the conditions: an agent knows that at least one of the preconditions is responsible, but does not know which. If $V(p, u) = l$ then this effect will not change p, regardless of whether φ. Let $\overset{con}{\mathcal{K}}{}^{\alpha}_{u}$ be the conjunction of all such learned effect conditions.

$$\overset{con}{\mathcal{K}}{}^{\alpha}_{u} = \left(\bigwedge_{p \in \mathcal{P} | u \models \phi_{u\alpha p}} \phi_{u\alpha p} \right) \wedge \left(\bigwedge_{p \in \mathcal{P} | u \not\models \phi_{u\alpha p}} \neg \phi_{u\alpha p} \right) \tag{4.9}$$

If an action effect precondition is satisfied (and thus the effect occurs), *and* if the effect actually changes something (it had a different value than that caused by the effect), then the observer comes to know that the effect precondition held. Similarly, if an effect condition was not satisfied, and the observer can determine this because the proposition's value did not change, then the observer learns that the effect condition did not hold. In either case, when an observer learns whether an effect condition held, we call such a condition an *indicated* effect condition. Thus $\overset{con}{\mathcal{K}}{}^{\alpha}_{u}$ is the conjunction of indicated effect conditions.

Third, $\overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}$ represents conditions on observability that an agent learns by knowing her own observer status. For example, consider an action with observability defined by "if $\neg battery\_dead$ then *Alice* observes *phonecall*". If Alice observes *phonecall*, she will come to know that $\neg battery\_dead$. To help with defining $\overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}$, we first define $\phi^{F}_{i\alpha}$ as the formula that conditions agent i's full observability of α, and $\phi^{P}_{i\alpha}$ as the formula that conditions agent i's partial observability of α.

$$\phi^{F}_{i\alpha} \coloneqq \bigvee_{\text{“if } \varphi \text{ then } i \text{ observes } \alpha\text{”} \in \mathcal{T}} \varphi$$

$$\phi^P_{i\alpha} \coloneqq \bigvee_{\text{“if } \varphi \text{ then } i \text{ is aware of } \alpha\text{”} \in \mathcal{T}} \varphi$$

$$\overset{obs}{\mathcal{K}}{}^{\alpha}_{iu} \coloneqq \begin{cases} \phi^F_{i\alpha} & \text{if } i \in F(\alpha, u) \\ \phi^P_{i\alpha} & \text{if } i \in P(\alpha, u) \\ \neg\phi^F_{i\alpha} \wedge \neg\phi^P_{i\alpha} & \text{if } i \in N(\alpha, u) \end{cases} \tag{4.10}$$

A full observer learns that at least one of the conditions of her being a full observer must be true. Similarly, a partial observer learns that at least one of the conditions on her being a partial observer must be true. Finally, non-observant agents knows that conditions on her full and partial observability must be false.

We combine these multiple sources of information into a single proposition formula, $\mathcal{K}^{\alpha}_{iu}$, representing knowledge that α causes agent i to gain when it occurs in world u.

$$\mathcal{K}^{\alpha}_{iu} \coloneqq \begin{cases} \overset{det}{\mathcal{K}}{}^{\alpha}_{u} \wedge \overset{eff}{\mathcal{K}}{}^{\alpha}_{u} \wedge \overset{con}{\mathcal{K}}{}^{\alpha}_{u} \wedge \overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}, & \text{if } i \in F(\alpha, u) \\ \overset{eff}{\mathcal{K}}{}^{\alpha}_{u} \wedge \overset{con}{\mathcal{K}}{}^{\alpha}_{u} \wedge \overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}, & \text{if } i \in P(\alpha, u) \\ \overset{obs}{\mathcal{K}}{}^{\alpha}_{iu} & \text{if } i \in N(\alpha, u) \end{cases} \tag{4.11}$$

Full observers learn sensed information, action effects, indicated effect conditions, and conditions on their observers status. Partial observers learn action effects, indicated effect conditions, and conditions on their observers status. Non-observant agents only learn conditions on their observer status. Note that a non-observant agent does not learn the value of observability conditions; she learns only that *in worlds where* the action occurred, conditions on her being a full or partial observer must be false. For example, even if Alice does not hear the phone ring, she comes to know that if someone *had in fact* called then the battery must be dead. An agent cannot gain false knowledge, i.e. $\mathcal{K}^{\alpha}_{iu}$ is consistent with u:

Theorem 4.3.1. *All acquired knowledge is true:* $\forall \alpha \in A, i \in \mathcal{G}, u \in M[W].\langle \mathcal{M}, u\rangle \models \mathcal{K}^{\alpha}_{iu}$

The proof appears in the Appendix (Section 9.2.1).

4.3.3 New World Construction

Without loss of generality let the names of agents be sequential integers: $A = (1, \ldots, n), n = |A| - 1$. We use the old knowledge relations K_i to construct the new knowledge relations K_i^α, removing connections between worlds according to $\mathcal{K}_{iu}^\alpha$. For example, if $\langle u, v\rangle \in K_i$ and agent i learns in world u that world v is not possible, i.e. $\langle M, v\rangle \not\models \mathcal{K}_{iu}^\alpha$, we construct new worlds u' and v' from u and v respectively, and have $\langle u', v'\rangle \notin K_i^\alpha$.

However, $\mathcal{K}$ may not act symmetrically: $\langle \mathcal{M}, v\rangle \not\models \mathcal{K}_{iu}^\alpha$ does not imply that $\langle \mathcal{M}, u\rangle \not\models \mathcal{K}_{iv}^\alpha$. For example, let world u be where Alice's cell phone battery has charge, and world v be where the battery is dead. An action occurs whereby someone calls the cell phone. In world u, the phone will ring and Alice will come to know that the battery is not dead and that world v is not possible. In world v, however, she will hear nothing, but cannot conclude that the battery is dead because, for all she knows, the action may not have occurred. Thus, in world v, she will not come to know that world u is impossible. Because K_i^α is an equivalence relation (and therefore symmetric), in order to represent such an asymmetric knowledge update we will duplicate world u. From v we create a new world v', and from u we create two new worlds: one will *not* be in the same equivalence class as v', representing the perspective from u, and the other *will* be in the same equivalence class as v', representing the perspective from v.

We start by defining *local equivalence classes* C_i^α for each agent $i \in \mathcal{G}$. A local equivalence class is an equivalence class from the perspective of some world that incorporates knowledge gained at that world.

$$C_{iu}^\alpha \coloneqq \{v \in W \mid \langle u, v\rangle \in K_i, \langle \mathcal{M}, v\rangle \models \mathcal{K}_{iu}^\alpha\} \tag{4.12}$$

$$C_i^\alpha \coloneqq \{C_{iu}^\alpha \mid u \in W\} \tag{4.13}$$

A single world may appear in multiple local equivalence classes (elements of C_i^α). Let Q_{iu}^α

be the set of agent i's local equivalence classes that contain world u.

$$Q_{iu}^{\alpha} \coloneqq \{c \in C_i^{\alpha} \mid u \in c\} \tag{4.14}$$

Let G_u^{α} be the set of assignments of world u to a local equivalence class for each agent:

$$G_u^{\alpha} = \{(c_1, \ldots, c_n) \mid \forall i \in A. c_i \in Q_{iu}^{\alpha}\} \tag{4.15}$$

Thus an element of G_u^{α} assigns world u to local equivalence class c_1 for agent 1, to c_2 for agent 2, and so forth, and G_u^{α} contains all such assignments.

For each old world $u \in W$ satisfying the action preconditions, and for each assignment to per-agent local equivalence classes in G_u^{α}, we construct a new world, $\langle u, \alpha, c_1, \ldots, c_n \rangle$, and W^{α} is the set of all such new worlds:

$$W^{\alpha} \coloneqq \{\langle u, \alpha, c_1, \ldots, c_n \rangle \mid u \in W, u \models pre(\alpha), (c_1, \ldots, c_n) \in G_u^{\alpha}\} \tag{4.16}$$

For the telephone example, we might have

$$W^{call} = \{(u, call, \{u\}), (u, call, \{u, v\}), (v, call, \{u, v\})\}.$$

For convenience, let m be a mapping from new (post-α) worlds to old (pre-α) worlds: $m(\langle u, \alpha, c_0, \ldots, c_n \rangle) = u$. We use square brackets to access a local equivalence class: $\langle u, \alpha, c_0, \ldots, c_n \rangle[c_i] = c_i$. Thus $m(u)$ and $u[c_i]$ are defined for new worlds $u \in W^{\alpha}$ for some $\alpha \in A$, and not for old worlds in W.

To compute the valuation of new worlds, we apply the action's ontic effects to the valuations of old worlds. For each new world corresponding to an old world that satisfies the *action* preconditions, we apply those action effects whose *effect* conditions hold. That is, for $p \in \mathcal{P}$ and $u^{\alpha} \in W^{\alpha}$, define V^{α} as follows.

$$V^{\alpha}(p, u^{\alpha}) \coloneqq \begin{cases} l, \text{ if } \exists \text{“if } \varphi \text{ then } \alpha \text{ causes } p \text{ to become } l\text{”} \in \mathcal{T}, m(u^{\alpha}) \models \varphi \\ V(p, m(u^{\alpha})), \text{ otherwise} \end{cases} \tag{4.17}$$

4.3.4 Knowledge Update

Let K_i^α be the new knowledge relations expressing the results of the action's occurrence. The names of the new worlds already encode the equivalence relations that should be expressed in each agent's new knowledge relation. For all $i \in \mathcal{G}$:

$$K_i^\alpha \coloneqq \{\langle u, v\rangle \mid u, v \in W^\alpha, u[c_i] = v[c_i]\} \tag{4.18}$$

4.3.5 Acquired Belief

In this section, we define $\mathcal{B}_{iu}^\alpha$ to represent the information that full observers may accept as new beliefs (about the state before α occurs) by virtue of α occurring in world u: things that an observer learns that *might possibly* be false.

$$\mathcal{B}_{iu}^\alpha \coloneqq \begin{cases} \bigwedge_{\text{“}\alpha \text{ announces } \psi\text{”}\in\mathcal{T}} \psi, \text{if } i \in F(\alpha, u) \\ \top, \text{otherwise} \end{cases} \tag{4.19}$$

4.3.6 Belief Update and Revision

Let B_i^α be the new belief relations resulting from α's execution. We construct B_i^α incrementally, performing belief revision as necessary. In $B_i^{\alpha 1}$ we copy the old belief relation to the new worlds, except to reflect newly acquired beliefs held in $\mathcal{B}_{iu}^\alpha$and knowledge represented in K_i^α.

$$B_i^{\alpha 1} \coloneqq \{\langle u, v\rangle \mid \langle u, v\rangle \in K_i^\alpha, \langle m(u), m(v)\rangle \in B_i, \langle M, m(v)\rangle \models \mathcal{B}_{im(u)}^\alpha\} \tag{4.20}$$

$B_i^{\alpha 1}$ may not be serial, for example if i learns information contradicting her existing beliefs. In that case, we use the knowledge relation to repair it, applying $\mathcal{B}_{iu}^\alpha$ to K_i^α, to construct

$B_i^{\alpha 2}$.

$$B_i^{\alpha 2} \coloneqq B_i^{\alpha 1} \cup \{\langle u, v\rangle \mid \forall w \in W^\alpha . \langle u, w\rangle \notin B_i^{\alpha 1}, \\ \langle u, v\rangle \in K_i^\alpha, \langle M, m(v)\rangle \models \mathcal{B}_{im(u)}^\alpha\} \tag{4.21}$$

It is possible that even $B_i^{\alpha 2}$ is not serial, for example because one of the newly learned beliefs in $\mathcal{B}_{iu}^\alpha$ contradicts old knowledge in K_i (e.g. an announcement that is a lie that i knows cannot possibly be true), or if there are multiple, contradictory announcements. In that case, we again use the knowledge relation to repair the belief relation, but ignore $\mathcal{B}_{iu}^\alpha$.

$$B_i^\alpha \coloneqq B_i^{\alpha 2} \cup \{\langle u, v\rangle \mid \forall w \in W^\alpha . (u, w) \notin B_i^{\alpha 2}, \langle u, v\rangle \in K_i^\alpha\} \tag{4.22}$$

The intermediate step $B_i^{\alpha 2}$ is valuable because it means an agent will reject an announcement if it contradicts her prior knowledge, but not if it simply contradicts her prior beliefs. Thus we have defined the intermediate state transition function.

$$M \bar{\times} \alpha \coloneqq (W^\alpha, V^\alpha, K_{i\in A}^\alpha, B_{i\in A}^\alpha) \tag{4.23}$$

4.4 State Transitions

We are now ready to define the application of action a in state $\langle M, u\rangle$ to generate a new state:

$$\langle M, u\rangle \times a \coloneqq \langle M', u'\rangle.$$

In the Appendix (Section 9.2.1), we prove that this process preserves the $S5$, $KD45$, $KB1$, and $KB2$ properties from K_i and B_i to K_i' and B_i'.

Theorem 4.4.1. *K' satisfies $S5_n$: for all $i \in \mathcal{G}$, K_i' is reflexive, symmetric, and transitive.*

Theorem 4.4.2. *B' satisfies $KD45_n$: for all $i \in \mathcal{G}$, B_i' is serial, transitive, and Euclidean.*

Theorem 4.4.3. *K_i' and B_i' satisfy the KB1 property that $B_i' \subseteq K_i'$.*

Theorem 4.4.4. *K'_i and B'_i satisfy the KB2 property that if $\langle u, v\rangle \in K'_i$ and $(v, w) \in B'_i$ then $\langle u, w\rangle \in B'_i$.*

As we intend that an action can only be executed if its preconditions are satisfied, we define the state transition function for just such cases;

$$\text{If } d \not\models pre(a) \text{ then } \langle M, d\rangle \times a \text{ is undefined.} \tag{4.24}$$

Otherwise, apply the intermediate transition function to construct the model M^a expressing the ontic and observed effects of the action: $M^a = M \bar{\times} a$ where $M^a = (W^a, V^a, B^a_{i\in A}, K^a_{i\in A})$. The model M^a contains the new designated world:

$$d' \coloneqq (d, a, C^a_{0d}, \ldots, C^a_{nd}). \tag{4.25}$$

If there is no world in $M^a[W]$ corresponding to a world in $M[W]$ where some agent is non-observant then we can simply return M^a. This may be the case even though there are non-observant agents at some worlds in W, provided that such worlds do not satisfy a's preconditions.

$$\text{If } \forall u \in M^a[W].N(a, m(u)) = \varnothing, \text{then } \langle M, d\rangle \times a = \langle M^a, d'\rangle. \tag{4.26}$$

Otherwise, an agent that is non-observant to an action will consider all actions that she believes *could* have occurred (including the possibility that nothing has occurred). We apply each such action to M, and will later connect the resulting submodels together with the knowledge and belief relations of non-observant agents to construct M'. We employ the action *No-op* to represent the perspective of non-observant agents, who believe that a did not occur. We assume that there are no statements in $\mathcal{T}$ that contain the term "*No-op*". Thus *No-op* has no ontic, sensing, or announcing effects, and no agent fully or partially observes it. Model M^0 represents the result of applying *No-op* in M and is thus semantically equivalent to M: $M^0 \coloneqq M \bar{\times}$ *No-op*.

Let $\mathcal{A}^0$ refer to all actions including *No-op*: $\mathcal{A}^0 \coloneqq A \cup \{$*No-op*$\}$. Let the set H_i hold sub-models corresponding to the occurrence of hypothetical actions (including the actually

occurring action a) considered possible by i at any world where i is non-observant of a. For every $i \in \mathcal{G}$:

$$H_i \coloneqq \{M \bar{\times} \alpha \mid \alpha \in \mathcal{A}^0, \exists u \in W. i \in N(a,u), i \in N(\alpha,u), u \models pre(\alpha)\} \tag{4.27}$$

In order to avoid a possible infinite regress, we will not represent that agents other than i may be non-observant of such actions that i hypothesizes.

Thus we have constructed several new sub-models which we will combine to produce M'. First, M^a represents the effects of the action which has occurred, and contains the new designated world. Second, M^0 represents the possibility, according to non-observant agents, that nothing has occurred. Finally, submodels in H_i represent the results of actions hypothesized by agent i (in some world) to have possibly occurred (even if i does not *believe* anything has occurred). If H_i is not empty (i does not observe the action) then it includes M^a and M^0, and by Equation 4.26 H_i is not empty for *some* $i \in \mathcal{G}$. Fully and partially observant agents will access only M^a, via knowledge and belief relations. A non-observant agent i will access only M^0 via her belief relation, but will access all models in H_i via her knowledge relation.

The worlds of the new state are W'.

$$W' \coloneqq \bigcup_{i \in \mathcal{G}} \bigcup_{M^H \in H_i} M^H[W] \tag{4.28}$$

Every world $u \in W'$ is also part of exactly one submodel, either M^a, M^0, or some member of H_i for some $i \in \mathcal{G}$, and we refer to that submodel as $M^{u[a]}$.

The new world valuations have been computed in the intermediate transition function. We combine them (treating functions as sets of ordered pairs) into V'.

$$V' \coloneqq \bigcup_{i \in \mathcal{G}} \bigcup_{M^H \in H_i} M^H[V] \tag{4.29}$$

Let K'_{iF} hold the new knowledge relations where agents are full or partial observers; these relate worlds only within each sub-model, and are copied from the relations computed

in the intermediate transition function.

$$K'_{iF} \coloneqq \{\langle u, v\rangle \mid \langle u, v\rangle \in M^{u[a]}[K_i] \wedge i \in F(u[a], m(u)) \cup P(u[a], m(u))\} \tag{4.30}$$

Let K'_{iN} hold the new knowledge relation where agents are non-observant, relating worlds both within and between sub-models. Within an agent's hypothesized action results H_i, an agent does not know which sub-model she is in (which action actually occurred), so we interconnect all sub-models in H_i. In submodels not in H_i (the result of an action hypothesized by some agent other than i) we do not represent i's non-observant status, i.e. no edges connect that sub-model to another.

$$\begin{aligned} K'_{iN} = \{\langle u, v\rangle \mid u, v \in W' \wedge i \in N(u[a], m(u)) \wedge \\ ((M^{u[a]}, M^{v[a]} \in H_i) \vee (M^{u[a]} = M^{v[a]} \notin H_i)) \wedge \\ (\exists x, y \in M^0[K_i].m(x) = m(u) \wedge m(y) = m(v))\} \end{aligned} \tag{4.31}$$

The final condition ($\exists x, y \in M^0[K_i]$) accounts for $\overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}$ (Equation 4.10), the information learned by non-observers as discussed in Section 4.3.2. The new knowledge relations K'_i combine K'_{iF} and K'_{iN}.

$$K'_i \coloneqq K'_{iF} \cup K'_{iN} \tag{4.32}$$

Let B'_{iF} hold the new belief relations where agents are full or partial observers; these relate worlds only within each sub-model, and are copied from the relations computed in the intermediate transition function.

$$B'_{iF} \coloneqq \{\langle u, v\rangle \mid \langle u, v\rangle \in M^{u[a]}[B_i] \wedge i \in F(u[a], m(u)) \cup P(u[a], m(u))\} \tag{4.33}$$

The relation B'_{iN} connects sub-models within H_i, expressing that a non-observant agent is unsure which of her hypothesized actions has occurred, and connects worlds within a sub-

model not in H_i, to accommodate the second-order beliefs (about i) of other non-observant agents.

$$B'_{iN} = \{\langle u, v\rangle \mid i \in N(u[a], u) \\ \wedge ((M^{u[a]} \in H_i \wedge M^{v[a]} = M^0) \vee (M^{u[a]} = M^{v[a]} \notin H_i)) \wedge \\ \exists (x, y) \in M^0[B_i].m(x) = m(u) \wedge m(y) = m(v)\} \quad (4.34)$$

We join relations for non-observant and observant agents to define B'_i.

$$B'_i = B'_{iF} \cup B'_{iN} \quad (4.35)$$

Thus we have defined the state transition operation.

$$\langle M, d\rangle \times a = \begin{cases} \langle M^a, d'\rangle, \text{if } \forall u \in M^a[W].N(a, m(u)) = \varnothing, \\ \langle\langle W', V', K'_{i\in A}, B'_{i\in A}\rangle, d'\rangle, \text{otherwise.} \end{cases}$$

4.4.1 Semantic Correctness

In the Appendix (Section 9.2.2) we prove the following propositions regarding the semantics of m$\mathcal{A}$-revise, where $\langle \mathcal{M}, u\rangle \times a = \langle M', u'\rangle$ and $M' = \langle W', V', \mathcal{B}'_1, \ldots, \mathcal{B}'_n, \mathcal{K}'_1, \ldots, \mathcal{K}'_n\rangle$.

Proposition 4.4.5. *Ontic effects alter the state if their conditions are met: if "if φ then a causes p to become l"$\in \mathcal{T}$ and $u \models \varphi$ then $p \in V'(u')$ iff $l = true$.*

Proposition 4.4.6. *Full and partial observers learn the effects of ontic actions: for every $i \in F(a, d) \cup P(a, d)$ and "if φ then a causes p to become l" $\in \mathcal{T}$, $\langle M', u'\rangle \models \boldsymbol{K}_i p \vee \boldsymbol{K}_i \neg p$.*

Proposition 4.4.7. *Full and partial observers learn that full and partial observers learn ontic effects: If "if φ then a causes p to become l"$\in \mathcal{T}$ and $u \models \varphi$ and $i \in F(a, u) \cup P(a, u)$ and $(\forall v \in M[W]$. if $\langle u, v\rangle \in M[B_i]$ then $j \in F(a, u) \cup P(a, u))$, then $\langle M', u'\rangle \models \boldsymbol{B}_i(\boldsymbol{K}_j p \vee \boldsymbol{K}_j \neg p)$.*

Proposition 4.4.8. *If one or more indicated effect conditions condition an effect, full and partial observers learn whether at least one of those conditions holds:* $\forall p \in \mathcal{P}.\langle M', u'\rangle \models \boldsymbol{K}_i\phi_{uap} \vee \boldsymbol{K}_i\neg\phi_{uap}$.

Proposition 4.4.9. *Full and partial observers know that full and partial observers learn whether indicated effect conditions hold: If* $i \in F(a,u) \cup P(a,u)$ *and* $\forall(u,v) \in M[B_i].j \in F(a,v) \cup P(a,v)$*, then* $\langle M', u'\rangle \models \boldsymbol{B}_i(\boldsymbol{K}_j\phi_{uap} \vee \boldsymbol{K}_j\neg\phi_{uap})$.

Proposition 4.4.10. *Full observers learn the value of sensed propositions:* $\forall$ *"*a *determines whether* φ*"* $\in \mathcal{T}.u' \models \boldsymbol{K}_i\varphi \vee \boldsymbol{K}_i\neg\varphi$.

Proposition 4.4.11. *Full and partial observers learn that full observers learn sensed propositions. If "*a *determines whether* φ*"*$\in \mathcal{T}$ *and* $i \in F(a,u) \cup P(a,u)$ *and* $\forall(u,v) \in M[B_i].j \in F(a,u)$*, then* $\langle M', u'\rangle \models \boldsymbol{B}_i(\boldsymbol{K}_j\varphi \vee \boldsymbol{K}_j\neg\varphi)$.

Proposition 4.4.12. *Full observers believe that the conjunction of all announced propositions* was *true if it is not self-contradictory and does not contradict their prior knowledge or current observations. If* i *is a full observer then in any world* $v' \in W'$ *such that* $\langle u', v'\rangle \in B'_i$*, it was the case before the action occurred that announcements were true:* $\langle M, m(v')\rangle \models \mathcal{B}^{\alpha}_{iu}$.

Proposition 4.4.13. *Full and partial observers know that full observers believe that the conjunction of all announced propositions* was *true if it is not contradictory and does not contradict their prior knowledge or current observations. If* i *is a full or partial observer and believes that* j *is a full observer then* $\forall\langle u', v'\rangle \in B'_i.\forall\langle v', w'\rangle \in B'_j.\langle M, m(w')\rangle \models \mathcal{B}^{\alpha}_{jm(v')}$.

Proposition 4.4.14. *Actions do not alter the beliefs of non-observant agents, except for beliefs about knowledge (e.g. they do come to believe that they know that the action could have occurred). Let* $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$*-*$K \subset \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$ *be the set of formulas that do not contain any subformula* $\boldsymbol{K}_j\varphi$ *for any* $j \in \mathcal{G}$ *and* $\varphi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$*. If* $i \in N(a,u)$ *then for all* $\psi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$*-*$K$*,* $\langle M, u\rangle \models \boldsymbol{B}_i\psi$ *iff* $\langle M', u'\rangle \models \boldsymbol{B}_i\psi$.

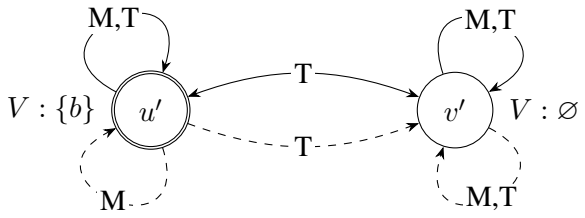


Figure 4.2: State $s'_1 = s_1 \times$ *mom_tell*, after M tells T that $\neg b$. For compactness, state names are abbreviated as follows: $u' = \langle u, mom\_tell, \{u, v\}_T, \{u\}_M \rangle$; $v' = \langle v, mom\_tell, \{u, v\}_T, \{u\}_M \rangle$.

4.4.2 *Bicycle* Transitions

Recall that state s_1 (Figure 4.1) represents the starting condition of the Bicycle Story, and that the *mom_tell* and *tim_look* actions (Section 4.2) represent M announcing $\neg b$ and T observing b, respectively. Applying *mom_tell* in s_1 results in state s'_1, shown in Figure 4.2, where T incorrectly believes $\neg b$. Applying *tim_look* in s'_1 results in state s''_1, shown in Figure 4.3, where M and T both believe that b, but M believes that T believes $\neg b$.

4.5 Local Observability

Like m$\mathcal{A}$-local, m$\mathcal{A}$-revise tracks action observability at the level of possible worlds (Equations 4.3, 4.4, and 4.5), making it possible to capture an agent's uncertainty or incorrect belief about another agent's observation of an action. This is demonstrated by another scenario, Bicycle-2, that involves conditional observability. Timmy's mom knows that Timmy looks in the basement, but does not know whether the basement door obstructs his view of the bicycle.

To keep this example simple, we ignore the belief relations and consider only the knowledge relations. We omit the mom_tell action and replace tim_look with tim_look_2. Unlike Bicycle, M knows that T performs this action. However, now T will see the bicycle only if the basement door is open (o). M doesn't know whether o, and therefore should not know whether T comes to know b. Figure 4.4 shows State s_2, the initial state for Bicycle-2. Action *tim_look_2* is defined with the following action theory:

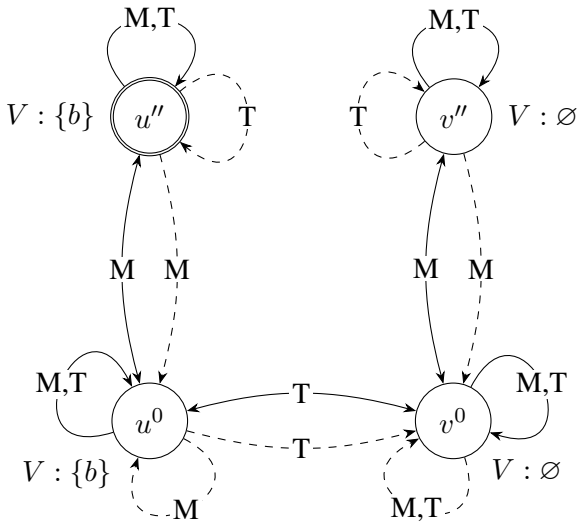


Figure 4.3: State $s_1'' = s_1' \times tim\_look$, after T determines that b. For compactness, state names are abbreviated as follows: $u'' = \langle u', tim\_look, \{u'\}_T, \{u'\}_M \rangle$; $v'' = \langle v', tim\_look, \{v'\}_T, \{v'\}_M \rangle$; $u^0 = \langle u', \textit{No-op}, \{u', v'\}_T, \{u'\}_M \rangle$; $v^0 = \langle v', \textit{No-op}, \{u', v'\}_T, \{v'\}_M \rangle$.

- "*tim_look_2* determines whether b"
- "if o then T observes *tim_look_2*"
- "if $\neg o$ then T is aware of *tim_look_2*"
- "M is aware of *tim_look_2*"

Figure 4.5 shows State s_2', the result of applying *tim_look_2* in s_2. While T now knows whether b, M does not know that T knows whether b. This representation of second-order uncertainty is made possible by the locality of the observability definition. T observes *tim_look_2* in the designated world of s_2, but not in the worlds where o is *false*. Therefore, M is uncertain about whether T is an observer, and, as a result, becomes uncertain about whether T knows whether b.

Because m$\mathcal{A}^*$ does not represent observability at this local level, it does not capture this second-order uncertainty. Figure 4.6 shows State s_2'', the result of applying *tim_look_2* to state s_2 using the state-transition rules for sensing actions provided by m$\mathcal{A}^*$. Here, M knows that T knows b, even though M did not (and still does not) know whether o.

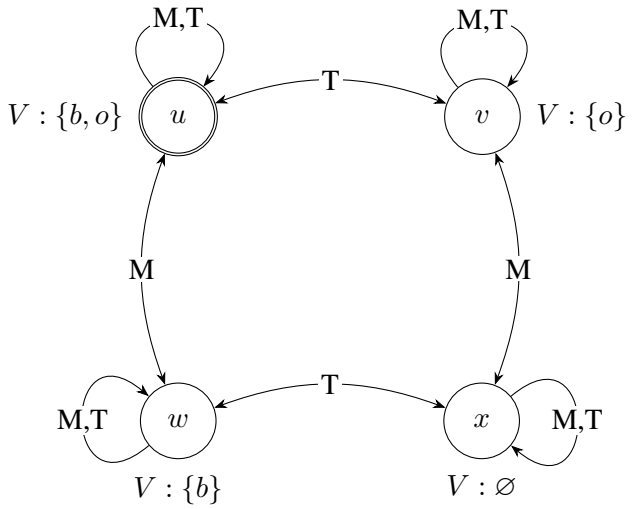


Figure 4.4: State s_2, the Bicycle-2 initial state.

4.6 Hypothetical Actions

m$\mathcal{A}$-revise is also an improvement upon the epistemic action language presented in Buckingham, Kasenberg, and Scheutz (2020). In that approach, non-observant agents do not consider the possibility of hypothetical actions except for the null action (they believe nothing occurs) and the action that actually occurs (they know that its occurrence is possible). Another variant of the Bicycle Story, called Bicycle-3, highlights this difference.

In Bicycle-3, a second action, *tim_play*, is available. Thus, T can either look for the bicycle or go play: M would not observe either action and therefore won't know which action is performed. We add the proposition p meaning "T is playing", and augment the action theory from Bicycle with the following statements:

- "T observes *tim_play*"
- "*tim_play* causes p to become *true*"

The Bicycle-3 initial state is s_1' (Figure 4.2), i.e. the state from the original Bicycle example after *mom_tell* but before *tim_look*. The inclusion of p does not change the figure, since p is not true in any world. Figure 4.7 shows the result of applying *tim_look*. M does not know whether T performed *tim_look*, *tim_play*, or no action (*No-op*), so M does not know whether p.

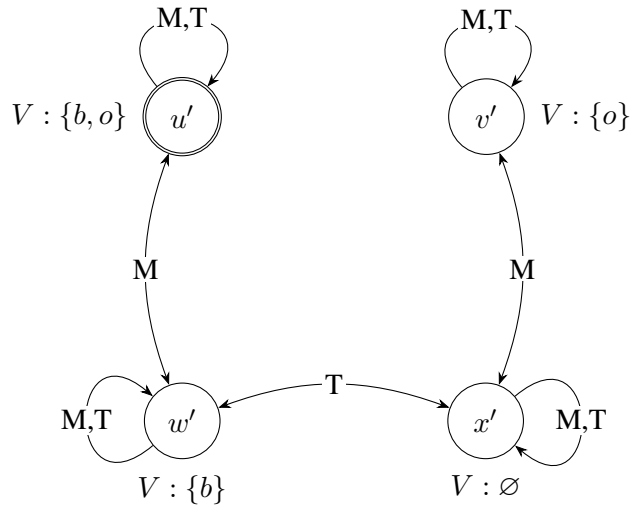


Figure 4.5: State $s'_2 = s_2 \times tim\_look\_2$, after T determines that b in Bicycle-2. For compactness, state names are abbreviated as follows: $u' = \langle u, tim\_look\_2, \{u\}_T, \{u, w\}_M \rangle$; $v' = \langle v, tim\_look\_2, \{v\}_T, \{v, x\}_M \rangle$; $w' = \langle w, tim\_look\_2, \{w, x\}_T, \{u, w\}_M \rangle$; $x' = \langle v, tim\_look\_2, \{w, x\}_T, \{v, x\}_M \rangle$.

As in m$\mathcal{A}$-revise, in Buckingham, Kasenberg, and Scheutz (2020) an agent that does not observe an action believes that no action has occurred, but comes to know that the action *might* have happened. However, she does not also consider that any other actions (i.e. that did not actually occur) could have occurred. Thus, she gains access to knowledge related to an action which she does not observe. Figure 4.8 shows state s'_3, the result of applying the *tim_look* action in Bicycle-3 using the transition system proposed in Buckingham, Kasenberg, and Scheutz (ibid.). Unlike State s_3, where M does not know whether p or $\neg p$, here M knows that $\neg p$. Nevertheless, incorporating non-observant agents' consideration of hypothetical actions can lead to very large states in some domains. In such domains it may be beneficial to modify m$\mathcal{A}$-revise to restrict hypothetical actions as Buckingham, Kasenberg, and Scheutz (ibid.) does, i.e. by replacing equation 4.27 with $H_i \coloneqq \{a, \textit{No-op}\}$.

4.7 Discussion

One strength of m$\mathcal{A}$-revise is how information learned by an agent is categorized into two tiers of reliability. Reliable information is stored as knowledge, less reliable information as beliefs. Beliefs can be strengthened to knowledge, e.g. via a sensing action, and knowledge

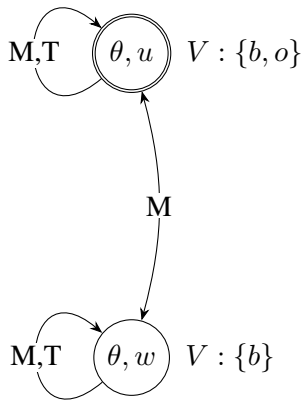


Figure 4.6: State s_2'', the result of applying *tim_look_2* in s_2 using m$\mathcal{A}^*$.

can be weakened to belief, e.g. when a non-observant agent does not know whether some ontic action has occurred.

Unlike m$\mathcal{A}$-local, m$\mathcal{A}$-revise offers a belief revision system that is more nuanced than that of m$\mathcal{A}^*$. It lets agents revise their beliefs based on information coming from many sources, including from effect conditions, which is not supported by m$\mathcal{A}^*$ or, as we will see in the next section, by m$\mathcal{B}$. The main limitation of m$\mathcal{A}$-revise is that it does not compile action representations to event models, isolating it from the extensive research into DEL. On the other hand, skipping the intermediate step of building event models while computing state transitions could save computational effort during planning.

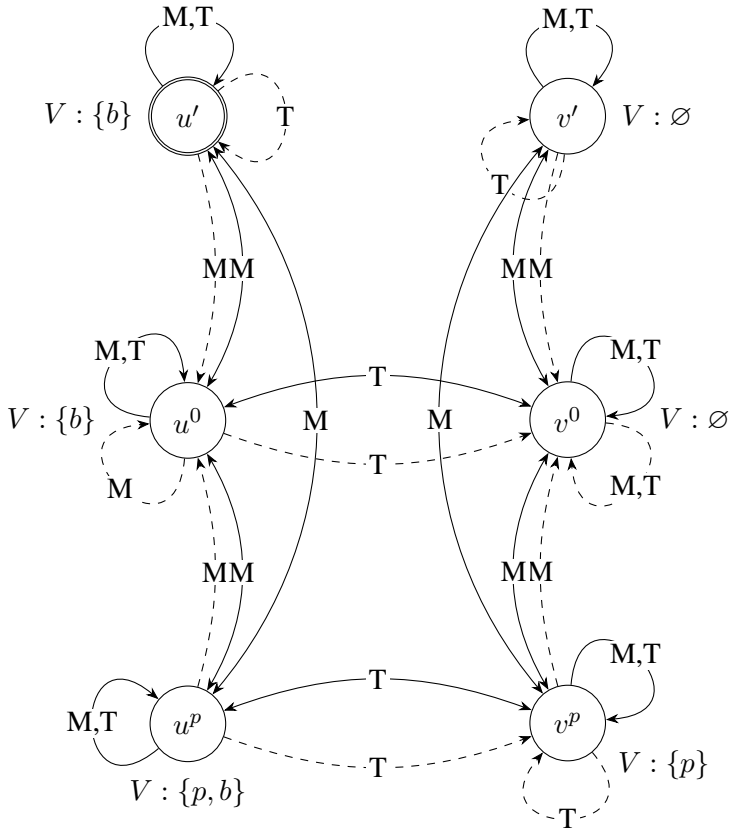


Figure 4.7: State $s_3 = s'_1 \times tim\_look$, after T determines that b in Bicycle-3. For compactness, state names are abbreviated as follows: $u' = \langle u, tim\_look, \{u\}_T, \{u\}_M \rangle$; $v' = \langle v, tim\_look, \{v\}_T, \{v\}_M \rangle$; $u^0 = \langle u, \textit{No-op}, \{u, v\}_T, \{u\}_M \rangle$; $v^0 = \langle v, \textit{No-op}, \{u, v\}_T, \{v\}_M \rangle$. $u^p = \langle u, tim\_play, \{u, v\}_T, \{u\}_M \rangle$; $v^p = \langle v, tim\_play, \{u, v\}_T, \{v\}_M \rangle$.

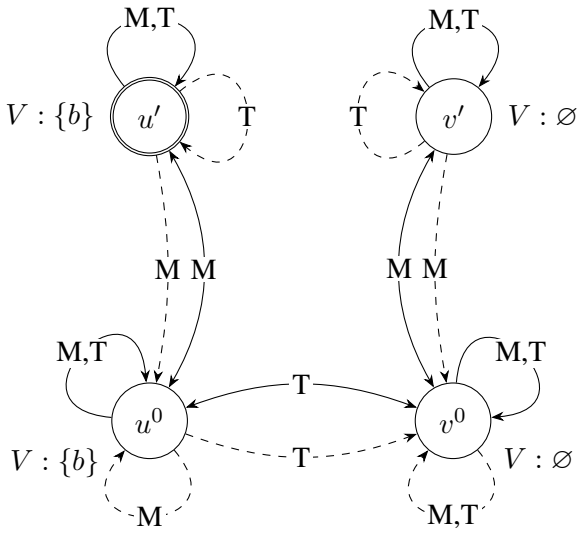


Figure 4.8: State s'_3, the result of applying *tim_look* to State s'_1 in the Bicycle-3 example with $H_M = \{tim\_look, \textit{No-op}\}$. For compactness, state names are abbreviated as follows: $u' = \langle u, tim\_look, \{u\}_T, \{u\}_M \rangle$; $v' = \langle v, tim\_look, \{v\}_T, \{v\}_M \rangle$; $u^0 = \langle u, \textit{No-op}, \{u, v\}_T, \{u\}_M \rangle$; $v^0 = \langle v, \textit{No-op}, \{u, v\}_T, \{v\}_M \rangle$.

Chapter 5

m$\mathcal{B}$

The m$\mathcal{B}$ language is an epistemic action language in the style of m$\mathcal{A}^*$ with semantics based in *plausibility preorders* (Baltag and Smets, 2006, 2008; Benthem and Smets, 2015). Like m$\mathcal{A}$-revise, it distinguishes between belief and knowledge. Unlike m$\mathcal{A}$-local, it allows false announcements. Its belief revision mechanism is simpler than that of m$\mathcal{A}$-revise, and handles observability in a way that allows second-order false belief (and uncertainty) tasks. m$\mathcal{A}^*$ and m$\mathcal{A}$-local build an action model from an action description paired with the state where the action is executed: action models are built dynamically to express the execution of an action in a particular state. In contrast, m$\mathcal{B}$ builds an action model once for each action definition and uses edge conditions (Bolander, 2018) to encode dynamic observability. Thus while m$\mathcal{A}^*$ and m$\mathcal{A}$-local builds event models "on-line" whenever an action is to be applied in some state, m$\mathcal{B}$ pre-builds action models "off-line".

Another system for epistemic planning using plausibility preorders is presented in Andersen, Bolander, and M. H. Jensen (2013). It focuses on using plausibility preorders to guide state search toward situations that the planner considers most plausible, and to avoid expending planning effort on future states that are possible, but unlikely. While plausibility preorders combine knowledge and belief representations, in Andersen, Bolander, and M. H. Jensen (ibid.) dynamic information must be epistemic, i.e. a plan can involve an agent gaining knowledge, but not (possibly false) beliefs.

A well-established approach to combining knowledge and belief (Hintikka, 1962) uses $S5$ relations over worlds to represent knowledge and $KD45$ relations over worlds to

represent belief, connecting knowledge and belief with epistemic-doxastic axioms ($KB1$ and $KB2$). While m$\mathcal{A}$-revise implements belief revision within that framework, it uses *ad-hoc* state transition rules instead of action models. The advantage of m$\mathcal{B}$ is that it supports belief revision based on established belief revision techniques (plausibility preorders) instead of on *ad-hoc* state transitions. As Benthem and Smets (2015) argue, "we want to put knowledge and belief side by side in a study of belief change... Hintikka's original models with a binary doxastic accessibility relation that drive such well-known systems as 'KD45' are not what we need."

Thus m$\mathcal{B}$ adopts the plausibility preorder (over worlds to represent state, over events to represent actions) approach advocated by Baltag, Smets, and others. Specifically, the language's semantics build action plausibility models like those described in those works but augmented with two additional features. First, in order to accommodate dynamic observability, and inspired by the use of edge conditions in $S5$ models (Bolander, 2018), we add equivalence edge conditions to our action models. Second, we follow Andersen, Bolander, and M. H. Jensen (2013) to accommodate ontic action effects by means of event postconditions.

There are three types of actions along the same lines as m$\mathcal{A}^*$ and m$\mathcal{A}$-local. An action is either an announcement action, a sensing action, or an ontic action. Announcement actions represent the sharing of information between agents. To accommodate untrue announcements, including lying, we treat announcements as unreliable "soft information" (Van Benthem, Van Eijck, and Kooi, 2006). An agent who is a *full observer* to an announcement action announcing formula φ will come to believe, but not know, φ, unless she knows φ to be false. A partial observer of an announcement announcing some formula φ will know that either φ or $\neg\varphi$ was announced, but not which, but will know that full observers know which. Sensing actions represent incontrovertible observations of the world. We treat the determinations of sensing actions as "hard facts" (ibid.). As with the other languages discussed in this work, sensing and announcement actions confer information about the state *before* the action occurs. An agent who is a full observer of a sensing action that senses formula φ will come to know (and believe) φ to be true. A partial observer will come to know that full observers learn whether φ, but will not themselves learn whether φ.

Ontic actions change the objective task environment, affecting knowledge and beliefs only indirectly. We do not distinguish full- from partial observers of ontic actions. If an ontic action causes proposition p to become *true* then an observer of the action will come to know that p. For all action types, an agent who is neither fully observant nor partially observant is *non-observant*.

Consider the Coin Lie scenario, which is similar to Second-Order Coin problem (Section 3.2.2) but with the addition of a false announcement, and captures some of the expressive power of m$\mathcal{B}$. Agents A, B, and C are in a room. A concealed coin lies heads-up (h), and it is common knowledge that A and B know whether h or $\neg h$, but that C does not. Agent A makes the untruthful public announcement that $\neg h$. After the announcement, C believes (wrongly) that $\neg h$, while B continues to believe and know h. Furthermore, A and B know that C wrongly believes $\neg h$. Then B distracts A, causing A to become distracted (d). Finally, C peeks at the coin, revising her false belief and coming to know that h. However, because d, A does not observe C peeking. Thus, A's belief that C believes $\neg h$ becomes a second-order false belief. On the other hand, B is partially observant of C peeking: B knows *that* the action occurs and that C comes to know whether h. This story involves the following key elements:

- an untrue announcement ($\neg h$),
- announcement, sensing, and ontic actions,
- dynamic observability (A becomes distracted),
- fully observant, partially observant, and non-observant agents,
- false belief (C believes $\neg h$),
- belief revision (C learns that h),
- second order false belief (A believes C believes $\neg h$).

After discussing some background about plausibility preorders and defining m$\mathcal{B}$, we will apply m$\mathcal{B}$ to the Coin Lie scenario.

5.1 Plausibility Models

5.1.1 States

The state representations used by m$\mathcal{B}$ employ *plausibility models* (Baltag and Smets, 2006, 2008; Benthem and Smets, 2015), restricted to finite sets of possible worlds. Given finite sets of propositions $\mathcal{P}$ and agents $\mathcal{G}$, a plausibility model is a tuple $\langle W, R, V\rangle$ consisting of a finite set of *worlds* W, a *locally well-preordered relation* over worlds for each agent $R : W \times W \to (\mathcal{G} \to \{true, false\})$, and a valuation function $V : \mathcal{P} \to 2^W$. We write $uR_iv \coloneqq R(u, v)(i) = true$ and say that "agent i considers world v to be at least as plausible as world u". A preorder is a binary relation that is reflexive and transitive. A locally well-preordered relation is a preorder whose restriction to any comparability class is connected, i.e. for every agent $i \in \mathcal{G}$, if some sequence of worlds $(w_0, w_1, \ldots, w_{n-1}, w_n)$ is sequentially related by R edges in either direction ($w_0R_iw_1$ or $w_1R_iw_0, \ldots, w_{n-1}R_iw_n$ or $w_nR_iw_{n-1}$), then either $w_0R_iw_n$ or $w_nR_iw_0$. A *state* $\langle\langle W, R, V\rangle, u\rangle$ consists of a plausibility model and designated world $u \in W$ corresponding to the "actual" state of affairs.

Let $u \sim_i v \coloneqq uR_iv$ or vR_iu. It follows that $\sim_i$ defines an equivalence relation. The *equivalence class* of a world u for agent i is $\sim_i^u \coloneqq \{v \in W \mid u \sim_i v\}$. For some set of worlds $C \subseteq W$ and some agent i, define $\to_i C \coloneqq \{u \in C \mid u'R_iu \text{ for all } u' \in C\}$. Note that if $C \neq \varnothing$ then $|\to_i C| > 0$. The *maximal plausibility class* of a world u for agent i is denoted $\to_i^u \coloneqq \to_i \sim_i^u$.

We compute bisimulation between plausibility models using the technique given in Andersen, Bolander, Ditmarsch, et al. (2013). Unfortunately, as discussed in that work, the algorithm is not complete in the multi-agent case. Thus, when using m$\mathcal{B}$ we may compute that two states are not bisimilar even when they are semantically equivalent.

Definition 1. A state $\langle M, u\rangle$ where $M = \langle W, R, V\rangle$ and world $u \in W$ models a formulas in $\mathcal{L}_{\mathcal{G}B}^{\mathcal{P}}$ as follows:

$$
\begin{aligned}
&\langle M, u\rangle \models p && \text{iff } p \in V(u)\\
&\langle M, u\rangle \models \neg\varphi && \text{iff } \langle M, u\rangle \not\models \varphi\\
&\langle M, u\rangle \models \varphi \wedge \psi && \text{iff } \langle M, u\rangle \models \varphi \text{ and } \langle M, u\rangle \models \psi\\
&\langle M, u\rangle \models \mathbf{K}_i\varphi && \text{iff for all } v \in \sim_i^u, \langle M, v\rangle \models \varphi\\
&\langle M, u\rangle \models \mathbf{B}_i\varphi && \text{iff for all } v \in \rightarrow_i^u, \langle M, v\rangle \models \varphi\\
&\langle M, u\rangle \models \mathbf{C}_g\varphi && \text{iff for all } v \in W, \text{if } u(\cup_{i\in g} \sim_i)^* v \text{ then } \langle M, v\rangle \models \varphi
\end{aligned}
$$

where $p \in \mathcal{P}$, $\varphi, \psi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$, $i \in \mathcal{G}$, $g \subseteq \mathcal{G}$, and $(\sim)^*$ is the transitive closure of $\sim$.

5.1.2 Actions

In m$\mathcal{B}$ actions are represented as *action plausibility models*, also called "plausibility event models" (Baltag and Smets, 2006, 2008; Benthem and Smets, 2015), augmented with event postconditions (ontic effects) as in Andersen, Bolander, and M. H. Jensen (2013) and with indistinguishability edge conditions in the style of Bolander (2018). An *action* is a tuple $\langle E, Q, \text{pre}, \text{add}, \text{del}, \Gamma\rangle$ consisting of a finite set of *events* E, an edge-conditioned locally well-preordered relation over events for each agent $Q : E \times E \to (\mathcal{G} \to \mathcal{L}^{\mathcal{P}})$, a *precondition map* $\text{pre} : E \to \mathcal{L}^{\mathcal{P}}$, an *add map* $\text{add} : E \to \mathcal{P}$, a *delete map* $\text{del} : E \to \mathcal{P}$, and a set of designated events $\Gamma \subseteq E$.

5.1.3 State transitions

An action $a = \langle E, Q, \text{pre}, \text{add}, \text{del}, \Gamma\rangle$ is *applicable* in state $\langle M, d\rangle$ if there exists exactly one designated event e in Γ such that $\langle M, u\rangle \models \text{pre}(e)$. It is straightforward to extend this to allow multiple designated events to match the designated world to accommodate nondeterministic actions. However, since m$\mathcal{B}$ is deterministic, we impose this constraint for simplicity. We define the following abbreviation to help with the state transition definition: given agent $i \in \mathcal{G}$, worlds $u, v \in W$, and events $e, f \in E$, let

$$e \overset{iuv}{\to} f \coloneqq \langle M, u\rangle \models Q(e,f)(i) \text{ and } \langle M, v\rangle \models Q(e,f)(i).$$

Intuitively, $e \overset{iuv}{\to} f$ means that in worlds u and v agent i considers event f to be at least as plausible as event e. By evaluating the edge conditions at both the world corresponding to the edge's origin event and at the world corresponding to the edge's destination event, we are treating edge conditions as conditions on the equivalence relation $\sim_i$.

Definition 2. If a is applicable in state $\langle M, d\rangle = \langle\langle W, R, V\rangle, d\rangle$ then define the state transition $\langle M, d\rangle \times a \coloneqq \langle\langle W', R', V'\rangle, d'\rangle$, where

- $W' = \{\langle u, e\rangle | u \in W, e \in E, \text{ and } \langle M, u\rangle \models \text{pre}(e)\}$,
- $R'(\langle u, e\rangle, \langle v, f\rangle)(i) = \Big((e \overset{iuv}{\to} f \text{ and not } f \overset{iuv}{\to} e) \text{ and } u \sim_i v\Big)$ or $\Big(e \overset{iuv}{\to} f \text{ and } f \overset{iuv}{\to} e \text{ and } uR_iv\Big)$,
- $V'(\langle u, e\rangle) = (V(u) \cup \text{add}(e)) \setminus \text{del}(e)$,
- $d' = \langle d, e\rangle$ such that $e \in \Gamma$.

The applicability requirement means that d' is well defined. The definitions of W', V', and d' are straightforward; R' warrants some discussion. There are two top-level disjuncts. The first disjunct includes $(e \overset{iuv}{\to} f$ and not $f \overset{iuv}{\to} e)$, i.e in worlds u and v agent i considers event f to be strictly more plausible than event e, so that as long as i could not epistemically distinguish u and v ($u \sim_i v$) she will come to consider the occurrence of event f in world v to be more plausible than the occurrence of event e in world u. On the other hand, if she could distinguish u from v, i.e. in u she knows v to be impossible and vice versa, then she will also be able to distinguish $\langle u, e\rangle$ from $\langle v, f\rangle$. Thus, by the first disjunct, the event plausibility relation overrides the state plausibility relation, and incoming information takes precedence over prior beliefs, as long as it does not contradict prior knowledge. The second disjunct has ($e \overset{iuv}{\to} f$ and $f \overset{iuv}{\to} e$ and uR_iv). If in u and v i thinks that events e and f are equally plausible, her plausibility preference between $\langle u, e\rangle$ and $\langle v, f\rangle$ will be the same as between u and v, i.e. beliefs held before the action occurs are preserved if the action doesn't alter them.

5.2 The m$\mathcal{B}$ Action Language

The action language m$\mathcal{B}$ is used to define a set of actions. We first present the syntax by listing the forms of the sentences in m$\mathcal{B}$. We then give semantics by constructing a set of actions from a set of sentences. The language m$\mathcal{B}$ is parameterized by a set of atomic propositions $\mathcal{P}$, a set of agent names $\mathcal{G}$, and a set of action names A, and consists of all sentences of the following forms:

1. "if φ then a is executable",
2. "if φ then i observes a",
3. "if φ then i is aware of a",
4. "a causes $l_0, \ldots, l_n$",
5. "a determines whether φ",
6. "a announces ψ".

where $a \in A$ is the name of an action, $i \in \mathcal{G}$ is an agent, $\varphi \in \mathcal{L}^{\mathcal{P}}$, $\psi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, and each l_j is a propositional literal. An *action theory* $\mathcal{T}$ is a subset of m$\mathcal{B}$, and specifies preconditions, ontic effects, sensing and announcement properties, and dynamic observability rules for each action in A. We define a_{pre} as the *precondition* of action a:

$$a_{pre} \coloneqq \bigvee_{\text{"if } \varphi \text{ then } a \text{ is executable"} \in \mathcal{T}} \varphi. \tag{5.1}$$

Observability statements should be consistent, as should ontic effects. Thus, we assume the following restrictions for any action theory $\mathcal{T}$:

- for each a in A, $\mathcal{T}$ contains exactly one sentence of form 4, 5, or 6,
- for every state $\langle M, u \rangle$ and every pair of statements "if φ then i observes a" and "if ψ then i is aware of a" in $\mathcal{T}$, $\langle M, u \rangle \not\models \varphi \wedge \psi$, and
- if "a causes $l_0, \ldots, l_n$" $\in \mathcal{T}$ then for every $j, k \in \{0, \ldots, n\}$ if $l_j = p$ and $l_k = \neg q$ then $p \neq q$.

Recall that Q maps each action plausibility model edge to an assignment of a propositional formula to each agent: $Q : E \times E \to (\mathcal{G} \to \mathcal{L}^{\mathcal{P}})$. In our action constructions Q will label each edge in $E \times E$ with either FPN, PN, or N, where for every agent $i \in \mathcal{G}$. Let

- $FPN(i) \coloneqq \top$,
- $PN(i) \coloneqq \neg \bigvee_{\text{“if } \varphi \text{ then } i \text{ observes } a\text{”} \in \mathcal{T}} \varphi$,
- $N(i) \coloneqq \neg \Big(\big(\bigvee_{\text{“if } \varphi \text{ then } i \text{ observes } a\text{”} \in \mathcal{T}} \varphi \big) \vee \big(\bigvee_{\text{“if } \varphi \text{ then } i \text{ is aware of } a\text{”} \in \mathcal{T}} \varphi \big) \Big)$.

Intuitively, FPN indicates that the edge exists for all agents: full observers, partial observers, and non-observers; PN indicates that the edge exists only for partial observers and non-observers; and N indicates that the edge exists only for non-observers.

For conciseness, we omit the following assumptions and simplifications from formalisms, figures, and explanations throughout this chapter. Every event $e \in E$ of every action has a trivially conditioned a reflexive edge for every agent, i.e. $Q(e, e) = FPN$. For any events $e, e' \in E$ where $e \neq e'$, and any agent $i \in \mathcal{G}$, $Q(e, e')(i) = \bot$ unless otherwise indicated. Every world $u \in W$ has reflexive edge for every agent $i \in \mathcal{G}$, i.e. uR_iu. Worlds that are disconnected from a state's designated world are not shown.

5.2.1 Announcement actions

An announcement action conveys to full observers either that some formula φ is true, or that it is false. Partial observers will not know which of these is the case, only that either φ or $\neg\varphi$ was announced. Non-observers will not know that anything was announced.

Definition 3. If $\mathcal{T}$ contains “a announces φ” then a names the announcement action $\langle E, Q, \text{pre}, \text{add}, \text{del}, \Gamma \rangle$ where

- $E = \{e^{\varphi}, e^{\neg\varphi}, e^{\top}\}$,
- $Q = \{\langle\langle e^{\varphi}, e^{\neg\varphi}\rangle, PN\rangle, \langle\langle e^{\neg\varphi}, e^{\varphi}\rangle, FPN\rangle, \langle\langle e^{\varphi}, e^{\top}\rangle, N\rangle, \langle\langle e^{\neg\varphi}, e^{\top}\rangle, N\rangle\}$,
- $\text{pre} = \{\langle e^{\varphi}, a_{pre} \wedge \varphi\rangle, \langle e^{\neg\varphi}, a_{pre} \wedge \neg\varphi\rangle, \langle e^{\top}, \top\rangle\}$,
- $\text{add} = \{\langle e^{\varphi}, \varnothing\rangle, \langle e^{\neg\varphi}, \varnothing\rangle, \langle e^{\top}, \varnothing\rangle\}$,

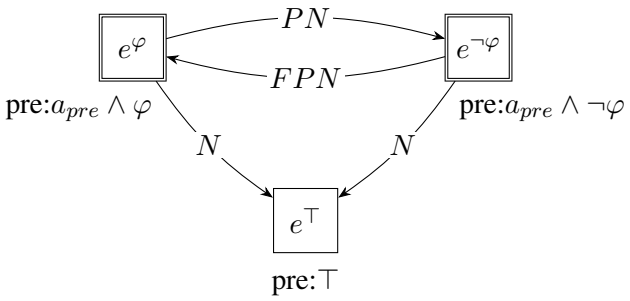


Figure 5.1: Announcement action construction.

- del $= \{\langle e^{\varphi}, \varnothing\rangle, \langle e^{\neg\varphi}, \varnothing\rangle, \langle e^{\top}, \varnothing\rangle\}$, and
- $\Gamma = \{e^{\varphi}, e^{\neg\varphi}\}$.

Figure 5.1 illustrates the construction of an announcement action, where squares are events, double borders indicate designated events, and labeled arrows show the edge-conditioned plausibility relations. Each event is labeled with its preconditions and add and delete effects. Intuitively, e^{φ} is the announcement that φ is true, $e^{\neg\varphi}$ is the announcement that φ is false, and $e^{\top}$ is nothing being announced. Full observers consider e^{φ} to be the most plausible: they will come to believe φ is true (unless they know that $\neg\varphi$). Partial observers will consider e^{φ} and $e^{\neg\varphi}$ equally plausible: they know that either φ or $\neg\varphi$ was announced, but not which. Non-observers consider $e^{\top}$, i.e that nothing occurs, to be most plausible.

5.2.2 Sensing actions

A sensing action conveys to full observers the value of some formula φ. If φ is true, then full observers learn that φ, if it is false, then they learn that $\neg\varphi$. Partial observers will not know which of these is the case, only that full observers learned either φ or $\neg\varphi$. Non-observers will not know that anything was sensed.

If $\mathcal{T}$ contains "a determines whether φ" then a names the sensing action $\langle E, Q, \text{pre}, \text{add}, \text{del}, \Gamma\rangle$ where

- $E = \{e^{\varphi}, e^{\neg\varphi}, e^{\top}\}$,
- $Q = \{\langle\langle e^{\varphi}, e^{\neg\varphi}\rangle, PN\rangle, \langle\langle e^{\neg\varphi}, e^{\varphi}\rangle, PN\rangle, \langle\langle e^{\varphi}, e^{\top}\rangle, N\rangle, \langle\langle e^{\neg\varphi}, e^{\top}\rangle, N\rangle\}$,
- pre $= \{\langle e^{\varphi}, a_{pre} \wedge \varphi\rangle, \langle e^{\neg\varphi}, a_{pre} \wedge \neg\varphi\rangle, \langle e^{\top}, \top\rangle\}$,

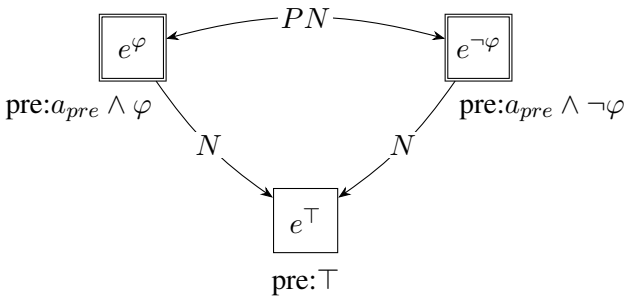


Figure 5.2: Sensing action construction.

- add $= \{\langle e^{\varphi}, \varnothing\rangle, \langle e^{\neg\varphi}, \varnothing\rangle, \langle e^{\top}, \varnothing\rangle\}$,
- del $= \{\langle e^{\varphi}, \varnothing\rangle, \langle e^{\neg\varphi}, \varnothing\rangle, \langle e^{\top}, \varnothing\rangle\}$, and
- $\Gamma = \{e^{\varphi}, e^{\neg\varphi}\}$.

Figure 5.2 illustrates a sensing action that senses whether φ is true. The construction is the same as for announcement actions, except that full observers can epistemically distinguish e^{φ} from $e^{\neg\varphi}$. Unlike announcements which only confer belief, can be lies, and are ignored when they contradict knowledge, sensing actions confer knowledge. Thus, full observers of a sensing action can determine whether event e^{φ} or $e^{\neg\varphi}$ occurs, and thereby come to know whether φ.

5.2.3 Ontic actions

An ontic action changes the valuations of one or more propositions. Full and partial observers are aware of these changes, while non-observers are not.

Definition 4. If $\mathcal{T}$ contains "a causes $l_0, \ldots, l_n$" then a names the ontic action $\langle E, Q, \text{pre}, \text{add}, \text{del}, \Gamma\rangle$. Let $\mathcal{P}^+ \coloneqq \{p \in \mathcal{P} \mid l_j = p \text{ for some } 0 \leq j \leq n\}$, and $\mathcal{P}^- \coloneqq \{p \in \mathcal{P} \mid l_j = \neg p \text{ for some } 0 \leq j \leq n$. Then

- $E = \{e^c, e^{\top}\}$,
- $Q = \{\langle\langle e^c, e^{\top}\rangle, N\rangle\}$,
- pre $= \{\langle e^c, a_{pre}\rangle, \langle e^{\top}, \top\rangle\}$,
- add $= \{\langle e^c, \mathcal{P}^+\rangle, \langle e^{\top}, \varnothing\rangle\}$,

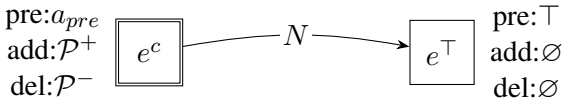


Figure 5.3: Ontic action construction.

- del $= \{\langle e^c, \mathcal{P}^-\rangle, \langle e^\top, \varnothing\rangle\}$, and
- $\Gamma = \{e^c\}$.

Figure 5.3 illustrates an ontic action. Full and partial observers consider e^c to be only possible event to occur, causing any ontic effects. Non-observers consider it most plausible that nothing changes.

5.2.4 Semantic Correctness

In the Appendix (Section 9.3) we prove the following propositions regarding the semantics of m$\mathcal{B}$, where $\mathcal{T}$ is an action theory and $\langle M, d\rangle \times a = \langle M', d'\rangle$ where $M = \langle W, R, V\rangle$, $d \in W$, $M' = \langle W', R', V'\rangle$, and $d' \in W'$.

Proposition 5.2.1. *Full observers come to believe announcements that do not contradict prior knowledge or the action's preconditions. If "if ψ then i observes a", "a announces φ"$\in \mathcal{T}$, and $\langle M, d\rangle \models \psi$, and $\langle M, d\rangle \not\models \boldsymbol{K}_i\neg(a_{pre} \wedge \varphi)$, then $\langle M', d'\rangle \models \boldsymbol{B}_i\varphi$.*

Proposition 5.2.2. *Full and partial observers know that full observers come to believe announcements unless they contradict prior knowledge or the action's preconditions. If either "if ω then j observes a" $\in \mathcal{T}$ or "if ω then j is aware of a" $\in \mathcal{T}$, and "if ψ then i observes a", "a announces φ"$\in \mathcal{T}$, and $\langle \mathcal{M}, d\rangle \models \omega$, and if for every $u \in{\sim}_j^d$ $\langle M, u\rangle \models \psi$ and $\langle M, u\rangle \not\models \boldsymbol{K}_i\neg(a_{pre} \wedge \varphi)$, then $\langle M', d'\rangle \models \boldsymbol{K}_j\boldsymbol{B}_i\varphi$.*

Proposition 5.2.3. *Full observers learn the value of sensed propositions. If "if ψ then i observes a", "a determines whether φ"$\in \mathcal{T}$ and $\langle \mathcal{M}, d\rangle \models \psi$, then $\langle \mathcal{M}', d'\rangle \models \boldsymbol{K}_i\varphi \vee \boldsymbol{K}_i\neg\varphi$.*

Proposition 5.2.4. *Full and partial observers learn that full observers learn sensed propositions. If either "if ω then j observes a" $\in \mathcal{T}$ or "if ω then j is aware of a" $\in \mathcal{T}$, and "if ψ then i observes a", "a determines whether φ"$\in \mathcal{T}$, and if for every $u \in{\sim}_j^d$, $\langle M, u\rangle \models \psi$, then $\langle M', d'\rangle \models \boldsymbol{K}_j(\boldsymbol{K}_i\varphi \vee \boldsymbol{K}_i\neg\varphi)$.*

Proposition 5.2.5. *Ontic effects update the state: if "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$ then for every k in $\{0, \ldots, n\}$, $\langle M', d' \rangle \models l_k$.*

Proposition 5.2.6. *Full and partial observers learn the effects of ontic actions: if "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$ and either "if ψ then i observes a"$\in \mathcal{T}$ or "if ψ then i is aware of a"$\in \mathcal{T}$ and $\langle M, d \rangle \models \psi$ then for every k in $\{0, \ldots, n\}$, $\langle M', d' \rangle \models \boldsymbol{K}_i l_k$.*

Proposition 5.2.7. *Full and partial observers learn that full and partial observers learn ontic effects: If either "if ψ then j observes a"$\in \mathcal{T}$, or "if ψ then j is aware of a"$\in \mathcal{T}$, and either "if ω then i observes a" $\in \mathcal{T}$ or "if ω then i is aware of a" $\in \mathcal{T}$, and $\langle M, d \rangle \models \psi$ and for every $u \in \sim_j^d$, $\langle M, u \rangle \models \omega$, and "$a$ causes $l_0, \ldots, l_n$"$\in \mathcal{T}$, then for every k in $\{0, \ldots, n\}$, $\langle M', d' \rangle \models \boldsymbol{K}_j \boldsymbol{K}_i l_k$.*

Proposition 5.2.8. *Actions do not alter the beliefs of non-observant agents, except for beliefs about knowledge (e.g. they do come to believe that they know that the action could have occurred). Let $\mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$-$K \subset \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$ be the set of formulas that do not contain any subformula $\boldsymbol{K}_i \varphi$ for any $i \in \mathcal{G}$ and $\varphi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$. If for every "if ψ then i observes a" or "if ψ then i is aware of a" in $\mathcal{T}$, $\langle \mathcal{M}, d \rangle \not\models \psi$, then for every $\varphi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$-$K$, $\langle \mathcal{M}', d' \rangle \models \boldsymbol{B}_i \varphi$ iff $\langle \mathcal{M}, d \rangle \models \boldsymbol{B}_i \varphi$.*

5.2.5 Coin Lie Scenario

Let $\mathcal{P} = \{h, d\}$ and $\mathcal{G} = \{A, B, C\}$. Start state s_0 is depicted in Figure 5.4. A coin is heads-up (h) and it is common knowledge that A and B know how the coin lies and that C does not.

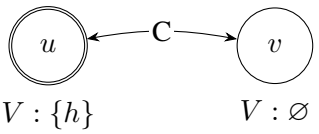


Figure 5.4: State s_0, the start state for the Coin Lie scenario.

Action theory $\mathcal{T}$ defines three actions. The announcement action *announce-not-heads* publicly announces that $\neg h$. The ontic action *distract-a* causes d (A is distracted) to become true. The sensing action *peek-c* causes full observer C to learn whether h is true or

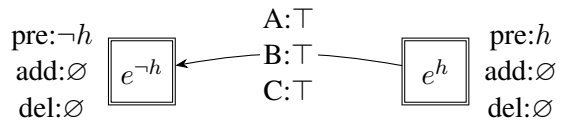


Figure 5.5: The *announce-not-heads* action.

false, partial observer B learns *that* C learns whether h, and conditionally partially observant agent A to learn that C learns whether h only if d. $\mathcal{T} =$

- *announce-not-heads* announces $\neg h$
- A observes *announce-not-heads*
- B observes *announce-not-heads*
- C observes *announce-not-heads*
- *distract-a* causes d
- A observes *distract-a*
- B observes *distract-a*
- C observes *distract-a*
- *peek-c* determines whether h
- A is aware of *peek-c* if $\neg d$
- B is aware of *peek-c*
- C observes *peek-c*

The announcement action *announce-not-heads* is depicted in figure 5.5.

Applying *announce-not-heads* in state s_0 results in state s_1 (Figure 5.6), where A and B still know that h, but C wrongly believes (but doesn't know) that $\neg h$.

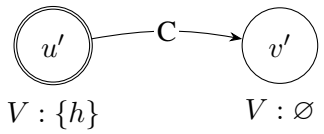


Figure 5.6: State $s_1 = s_0 \times$ *announce-not-heads*. State names are abbreviated as follows: $u' = \langle u, e^h \rangle$; $v' = \langle v, e^{\neg h} \rangle$.

Action *distract-a* is depicted in figure 5.7.

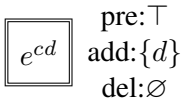


Figure 5.7: The *distract-a* action.

The result of applying *distract-a* in state s_1 is state s_2, depicted in Figure 5.8, where d is true.

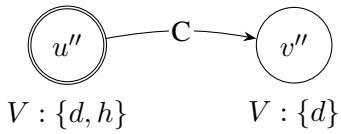


Figure 5.8: State $s_2 = s_1 \times$ *distract-a*. State names are abbreviated as follows: $u'' = \langle u', e^d \rangle$; $v'' = \langle v', e^d \rangle$.

The sensing action *peek-c* is depicted in figure 5.9. The result of applying *peek-c* in state s_2 is state s_3, depicted in Figure 5.10. All agents know that h, but A believes that C believes $\neg h$.

5.3 Discussion

To simplify presentation, m$\mathcal{B}$ does not have conditional effects. It would be straightforward to add effect conditions, although care must be taken for observers to learn information about effect conditions when it is appropriate to do, for example as discussed in m$\mathcal{A}$-revise in Section 4.3.2. Alternatively, a planning domain with event conditions can be rewritten into a domain without effect conditions by embedding effect conditions into action preconditions.

The advantage of m$\mathcal{B}$ over m$\mathcal{A}$-revise is that it supports belief revision based on an established semantics (plausibility preorders) instead of on the ad-hoc state transitions given by m$\mathcal{A}$-revise. One limitation of m$\mathcal{B}$ with respect to the other systems discussed so far is that bisimulation for multi-agent plausibility models is not complete. Another limitation is with agents' nested perspectives about announcement actions. As shown in Proposition 5.2.2, given an announcement action that announces φ, full observer j, and partial observer i, we have that j comes to know that i believes *that* φ (and not just *whether* φ). Future work should address this limitation by adjusting or expanding m$\mathcal{B}$.

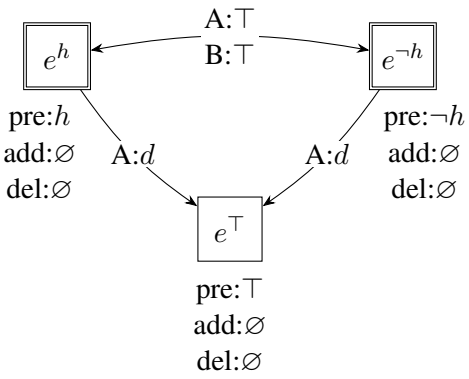


Figure 5.9: The *peek-c* action.

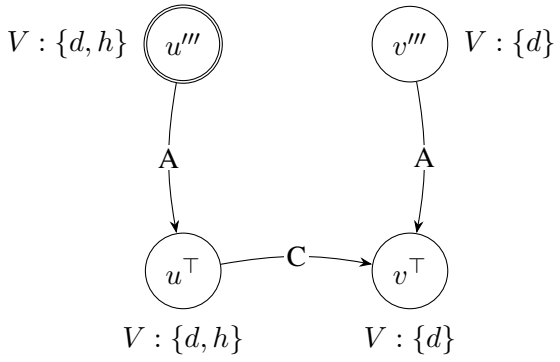


Figure 5.10: State $s_3 = s_2 \times$ *peek-c*. State names are abbreviated as follows: $u''' = \langle u'', e^h \rangle$; $v''' = \langle v'', e^{\neg h} \rangle$; $u^\top = \langle u'', e^\top \rangle$; $v^\top = \langle v'', e^\top \rangle$.

Chapter 6

Cooperation-Agnostic Search

This chapter presents the Cooperation-Agnostic Search algorithm, which can employ any of the state and action representations that have been presented in this work to solve multi-agent Epistemic Planning problems. The algorithm employs generic epistemic state and action representations, allowing specific state and action implementations to be deployed as needed, for example as appropriate for a given planning domain. This approach plans for both system agents under the control of the planner, and environment agents that the planner does not control, but whose expected behavior is given by predictive models. We build conditional plans that distinguish plan-time state information available to the planner from plan-execution-time state information available to system agents by ensuring that actions are conditioned by execution-time perspectives.

We do offline planning where a planner determines the actions of one or more system agents before any actions are made. With offline planning there are two distinct but interacting forms of uncertainty: plan-time uncertainty (of the planner) and execution-time uncertainty (of agents). Possible topics of uncertainty include uncertainty about what is the starting state of the planning problem, uncertainty about the outcome of nondeterministic actions or of observations, and uncertainty about the future behavior of environment agents. In addition, the planner may have plan-time uncertainty about agents' execution-time uncertainty. For example, the planner may not know whether an agent will be paying attention when a hidden coin is revealed, and therefore be uncertain about whether the agent will be uncertain about whether the coin lies heads- or tails-up.

The technique presented here accommodates the distinction between plan-time and execution-time uncertainty by employing two kinds of epistemic states. A *ground state* (g-state) represents the state of the task environment, including the beliefs and knowledge of each agent, from the perspective of the planner. The planner uses g-states to represent the starting state as well as hypothetical future states resulting from actions. Ground states are deterministic, so plan-time uncertainty is expressed by sets of g-states. A *perspective state* (p-state) represents a particular agent's perspective of a g-state. If a g-state models that it is raining, but that some agent doesn't know whether it is raining, then the corresponding p-state for that agent should model that it may or may not be raining. Thus, a single p-state can express an agent's execution-time uncertainty, or even false beliefs. Plan-time uncertainty about the beliefs and knowledge of an agent can be represented equivalently as a set of g-states or a set of p-states. Note that g-states may be more finely granular than p-states: two unique g-states (e.g. one where a coin is heads, and one where it is tails) may correspond to the same p-state (where how the coin lies is unspecified). The reverse is not true, a g-state encodes a single p-state for each agent.

Our approach accommodates an arbitrary finite number of system agents, also called *s-agents*, and an arbitrary finite number of environment agents, also called *e-agents*. In general, *agents* refers to both s-agents and e-agents. It is assumed that all agents act atomically and sequentially according to a pre-specified order. A generated plan specifies the actions that s-agents should take, and relies upon predictive models to determine how e-agents will act.

One challenge in distinguishing plan-time state information available to the planner from execution-time state information available to the agents is to ensure that s-agents have the information necessary to execute a generated plan, in particular, to determine the value of conditions in a conditional plan or to determine which entry in a policy to execute. For example, if the planner is uncertain whether an s-agent will know how a coin lies, it should not construct a conditional plan that specifies to do one action if the coin is heads-up, but another action if it is tails-up. As Levesque (1996) explains: "We need a program that does not contain conditions whose truth value is unknown to the agent at the required time: that is, the agent needs to know how to execute the program." Similarly,

De Lima (2007) argues that in "meaningful plans" action conditions must be "epistemically interpretable formulas." Herzig, Lang, et al. (2000) develop a logical framework to describe conditional epistemic plans where actions are conditioned only by the knowledge of the acting agent. In Andersen, Bolander, and M. H. Jensen (2012), for example, action conditions must be formulas expressing execution-time knowledge of the agent. This is in contrast to approaches such as C.-M. Huang and Mutlu (2016), who construct multi-agent conditional epistemic plans "from the viewpoint of a third person" with action conditions not necessarily interpretable by the agents. We will follow De Lima (2007) by calling plans that have this property *meaningful*. Our approach accomplishes this by building a policy whose keys are p-states.

In nondeterministic planning, a planner must account for actions that have multiple possible outcomes. This could be because an action's effects are nondeterministic (flipping a coin), or because it is unknown at plan time what information the action will reveal (peeking at a concealed coin). In the current work, nondeterminism in the state (plan-time uncertainty about what the actual state will be during execution time) also emerges from nondeterminism in the environment agent prediction models. Muise, Felli, et al. (2015) show how to treat a multi-agent planning problem involving environment agents whose expected behavior is nondeterministic as a single-agent fully-observable nondeterministic (FOND) planning problem. This approach treats uncertainty about the behavior of environment agents as nondeterminism in the effects of actions performed by system agents. In other words, the planner plans as if the result of a system agent action may result in any of several possible outcomes, whereas "under the hood" one or more environments agents may act, leading to one of several possible results.

Andersen, Bolander, and M. H. Jensen (2012) builds conditional plans for single-agent tasks, solving problems where plan-time uncertainty is resolved by an agent at execution time to determine which branches of the plan to follow. In contrast, we focus on multi-agent planning. Engesser, Bolander, et al. (2017) builds conditional plans for multi-agent epistemic planning where the planner adopts the perspective of an environment agent to plan for how that agent will act based on her possibly-incomplete knowledge, and not assuming that she will have perfect knowledge. This is similar to the present work in

that it uses AND-OR search over epistemic states to accommodate nondeterminism while planning for both system agents' and environment agents' execution time uncertainty. It differs from the present work in that it assumes that environment agents act in pursuit of the planer's goals according to the same AND-OR search algorithm, and relies upon DEL state and action representations. Thus, the proposed approach generalizes prior work by relaxing several assumptions: (1) instead of requiring a specific state representation, it allows arbitrary epistemic state representations, (2) instead of requiring a specific action representation, it allows arbitrary action representations, and (3) instead of assuming that environment agents will act according to the same reasoning as the planner uses, and toward the same goal, it allows arbitrary environment agent policies as predictors.

6.1 Inputs

Following a common practice in automated planning (Ghallab, Nau, and Traverso, 2004), for example as in PDDL (Ghallab, Howe, et al., 1998), we distinguish two aspects of the input to the planner. A planning *domain* describes the task environment, including the state space, the agents involved, and available actions. Within the context of a domain, a planning *problem* specifies the current (starting) state, the goal, and in this work, the order in which agents shall act. Thus agents act discretely in a pre-specified order, and we do not consider, for example, joint actions or durative actions. A planning domain consists of:

- $\mathcal{P}$, a finite set of atomic propositions,
- $\mathcal{G}_s$, a finite set of s-agent names,
- $\mathcal{G}_e$, a finite set of e-agent names where $\mathcal{G}_s \cap \mathcal{G}_e = \varnothing$,
- S, a set of ground states,
- $\widehat{S}$, a set of perspective states,
- $\sim_g$ and $\sim_p$, bisimulation operations over pairs of g-states and pairs of p-states respectively. When the context is clear, we simply write $\sim$,
- $\models_g$ and $\models_p$, entailment definitions for g-states and p-states respectively. When the

context is clear, we simply write $\models$. Given (g- or p-) state s and formula $\varphi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, $s \models \varphi$ means that s models φ, or that φ holds, or is true, in s,

- a state perspective shift function that defines s^i where s is a ground state, i is an agent, and s^i is a p-state representing i's beliefs and/or knowledge in s.
- A, a finite set of action names,
- $\times$, a ground state transition operation that maps a ground state and an action to a set of ground states. For example, given ground state s and action a, $s \times a = \{s', s''\}$ means that when a occurs in s the result will be either s' or s'',
- $app : S \times \mathcal{G}_s \cup \mathcal{G}_e \rightarrow 2^A$, a function specifying which actions are *applicable* for each agent, i.e. which actions an agent can perform in a given state.
- $\beta : S \times \mathcal{G}_e \rightarrow 2^A$, a behavior prediction model for each environment agent i. It is assumed that for any g-state s and agent i, $\beta(s, i) \subseteq app(s, i)$.

A planning problem consists of

- a planning domain,
- $\mathcal{G}$, a sequential ordering of $\mathcal{G}_s \cup \mathcal{G}_e$,
- $S_0 \subset S$, the initial states, and
- $\varphi_{goal} \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, a goal formula.

While S and $\widehat{S}$ could theoretically be explicitly enumerated (if they are finite), in practice S and $\widehat{S}$ will be provided by formal definitions which induce (possibly infinite) state spaces. The agents in $\mathcal{G}$ can be accessed with square brackets, modulo the size of $\mathcal{G}$. For example, if there are two system agents and two environment agents then there are four agents in total, and $\mathcal{G}[6]$ refers to the third agent in $\mathcal{G}$. Action applicability incorporates the concepts of ability (what is the agent capable of doing) as well as action preconditions (e.g. a door can be opened only if it is not locked). The bisimulation operations allow the planner to determine whether two g-states or two p-states are semantically equivalent. It is assumed that both operations are sound. Thus for any g-states s and s', if $s \sim_g s'$ then for all $\varphi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, $s \models \varphi$ iff $s' \models \varphi$, and similarly for p-states and $\sim_p$. On the other hand, it is

not assumed that the bisimulation operators are complete, as discussed in Section 6.4. It is intended that agents will act sequentially in the order indicated by $\mathcal{G}$.

The p-states are used to represent the mental perspectives of s-agents. The generated polices have keys that are pairs of p-states and timesteps. By including timesteps in the policy keys, we assume that agents will keep track of time and be able to distinguish between otherwise identical p-states occurring at different timesteps. This is necessary because the distinction between p-states and g-states means that it may appear to an agent that she is in a state that she as already visited even when the plan is not, in fact, cyclic.

To build such a policy the algorithm conducts a forward search over p-states. However, to determine how an action applied in a p-state at s-agent i's turn will lead to another p-state at s-agent i''s turn, the planner must consider the *actual* state of affairs at i's turn, not the p-state expressing i's perspective. Therefore the planner maintains references from each p-state to a set of associated g-states, and uses the g-states to determine what the effect of the action will be, as well as the effects of the actions of any e-agents who act between i and i'. Thus the algorithm searches the tree over p-states to determine which actions to put in the policy by building a tree of g-nodes in order to compute the affects of actions.

A g-node is a pair, $\langle s, n \rangle$, where s is a g-state and n is a parent g-node (or Null if $s \in S_0$).

We use square brackets to access the states of g-nodes: if g-node $n = \langle s', n' \rangle$ then $n[s] = s'$. Furthermore, given a set of g-nodes N, square brackets can access generate the set consisting of their g-states, $N[S] = \{n[s] \mid n \in N\}$. Let g-node $n = \langle s, n' \rangle$. The *ancestors* of n are inductively defined as n' and all of the ancestors of n'. If n' is Null, then n has no ancestors. Let the *depth* of n, written $d(n)$, be the number of ancestors of n. In practice, a depth can be stored within a g-node to avoid having to compute it. If some set of g-nodes N have the same depth, then $d(N)$ is that depth, otherwise, it is undefined. Then n'' is the *m-th degree ancestor* of n if it is an ancestor of n and $d(n) - d(n'') = m$.

We say that n is *$\sim_g$-cyclic* if some g-node $n' \neq n$ is the m-th degree ancestor of n, $m \bmod |\mathcal{G}| = 0$, and $n[s] \sim_g n'[s]$, i.e. if a bisimilar ancestor occurs on the same agent's turn.

A p-node is a pair, $\langle \hat{s}, N \rangle$ where $\hat{s}$ is a p-state and N is a set of g-nodes. We use

square brackets to access the states of g-nodes: if p-node $\hat{n} = \langle \hat{s}', N \rangle$ then $\hat{n}[\hat{s}] = \hat{s}'$. We say that p-node $\langle \hat{s}, N \rangle$ *references* g-nodes N. Intuitively, referenced g-nodes represent the intersection of those states that are compatible with the p-node, i.e. that the s-agent at execution-time thinks may be the case, with those states that the planner considers may be possible at that time.

6.2 Procedures

Procedure 10 computes whether a g-node is $\sim_g$-cyclic.

```
1:  procedure IsCyclic(⟨s, n⟩)
2:      ancestor ← n
3:      d ← 1
4:      while ancestor ≠ Null do
5:          ⟨s′, n′⟩ ← ancestor
6:          if d mod |G| = 0 and s ∼_g s′ then
7:              return true
8:          ancestor ← n′
9:          d ← d + 1
10:     return false;
```

A g-node is expanded into a set of descendant g-nodes by applying an action. The g-node transition operator defines the application of an action to a g-node. It maps from pairs of g-nodes and actions to sets of g-nodes and employs the g-state transition operator.

$$T(\langle s, n \rangle, a) \coloneqq \{\langle s', \langle s, n \rangle\rangle \mid s' \in (s \times a)\} \tag{6.1}$$

A g-node n is an *or-node* if its depth corresponds to an s-agent's turn to act, i.e. if $\mathcal{G}[d(n)] \in \mathcal{G}_s$. It is an *and-node* otherwise, i.e. if it corresponds to an e-agent's turn. At an or-node the planner must decide which action the corresponding s-agent should perform, and so only a single action leading to the goal needs to be found. At an and-node, the planner

must consider all actions that the corresponding e-agent might perform, so all such actions must lead to the goal. By relying on the e-agent predictor models, the planner considers only those actions predicted by the model instead of all applicable actions. Thus, a g-node $n = \langle s, n' \rangle$ is *solved* if any one of the following conditions holds, where $i = \mathcal{G}[d(n)]$: either

- $s \models \varphi_{goal}$, or
- $i \in \mathcal{G}_s$ and there exists some action $a \in app(s, i)$ such that $T(n, a)$ is solved, or
- $i \in \mathcal{G}_e$ and for every action $a \in \beta(s, i)$, $T(n, a)$ is solved.

The DESCEND procedure takes a set of and-nodes (N) of the same depth and repeatedly (recursively) applies T using the actions predicted by β to continue transitioning to descendant and-nodes until reaching a depth where it is an s-agent's turn to act (i.e. until finding a set of or-nodes), and then returns those or-nodes. If a search depth limit D is reached before the next s-agent timestep, or if any path through the g-nodes is cyclic, DESCEND returns Null, indicating failure. If every path through the g-nodes reaches a goal before (or at) the next s-agent's turn, then DESCEND returns the empty set, indicating that the g-nodes it started with are solved. This is the "and" part of the search: it checks whether every predicted e-agent action leads to a goal.

Let N be a set of g-nodes of the same depth, $\varphi_{goal} \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$ be the goal formula, and L be the search depth limit.

1: **procedure** DESCEND(N, φ_{goal}, L)
2: **if** $d(N) > L$ **then**
3: **return** Null # failure
4: **for** $n \in N$ **do**
5: **if** ISCYCLIC(n) **then**
6: **return** Null # failure
7: define $i \coloneqq \mathcal{G}[d(N)]$,
8: **if** $i \in \mathcal{G}_s$ **then**
9: **return** $\{\langle s, n\rangle \in N \mid s \not\models \varphi_{goal}\}$ # or-nodes
10: $N' \leftarrow \varnothing$
11: **for** $n = \langle s, n'\rangle \in N$ such that $s \not\models \varphi_{goal}$ **do**
12: **for** $a \in \beta(s, i)$ **do** # query e-agent models
13: $N'' \leftarrow$ DESCEND($T(n, a), \varphi_{goal}, L$)
14: **if** N'' = Null **then**
15: **return** Null
16: $N' \leftarrow N' \cup N''$
17: **return** N' # descendant or-nodes

Thus given a set of g-nodes N, DESCEND returns a set of or-nodes N' representing the possible outcomes of actions performed by environment agents, or Null if a cycle is found or the depth limit is reached, such that either $N' = N$ or every n in N' has an ancestor in N.

The PERSPECTIVE procedure takes a set of g-nodes (N) of the same depth and computes a set of p-nodes ($\widehat{N}$) whose p-states model the current agent's perspective of $N[S]$. Differences between the g-nodes express plan-time uncertainty about the future, while each p-node expresses i's execution-time uncertainty. If i's execution-time uncertainty coincides with the planner's plan-time uncertainty, then the size of $\widehat{N}$ may be smaller than the size of N. For example, consider that $N = \{n, n'\}$, where $n[s]$ models that it is raining (r)

and that i does not know whether r, $n[s]$ models that it is sunny ($\neg r$) and that i does not know whether r. Since i cannot distinguish the two g-states, *Perspective* will output a single p-node, $(\hat{s}, \{n, n'\})$, where $n^i \sim_g n'^i$ and either $\hat{s} = n^i$ or $\hat{s} = n'^i$.

Let N be a set of g-nodes with the same depth.

1: **procedure** PERSPECTIVE(N)
2: $\quad i \leftarrow \mathcal{G}[d(N)]$
3: $\quad \widehat{N} \leftarrow \varnothing$
4: $\quad$ **for** $n = \langle s, n' \rangle \in N$ **do**
5: $\quad\quad$ **if** for any $\langle \hat{s}, N' \rangle \in \widehat{N}$, $s^i \sim_p \hat{s}$ **then**
6: $\quad\quad\quad N' \leftarrow N' \cup \{n\}$
7: $\quad\quad$ **else**
8: $\quad\quad\quad \widehat{N} \leftarrow \widehat{N} \cup \{\langle s^i, \{n\} \rangle\}$
9: $\quad$ **return** $\widehat{N}$

Thus PERSPECTIVE(N) returns a set of p-nodes $\widehat{N}$ representing i's perspectives of N, with $|\widehat{N}| > 1$ indicating plan-time uncertainty about i's execution-time knowledge or beliefs.

We now define the APPLY procedure that applies an action to a p-node to transition to a new set of p-nodes. This abstracts the g-node transitions, allowing search over the space of p-nodes. Given p-node $\langle \hat{s}, N \rangle$ and action a, APPLY first uses the DESCEND procedure to determine how the action's application will transition the ground states $N[S]$, resulting in a set of or-nodes at the next system agent's turn. The PERSPECTIVE procedure finds that next s-agent's perspective.

Let $\langle \hat{s}, N \rangle$ be a p-node, $a \in A$ an action, $\varphi_{goal} \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$ the goal formula, and L the search depth limit.

1: **procedure** APPLY($\langle \hat{s}, N \rangle$, a, φ_{goal}, L)
2: $N' \leftarrow \{T(n, a) \mid n \in N\}$
3: $N'' \leftarrow$ DESCEND(N', φ_{goal}, L)
4: **if** $N'' = Null$ **then**
5: **return** Null
6: **return** PERSPECTIVE(N'')

Figure 6.1 illustrates APPLY($\langle \hat{s}, N \rangle$, a,φ_{goal}, L) in a domain having one s-agent and one e-agent. In this example, action a is applied in p-state $\hat{s}$ at timestep t. Let the s-agent whose turn it is to act be $i = \mathcal{G}[t]$. The p-nodes p_1, p_2, and p_3 are shown with dashed borders, while g-nodes g_1 through g_9 have solid borders. The p-node $p_1 = \langle \hat{s}, N \rangle$ contains the starting p-state (before the action occurs). The dashed arrows from p_1 to g_1 and g_2 indicate that $N = \{g_1, g_2\}$. Nodes g_1 and g_2 are or-nodes, because at timestep t it is the s-agent's turn to act. The APPLY procedure begins using the g-node transition operator to transition to the next layer of g-states, as indicated by the solid arrows from g_1 and g_2 to g_3, g_4, and g_5. For example, $T(g_1 \times a) = \{g_3, g_4\}$. Nodes g_3, g_4, and g_5 are and-nodes. The solid arrows from g_3, g_4, and g_5 to nodes g_6 through g_9 indicate the application of the g-node transition operator by the DESCEND procedure, using actions determined by the predictive model β for the e-agent. For example, the edges from g_5 to g_8 and g_9 could either indicate that β returns two actions for state $g_5[s]$, or that it returns a single action with nondeterministic effects leading to two possible outcomes. The dashed arrows leading from the or-nodes at $t + 2$ to p-nodes p_2 and p_3 show the application of PERSPECTIVE to g-nodes g_6, g_7, g_8, and g_9. Thus, the planner determines that if the s-agent performs a at timestep t while she has perspective $p_1[\hat{s}]$, then after an intervening e-agent action the resulting actual g-state will be will be one of $g_6[s]$, $g_7[s]$, $g_8[s]$, or $g_9[s]$, and the s-agent's perspective will be one of $p_2[\hat{s}]$, $p_3[\hat{s}]$, or $p_4[\hat{s}]$,

With APPLY abstracting the various sources of nondeterminism (What is the current

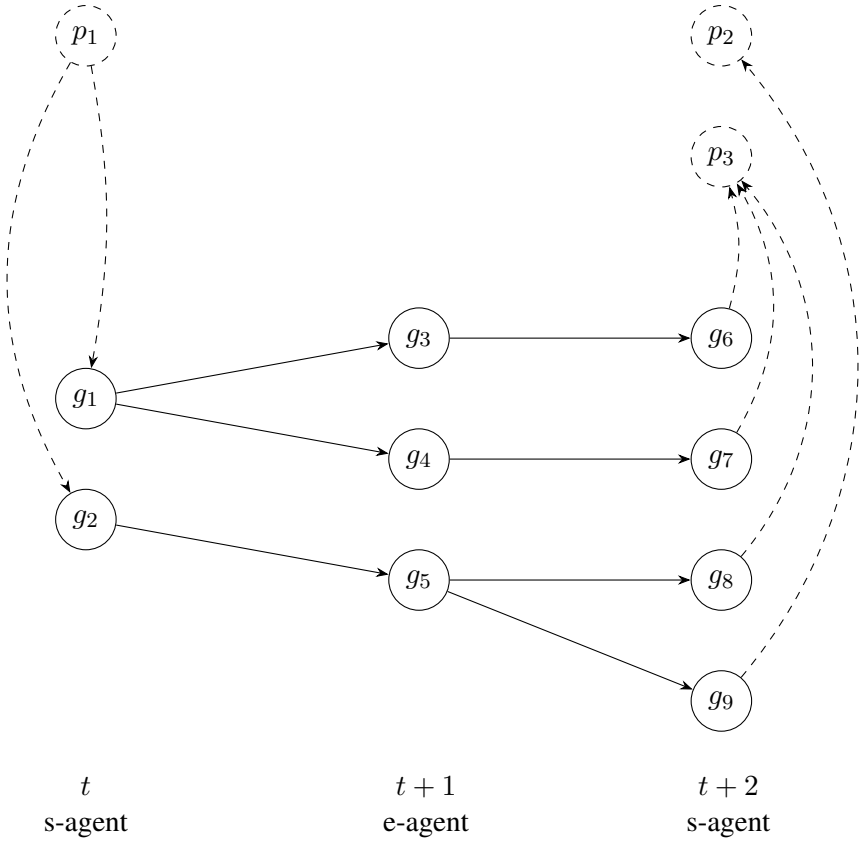


Figure 6.1: Example p-node transition

g-state? What will e-agents do? What will be the outcome of nondeterministic actions?), we can treat them all simply as nondeterminism in action effects. In this sense, applying a to p_1 yields three possible outcomes: p_2, p_3, and p_4. Taking this perspective, the EVALUATE procedure searches over p-nodes to find a policy mapping p-states (paired with timesteps) to actions. This is the "or" part of the search: it finds a single action for each p-node that leads to a goal.

Let $\widehat{N}$ be a (possibly Null) finite set of p-nodes, $\varphi_{goal} \in \mathcal{L}^{\mathcal{P}}_{\mathcal{GB}}$ the goal formula, and L the search depth limit.

1: **procedure** EVALUATE($\widehat{N}$,φ_{goal},L)
2: **if** $\widehat{N} = Null$ **then**
3: **return** $Null$
4: $i \leftarrow \mathcal{G}(t)$
5: $\Pi \leftarrow \varnothing$
6: **for** $\hat{n} \in \widehat{N}$ **do**
7: define $\pi(a) \coloneqq$ EVALUATE(APPLY($\hat{n}, a, \varphi_{goal}, L$), φ_{goal}, L)
8: **if** for every $a \in app(\hat{n}, i)$, $\pi(a) = Null$ **then**
9: **return** Null
10: select some $a \in app(\hat{n}, i)$ such that $\pi(a) \neq Null$
11: $\Pi \leftarrow \Pi \cup \{\langle\langle \hat{n}[\hat{s}], d(\hat{n}[N])\rangle, a\rangle\} \cup \pi(a)$
12: **return** Π

When APPLY finds only solution nodes, the recursive call to EVALUATE passes an empty set of p-nodes and the recursion bottoms out. Choosing which action to apply (line 10) can be accomplished in several ways. We could simply try applicable actions and use the first one that succeeds. This approach minimizes planning time by accepting the first solution found. Alternatively, we could try all actions, and chose the one with the shortest solution *in the worst case*, i.e. the action that leads to the solution that is shortest if all nondeterminism maximizes the number of actions needed to reach a goal. Another option is to try all actions and chose the one with the shortest solution *in the best case*, i.e. the action that leads to a solution that is shortest if all nondeterminism minimizes the number of actions needed to reach a goal. These latter two options require extra bookkeeping, such as returning extra information with a policy (its best- or worst-case length) or computing that information from a policy after it is returned.

Finally, a top-level procedure initializes the search algorithm, and then iteratively calls EVALUATE while incrementing the depth limit. Thus, this procedure runs iterated

depth-first search (IDFS) over p-nodes. Instead of a single start state, IDFS admits possibly multiple starting ground states to express plan-time uncertainty about the starting state.

Let S_0 be a set of starting ground states and φ_{goal} the goal formula.

1: **procedure** IDFS(S_0, φ_{goal})
2: $N_0 \leftarrow \{\langle s, Null\rangle \mid s \in S_0\}$ # initial set of g-nodes
3: $L \leftarrow 0$
4: **while** $\mathcal{G}[L] \notin \mathcal{G}_s$ **do**
5: $L \leftarrow L + 1$ # set L to the index of the first s-agent
6: $N \leftarrow \textsc{Descend}(N_0, \varphi_{goal}, L)$
7: **if** $N = Null$ **then**
8: **return** Null # failed from cycles before first s-agent turn
9: **if** $N = \varnothing$ **then**
10: **return** $\varnothing$ # trivial solution
11: $\widehat{N}_0 \leftarrow \textsc{Perspective}(N)$
12: $\Pi \leftarrow Null$
13: **while** $\Pi = Null$ **do** # until a solution is found
14: $\Pi \leftarrow \textsc{Evaluate}(\widehat{N}_0, \varphi_{goal}, L)$
15: $L \leftarrow L + 1$
16: **return** Π

6.3 Correctness

These propositions pertaining to the correctness of the algorithm are proved in the Appendix (Section 9.4).

Proposition 6.3.1. *If a policy exists that solves S_0 for φ, then* IDFS(S_0, φ) *returns a solution.*

Proposition 6.3.2. *If* IDFS(S_0, φ_{goal}) *returns policy* Π*, then* Π *is a solution.*

6.4 Discussion

This algorithm could employ any of the state and action systems discussed in this work. Using standard DEL, for example, g-states could be pointed Kripke structures, actions could be event models, and p-states could be multi-pointed Kripke structures with a perspective shift function such as $\langle\langle S, V, R_{j\in\mathcal{G}}\rangle, u\rangle^i = \langle\langle S, V, R_{j\in\mathcal{G}}\rangle, \{v \in S \mid \langle u, v\rangle \in R_i\}\rangle$. To employ one of the action languages, actions could be sets of statements from an action theory.

Let ND (action nondeterminism) be the maximum size of the set of g-states resulting from applying an action in a g-state: $ND \coloneqq Max_{a\in A, s\in S}(|T(s, A)|)$. Let APP be the maximum number of actions that are applicable in any state: $APP \coloneqq Max_{i\in\mathcal{G}, s\in S}(|app(s, i)|)$. Let GD (goal depth) be the maximum length of a g-node path from any start g-state to the first goal-satisfying g-node found. Let TR be the maximum complexity of computing a state transition $s \times a$. Let $PRED$ (predict) be the maximum complexity of querying an e-agent behavior prediction model. Let PER be the maximum complexity of computing a perspective shift, i.e. computing s^i from s. Let $BIS\text{-}g$ be the maximum cost of computing $\sim_g$. Let $BIS\text{-}p$ be the maximum cost of computing $\sim_p$.

Each g-node has at most $APP * ND$ children, so a search to depth GD has at most $(APP * ND)^{GD}$ leaves and $O(APP * ND)^{GD}$ g-nodes. Constructing each node consists of computing a state transition, checking for cycles, possibly consulting a predictor model (and-nodes), and possibly computing a perspective shift (or-nodes). Thus the cost of searching the g-nodes is $O((TR + (BIS\text{-}g * d) + PRED + PER) * (APP * ND)^{GD})$. In addition, when building p-nodes the algorithm checks for bisimulation between all elements of sets of p-states. This could be as much as $O(BIS\text{-}p * (APP * ND)^{GD})^2$. The complexity of searching over the g-nodes is necessary given the approach of searching over epistemic states without heuristics, although opting not to check for cycles could save time on some problems. The high cost of checking semantic equivalence to construct p-nodes is main limiting factor of this algorithm. This is especially so because computing equivalence can be an expensive operation for many state representations.

The g-state bisimulation operation is complete iff [for every φ in $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, $s \models \varphi$ iff

$s' \models \varphi$] implies that $s \sim_g s'$. Similarly the p-state bisimulation operation is complete iff [for every φ in $\mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, $\hat{s} \models \varphi$ iff $\hat{s}' \models \varphi$] implies that $\hat{s} \sim_p \hat{s}'$. If $\sim_g$ is complete, then the generated plan is acyclic, otherwise it may contain cycles. This is because the DESCEND procedure uses $\sim_g$ to check if a node is cyclic. If $\sim_g$ is not complete, the procedure may not find a g-node to be $\sim_g$-cyclic even though it has a semantically equivalent ancestor. If $\sim_p$ is complete, then a generated policy is strong, otherwise it is weak. This is because PERSPECTIVE uses $\sim_p$ to determine if multiple g-nodes should be associated with a single p-node. If for g-states s and s', agent i, and timestep t it is the case that $s^i \not\sim_p s'^i$ even though s^i and s'^i are semantically equivalent, then a solution policy may contain separate entries $\langle\langle s^i, t\rangle, a\rangle$ and $\langle\langle s'^i, t\rangle, a'\rangle$ where $a \neq a'$ where at execution time at t agent i cannot distinguish the two entries (their p-states are equivalent). If she executes a when the actual state of the task environment corresponds to s', or if she executes a' when the task environment corresponds to s, the results may not lead to the goal.

The generated policies map p-states to actions, ensuring that if two entries in the policy specify different actions, then the agent executing the policy will be able (at execution time) to distinguish (epistemically) the states of those entries, i.e. they are meaningful. Since p-states may be coarser than g-states, MECA can use finer state representations to generate a plan which is then executed by a robot that maintains a coarser state representation. For example, using Kripke structures for g-states and finite sets of propositions for p-states, the planner can reason about agents' nested knowledge (using the finer Kripke structures) while building a plan that can be executed by a robotic system, such as DIARC (Scheutz, Williams, et al., 2019), that represents states with (the coarser) sets of propositions.

As described, the algorithm does not distinguish between failing due to detecting cycles and failing due to reaching the maximum depth. Even if it is impossible to reach a goal because all actions are found to lead to cycles, the algorithm continues to run indefinitely while increasing the depth limit. It would be straightforward to instead quit early when all execution paths lead to cycles.

Chapter 7

MECAP

The Multiagent Epistemic Cooperation-Agnostic Planner (MECAP)[1] implements the cooperation-agnostic search algorithm presented in Chapter 6 and can be parameterized with various state and action representations, including those used in DEL, m$\mathcal{A}^*$, m$\mathcal{A}$-local, m$\mathcal{A}$-revise, and m$\mathcal{B}$. This chapter reports several evaluations of this system. First, a demonstration with a s-agent robot controlled by MECAP and an experimenter environment agent shows how the MECAP planning algorithm accommodates diverse inter-agent relationships, including cooperative, antagonistic, and self-interested dynamics. Second, MECAP generates plans used in a human study involving a robot system agent and a human subject environment agent. This experiment demonstrates the use of a predictor model for HRI and provides insight into social dynamics emerging from the use of MECAP. In both of these scenarios, the robot must influence the human's behavior in order to accomplish its goals. Third, MECAP is compared against several epistemic planners. While state-of-the-art epistemic planners out-perform MECAP in head-to-head comparisons, this section discusses important classes of problems that MECAP can solve, but which other planners cannot.

7.1 Cooperation-Agnostic Planning Demonstration

This section reports a demonstration of MECAP in three scenarios with cooperating, competing, and uninvolved humans. Ground states are pointed Kripke structures. An action

[1] github.com/DaveBuckingham/mecaPlanner

is a subset of an m$\mathcal{A}^*$ action theory, and m$\mathcal{A}^*$ gives the g-state transition function. Perspective states are finite sets of fluents, and the perspective shift function is defined as $\langle\langle W, V, R_{i\in\mathcal{G}}\rangle, u\rangle^i \coloneqq V(u)\rangle$.

This perspective shift function ignores some of the information contained in the g-state, and p-states are more coarsely granular than g-states. Since p-states are simply sets of propositions, so are the keys of the generated policy. This means that the planner uses Kripke structures with all of their semantic power (g-states), e.g. representing unbounded nested belief, to generate a policy that can be executed by an automated system that represents states as sets of propositions (p-states), as was the was the case in this demonstration.

7.1.1 Setup

This domain contains two agents: a robot (P) and a human (H). We installed our planner on a NAO robot running the ADE robotic architecture (Ferreira, Sequeira, and Ventura, 2019; Scheutz, 2006). The robot was able to speak, turn, walk forward, and sense obstacles. We placed the robot in a toy airfield domain (Figure 7.1) with tasks involving a cooperative human, a self-interested human, and an antagonistic human (each corresponding to a unique behavior model). A toy airplane is affixed to the robot by a string such that when the robot moves it pulls the airplane. The domain contains three locations: Airfield (AF), Hangar 1 (H1), and Hangar 2 (H2), each indicated by paper labels on the floor. There is a door separating Hangar 1 from the airfield (a cardboard box), but there is no door for Hangar 2. The propositional atoms $\mathcal{P}$ used in this domain are listed below.

- for every $i \in \{P, H\}$, $x \in \{A, H1, H2\}$, $at(i, x)$
- $door\text{-}is\text{-}open$
- $brownies$
- $did\text{-}request - door$

The following action theory (given in the sublists) defines the actions in the domain.

- P and H can move between AF and H1: for every $i, j \in \{P, H\}, i \neq j$ and every $x, y \in \{AF, H1\}, x \neq y$

  - executable $move(i, x, y)$ if $door\text{-}open$
  - $move(i, x, y)$ causes $\neg at(i, x)$
  - $move(i, x, y)$ causes $at(i, y)$
  - i observes $move(i, x, y)$
  - j observes $move(i, x, y)$ if $at(j, x) \vee at(j, y)$

- P and H can move between AF and H2: for every $i, j \in \{P, H\}, i \neq j$ and every $x, y \in \{AF, H2\}, x \neq y$
  - $move(i, x, y)$ causes $\neg at(i, x)$
  - $move(i, x, y)$ causes $at(i, y)$
  - i observes $move(i, x, y)$
  - j observes $move(i, x, y)$ if $at(j, x) \vee at(j, y)$

- the robot can request that the human open the door
  - $request\text{-}door$ causes $did\text{-}request - door$
  - P observes $request\text{-}door$
  - H observes $request\text{-}door$

- the robot can announce that there are brownies at the airfield
  - $announce\text{-}brownies$ announces $brownies$
  - P observes $announce\text{-}brownies$
  - H observes $announce\text{-}brownies$

- the human can open the door
  - $open\text{-}door$ causes $door\text{-}is\text{-}open$
  - P observes $door\text{-}is\text{-}open$
  - H observes $door\text{-}is\text{-}open$

- both agents can wait and do nothing

- P observes $wait$
- H observes $wait$

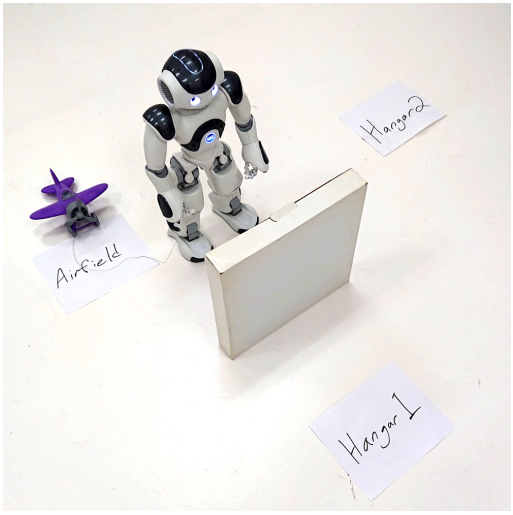


Figure 7.1: The toy airfield domain with the NAO robot

7.1.2 Teammate task

The teammate task demonstrates a collaborative relationship, with an e-agent prediction model that returns actions requested by the robot. Specifically

$$\beta_H(s) = \begin{cases} \{open\text{-}door\}, \text{ if } s \models did\text{-}request\text{-}door \\ \{wait\}, \text{ otherwise} \end{cases}$$

The start state is $\langle\langle W, V, R_P, R_H\rangle, u\rangle$ where $W = \{u\}$, $V(u) = \{at(P, AF)\}$, and $R_P = R_H = \{(u, u\}$.

The planner finds the policy shown in Table 7.1, where the robot first asks the human to open the door. According to the behavior model, this will cause the human to open the door. Then, with the door open, the robot pulls the airplane into the hangar.

In the demonstration, the robot senses the cardboard box and says "Please open the door to hangar one." Then the cardboard box representing the door is removed by the experimenter. Finally, sensing the absence of the door, the robot walks forward, pulling the toy airplane to the space marked "Hangar 1".

Table 7.1: Solution policy for the teammate scenario

state	timestep	action
$at(P, AF)$	0	$request\text{-}door$
$at(P, AF)$, $did\text{-}request\text{-}door$, $door\text{-}is\text{-}open$	2	$move(P, AF, H1)$

7.1.3 Self-interested task

The Bystander task demonstrates a self-interested relationship, requiring the robot to achieve implicit coordination with the human without any intention by the human to help the robot. The human will not open the door just because the robot asks. However, the human does have her own goal to eat any brownies. She doesn't know about the brownies located on the airfield, but if she did, she would move to the airfield, inadvertently helping the robot. The goal φ_{goal} is the same as in the *teammate* scenario: $at(P, H1)$. According to the e-agent prediction model, the human will move to the airfield if she believes that there are brownies there (opening the door if it is in the way), and will otherwise do nothing.

$$\beta_H(s) = \begin{cases} \{open\text{-}door\}, \text{ if } s \models \mathbf{B}_H(brownies) \wedge \neg door\text{-}is\text{-}open \\ \{move(H, H1, AF)\}, \text{ if } s \models \mathbf{B}_H(brownies) \wedge door\text{-}is\text{-}open \\ \{wait\}, \text{ otherwise} \end{cases}$$

The start state is $\langle\langle W, V, R_P, R_H\rangle, u\rangle$ where

- $W = \{u, v\}$,
- $V(u) = \{at(P, AF), at(H, H1), brownies\}$,
- $V(v) = \{at(P, AF), at(H, H1)\}$,
- $R_P = \{u, u\}, \{v, v\}$, and
- $R_H = \{(u, u\}, \{u, v\}, \{v, v\}, \{v, u\}$.

The planner returns a policy shown in Table 7.3, where the robot first announces that the brownies are on the airfield, which, according to the behavior model, will cause the human to open the door. Then, with the door open, the robot pulls the airplane into the hangar.

Table 7.2: Solution policy for the bystander scenario

state	timestep	action
$at(P, AF), at(H, H1), brownies,$	0	$announce\text{-}brownies$
$at(P, AF), at(H, H1), brownies, door\text{-}is\text{-}open$	2	$move(P, AF, H1)$

In the demonstration, the robot first senses the door and announces that "there are some brownies at the airfield." Then the cardboard box representing the door is removed by the experimenter. Finally, sensing the absence of the door, the robot walks forward, pulling the toy airplane to the space marked "Hangar 1".

7.1.4 Opponent task

The Bystander task demonstrates a self-interested relationship, requiring the robot to achieve implicit coordination with the human without any intention by the human to help the robot. The human will not open the door just because the robot asks. However, the human does have her own goal to eat any brownies. She doesn't know about the brownies located on the airfield, but if she did, she would move to the airfield, inadvertently helping the robot. The goal φ_{goal} is the same as in the *teammate* scenario: $at(P, H1)$. According to the e-agent prediction model, the human will move to the airfield if she believes that there are brownies there (opening the door if it is in the way), and will otherwise do nothing.

$$\beta_H(s) = \begin{cases} \{open\text{-}door\}, \text{ if } s \models \mathbf{B}_H(brownies) \wedge \neg door\text{-}is\text{-}open \\ \{move(H, H1, AF)\}, \text{ if } s \models \mathbf{B}_H(brownies) \wedge door\text{-}is\text{-}open \\ \{wait\}, \text{ otherwise} \end{cases}$$

The start state is $\langle\langle W, V, R_P, R_H\rangle, u\rangle$ where

- $W = \{u, v\}$,
- $V(u) = \{at(P, AF), at(H, H1), brownies\}$,
- $V(v) = \{at(P, AF), at(H, H1)\}$,
- $R_P = \{u, u\}, \{v, v\}$, and
- $R_H = \{(u, u\}, \{u, v\}, \{v, v\}, \{v, u\}$.

The planner returns a policy shown in Table 7.3, where the robot first announces that the brownies are on the airfield, which, according to the behavior model, will cause the human to open the door. Then, with the door open, the robot pulls the airplane into the hangar.

Table 7.3: Solution policy for the bystander scenario

state	timestep	action
$at(P, AF)$, $at(H, H1)$, $brownies$,	0	$announce\text{-}brownies$
$at(P, AF)$, $at(H, H1)$, $brownies$, $door\text{-}is\text{-}open$	2	$move(P, AF, H1)$

In the demonstration, the robot first senses the door and announces that "there are some brownies at the airfield." Then the cardboard box representing the door is removed by the experimenter. Finally, sensing the absence of the door, the robot walks forward, pulling the toy airplane to the space marked "Hangar 1".

7.1.5 Opponent task

In the Opponent task the robot's goal is to prevent the human from reaching the airplane. We add a time constraint goal requiring that the human remain separated from the airplane for at least one time step. The scenario begins at timestep 0 and each action (by either agent) increases the timestep by one. The goal φ_{goal} is ¬(at(human,X) ∧ at(plane,X)) for $X \in$ {hangar1, hangar2, airfield} and timestep ≥ 1.

$$\beta_H(s) = \begin{cases} \{open\text{-}door\}, \text{ if } s \models \mathbf{B}_H(brownies) \wedge \neg door\text{-}is\text{-}open \\ \{move(H, H1, AF)\}, \text{ if } s \models \mathbf{B}_H(brownies) \wedge door\text{-}is\text{-}open \\ \{wait\}, \text{ otherwise} \end{cases}$$

The start state is $\langle\langle W, V, R_P, R_H\rangle, u\rangle$ where $W = \{u\}$, $V(u) = \{at(P, AF), at(H, H1)\}$, $R_P = R_H = \{(u, u\}$.

The human model is:

for $X, Y \in$ {hangar1,hangar2,airfield} | $X \neq Y$ **do**

if at(human1,X) ∧ at(airplane,Y) **then**

Table 7.4: Solution policy for the opponent scenario

state:	at(robot,airfield), at(plane,airfield), at(human,hangar1)
action:	pull_plane(airfield,hangar2)

if $X, Y \in \{$hangar1,airfield$\}$ and $\neg$is_open(door) **then**

return open(door)

return *move(X,Y)*

return *wait()*

The planner returns a policy with a single entry, shown in Table 7.4, where the robot pulls the airplane to Hangar 2. The human does not observe this action because she is at Hangar 1, and by the time she reaches the airfield the airplane is at Hangar 2. Not knowing where the airplane is, she then waits at the airfield. In the demonstration, the robot turns ninety degrees to the left and then walks forward, pulling the toy airplane to the space marked "Hangar 2".

7.1.6 Discussion

This demonstration shows how MECAP can be applied to diverse teaming relations. It corresponds to three applications of Epistemic Planning identified by Baral, Bolander, et al. (2017): cooperative agents, purely adversarial games, and problems that are "neither purely cooperative nor purely adversarial." It also shows how, with MECAP, cooperative, competitive, and self-interested relationships can be expressed as relationships between the goals of different agents, which can be identical, distinct, overlapping, conflicting, or some combination thereof.

Each scenario involves explicit modeling of the human's belief state: the human will open the door only if she knows about the robot's request (*teammate* scenario) or about the brownies (*bystander* scenario). The *opponent* scenario requires the planner to model that the human knows about the airplane's starting position and will be unaware of the effects of unobserved robot actions (pulling the airplane to Hangar 2). Thus, our scenarios can only be solved by a planner that uses predictions of system agents' actions based on their beliefs.

A planner without models to predict the human's actions is unable to produce *safe* plans for any of the scenarios. For the *teammate* and *bystander* scenarios, the human could wait indefinitely and not open the door. It is only because the model predicts that the human will open the door in response to specific robot actions that the planner can find safe solutions. For the *opponent* scenario, the human could move to any of the locations, including Hangar 2, and there would be no policy for the robot that guarantees the human will not reach the plane. A second problem is the explosion of the size of the search space if the number of possible human actions are not constrained. Even in our extremely limited domain, there are three to four actions that the human *could* perform (has satisfied preconditions) in any given state, whereas our models predict only a single action. The more a predictive model reduces the branching factor of the state space search by constraining the likely human actions (while not failing to predict what the human actually does), the better this approach will scale to more complex domains.

7.2 Human-Robot Interaction Study

This section reports a human study first presented in Buckingham, Chita-Tegmark, and Scheutz (2020), where a teleoperated robot executes a planner-generated policy to influence the behavior of human participants. Since the robot is not capable of performing some of the actions necessary to achieve its goal, the robot tries to cause the human to perform those actions. For this study, the state representations are simply finite sets of propositions (things that are true in a state). Perspective states are equivalent to ground states and the perspective shift function trivially returns its input, i.e. it is assumed that environment agents have perfect knowledge and observation. The e-agent predictive model uses a basic forward-search planner to predict how the human will pursue her goals.

Ground states and perspective states are finite sets of atomic propositions, i.e. subsets of $\mathcal{P}$. We assume two agents: a robot R and a human H. An action a is a 3-tuple $\langle a_{\text{pre}}, a_{\text{add}}, a_{\text{del}} \rangle$, where a_{pre} is a set of precondition propositions, a_{add} is the add list: the set of propositions the action causes to be true, and a_{del} is the delete list: the set of propositions the action causes to be false. The g-state transition function is performs a^R in s. It is defined

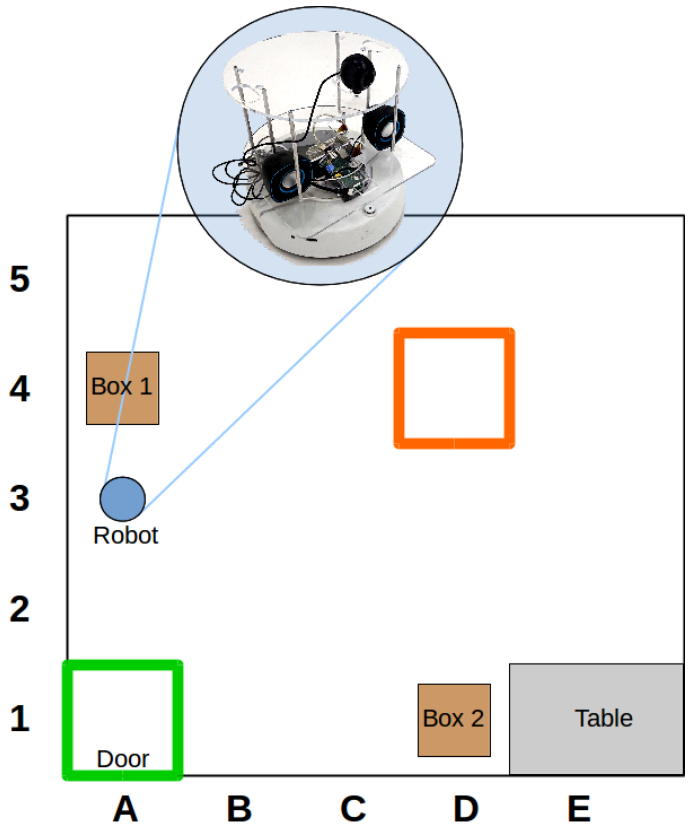


Figure 7.2: Setup for the HRI experiment, including the iRobot Create robotic platform.

as $s \times a := s \setminus a_{\text{del}} \cup a_{\text{add}}$. The perspective shift function is trivial, for every $i \in \mathcal{G}$, $s^i := s$.

7.2.1 Experiment

Figure 7.2 shows the experimental setup. The robot is an iRobot Create, a 32cm-diameter circular robot driven by two wheels, which was able to move, turn and push objects. Two three-cubic-foot ($18 \times 18 \times 16$ inches) empty cardboard boxes were positioned on the floor of a large room at equal distances (3 meters) from the door and from a target area. The human's starting position was at the door, marked with a green square drawn on the floor (space 1-A in the figure). The target area was marked with an orange square drawn on the floor (space 4-D in the figure). Three whiteboard markers were placed on top of Box 2 to incentivize the human to to pick up Box 1. A table was placed behind Box 2.

In the domain representation, each agent can move between adjacent spaces and both agents can occupy the same space. Agents act simultaneously. The human can pick up a box

Table 7.5: Solution policy for the box-moving scenario

state:	at(robot, spaceA3), at(human, spaceA1), at(box1, spaceA4), at(box2, spaceD1)
action:	push box1 to spaceA5 and move to spaceA4
state:	at(robot, spaceA4), at(human, spaceB1), at(box1, spaceA5), at(box2, spaceD1)
action:	move to spaceB4
state:	at(robot, spaceB4), at(human, spaceC1), at(box1, spaceA5), at(box2, spaceD1)
action:	move to spaceC4
state:	at(robot, spaceC4), at(human, spaceD1), at(box1, spaceA5), at(box2, spaceD1)
action:	move to spaceC3
state:	at(robot, spaceC3), at(human, spaceD1), at(box1, spaceA5), has(human,box)
action:	move to spaceC2
state:	at(robot, spaceC2), at(human, spaceD2), at(box1, spaceA5), has(human,box)
action:	move to spaceC1
state:	at(robot, spaceC1), at(human, spaceD2), at(box1, spaceA5), has(human, box)
action:	move to spaceD1

in a space she occupies. The robot cannot pick up boxes or move into a space containing a box, but can push boxes. When the robot pushes a box, it moves into the space where the box had been, and the box moves one space in the same direction as the robot moves (pushing isn't allowed if the there is no space for the box to move into). The task of the robot was to move to a position on the floor marked with an X, which was under Box 2. In order for the robot to reach that position, the human needed to move Box 2, which the robot was unable to push because of the wall and the table.

The human was given the following instructions by an experimenter: "Please enter the white room and stand in the green square by the door. Stay there until I say begin. There are two cardboard boxes on the floor. Do you see them? In addition to the green square where you are standing, there is an orange square on the floor. Do you see it? Your task is to retrieve one of the boxes, and place it on the floor in the orange square. Begin." All subjects responded affirmatively to both questions, and no other information about the robot or the planner was provided.

The predictive model β is a basic forward state-space search (breadth-first search) planner. Inputs are the human's goals (which do not change) and a world state (perfect knowledge is assumed). The model finds the set of minimal-cost plans, where a plan is a sequence of human actions that transitions the system to a state that satisfies the human's goals, and returns the set containing the first action of each of those plans. This model assumes that the human has perfect knowledge of the state of the system at all times, will

plan optimally, and will replan as necessary in response to unexpected state changes.

MECAP returns a policy (Table 7.5) that allows the robot to achieve its goal. Timesteps are omitted from the policy in this early version of MECA. From the start state, the robot pushes Box 1 to space A5. Now that Box 1 is further away from the human, β_H predicts that the human will move toward Box 2, and as the human moves toward Box 2, and eventually picks it up and moves toward the orange square, the robot moves to space D1.

The robot began moving as the experimenter said "begin". The robot was remotely controlled according to the policy in order to avoid unrelated robot control issues such as collision detection, orientation detection, and localization. Thus, the experimenter manually implemented the policy calculated by the planner. A camera mounted on the wall above the door allowed the experimenter controlling the robot to observe the robot's movement.

The model predicts that, with no robot intervention, the human will pick up either Box 1 or Box 2. However, in a state where Box 1 has been moved to space A5, the model predicts that the human will necessarily pick up Box 2 (since it is closer). Thus, the planner finds the safe solution to push Box 1, whereby the human is predicted pick up Box 2.

7.2.2 Results

Twenty adult subjects participated in the study (11 male, 8 female, 1 preferred not to answer, mean age = 31 years, standard deviation of age = 12.16 years). After consenting to participate in the study, subjects were guided to the experiment room. Following the completion of the experimental task, the subject filled out a questionnaire with the following questions: Which box did you retrieve? Did you change your mind about which box to retrieve? Did the robot cause you to change your mind about which box to retrieve? Why did you select the box(es) you did? Why do you think the robot behaved as it did?

Seventeen subjects moved Box 2, the remaining three moved Box 1. Of the subjects that moved Box 2, twelve said that the robot caused them to change their mind about which box to retrieve, although one of those specified that they "picked my choice before the robot started to move" (we interpret this to mean that the robot's presence, but not actions, influenced the subject's decision). The other eight subjects (five of whom moved Box 2, and 3 of whom moved box 1) said that the robot did not cause them to change their mind. Thus

Table 7.6: Selected responses to the question "why did you select the box(es) you did?"

Motivating factor	**Subject response**
physical obstruction	"Robot was between me and the box."
visibility	"It was the one directly in front of me." "Because it was the first one I saw."
box motion	"I didn't care to pick up a moving box."
perceived distance	"It [Box 2] seemed to me to be closer to the orange square in which I was supposed to place the box." "Visually box 1 seemed closer." "Even when the robot started to move, that box [Box 1] was closer."
markers on Box 2	"I initially chose box 1 because it didn't have a bunch of stuff on top." "It didn't have markers on it and so was easier to get and move, as I didn't need to move the markers." "Initially I was going to select box 1, because box 2 had markers on it which I didn't want to knock off."
reluctance to interfere with the robot	"As soon as robot went for first, I went for second." "The robot started moving toward box 1." "I retrieved box 2 because the robot was moving towards box 1."

we conclude that the robot's behavior successfully altered the human's actions to allow the robot to accomplish its goal in 11 out of 20 runs.

From subject responses to the question "why did you select the box(es) you did?" we discern six main motivating factors. These factors, and selected relevant subject responses, are reported in Table 7.6. In response to the question "why do you think the robot behaved as it did?" most subjects referenced the intentions of either researcher, the robot programmer, or the robot itself. Selected responses are reported in Table 7.7. Two of the runs suffered technical difficulties. In one run the robot halted after pushing Box 1 one meter and never turned or moved toward the Box 2 start location. In another run the robot failed to move altogether, and the subject "did not notice the robot." In both of these runs, the subject moved Box 2 and said that they did not change their mind.

Interestingly, two subjects thought that Box 1 was closer to them, even though the starting distances between each box and each square were equal and the robot caused Box 1 to move strictly away from the green square. This result reinforces the limitations of relying on *post factum* self-reporting to understand human decision-making. None of the subject responses indicate that they thought that the robot behaved as it did in order to incentivize the human to move Box 2.

Table 7.7: Selected responses to the question "Why do you think the robot behaved as it did?"

Assigned intention	Subject response
researcher	"To test how people perceive objects that are in use by robot, which would give a sense of how much people see robots as autonomous beings that have motivations for performing actions." "Perhaps the researchers are trying to see if anti-social behavior on the part of the robot causes people to change their courses of action."
programmer	"I assume it was programmed to push box 1." "It seemed programmed to push the box in that direction. . . "
robot	"It wanted to move the box to a new location." "It might have been stuck behind box 1, or it might have been intentionally pushing it somewhere. . . " "Maybe it also wanted to move the box? Maybe it was trying to help me move it?" "It seemed like it had some goal to move the boxes."

7.2.3 Discussion

Several subject responses suggest that they considered trade-offs between two or more factors. For example, one participant responded: "Initially I was going to select box 1, because box 2 had markers on it which I didn't want to knock off. When the robot moved box 1, I changed my mind because moving box 2 without knocking off the markers now seemed easier." Perhaps the most interesting trade-offs are between interfering with the robot and facing physical difficulties. It seems that some people preferred the cost of dealing with difficulties presented by the physical environment to the mental and emotional cost of dealing with dynamic agents that are harder to predict: "Because the robot was going after box 1 so I chose the other one. It was just easier."

Furthermore, it seems that not only did the robot model the humans, but the humans modeled the robot's goals and intentions: "It seemed like the robot was doing something with box 1, so it was easier to perform the task I was asked to do with box 2. That way I didn't have to compete with the robot." "The other box was in use by the robot so I felt bad about taking the box away from it, so I figured it would just be easier to move the other box." "It was easier to get Box 2 because the robot wasn't interfering [with] it."

It is remarkable that, even in this simple task and with a non-humanoid robot, participants had some degree of emotional involvement, for example "feeling bad about taking the box away." Participants described the robot with anthropomorphic terms, for

example discussing the robot's wants: "It did not have stuff on it but I almost changed my mind because the robot seemed to want my box but then I decided to do it anyway." Participants discuss interactions with the robot in ways similar to social interactions between humans, referring to inconveniencing the robot: "I did not want to disturb the robot." "I chose box 2 because I figured there was a reason there was a robot between me and box 1, and I didn't want to mess with it." One participant changed their mind twice, apparently motivated by such human considerations: "I initially chose box 1 because it didn't have a bunch of stuff on top. Then it seemed like the robot might be engaged in some sort of task with box 1, so I considered choosing box 2 as not to inconvenience it. Then it appeared the robot might be stuck on box 1 so I chose box 1 in the hope that I might also be doing the robot a favor in removing the obstruction." These results demonstrate the potential for a robot to use MECAP to influence what actions humans take and provide evidence of the utility of predictive models for future robots operating in social contexts.

7.3 Comparison to other planners

We evaluate MECAP alongside three other planners. EFP2.0 (Fabiano et al., 2020) conducts forward search over epistemic states represented by *possibilities*, non-well-founded sets that give a bisimulation-minimal representation of Kripke structures. It uses the m$\mathcal{A}^{\rho}$ action language to define actions and the start state. RP-MEP (Muise, Belle, Felli, McIlraith, Miller, A. R. Pearce, et al., 2022) compiles EP problems to classical planning problems which are then solved by an off-the-shelf classical planner. MEPK (Wan, Fang, and Liu, 2021) maintains formulas in alternating cover disjunctive form for use with the PrAO (To, Son, and Pontelli, 2011) search algorithm.

One planning problem that is useful for comparing epistemic planners is the Grapevine problem (Muise, Felli, et al., 2015), a combination of the simpler Corridor and Gossip problems. In Grapevine, a group of agents each have a secret. Each agent can move between some number of rooms and can announce her secret. All agents in the same room as an agent announcing her secret will come to know the secret. Parameters to the problem include the number of agents, the number of rooms, the starting condition, and goals. For

this comparison, we used three agents, A, B, and C, and two rooms, 1 and 2. The agents secrets are a, b and c respectively. All agents begin in one room and can move between the rooms freely. The goal is $\Box_A b \wedge \neg\Box_A c \wedge \Box_B c \wedge \neg\Box_B a \wedge \Box_C a \wedge \neg\Box_C b$.

In addition to agents as used in Chapter 6, MECA allows a second type of agent called *passive agents*. Passive agents do not act, but they are included in epistemic state representations. One use of passive agents is to relax the requirement for system agents to act in a prespecified order. Given a domain with multiple system agents, we create a single "root" agent to stand in as the sole system agent for the cooperation-agnostic search algorithm. Every action in the domain is annotated with the name of an s-agent who performs the action. We use this approach for the Grapevine benchmark because it involves multiple system agents who are not restricted to a turn order.

Our next benchmark is Finding-the-truth (Wan, Fang, and Liu, 2021). There is one agent, some number of boxes, and some number of rooms. Each box is placed in a room. The agent begins with incorrect beliefs about the locations of the boxes. She can move between rooms to observe the boxes, and the goal is for the agent to know the true locations of the boxes. We run this benchmark with one box and two rooms (1,2), and with two boxes and three rooms (2,3).

The Coin-in-the-box benchmark is based on Baral, G. Gelfond, Pontelli, et al. (2015), and has been evaluated elsewhere, e.g. in Wan, Fang, and Liu (2021). There are three agents (A, B, and C), although only A performs actions, and a box containing a coin. Actions available to agent A are to peek into the box and observe the coin, and to distract either of the other agents. When A peeks in the box, a distracted agent won't come to know that A knows how the coin lies. The goal is for A to know how the coin lies, for B to know that A knows how the coin lies, and for C not to know that A knows how the coin lies.

Table 7.8 shows the time taken to solve Grapevine.

Benchmark	MECAP	EFP2.0	RP-MEP	MEPK
Grapevine	101.52	18.12	0.69	0.076
Finding-the-truth(1,2)	0.507	0.007	0.094	0.023
Finding-the-truth(2,3)	0.938	0.015	0.118	0.081
Coin-in-the-box	0.524	0.008	0.068	0.055

Table 7.8: Time in seconds to solve the benchmarks.

MECAP is parameterized with the state and action representations given by m$\mathcal{B}$ (Chapter 5). It performs significantly worse on the benchmarks than the other planners (Table 7.8). This is unsurprising given the computational complexity of its search algorithm. The output from MECA for the Grapevine benchmark is shown in Table 7.9 (since the solution is sequential, we give a list of actions instead of a policy).

Table 7.9: Solution for Grapevine

timestep	action
0	move(C)
1	announce(B)
2	move(A)
3	announce(A)
4	move(C)
5	announce(C)

The other planners, except for RP-MEP, gave equivalent solutions. RP-MEP did not find an optimal solution. Its output requires eight actions, and is shown in Table 7.10. For the other problems, all planners gave equivalent minimal solutions. For Finding-the-truth, this involves moving from the agent's starting room to one of the other rooms and sensing which boxes are there. For Coin-in-the-box, the solution is for A to distract C and then peek in the box.

Table 7.10: RP-MEP solution for Grapevine

timestep	action
0	move_c_r1_r2
1	share_b_b_r1
2	move_a_r1_r2
3	move_c_r2_r1
4	share_c_c_r1
5	move_b_r1_r2
6	move_a_r2_r1
7	share_a_a_r1

While other planners outperform MECAP on these benchmarks, there are many problems that MECAP can solve but which the other planners cannot. Because problems such as Secret Distract (Section 3.2.4) and Bicycle-2 (Section 4.5) involve second-order false belief arising from false belief about action observability, m$\mathcal{A}^*$, and thus EFP2.0, which uses an action language (m$\mathcal{A}^\rho$) that is derived from m$\mathcal{A}^*$, cannot solve them. RP-MEP and MEPK cannot solve problems involving common knowledge, which MECAP can because it

uses relational structures to represent epistemic state. None of EFP2.0, RP-MEP, and MEPK support behavior prediction models for environment agents as are used in the Teammate (Section 7.1.2), Self-interested (Section 7.1.3), and Opponent (Section 7.1.4) tasks. Thus, the main advantage of MECAP with respect to other epistemic planners is the flexibility arising from the the combination of relational structures, novel action languages, and support for both system and environment agents.

Chapter 8

Discussion

Robots are increasingly embedded in human societies where they encounter human collaborators, potential adversaries, and even uninvolved by-standers. Such robots must plan to accomplish joint goals with teammates while avoiding interference from competitors, and possibly utilizing bystanders to advance the robot's goals. This requires formal representations of actions that include dynamic observability, epistemic and doxastic effects, false beliefs (including second-order false beliefs), and belief revision. The epistemic action languages m$\mathcal{A}$-local, m$\mathcal{A}$-revise, and m$\mathcal{B}$ provide these representations (each with their own strengths and trade-offs) while maintaining the expressive power of relational models in state representations. AI practitioners working to equip automated agents with the capacity to reason, plan, and act based on the changing beliefs and knowledge of multiple agents will benefit from such languages: tasks involving false beliefs and second-order false belief abound, from planning covert activities to identifying and correcting misinformation. In MECAP, the cooperation-agnostic epistemic search algorithm applies these languages to problems involving environment agents whom the planner does not control, and uses behavior prediction models to solve problems involving diverse teaming relations between system and environment agents. These include tasks involving deception, exploitation, coordination, cooperation, and helping, with belief and knowledge interacting in scenarios that other planners cannot handle.

It is possible to impose desired teaming dynamics when specifying the planner's goals and models. For example, assuming goal-oriented human behavior and a planner-based

predictive model, strictly cooperative robot behavior can be achieved by assigning any human goals to the robot. Oppositional dynamics, on the other hand, can be imposed by assigning the robot's goals to be the negation of the modeled human goals, or implicitly by arranging the domain representation such that the robot's goals and the human's goals are incompatible. In all cases, the planner searches for a policy that will allow the robot to accomplish its goals, possibly by making changes to the environment that will influence the (predicted) behavior of other agents.

This work does not address the question of where environment agent prediction models come from. Predictive models could be built within a *mental model* (Chakraborti and Kambhampati, 2018; Scheutz, 2013; Scheutz, DeLoach, and Adams, 2017) framework, involving a human's goals, perceptual and action capabilities, knowledge, team roles, attitudes, preferences, and other factors (Scheutz, DeLoach, and Adams, 2017). Such a model could use a cognitive architecture, such as ACT-R (Anderson, 1993) or SOAR (Laird, 2012), to emulate human cognitive processes. In domains where a human is likely to perform goal-oriented behavior, the model could involve a planner. For example, Talamadupula et al. (2014) construct a planning problem from a human's beliefs to predict the human's plan. This could be the same planner used by the robot, as in Muise, Belle, Felli, McIlraith, Miller, A. Pearce, et al. (2015), where a single planner switches perspective to reason "as if" it were another agent. In general, however, a human may plan non-optimally, or otherwise differently than the robot, and a specialized planner might work better. In addition to mental state information, a model could incorporate physiological factors such as human fatigue or eye gaze (C.-M. Huang and Mutlu, 2016), or a robot could infer a human's plans by overhearing conversations as in Kim et al. (2018).

One simple variation is to attach the environment agent models to ground states, instead of using global models. The planner could thus predict how the models will update when actions occur, and could plan for model updates. For example, a robot could make a plan that involves provoking behavior from a human that the robot can use to improve its predictive model of the human, before continuing to a part of the plan that requires the improved model. Predicting a human's behavior may involve inferring her goals, which might be unknown, or only partially known, to the robot. Geib, Craenen, and Petrick

(2016a,b) present a framework that combines plan recognition, goal inference, and planning for a robot. They use probabilistic techniques to determine the most likely human goal. While that approach assumes collaboration, future work can combine behavior model inference techniques with planning for non-cooperative tasks. A probabilistic model could express that some human actions are more likely than others. This could accommodate models based on distributions of possible human goals. Given a probabilistic human behavior model, the robot planner could prioritize exploring state spaces resulting from highly likely human actions over those caused by less likely actions.

While a plan is being executed it is possible that system agents will not behave as predicted by the predictor models, performing actions that the planner did not consider. This could cause the plan not to lead to a goal, i.e. a policy with no entry for the state resulting from an unexpected action. This problem is not unique to the system agent predictor models: any incongruence between the formal representations used by an automated planner and the reality of the task environment can cause a plan to fail. In general, any computational model of a real-world problem can be invalidated by observation. When this happens the robot may have to replan. In the event that replanning is necessary because a system agent did not act as expected, this may be an opportunity to update the predictor model before proceeding. The amount of nondeterminism in the predictor models will largely determine how effective and useful they are. At one extreme, a trivial model could simply predict all of the actions that are applicable for the system agent in a given state. In that case, the predictor can never be "wrong", and if it fails to predict an action that the system agent performs it is because of a problem with the action applicability formalism. Thus, there is a tradeoff between models returning fewer actions to help make the search more efficient, or returning more actions to decrease the chance that a system agent will act in a way that breaks a plan.

The modular nature of MECAP can facilitate future development along multiple paths. Action languages should continue to be developed, for example attempting to bring the completeness properties of DER (Rajaratnam and Thielscher, 2021) to the belief-revision-supporting plausibility preorders used by m$\mathcal{B}$. The cooperation-agnostic search algorithm could employ search heuristics, such as the domain-independent heuristics for Kripke structures presented in Le et al. (2018). It could also be developed to incorporate social

norms, as in Kasenberg and Scheutz (2018). Having been effectively used on robots within laboratory settings, MECAP should be deployed in open-world public settings to further evaluate its performance and social impact.

Chapter 9

Conclusion

Action languages allow us to easily define, reason about, and evaluate the actions available to automated systems. We have developed three new action languages for Epistemic Planning that improve upon prior systems. First, m$\mathcal{A}$-local updates the semantics of an existing language to admit a new class of problems: those involving second-order false beliefs. Second, m$\mathcal{A}$-revise gives an *ad hoc* state transition system that allows belief revision that is more nuanced and powerful than other epistemic action languages. Finally, m$\mathcal{B}$ uses plausibility preorders to perform belief revision with relational models of actions. We then presented the cooperation-agnostic search algorithm, which uses arbitrary state and action representations to plan for domains involving possibly multiple system agents and environment agents. This algorithm relies on oracular behavior prediction models to plan for the expected behavior of environment agents. This includes environment agent actions that can affect a system agent, as well as possibly planning for a system agent to influence the behavior of environment agents. This algorithm and the presented state and action representations have been implemented in the Multiagent Epistemic Cooperation-Agnostic Planner. We presented several evaluations of MECAP. First, an experimental demonstration showed how diverse relation types, from cooperation to competition, can emerge between a human and a robot equipped with MECAP. Second, we reported a study involving naive human participants that demonstrated how a robot can influence the behavior of humans in order to achieve its goals. Finally, a benchmark comparison against several state-of-the art planners revealed that MECAP does not perform competitively against other systems.

Indeed the strength of MECAP is not speed, but that it expands the bounds of what kinds of multi-agent epistemic planning problems can be modeled and solved.

The dynamics of human interrelations are nuanced and varied. We compete. We help each other. We go our own ways. We cooperate to achieve our shared aspirations. Entering into these social relations, not as mere tools but as intentional agents, robots will have to deliberate about the knowledge and the beliefs of the people around them, anticipate what others may do, consider how their own actions will impact the thoughts and the behaviors of others, and plan accordingly. Epistemic Planning is rapidly developing in the direction of effective deployment in AI systems. The distance between the theoretically robust semantic approach and the computationally efficient syntactic approach is closing, with increasingly efficient semantic models (e.g. Fabiano et al. (2020)) and increasingly powerful syntactic techniques (e.g. Lorini (2020)). This progress is moving us toward the development of systems that can tackle socially compelling epistemic problems in real-time: we will be interacting with robots that make plans involving not just physical objects (like current factory robots), but also the mental states of people, animals, and other robots. Humans are heavily involved in thinking about the thoughts of others, debating differences in beliefs, and building shared knowledge, and robots that engage in this realm of goal-oriented nested mental modeling will be a major step toward human-like artificial intelligence.

Bibliography

Alchourrón, Carlos E., Peter Gärdenfors, and David Makinson (1985). "On the Logic of Theory Change: Partial Meet Contraction and Revision Functions". In: *The Journal of Symbolic Logic* 50.2, pp. 510–530. ISSN: 00224812.

Andersen, Mikkel Birkegaard, Thomas Bolander, Hans van Ditmarsch, et al. (2013). "Bisimulation for single-agent plausibility models". In: *Australasian Joint Conference on Artificial Intelligence*. Springer, pp. 277–288.

Andersen, Mikkel Birkegaard, Thomas Bolander, and Martin Holm Jensen (2012). "Conditional Epistemic Planning". In: *Logics in Artificial Intelligence*. Ed. by Luis Fariñas del Cerro, Andreas Herzig, and Jérôme Mengin. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 94–106. ISBN: 978-3-642-33353-8.

— (Oct. 2013). "Don't Plan for the Unexpected: Planning Based on Plausibility Models". In: *Logique et Analyse* 58. DOI: 10.2143/LEA.230.0.3141807.

Anderson, John R (1993). *Rules of the mind*. New York: Psychology Press. ISBN: 9781315806938.

Arslan, Burcu, Niels Taatgen, and Rineke Verbrugge (2013). "Modeling developmental transitions in reasoning about false beliefs of others". In: *Proceedings of the 12th International Conference on Cognitive Modeling, Ottawa: Carleton University*. Citeseer, pp. 77–82.

Asimov, Isaac (1950). *I, Robot*. New York: Gnome Press.

Aucher, Guillaume (2003). "A combined system for update logic and belief revision. Master's thesis". In:

Baltag, Alexandru, Lawrence S Moss, and Slawomir Solecki (1998). "The logic of common knowledge, public announcements, and private suspicions". In: *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98)*. Ed. by Itzhak Gilboa. San Francisco, CA: Morgan Kaufmann, pp. 43–56.

Baltag, Alexandru and Sonja Smets (2006). "Dynamic belief revision over multi-agent plausibility models". In: *Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision Theory (LOFT7)*. Vol. 6. University of Liverpool, pp. 11–24.

— (2008). *A qualitative theory of dynamic interactive belief revision*.

Baral, Chitta, Thomas Bolander, et al. (2017). "Epistemic Planning (Dagstuhl Seminar 17231)". In: *Dagstuhl Reports* 7.6. Ed. by Chitta Baral et al., pp. 1–47. ISSN: 2192-5283. DOI: 10.4230/DagRep.7.6.1.

Baral, Chitta and Gregory Gelfond (2011). "On representing actions in multi-agent domains". In: pp. 213–232.

Baral, Chitta, Gregory Gelfond, Enrico Pontelli, et al. (2010). "Logic programming for finding models in the logics of knowledge and its applications: A case study". In: *Theory and Practice of Logic Programming* 10.4-6, pp. 675–690.

— (2015). "An Action Language for Multi-Agent Domains: Foundations". In: *CoRR* abs/1511.01960.

— (2022). "An action language for multi-agent domains". In: *Artificial Intelligence* 302, p. 103601. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2021.103601.

Baral, Chitta, Gregory Gelfond, Tran Cao Son, et al. (2010). "Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. AAMAS 2010, pp. 259–266.

Baral, Chitta, Tran Cao Son, and Enrico Pontelli (2010). "Reasoning about Multi-agent Domains Using Action Language $\mathcal{C}$: A Preliminary Study". In: *Computational Logic in Multi-Agent Systems*. Ed. by Jürgen Dix, Michael Fisher, and Peter Novák. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 46–63. ISBN: 978-3-642-16867-3.

Ben Larbi, Ramzi, Sébastien Konieczny, and Pierre Marquis (2007). "Extending Classical Planning to the Multi-agent Case: A Game-Theoretic Approach". In: *Symbolic and*

Quantitative Approaches to Reasoning with Uncertainty. Ed. by Khaled Mellouli. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 731–742. ISBN: 978-3-540-75256-1.

Benthem, Johan van and Sonja Smets (2015). "Dynamic logics of belief change". In: *Handbook of Epistemic Logic*. Ed. by Hans van Ditmarsch et al. College Publications. Chap. 7, pp. 213–393.

Bolander, Thomas (Jan. 2014). "Seeing is Believing: Formalising False-Belief Tasks in Dynamic Epistemic Logic". In: *CEUR Workshop Proceedings*. Vol. 1283.

— (Mar. 2017). "A Gentle Introduction to Epistemic Planning: The DEL Approach". In: *Electronic Proceedings in Theoretical Computer Science* 243, pp. 1–22. ISSN: 2075-2180. DOI: 10.4204/eptcs.243.1.

— (2018). "Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic". In: *Jaakko Hintikka on knowledge and game-theoretical semantics*. Springer, pp. 207–236.

Bolander, Thomas et al. (2020). *Tutorial on Epistemic Planning*. The Thirtieth International Conference on Automated Planning and Scheduling (ICAPS-2020). https://www.youtube.com/watch?v=lmimHMB3X5M.

Booth, Serena et al. (2017). "Piggybacking Robots: Human-Robot Overtrust in University Dormitory Security". In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '17. Vienna, Austria: ACM, pp. 426–434. ISBN: 978-1-4503-4336-7. DOI: 10.1145/2909824.3020211.

Bowling, Michael, Rune Jensen, and Manuela Veloso (2003). "A formalization of equilibria for multiagent planning". In: *IJCAI*, pp. 1460–1462.

— (2006). "Multi-Agent Planning in the Presence of Multiple Goals". In: *Planning in Intelligent Systems: Aspects, Motivations and Methods*. John Wiley and Sons, Ltd. Chap. 10, pp. 301–325. ISBN: 9780471781264. DOI: 10.1002/0471781266.ch10.

Brafman, Ronen I. and Carmel Domshlak (2008). "From One to Many: Planning for Loosely Coupled Multi-Agent Systems". In: *Proceedings of the Eighteenth International Conference on International Conference on Automated Planning and Scheduling*. ICAPS'08. Sydney, Australia: AAAI Press, pp. 28–35. ISBN: 9781577353867.

Brafman, Ronen I., Carmel Domshlak, et al. (2009). "Planning Games." In: *IJCAI*, pp. 73–78.

Braüner, Torben, Patrick Blackburn, and Irina Polyanskaya (2016). "Second-order false-belief tasks: Analysis and formalization". In: *International Workshop on Logic, Language, Information, and Computation*. Springer, pp. 125–144.

— (2020). "Being Deceived: Information Asymmetry in Second-Order False Belief Tasks". In: *Topics in Cognitive Science* 12.2, pp. 504–534. DOI: https://doi.org/10.1111/tops.12422.

Buckingham, David, Meia Chita-Tegmark, and Matthias Scheutz (2020). "Robot Planning with Mental Models of Co-present Humans". In: *International Conference on Social Robotics*. Springer, pp. 566–577.

Buckingham, David, Daniel Kasenberg, and Matthias Scheutz (2020). "Simultaneous Representation of Knowledge and Belief for Epistemic Planning with Belief Revision". In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. Vol. 17, pp. 172–181.

Buckingham, David and Matthias Scheutz (May 2017). "Getting help without asking: Stigmergic planning for human-robot collaboration (Extended Abstract)". In: AAMAS'17. Ed. by S. Das et al.

Buzing, Pieter et al. (Mar. 2006). "Coordinating Self-interested Planning Agents". In: *Autonomous Agents and Multi-Agent Systems* 12.2, pp. 199–218. ISSN: 1573-7454. DOI: 10.1007/s10458-005-6104-4.

Chakraborti, Tathagata, Gordon Briggs, et al. (Sept. 2015). "Planning for serendipity". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5300–5306. DOI: 10.1109/IROS.2015.7354125.

Chakraborti, Tathagata and Subbarao Kambhampati (2018). "Algorithms for the Greater Good! On Mental Modeling and Acceptable Symbiosis in Human-AI Collaboration". In: *CoRR* abs/1801.09854.

Chakraborti, Tathagata, Subbarao Kambhampati, et al. (2017). "AI challenges in human-robot cognitive teaming". In: *arXiv preprint arXiv:1707.04775*.

Chen, Min et al. (2018). “Planning with Trust for Human-Robot Collaboration”. In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’18. Chicago, IL, USA: ACM, pp. 307–315. ISBN: 978-1-4503-4953-6. DOI: 10.1145/3171221.3171264.

Chidambaram, Vijay, Yueh-Hsuan Chiang, and Bilge Mutlu (Mar. 2012). “Designing persuasive robots: How robots might persuade people using vocal and nonverbal cues”. In: *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 293–300. DOI: 10.1145/2157689.2157798.

Cirillo, Marcello, Lars Karlsson, and Alessandro Saffiotti (Dec. 2010). “Human-aware Task Planning: An Application to Mobile Robots”. In: *ACM Trans. Intell. Syst. Technol.* 1.2, 15:1–15:26. ISSN: 2157-6904. DOI: 10.1145/1869397.1869404.

De Lima, Tiago (2007). “Optimal methods for reasoning about actions and plans in multi-agent systems”. PhD thesis. Université de Toulouse, Université Toulouse III-Paul Sabatier.

Devin, Sandra and Rachid Alami (2016). “An Implemented Theory of Mind to Improve Human-Robot Shared Plans Execution”. In: *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. HRI ’16. Christchurch, New Zealand: IEEE Press, pp. 319–326. ISBN: 978-1-4673-8370-7.

Ditmarsch, Hans van (Nov. 2005). “Prolegomena to Dynamic Logic for Belief Revision”. In: *Synthese* 147.2, pp. 229–275. ISSN: 1573-0964. DOI: 10.1007/s11229-005-1349-7.

Ditmarsch, Hans van, Wiebe van der Hoek, and Barteld Kooi (2004). “Public announcements and belief expansion”. In: *Proceedings of AiML-2004 (Advances in Modal Logic), University of Manchester*, pp. 62–73.

Ditmarsch, Hans van and Barteld Kooi (2008). “Semantic results for ontic and epistemic change”. In: *Logic and the foundations of game and decision theory (LOFT 7)* 3, pp. 87–117.

Ditmarsch, Hans van and Willem Labuschagne (2003). “A multimodal language for revising defeasible beliefs”. In: *Proceedings of the 12th international congress of logic, methodology, and philosophy of science (LMPS)*, pp. 140–141.

Dragan, Anca D. (2017). *Robot Planning with Mathematical Models of Human State and Action*. DOI: 10.48550/ARXIV.1705.04226.

Eijck, Jan van (2017). "Public announcements and public lies". In: Technical report, Lying Workshop.

Engesser, Thorsten, Thomas Bolander, et al. (Mar. 2017). "Cooperative Epistemic Multi-Agent Planning for Implicit Coordination". In: *Electronic Proceedings in Theoretical Computer Science* 243, pp. 75–90. ISSN: 2075-2180. DOI: 10.4204/eptcs.243.6.

Engesser, Thorsten, Robert Mattmüller, et al. (July 2018). "Game Description Language and Dynamic Epistemic Logic Compared". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, pp. 1795–1802. DOI: 10.24963/ijcai.2018/248.

Fabiano, Francesco et al. (June 2020). "EFP 2.0: A Multi-Agent Epistemic Solver with Multiple E-State Representations". In: *Proceedings of the International Conference on Automated Planning and Scheduling* 30.1, pp. 101–109.

Fagin, Ronald et al. (1995). *Reasoning About Knowledge*. Cambridge, MA, London, England: MIT Press.

Ferreira, Maria Isabel Aldinhas, João Silva Sequeira, and Rodrigo Ventura (2019). *Cognitive Architectures*. Springer.

Fikes, Richard E. and Nils J. Nilsson (1971). "Strips: A new approach to the application of theorem proving to problem solving". In: *Artificial Intelligence* 2.3, pp. 189–208. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(71)90010-5.

Fisac, Jaime F. et al. (2018). "Probabilistically Safe Robot Planning with Confidence-Based Human Predictions". In: *CoRR* abs/1806.00109.

Fischer, Michael J. and Richard E. Ladner (1979). "Propositional dynamic logic of regular programs". In: *Journal of Computer and System Sciences* 18.2, pp. 194–211.

Flobbe, Liesbeth et al. (2008). "Children's application of theory of mind in reasoning and language". In: *Journal of Logic, Language and Information* 17.4, pp. 417–442.

Geib, Christopher, Bart Craenen, and Ronald P. A. Petrick (July 2016a). "Combining Plan Recognition, Goal Reasoning, and Planning for Cooperative Task Behaviour". In:

Proceedings of the IJCAI 2016 Workshop on Goal Reasoning. New York City, New York, USA.

Geib, Christopher, Bart Craenen, and Ronald P. A. Petrick (June 2016b). "Generating Collaborative Behaviour through Plan Recognition and Planning". In: *Proceedings of the ICAPS 2016 Workshop on Distributed and Multi-Agent Planning (DMAP)*. London, England, United Kingdom, pp. 98–105.

Gelfond, Michael and Yulia Kahl (2014). *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press.

Gelfond, Michael and Vladimir Lifschitz (1998). "Action languages". In: Linköping Electronic Articles in Computer and Information Science 3.16. ISSN: 1401-9841.

Gerbrandy, Jelle (1999). "Dynamic Epistemic Logic". In: *Logic, Language and Information* 2. Ed. by L.S. Moss, J. Ginzburg, and M. de Rijke, pp. 67–84.

Ghallab, Malik, Adele Howe, et al. (1998). "PDDL-the planning domain definition language". In: *AIPS-98 Planning Committee*.

Ghallab, Malik, Dana Nau, and Paolo Traverso (2004). *Automated Planning: theory and practice*. Elsevier.

— (2016). *Automated Planning and Acting*. Cambridge University Press. DOI: 10.1017/CBO9781139583923.

Görür, O. Can et al. (2018). "Social Cobots: Anticipatory Decision-Making for Collaborative Robots Incorporating Unexpected Human Behaviors". In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '18. Chicago, IL, USA: ACM, pp. 398–406. ISBN: 978-1-4503-4953-6. DOI: 10.1145/3171221.3171256.

Gray, Jesse and Cynthia Breazeal (Aug. 2014). "Manipulating Mental States Through Physical Action". In: *International Journal of Social Robotics* 6.3, pp. 315–327. ISSN: 1875-4805. DOI: 10.1007/s12369-014-0234-2.

Herzig, Andreas, Jérôme Lang, et al. (2000). "A logic for planning under partial observability". In: *AAAI/IAAI*, pp. 768–773.

Herzig, Andreas, Emiliano Lorini, et al. (2010). “A logic of trust and reputation”. In: *Logic Journal of the IGPL* 18.1, pp. 214–244.

Hintikka, Jaakko (1962). “Knowledge and Belief: An Introduction to the Logic of the Two Notions”. In: *Journal of Symbolic Logic* 29.3, pp. 132–134. DOI: 10.2307/2271621.

Huang, Chien-Ming and Bilge Mutlu (2016). “Anticipatory Robot Control for Efficient Human-Robot Collaboration”. In: *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. HRI ’16. Christchurch, New Zealand: IEEE Press, pp. 83–90. ISBN: 978-1-4673-8370-7.

Huang, Xiao et al. (2017). “A General Multi-agent Epistemic Planner Based on Higher-order Belief Change”. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. IJCAI ’17. Melbourne, Australia: AAAI Press, pp. 1093–1101. ISBN: 978-0-9992411-0-3.

Kasenberg, Daniel and Matthias Scheutz (Apr. 2018). “Norm Conflict Resolution in Stochastic Domains”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1. DOI: 10.1609/aaai.v32i1.11295.

Katsuno, Hirofumi and Alberto O. Mendelzon (1992). “On the difference between updating a knowledge base and revising it”. In: *Belief Revision*. Ed. by PeterEditor Gärdenfors. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, pp. 183–203. DOI: 10.1017/CBO9780511526664.007.

Kim, Joseph et al. (July 2018). “Learning to Infer Final Plans in Human Team Planning”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, pp. 4771–4779. DOI: 10.24963/ijcai.2018/663.

Kraus, Sarit and Daniel Lehmann (1988). “Knowledge, belief and time”. In: *Theoretical Computer Science* 58.1-3, pp. 155–174.

Kulkarni, Anagha, Siddharth Srivastava, and Subbarao Kambhampati (2018). “Implicit Robot-Human Communication in Adversarial and Collaborative Environments”. In: *CoRR* abs/1802.06137.

Laird, John E. (2012). *The Soar cognitive architecture*. MIT press.

Le, Tiep et al. (2018). “EFP and PG-EFP: Epistemic Forward Search Planners in Multi-Agent Domains”. In: *Proceedings of the twenty-eighth international conference on automated planning and scheduling*.

Levesque, Hector J (1996). “What is planning in the presence of sensing?” In: *Proceedings of the National Conference on Artificial Intelligence*, pp. 1139–1146.

Lorini, Emiliano (2020). “Rethinking epistemic logic with belief bases”. In: *Artificial Intelligence* 282, p. 103233.

Lorini, Emiliano and Éloan Rapion (2022). “Logical Theories of Collective Attitudes and the Belief Base Perspective”. In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 833–841.

Lorini, Emiliano and Francois Schwarzentruber (2021). “Multi-Agent Belief Base Revision.” In: *IJCAI*, pp. 1959–1965.

Miller, Tim, Paolo Felli, et al. (2016). “’Knowing whether’in proper epistemic knowledge bases”. In: *Thirtieth AAAI Conference on Artificial Intelligence*.

Miller, Tim and Christian Muise (2016). “Belief Update for Proper Epistemic Knowledge Bases”. In: *IJCAI*, pp. 1209–1215.

Milliez, G. et al. (Mar. 2016). “Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring”. In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 43–50. DOI: 10.1109/HRI.2016.7451732.

Muise, Christian, Vaishak Belle, Paolo Felli, Sheila McIlraith, Tim Miller, Adrian Pearce, et al. (2015). “Planning Over Multi-Agent Epistemic States: A Classical Planning Approach”. In: *Proceedings of AAAI’12, the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Austin, Texas, pp. 3327–3334. ISBN: 0-262-51129-0.

Muise, Christian, Vaishak Belle, Paolo Felli, Sheila McIlraith, Tim Miller, Adrian R. Pearce, et al. (2022). “Efficient multi-agent epistemic planning: Teaching planners about nested belief”. In: *Artificial Intelligence* 302, p. 103605. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2021.103605.

Muise, Christian, Paolo Felli, et al. (2015). "Leveraging FOND Planning Technology to Solve Multi-Agent Planning Problems". In: *Workshop on Distributed and Multi-Agent Planning (DMAP'15)*.

Nikolaidis, Stefanos, David Hsu, and Siddhartha Srinivasa (2017). "Human-robot mutual adaptation in collaborative tasks: Models and experiments". In: *The International Journal of Robotics Research* 36.5-7, pp. 618–634. DOI: 10.1177/0278364917690593.

Nikolaidis, Stefanos, Minae Kwon, et al. (Nov. 2018). "Planning with Verbal Communication for Human-Robot Collaboration". In: *ACM Trans. Hum.-Robot Interact.* 7.3, 22:1–22:21. ISSN: 2573-9522. DOI: 10.1145/3203305.

Nikolaidis, Stefanos, Yu Xiang Zhu, et al. (2017). "Human-robot mutual adaptation in shared autonomy". In: *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, pp. 294–302.

Nissim, Raz, Ronen I. Brafman, and Carmel Domshlak (2010). "A General, Fully Distributed Multi-agent Planning Algorithm". In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*. AAMAS '10. Toronto, Canada: International Foundation for Autonomous Agents and Multiagent Systems, pp. 1323–1330. ISBN: 978-0-9826571-1-9.

Pednault, Edwin P. (1987). "Formulating multiagent, dynamic-world problems in the classical planning framework". In: *Reasoning about actions & plans*. Elsevier, pp. 47–82.

Premack, David and Guy Woodruff (1978). "Does the chimpanzee have a theory of mind?" In: *Behavioral and Brain Sciences* 1.4, pp. 515–526. DOI: 10.1017/S0140525X00076512.

Prunera, Jordán Jaume Magí (2017). "Non-Cooperative Games for Self-Interested Planning Agents". PhD thesis. Universitat Politècnica de València. DOI: 10.4995/Thesis/10251/90417.

Rajaratnam, David and Michael Thielscher (2021). "Representing and Reasoning with Event Models for Epistemic Planning". In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. Vol. 18, pp. 519–528.

Rosenthal, Stephanie and Manuela Veloso (July 2012). "Mobile Robot Planning to Seek Help with Spatially-Situated Tasks". In: *Proceedings of AAAI'12, the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Toronto, Canada.

Rosenthal, Stephanie, Manuela Veloso, and Anind Dey (Mar. 2011a). "Hello? Is Someone in this Office Available to Help Me? Proactively Seeking Help from Spatially-Situated Humans". In: *Young Pioneers Workshop, International Conference on Human Robot Interaction (HRI 2011)*. Invited talk.

— (Aug. 2011b). "Learning Accuracy and Availability of Humans who Help Mobile Robots". In: *Proceedings of AAAI'11, the Twenty-Fifth Conference on Artificial Intelligence*. San Francisco, CA.

Sadigh, Dorsa et al. (2016). "Planning for Autonomous Cars that Leverage Effects on Human Actions". In: *Robotics: Science and Systems*.

Sandoval, Eduardo Benítez, Jürgen Brandstetter, and Christoph Bartneck (2016). "Can a Robot Bribe a Human?: The Measurement of the Negative Side of Reciprocity in Human Robot Interaction". In: *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. HRI '16. Christchurch, New Zealand: IEEE Press, pp. 117–124. ISBN: 978-1-4673-8370-7.

Saunderson, Shane P and Goldie Nejat (2021). "Persuasive robots should avoid authority: The effects of formal and real authority on persuasion in human-robot interaction". In: *Science Robotics* 6.58, eabd5186.

Scheutz, Matthias (2006). "ADE - Steps Towards a Distributed Development and Runtime Environment for Complex Robotic Agent Architectures". In: *Applied Artificial Intelligence* 20.4-5.

— (2013). "Computational Mechanisms for Mental Models in Human-Robot Interaction". In: *HCI International*, pp. 304–312.

Scheutz, Matthias, Scott DeLoach, and Julie Adams (2017). "A Framework for developing and using shared mental models in human-agent teams". In: *Journal of Cognitive Engineering and Decision Making* 11.3, pp. 203–224.

Scheutz, Matthias, Evan Krause, et al. (2017). “Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture”. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*.

Scheutz, Matthias, Thomas Williams, et al. (2019). “An overview of the distributed integrated cognition affect and reflection diarc architecture”. In: *Cognitive architectures*, pp. 165–193.

Siegel, M., C. Breazeal, and M. I. Norton (Oct. 2009). “Persuasive Robotics: The influence of robot gender on human behavior”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2563–2568. DOI: 10.1109/IROS.2009.5354116.

Son, Tran Cao and Chitta Baral (2001). “Formalizing sensing actions—A transition function based approach”. In: *Artificial Intelligence* 125.1-2, pp. 19–91.

Son, Tran Cao, Enrico Pontelli, Chitta Baral, et al. (2014). “Finitary S5-Theories”. In: *Logics in Artificial Intelligence*. Ed. by Eduardo Fermé and João Leite. Cham: Springer International Publishing, pp. 239–252. ISBN: 978-3-319-11558-0.

Son, Tran Cao, Enrico Pontelli, Ngoc-Hieu Nguyen, et al. (2014). “Formalizing negotiations using logic programming”. In: *ACM Transactions on Computational Logic (TOCL)* 15.2, pp. 1–30.

Son, Tran Cao, Enrico Pontelli, and Chiaki Sakama (2009). “Logic Programming for Multiagent Planning with Negotiation”. In: *Logic Programming*. Ed. by Patricia M. Hill and David S. Warren. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 99–114. ISBN: 978-3-642-02846-5.

Son, Tran Cao and Chiaki Sakama (2010). “Reasoning and Planning with Cooperative Actions for Multiagents Using Answer Set Programming”. In: *Declarative Agent Languages and Technologies VII*. Ed. by Matteo Baldoni et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 208–227. ISBN: 978-3-642-11355-0.

Spohn, Wolfgang (2012). *The laws of belief: Ranking theory and its philosophical applications*. Oxford University Press.

Sullivan, Kate, Deborah Zaitchik, and Helen Tager-Flusberg (1994). “Preschoolers can attribute second-order beliefs.” In: *Developmental psychology* 30.3, p. 395.

Talamadupula, Kartik et al. (2014). “Coordination in human-robot teams using mental modeling and plan recognition”. In: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, pp. 2957–2962.

Thielscher, Michael (2017). “GDL-III: A description language for epistemic general game playing”. In: *The IJCAI-16 workshop on general game playing*, p. 31.

To, Son Thanh, Tran Cao Son, and Enrico Pontelli (2011). “Contingent planning as and/or forward search with disjunctive representation”. In: *Twenty-First International Conference on Automated Planning and Scheduling*.

Torreño, Alejandro et al. (2017). “Cooperative Multi-Agent Planning: A Survey”. In: *CoRR* abs/1711.09057.

Van Benthem, Johan (2007). “Dynamic logic for belief revision”. In: *Journal of applied non-classical logics* 17.2, pp. 129–155.

Van Benthem, Johan, Jan Van Eijck, and Barteld Kooi (2006). “Logics of communication and change”. In: *Information and computation* 204.11, pp. 1620–1662.

Van Ditmarsch, Hans, Wiebe van Der Hoek, and Barteld Kooi (2007). *Dynamic epistemic logic*. Vol. 337. Springer Science & Business Media.

Vega Magro, Araceli et al. (Jan. 2019). “Planning Human-Robot Interaction for Social Navigation in Crowded Environments”. In: *Proceedings of the 19th International Workshop of Physical Agents (WAF 2018), November 22-23, 2018, Madrid, Spain*, pp. 195–208. ISBN: 978-3-319-99884-8. DOI: 10.1007/978-3-319-99885-5_14.

Wan, Hai, Biqing Fang, and Yongmei Liu (2021). “A general multi-agent epistemic planner based on higher-order belief change”. In: *Artificial Intelligence* 301, p. 103562. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2021.103562.

Weerdt, Mathijs de and Brad Clement (2009). “Introduction to planning in multiagent systems”. In: *Multiagent and Grid Systems* 5, pp. 345–355.

Yang, Boling, Patrick Lancaster, and Joshua R Smith (2018). “Physical Human-Robot Adversarial Gameplay (Extended Abstract)”. In: *Robotics: Science and Systems*.

Appendix

9.1 m$\mathcal{A}$-local Proofs

Lemma 3.1.1.

$$\forall u \in W.u \models \Omega(F, P, O) \textit{ iff } \pi(u) = \langle F, P, N \rangle.$$

Proof. First we prove the forward direction. Assume that for some world u in W, $u \models \Omega(F, P, N)$. $\Omega(F, P, N)$ is a conjunction of three conjunctions. Consider agent $i \in F$, corresponding to a disjunction that is conjoined by the first conjunction in $\Omega(F, P, N)$. If that disjunction disjoins zero formulas, then $\Omega(F, P, N)$ evaluates to $\bot$. There is no world u such that $u \models \bot$. Thus we consider the case that the disjunction corresponding to i disjoins one or more fluent formulas ω such that "if φ then i observes a"$\in \mathcal{T}$. It follows that at least one of those formulas holds in u. Thus, by definition, $i \in F(u)$. Now consider $i \in P$, corresponding to one of the disjunctions that are conjoined by the second conjunction in $\Omega(F, P, N)$. If that corresponding disjunction disjoins zero formulas, then again $\Omega(F, P, N)$ evaluates to $\bot$, and we have violated our assumption. If the disjunction corresponding to i disjoins one or more fluent formulas ω such that "if φ then i is aware of a"$\in \mathcal{T}$, then at least one of those formulas holds in u. Thus, by definition, $i \in P(u)$. Finally, consider $i \in N$. For all statements of the form "if φ then i observes a" or of the form "if φ then i is aware of a" in the theory it will be the case that $u \models \neg\omega$. Thus, by definition, and since $u \models \neg\omega \rightarrow u \not\models \omega$, it follows that $i \in N(u)$. □

Lemma 3.1.4. *The* sub *function for θ-events for ontic actions 1) causes proposition p to become true where formula φ holds if "if φ then a causes p to become* true*"$\in \mathcal{T}$, 2) causes p to become false where φ holds if "if φ then a causes p to become* false*"$\in \mathcal{T}$, and 3)*

preserves the value of p otherwise.

Proof. Consider world u and ontic θ-event $\theta_{\pi(u)}$. According to the definition of sub, for every fluent p, $sub(\theta_{\pi(u)})$ contains $p \rightarrow \Psi(p,\alpha)$ where $\Psi(p,\alpha) = \Psi^{+}(p,\alpha) \vee (p \wedge \neg\Psi^{-}(p,a))$.

If $\exists$ "if φ then a causes p to become *true*"$\in \mathcal{T}.u \models \phi$, then it follows that $u \models \Psi^{+}(p,\alpha)$ and therefore $u \models \Psi(p,\alpha)$. Thus by Equation (2.3) $p \in V'((u,\theta_{\pi(u)}))$.

Alternatively, if $\exists$ "if φ then a causes p to become *false*"$\in \mathcal{T}.u \models \phi$, then it follows that $u \models \Psi^{-}(p,\alpha)$ and therefore $u \not\models \neg\Psi^{-}(p,\alpha)$. Because of the requirement against contradictory observability statements in $\mathcal{T}$, also $u \not\models \Psi^{+}(p,\alpha)$, and it follows that $u \not\models \Psi(p,\alpha)$. Thus by Equation (2.3) $p \notin V'((u,\theta_{\pi(u)}))$.

Finally, if $\forall$"if φ then a causes p to become *true*" $\in \mathcal{T}.u \not\models \phi$ and $\forall$"if φ then a causes p to become *false*" $\in \mathcal{T}.u \not\models \phi$, it follows that $u \models \Psi(p,\alpha)$ iff $u \models p$. Thus by Equation (2.3) $p \in V'((u,\theta_{\pi(u)}))$ iff $p \in V(u)$. □

Lemma 3.1.5. *For ontic actions, the application of the ϵ-event preserves the valuation of formulas. For every $\varphi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$ and $u \in S$, $\langle M,u\rangle \models \varphi$ iff $\langle M',\langle u,\epsilon\rangle\rangle \models \varphi$.*

Proof. Since $pre(\epsilon) = \top$, for every world $u \in W$ there exists $(u,\epsilon) \in W'$. Since $sub(\epsilon) = \{p \rightarrow p \mid p \in \mathcal{P}\}$, $V((u,\epsilon)) = V(u)$. Since $(\epsilon,\epsilon) \in R_i^{\Sigma}$, for every $(u,v) \in R_i$, there is $((u,\epsilon),(v,\epsilon)) \in R_i'$. Since $\forall(\epsilon,e) \in R_i^{\Sigma}.e = \epsilon$, for every $((u,\epsilon),(v,e)) \in R_i', e = \epsilon$ and $(u,v) \in R_i$. It follows that $\langle M,u\rangle$ is bisimilar to $(M',(u,\epsilon))$. □

Proposition 3.1.6. *If $\mathcal{T}$ contains the statement "a causes l if φ" (a is an ontic action) and a is applied in state $\langle M,s\rangle$ resulting in state $\langle M',s'\rangle$, then it holds that*

1. *Ontic effects alter the state if their conditions are met: if $\langle M,s\rangle \models \varphi$ then $\langle M',s'\rangle \models l$,*

2. *Observers learn about effects they believe occur: for very $i \in F(s)$, if $\langle M,s\rangle \models \boldsymbol{B}_i\varphi$ then $\langle M',s'\rangle \models \boldsymbol{B}_i l$,*

3. *Non-observant agents are unaffected by ontic events: for every $i \in N(s)$ and $\omega \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$, $\langle M',s'\rangle \models \boldsymbol{B}_i\omega$ iff $\langle M,s\rangle \models \boldsymbol{B}_i\omega$,*

4. *Observers know that observers learn ontic effects: for every* $i \in F(s)$ *and* $j \in F^i(M,s)$*, if* $\langle M,s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\varphi$ *then* $\langle M',s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j l$*, and*

5. *Observers know that non-observant agents are unaffected: for every* $i \in F(s)$ *and* $j \in N^i(M,s)$ *and* $\omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{GB}}$*,* $\langle M',s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$ *iff* $\langle M,s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$*.*

Proof.

1. This follows directly from Lemma 3.1.4.

2. Consider agent $i \in F(s)$ and $(s',u') \in R'_i$ where $u' = (u,e)$. It follows from Equation (2.2) that $(s,u) \in R_i$ and $(\theta_{\pi(s)},e) \in R_i^{\Sigma}$. By Equation (3.3) we know that e is a θ-event, i.e. $e = \theta_{\pi(u)}$. Since $(M,s) \models \mathbf{B}_i\varphi$, $\langle M,u\rangle \models \varphi$. By Lemma 3.1.4 $u' \models l$.

3. Consider agent $i \in N(s)$ and $(s,u) \in R_i$. Because $pre(\epsilon) = \top$, $\exists (u,\epsilon) \in W'$. By Equations (3.3) and 2.2, $((s,\theta_{\pi(s)}),(u,\epsilon)) \in R'_i$ and by Lemma 3.1.5 (u,ϵ) is bisimilar to u. Now consider $((s,\theta_{\pi(s)}),(v,e)) \in R'_i$. By Equations (3.3) and 2.2, $e = \epsilon$, $(s,v) \in R'_i$, and by Lemma 3.1.5 v is bisimilar to (v,e).

4. Consider agent $i \in F(s)$ and $(s',u') \in R'_i$, where $u' = (u,e)$. By Equation (3.3), e is a θ-event, i.e. $u' = (u,\theta_{\pi(u)})$. It follows from Equation (2.2) that $(s,u) \in R_i$. Since $j \in F^i(M,s)$, it follows that $j \in F(u)$. If $(M,s) \models \mathbf{B}_i\mathbf{B}_j\varphi$, then $\langle M,u\rangle \models \mathbf{B}_j\varphi$. By the same reasoning as for part 2 of this proof, it follows that $(M',u') \models \mathbf{B}_j l$, and thus $(M',s') \models \mathbf{B}_i\mathbf{B}_j l$.

5. Consider agent $i \in F(s)$ and $(s',u') \in R'_i$, where $u' = (u,e)$. By Equation (3.3), e is a θ-event, i.e. $u' = (u,\theta_{\pi(u)})$. It follows from Equation (2.2) that $(s,u) \in R_i$. Consider agent $j \in N^i(M,s)$; it follows that $j \in N(u)$. By the same reasoning as for part 3 of this proof, for any formula ψ, $(M',u') \models \mathbf{B}_j\psi$ iff $\langle M,u\rangle \models \mathbf{B}_j\psi$. It follows that $(M',s') \models \mathbf{B}_i\mathbf{B}_j\psi$ iff $(M,s) \models \mathbf{B}_i\mathbf{B}_j\psi$.

□

Proposition 3.1.2. *Given action theory $\mathcal{T}$, if $\mathcal{T}$ contains the statement "a determines whether φ" (a is a sensing action) and a is applied in state $\langle M, s\rangle$ resulting in state $\langle M', s'\rangle$, then it holds that*

1. *Full observers learn true sensed formulae: $\forall i \in F(s)$. if $s \models \varphi$ then $s' \models \boldsymbol{B}_i\varphi$;*
2. *Full observers learn false sensed formulae: $\forall i \in F(s)$. if $\langle M, s\rangle \models \neg\varphi$ then $\langle M', s'\rangle \models \boldsymbol{B}_i\neg\varphi$;*
3. *Non-observant agents are unaffected by sensing events: $\forall i \in N(s), \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$.$\langle M', s'\rangle \models \boldsymbol{B}_i\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\omega$;*
4. *Full and partial observers learn that full observers learn sensed formulae: $\forall i \in F(s) \cup P(s), j \in F^i\langle M, s\rangle$.$\langle M', s'\rangle \models \boldsymbol{B}_i\big((\varphi \wedge \boldsymbol{B}_j\varphi) \vee (\neg\varphi \wedge \boldsymbol{B}_j\neg\varphi)\big)$;*
5. *Full and partial observers know that non-observant agents are unaffected: $\forall i \in F(s) \cup P(s), j \in N^i(M, s), \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}$.$\langle M', s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$.*

Proof.

1. Consider agent $i \in F(s)$ and $(s', u') \in R'_i$. If $s \models \omega$ then $s' = (s, \theta_{\pi(s)})$. By Equation (3.2) $u' = (u, \theta_{\pi(u)})$, and since $pre(\theta_{\pi(u)}) \models \omega$, necessarily $u \models \omega$, and since $sub(\theta_{\pi(u)})$ is the identity relation, $u' \models \omega$.
2. Consider agent $i \in F(s)$ and $(s', u') \in R'_i$. If $s \models \neg\omega$ then $s' = (s, \tau_{\pi(s)})$. By Equation (3.2) $u' = (u, \tau_{\pi(u)})$. Since $pre(\tau_{\pi(u)}) \models \neg\omega$, necessarily $u \models \neg\omega$, and since $sub(\tau_{\pi(u)})$ is the identity relation, $u' \models \neg\omega$.
3. The proof is the same as for part 3 of Proposition 3.1.6.
4. Consider agent $i \in F(s) \cup P(s)$ and $(s', u') \in R'_i$. Since the designated events are $\{\theta_{\pi(s)}, \tau_{\pi(s)}\}$, either $s' = (s, \theta_{\pi(s)})$, or $s' = (s, \tau_{\pi(s)})$. In either case, by Equations (3.2) and 2.2, u' is either a θ-world or u' is a τ-world, i.e. either $u' = (u, \theta_{\pi(u)})$ or $u' = (u, \tau_{\pi(u)})$, where $u \in W$. It follows from the first case that $u' \models \omega$ (since $pre(\theta_{\pi(u)}) \models \omega$ and $sub(\theta_{\pi(u)})$ is the identity relation), and by the same reasoning as in part 1 of this proof, that $(M'u') \models \mathbf{B}_j\omega$. It follows from the second case that

$u' \models \neg\omega$ (since $pre(\tau_{\pi(u)}) \models \neg\omega$ and $sub(\tau_{\pi(u)})$ is the identity relation), and by the same reasoning as in part 2 of this proof, that $(M'u') \models \mathbf{B}_j\neg\omega$.

5. The proof is the same as for part 5 of Proposition 3.1.6 except that s' can be either $(s, \theta_{\pi(s)})$ or $(s, \tau_{\pi(s)})$ and u' can be either $(u, \theta_{\pi(u)})$ or $(u, \tau_{\pi(u)})$.

□

Proposition 3.1.3. *If $\mathcal{T}$ contains the statement "a announces ψ"(a is an announcement action), then it holds that*

1. *Full observers learn announcements: $\forall i \in F(s).\langle M', s'\rangle \models \boldsymbol{B}_i\varphi$,*
2. *Non-observant agents are unaffected by announcements: $\forall i \in N(s), \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}.\langle M', s'\rangle \models \boldsymbol{B}_i\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\omega$,*
3. *Full and partial observers learn that full observers learn announcements: $\forall i \in F(s) \cup P(s), j \in F^i\langle M, s\rangle.\langle M', s'\rangle \models \boldsymbol{B}_i\big((\varphi \wedge \boldsymbol{B}_j\varphi) \vee (\neg\varphi \wedge \boldsymbol{B}_j\neg\varphi)\big)$,*
4. *Full and partial observers know that non-observant agents are unaffected: $\forall i \in F(s) \cup P(s), j \in N^i\langle M, s\rangle, \omega \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}}.\langle M', s'\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$ iff $\langle M, s\rangle \models \boldsymbol{B}_i\boldsymbol{B}_j\omega$.*

Proof.

1. Consider agent $i \in F(s)$ and $(s', u') \in R'_i$, $s' = (s, \theta_{\pi(s)})$, and by Equation (3.2), $u' = (u, \theta_{\pi(u)})$. Since $pre(\theta_{\pi(u)}) \models \omega$, necessarily $u \models \omega$, and since $sub(\theta_{\pi(u)})$ is the identity relation, it follows that $u' \models \omega$.
2. The proof is the same as for part 3 of Proposition 3.1.6.
3. The proof is the same as for part 4 of Proposition 3.1.2 except that since $\Gamma = \{\theta_{\pi(s)}\}$, necessarily $s' = (s, \theta_{\pi(s)})$.
4. The proof is the same as for part 5 of Proposition 3.1.6, except that u' can be either $(u, \theta_{\pi(u)})$ or $(u, \tau_{\pi(u)})$.

□

9.2 m$\mathcal{A}$-revise Proofs

9.2.1 Proofs of Frame Restriction Preservation

All proofs assume the following for all $i \in \mathcal{G}$:

- K_i satisfies $S5$, i.e. K_i is reflexive, transitive, and symmetric.
- B_i satisfies $KD45$, i.e. K_i is serial, transitive, and Euclidean.
- K_i and B_i satisfy KB1, i.e. $B'_i \subseteq K'_i$.
- K_i and B_i satisfy KB2, i.e. $\langle u, v\rangle \in K_i, \langle v, w\rangle \in B_i \rightarrow \langle u, w\rangle \in B_i$.

Theorem 4.3.1. *All acquired knowledge is true:* $\forall \alpha \in A, i \in \mathcal{G}, u \in M[W].\langle \mathcal{M}, u\rangle \models \mathcal{K}^{\alpha}_{iu}$

Proof. Since we have defined $\mathcal{K}^{\alpha}_{iu}$ to contain only fluent (non-modal) formulae, it will suffice to show that $u \models \mathcal{K}^{\alpha}_{iu}$. Note that if φ is a fluent formula, then $u \not\models \varphi \rightarrow u \models \neg\varphi$. For each conjunct φ of $\overset{det}{\mathcal{K}}{}^{\alpha}_{u}$, $u \models \varphi$ (Equation 4.6). Thus $u \models \overset{det}{\mathcal{K}}{}^{\alpha}_{u}$. For each conjunct φ of $\overset{eff}{\mathcal{K}}{}^{\alpha}_{u}$, $u \models \varphi$ (Equation 4.7). Thus $u \models \overset{det}{\mathcal{K}}{}^{\alpha}_{u}$. For each conjunct $\psi \in \{\phi_{u\alpha p}, \neg\phi_{u\alpha p}\}$ of $\overset{con}{\mathcal{K}}{}^{\alpha}_{u}$, $u \models \psi$ (Equation 4.9). Thus $u \models \overset{con}{\mathcal{K}}{}^{\alpha}_{u}$. $\overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}$ is composed of parts $\phi^{F}_{i\alpha}$and $\phi^{P}_{i\alpha}$(Equation 4.10). If $i \in F(\alpha, u)$ then $u \models \phi^{F}_{i\alpha}$ because $\exists$ "if φ then i observes α" $\in \mathcal{T}.u \models \varphi$ (Equation 4.3), and $u \not\models \phi^{P}_{i\alpha}$ because $\neg\exists$ "if φ then i observes α" $\in \mathcal{T}.u \models \varphi$ (Equations 4.3, 4.4, 4.3.1). If $i \in P(\alpha, u)$ then $u \models \phi^{P}_{i\alpha}$ (Equation 4.4), and $u \models \neg\phi^{F}_{i\alpha}$ (Equations 4.3, 4.4, 4.3.1). If $i \in N(\alpha, u)$ then $u \models \neg\phi^{F}_{i\alpha}$ and $u \models \neg\phi^{P}_{i\alpha}$ (Equations 4.3, 4.4, 4.5). It follows that $u \models \overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}$ (Equation 4.10). Since $u \models \overset{det}{\mathcal{K}}{}^{\alpha}_{u}, u \models \overset{eff}{\mathcal{K}}{}^{\alpha}_{u}, u \models \overset{obs}{\mathcal{K}}{}^{\alpha}_{iu}$, it follows that $u \models \mathcal{K}^{\alpha}_{iu}$ (Equation 4.11). □

Lemma 9.2.1. $(u, v) \in K'_i \rightarrow (m(u), m(v)) \in K_i$

Proof. Consider $(u, v) \in K'_i$. Either $(u, v) \in K'_{iF}$ or $(u, v) \in K'_{iN}$ (Equation 4.32). In either case, $\exists \alpha \in A, (w, x) \in M^{\alpha}[K_i]$, m(u)=m(w),m(v)=m(x) (if $(u, v) \in K'_{iF}$ then $w = u$, $x = v$ (Equation 4.30), if $(u, v) \in K'_{iN}$ then $\alpha =$ *No-op* (Equation 4.31)). It follows that $u[c_i] = v[c_i]$ (Equation 4.18), i.e. that u and v are both members of some local equivalence class C_{iz} defined with respect to some $z \in W$ (Equations 4.12–4.16). It follows by definition of C_{iz} that $(z, u), (z, v) \in K_i$ (Equation 4.12). By symmetry and transitivity of K_i, $(u, v) \in K_i$. □

Lemma 9.2.2. $\forall \alpha \in A.(u', v') \in K_i^\alpha.\ \ i \in F(\alpha, m(u')) \leftrightarrow i \in F(\alpha, m(v'))$, $i \in P(\alpha, m(u')) \leftrightarrow i \in P(\alpha, m(v'))$, $i \in N(\alpha, m(u')) \leftrightarrow i \in N(\alpha, m(v'))$.

Proof. Consider $(u', v') \in K_i^\alpha, u = m(u'), v = m(v')$. By definition $u'[c_i] = v'[c_i]$ (Equation 4.18), i.e. u' and v' are members of some local equivalence class C_{iw} for some $w \in W$ (Equations 4.12–4.16). By definition of C_{iw} $\langle M, u\rangle \models \mathcal{K}_{iw}^\alpha, \langle M, v\rangle \models \mathcal{K}_{iw}^\alpha$. Therefore, by the definition of $\mathcal{K}_{iw}^\alpha$ (Equation 4.11), $\langle M, u\rangle \models \overset{obs}{\mathcal{K}}{}_{iw}^\alpha, \langle M, v\rangle \models \overset{obs}{\mathcal{K}}{}_{iw}^\alpha$.

By Theorem 4.3.1 $\langle M, u\rangle \models \mathcal{K}_{iu}^\alpha$, and thus $\langle M, u\rangle \models \overset{obs}{\mathcal{K}}{}_{iu}^\alpha$ (Equation 4.11). From Equation 4.10, because $\phi_{i\alpha}^F \wedge \neg\phi_{i\alpha}^P$, $\neg\phi_{i\alpha}^F \wedge \phi_{i\alpha}^P$, and $\neg\phi_{i\alpha}^F \wedge \neg\phi_{i\alpha}^P$ are mutualy exclusive, it follows that $\overset{obs}{\mathcal{K}}{}_{iu}^\alpha = \overset{obs}{\mathcal{K}}{}_{iw}^\alpha$. Simmilarly, since $\langle M, v\rangle \models \overset{obs}{\mathcal{K}}{}_{iw}^\alpha$, it follows by Equation 4.10 that $\overset{obs}{\mathcal{K}}{}_{iv}^\alpha = \overset{obs}{\mathcal{K}}{}_{iw}^\alpha$, and thus $\overset{obs}{\mathcal{K}}{}_{iu}^\alpha = \overset{obs}{\mathcal{K}}{}_{iv}^\alpha$.

Now there are three possible values for $\overset{obs}{\mathcal{K}}{}_{iu}^\alpha$ and $\overset{obs}{\mathcal{K}}{}_{iv}^\alpha$ (Equation 4.10). If $\overset{obs}{\mathcal{K}}{}_{iu}^\alpha = \overset{obs}{\mathcal{K}}{}_{iv}^\alpha = \phi_{i\alpha}^F \wedge \neg\phi_{i\alpha}^P$ then $i \in F(\alpha, u)$ and $i \in F(\alpha, v)$. If $\overset{obs}{\mathcal{K}}{}_{iu}^\alpha = \overset{obs}{\mathcal{K}}{}_{iv}^\alpha = \neg\phi_{i\alpha}^F \wedge \phi_{i\alpha}^P$ then $i \in P(\alpha, u)$ and $i \in P(\alpha, v)$. If $\overset{obs}{\mathcal{K}}{}_{iu}^\alpha = \overset{obs}{\mathcal{K}}{}_{iv}^\alpha = \neg\phi_{i\alpha}^F \wedge \neg\phi_{i\alpha}^P$ then $i \in N(\alpha, u)$ and $i \in N(\alpha, v)$. □

Lemma 9.2.3. *An action causes agents to know their own observer status:* $\forall (u, v) \in K_i'.\ i \in F(\alpha, m(u)) \leftrightarrow i \in F(\alpha, m(v))$, $i \in P(\alpha, m(u)) \leftrightarrow i \in P(\alpha, m(v))$, $i \in N(\alpha, m(u)) \leftrightarrow i \in N(\alpha, m(v))$.

Proof. Consider $(u, v) \in K_i', u = m(u), v = m(v)$. Either $(u, v) \in K_{iF}'$ or $(u, v) \in K_{iN}'$ (Equation 4.32). In either case, $\exists \alpha \in A.(w, x) \in M^\alpha[K_i]$: if $(u, v) \in K_{iF}'$ then $w = u$ and $x = v$ (Equation 4.30), if $(u, v) \in K_{iN}'$ then $\alpha =$ *No-op* (Equation 4.31). By Lemma 9.2.2, $i \in F(\alpha, m(u)) \leftrightarrow i \in F(\alpha, m(v))$, $i \in P(\alpha, m(u)) \leftrightarrow i \in P(\alpha, m(v))$, $i \in N(\alpha, m(u)) \leftrightarrow i \in N(\alpha, m(v))$. □

Theorem 4.4.1. *K' satisfies $S5_n$: for all $i \in \mathcal{G}$, K_i' is reflexive, symmetric, and transitive.*

Proof.

Reflexivity Consider world $u' \in W', u = m(u'), i \in \mathcal{G}$. By definition of W' (Equation 4.28), $\exists \alpha \in\ W^\alpha$. Since $K_i^\alpha$is clearly reflexive (Equation 4.18), $(u', u') \in K_i^\alpha$. If $i \in F(\alpha, u') \cup P(\alpha, u')$ then $(u', u') \in K_{iF}' \subseteq K_i'$ (Equation 4.30). If $i \in N(\alpha, u')$, then $(u', u') \in K_{iN}' \subseteq K_i'$ (Equation 4.31).

Symmetry Consider $(u', v') \in K'_i$, $u = m(u')$, $v = m(v')$. By Lemma 9.2.1 $(u, v) \in K_i$ and by symmetry of K_i $(v, u) \in K_i$. Either $(u', v') \in K'_{iF}$ or $(u', v') \in K'_{iN}$ (Equation 4.32). If $(u', v') \in K'_{iF}$ then $i \in F(\alpha, v) \cup P(\alpha, v)$ (Lemma 9.2.3), and $\exists \alpha \in \mathcal{A}^0.(u', v') \in M^\alpha[K_i]$ (Equation 4.30). Since K_i^α is clearly symmetric (Equation 4.18), $(v', u') \in K_i^\alpha$, and it follows that $(v', u') \in K'_{iF} \subseteq K'_i$. If $(u', v') \in K'_{iN}$ (Equation 4.31) then $i \in N(\alpha, v)$ (Lemma 9.2.3). Since $M^0[K_i]$ is clearly reflexive (Equation 4.18), $(\exists (x, w) \in M^0[K_i].m(w) = m(u') \wedge m(x) = m(v'))$. By definition $(v', u') \in K'_{iN} \subseteq K'_i$ (Equation 4.31).

Transitivity Consider $(u', v'), (v', w') \in K'_i, u = m(u'), v = m(v'), w = m(w')$, If $(u', v') \in K'_{iF}$ then $u' \notin N(\alpha, u)$ (Equation 4.30) and $v', w' \notin N(\alpha, u)$ (Lemma 9.2.3), so necessarily $(v', w') \notin K'_{iN}$ (Equation 4.31) and $(v', w') \in K'_{iF}$ (Equation 4.32). Thus, $(u', v,') \in K_i^\alpha$ for some $\alpha \in A$ and $(v', w') \in K_i^{\alpha'}$ for some $\alpha' \in A$ (Equation 4.30). Since $v'[\alpha] = v'[\alpha]$, it follows that $K_i^\alpha = K_i^{\alpha'}$ (Equation 4.18) and $(u', v'), (v', w') \in K_i^\alpha$. Since $i \notin N(\alpha, u)$, and since $K_i^\alpha$is clearly transitive (Equation 4.18), $(u', w') \in K_i^\alpha$. Thus by definiton of K'_{iF}, $(u', w') \in K'_{iF} \subseteq K'_i$ (Equation 4.30).

If $(u', v') \in K'_{iN}$ then $i \in N(\alpha, u)$ (Equation 4.31) and $i \in N(\alpha, v')$ (Lemma 9.2.3). Thus $(v', w') \notin K'_{iF}$ (Equation 4.30) and necessarily $(v', w') \in K'_{iN}$ (Equation 4.32) and $i \in N(\alpha, w)$ (Lemma 9.2.3). It follows from the definition of K'_{iN} that $\exists (x, y, z) \in M^0[K_i].m(x) = m(u) \wedge m(y) = m(v) \wedge m(z) = m(wx)$ (Equation 4.30). If $M^{v'[\alpha]} \in H_i$ then also $M^u, M^w \in H_i$; if $M^{v'[\alpha]} \notin H_i$ then $M^u = M^v = M^w \notin H_i$. Either way, it follows by defintion of K'_{iN} that $(u', w') \in K'_{iN} \subseteq K'_i$ (Equation 4.31).

□

Theorem 4.4.3. *K'_i and B'_i satisfy the KB1 property that $B'_i \subseteq K'_i$.*

Proof. Consider $(u', v') \in B'_i$. Either $(u', v') \in B'_{iF}$ or $(u', v') \in B'_{iN}$ (Equation 4.35). If $(u', v') \in B'_{iF}$ then $u' \in F(a, u) \cup P(a, u)$ and $(u', v') \in M^\alpha[B_i]$ for some $\alpha \in \mathcal{A}^0$. Therefore $(u', v') \in K_i^\alpha$ (Equations 4.20, 4.21, 4.22). It follows that $(u', v') \in K'_{iF} \subseteq K'_i$ (Equation 4.30). If $(u', v') \in B'_{iN}$ then $u' \in N(a, u)$ and $\exists (x, y) \in M^0[B_i].m(x) = m(u') \wedge m(y) = m(v')$. It follows that $(x, y) \in M^0[K_i]$ (Equations 4.20, 4.21, 4.22). If

$M^u \in H_i \wedge M^v = M^0$ then $M^u, M^v \in H_i$ (since $M^0 \in H_i$, see Equation 4.27). Otherwise, $M^u = M^v \notin H_i$. It follows by definition that $(u', v') \in K'_{iN} \subseteq K'_i$ (Equation 4.31). $\square$

Lemma 9.2.4. *For every $\alpha \in A$ if $(u, v) \in K_i^\alpha \subseteq K'_i$ then $(M, w) \models \mathcal{B}_{iu}^\alpha \leftrightarrow (M, w) \models \mathcal{B}_{iv}^\alpha$.*

Proof. By Lemma 9.2.2, $i \in F(\alpha, m(u)) \leftrightarrow i \in F(\alpha, m(v))$. It follows by Equation 4.19 that $\mathcal{B}_{iu}^\alpha = \mathcal{B}_{iv}^\alpha$. $\square$

Lemma 9.2.5. *For all $\alpha \in \mathcal{A}^0$, K_i^α and B_i^α satisfy the KB2 property that $(u', v') \in K_i^\alpha, (v', w') \in B_i^\alpha \rightarrow (u', w') \in B_i^\alpha$.*

Proof. Consider $(u', v') \in K_i^\alpha$ and $(v', w') \in B_i^\alpha$, with $u = m(u'), v = m(v'), w = m(w')$. It follows from the definition of B_i^α (Equations 4.20, 4.21, 4.22) that $(v', w') \in K_i^\alpha$, and since K_i^α is clearly transitive (Equation 4.18), $(u', w') \in K_i^\alpha$. Since $K_i^\alpha \subseteq K'_i$, by Lemma 9.2.1 $(u, v) \in K_i$. Either $(v', w') \in B_i^{\alpha 1}$ (Equation 4.20) or $(v', w') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ (Equation 4.21) or $(v', w') \in B_i^\alpha \setminus B_i^{\alpha 2}$ (Equation 4.22).

If $(v', w') \in B_i^{\alpha 1}$ then $(v, w) \in B_i$ (Equation 4.20), and it follows by KB2 of K_i and B_i that $(u, w) \in B_i$. From $(v', w') \in B_i^{\alpha 1}$ it follows that $(M, w) \models \mathcal{B}_{iv}^\alpha$ and by Lemma 9.2.4 and $(M, w) \models \mathcal{B}_{iu}^\alpha$. Thus by definition $(u', w') \in B_i^{\alpha 1} \subseteq B_i^\alpha$ (Equation 4.20).

If $(v', w') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ then $\forall x' \in W^\alpha.(v', x') \notin B_i^{\alpha 1}$, i.e. v' is non-serial in $B_i^{\alpha 1}$ (Equation 4.21). We will now show that this implies that u' is also non-serial in $B_i^{\alpha 1}$. Assume that there exists some $x' \in W', x = m(x')$ such that $(u', x') \in B_i^{\alpha 1}$. It follows that $(u', x') \in K_i^\alpha$, and since K_i^α is clearly transitive and symmetric (Equation 4.18), that $(v', x') \in K_i^\alpha$. Since $(v, u) \in K_i$ (symmetry of K_i), and $(u, x) \in B_i$ (Equation 4.20), it follows from the KB1 property of K_i and B_i that $(v, x) \in B_i$. From $(u', x') \in B_i^{\alpha 1}$ it follows that $(M, x) \models \mathcal{B}_{iu}^\alpha$ and by Lemma 9.2.4 $(M, x) \models \mathcal{B}_{iv}^\alpha$. Thus, by definition (Equation 4.20), $(v', x') \in B_i^{\alpha 1}$, a contradiction since $(v', w') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ implies that v' is non-serial in $B_i^{\alpha 1}$, and we must reject our assumption. Thus $\forall x' \in W^\alpha.(u', x') \notin B_i^{\alpha 1}$. From $(v', w') \in B_i^{\alpha 2}$ it follows that $(M, w) \models \mathcal{B}_{iv}^\alpha$ and by Lemma 9.2.4 $(M, w) \models \mathcal{B}_{iu}^\alpha$. Thus by definition $(u', w') \in B_i^{\alpha 2} \subseteq B_i^\alpha$ (Equation 4.21).

If $(v', w') \in B_i^\alpha \setminus B_i^{\alpha 2}$, then $\forall x' \in W^\alpha.(u', x') \notin B_i^{\alpha 2}$ according to the same

reasoning used in the previous condition to show that $(v', w') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1} \rightarrow \forall x' \in W^\alpha.(u', x') \notin B_i^{\alpha 1}$. As we have shown that $(u', w') \in K_i^\alpha$, it follows by definition that $(u', w') \in B_i^\alpha$ (Equation 4.22). $\square$

Theorem 4.4.4. *K_i' and B_i' satisfy the KB2 property that if $\langle u, v\rangle \in K_i'$ and $(v, w) \in B_i'$ then $\langle u, w\rangle \in B_i'$.*

Proof. Consider $(u', v') \in K_i', (v', w') \in B_i', u = m(u'), v = m(v'), w = m(w')$. Either $(u', v') \in K_{iF}'$ or $(u', v') \in K_{iN}'$ (Equation 4.32).

If $(u', v') \in K_{iF}'$, then $i \in F(\alpha, u) \cup P(\alpha, u)$ and $(u', v') \in K_i^\alpha$ where $\alpha = u'[\alpha]$ (Equation 4.30). Since $(u', v') \in K_i^\alpha$, $v \in W^\alpha$ and $v'[\alpha] = \alpha$. Since $i \in F(\alpha, v) \cup P(\alpha, v)$ (Lemma 9.2.3), $(v', w') \in B_{iF}'$. Thus $(v', w,') \in B_i^\alpha$ and $(u', v'), (v', w') \in K_i^\alpha$. By Lemma 9.2.5, $(u', w') \in K_i^\alpha \subseteq K_{iF}' \subseteq K_i'$.

If $(u', v') \in K_{iN}'$, then $i \in N(\alpha, u)$ and $\exists (x, y) \in M^0[K_i].m(x) = u \wedge m(y) = v$ (Equation 4.31). Furthermore, $i \in N(\alpha, v)$ (Lemma 9.2.3), and therefore $(v', w') \in B_{iN}'$. Thus $\exists (y, z) \in M^0[B_i].m(z) = w$ (Equation 4.34). By Lemma 9.2.5, $(x, z) \in M^0[B_i]$. By Equation 4.34, either $M^{v'[\alpha]} \in H_i \wedge M^{w'[\alpha]} = M^0$ or $M^{v'[\alpha]} = M^{w'[\alpha]} \notin H_i$. If $M^{v'[\alpha]} \in H_i \wedge M^{w'[\alpha]} = M^0$, then $M^{u'[\alpha]} \in H_i$ (Equation 4.31), so $M^{u'[\alpha]} \in H_i \wedge M^{w'[\alpha]} = M^0$ and it follows by definition that $(u', w') \in B_{iN}' \subseteq B_i'$ (Equation 4.34). If $M^{v'[\alpha]} = M^{w'[\alpha]} \notin H_i$, then then $M^{u'[\alpha]} = M^{v'[\alpha]} \notin H_i$ (Equation 4.31), and thus $M^{u'[\alpha]} = M^{w'[\alpha]} \notin H_i$, and it follows by definition that $(u', w') \in B_{iN}' \subseteq B_i'$ (Equation 4.34). $\square$

Lemma 9.2.6. *For all $\alpha \in \mathcal{A}^0$, B_i^α satisfies $KD45_n$, i.e. for all $i \in \mathcal{G}$, B_i^α is serial, transitive, and Euclidean.*

Proof.

Seriality Consider $u' \in W^\alpha, u = m(u')$. If $\exists v' \in W^\alpha.(u', v') \in B_i^{\alpha 2}$, then $(u', v') \in B_i^{\alpha 2} \subseteq B_i^\alpha$ (Equation 4.21). Otherwise, since K_i^α is clearly reflexive (Equation 4.18), by definition $(u', u') \in B_i^\alpha$ (Equation 4.22).

Transitivity Consider $i \in \mathcal{G}$, $u', v', w' \in W', (u', v'), (v', w') \in B_i^\alpha, m(u') = u, m(v') = v, m(w') = w$. By definition $(u', v'), (v', w') \in K_i^\alpha$ (Equations 4.20, and 4.21,

4.22). Since K_i^α is clearly transitive (Equation 4.18), $(u', w') \in K_i^\alpha$. Either $(u', v') \in B_i^{\alpha 1}$ (Equation 4.20) or $(u', v') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ (Equation 4.21) or $(u', v') \in B_i^\alpha \setminus B_i^{\alpha 2}$ (Equation 4.22).

If $(u', v') \in B_i^{\alpha 1}$ then $(u, v) \in B_i$ and $\langle M, v\rangle \models \mathcal{B}_{iu}^\alpha$ and by Lemma 9.2.4 $\langle M, v\rangle \models \mathcal{B}_{iv}^\alpha$. By transitivity, seriality, and Euclideanness of B_i, $(v, v) \in B_i$. By reflexivity of K_i, $(v, v) \in K_i$. It follows by definition that $(v', v') \in B_i^{\alpha 1}$ (Equation 4.20). Thus, v' is serial in $B_i^{\alpha 1}$, so (v', w') cannot be in $B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ or $B_i^\alpha \setminus B_i^{\alpha 2}$, and necessarily $(v', w') \in B_i^{\alpha 1}$. Thus $(u, v), (v, w) \in B_i$ and by transitivity of B_i, $(u, w) \in B_i$ and $(M, w) \models \mathcal{B}_{iv}^\alpha$. Simmilarly $(u', v'), (v', w') \in K_i^\alpha$ and, by Theorem 4.4.1, $(u', w') \in K_i^\alpha$. By Lemma 9.2.4 $(M, w) \models \mathcal{B}_{iu}^\alpha$. It follows by definition that $(u', w') \in B_i^{\alpha 1}$ (Equations 4.20).

If $(u', v') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ then $\langle M, v\rangle \models \mathcal{B}_{iu}^\alpha$ and u' is non-serial in $B_i^{\alpha 1}$ (Equation 4.21). It follows that v' is serial in $B_i^{\alpha 2}$: if v' is non-serial in $B_i^{\alpha 1}$ then (v', v') will be put in $B_i^{\alpha 2}$ since K_i^α is clearly reflexive (Equation 4.18) and $\langle M, v\rangle \models \mathcal{B}_{iv}^\alpha$ (Lemma 9.2.4). Thus (v', w') cannot possibly be in $B_i^\alpha \setminus B_i^{\alpha 2}$ and must be in either $B_i^{\alpha 1}$ or $B_i^{\alpha 2} \setminus B_i^{\alpha 1}$. It follows that $(M, w) \models \mathcal{B}_{iv}^\alpha$ and therefore $(M, w) \models \mathcal{B}_{iu}^\alpha$. Thus by definition $(u', w') \in B_i^{\alpha 2} \subseteq B_i^\alpha$ (Equation 4.21).

If $(u', v') \in B_i^\alpha \setminus B_i^{\alpha 2}$ then u' is non-serial in $B_i^{\alpha 1}$, and so by definition $(u', w') \in B_i^\alpha$ (Equation 4.22).

Euclideanness Consider $i \in \mathcal{G}$, $u', v', w' \in W^\alpha, (u', v'), (u', w') \in B_i^\alpha, m(u') = u, m(v') = v, m(w') = w$, and $u, v, w \in W$. By definition $(u', v'), (v', w') \in K_i^\alpha$ (Equations 4.20, and 4.21, 4.22). Since K_i^α is clearly Euclidean (Equation 4.18), $(v', w') \in K_i^\alpha$. Either $(u', v') \in B_i^{\alpha 1}$ (Equation 4.20) or $(u', v') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ (Equation 4.21) or $(u', v') \in B_i^\alpha \setminus B_i^{\alpha 2}$ (Equation 4.22).

If $(u', v') \in B_i^{\alpha 1}$, then u' is serial in $B_i^{\alpha 1}$ so (u', w') cannot be in $B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ or $B_i^\alpha \setminus B_i^{\alpha 2}$ and necessarily $(u', w') \in B_i^{\alpha 1}$. Thus $(u, v), (u, w) \in B_i$ and by Euclideanness of B_i, $(v, w) \in B_i$. Now $(u', w') \in B_i^{\alpha 1}$ implies that $(M, w) \models \mathcal{B}_{iu}^\alpha$ and by Lemma 9.2.4 and $(M, w) \models \mathcal{B}_{iv}^\alpha$. It follows by definition that $(v', w') \in B_i^{\alpha 1} \subseteq B_i^\alpha$ (Equation 4.20).

If $(u', v') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$, then u' is nonserial in $B_i^{\alpha 1}$ and serial in $B_i^{\alpha 2}$, so necessarily $(u', w') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$. We will show that v' is non-serial in $B_i^{\alpha 1}$. Assume that v' is serial

in $B_i^{\alpha 1}$: $\exists x' \in W^\alpha.(v', x') \in B_i^{\alpha 1}, m(x') = x \in W$. By definition $(v, x) \in B_i$ and $(M, x) \models \mathcal{B}_{iv}^\alpha$ and by Lemma 9.2.4 and $(M, w) \models \mathcal{B}_{iu}^\alpha$. Since $(u, v) \in K_i$, by KB1 of K_i and B_i, $(u, x) \in B_i$. It follows by definition that $(u', x') \in B_i^{\alpha 1}$ (Equation 4.20), which contradicts the assumption that $(u', v') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$ and u' is nonserial in $B_i^{\alpha 1}$. Thus, we must reject our assumption that $(v', x') \in B_i^{\alpha 1}$ and conclude that v' is nonserial in $B_i^{\alpha 1}$. It follows by definition that $(v', x') \in B_i^{\alpha 2} \subseteq B_i^\alpha$ (equation 4.21).

If $(u', v') \in B_i^\alpha \setminus B_i^{\alpha 2}$, then u' is nonserial in $B_i^{\alpha 2}$, so necessarily $(u', x') \in B_i^\alpha \setminus B_i^{\alpha 2}$. We will show that v' is non-serial in $B_i^{\alpha 2}$. Assume that v' is serial in $B_i^{\alpha 2}$: $\exists x' \in W'.(v', x') \in B_i^{\alpha 2}, m(x') = x \in W$. Either $(v', x') \in B_i^{\alpha 1}$ or $(v', x') \in B_i^{\alpha 2} \setminus B_i^{\alpha 1}$. If $(v', x') \in B_i^{\alpha 1}$, then by definition $(v, x) \in B_i$ and $(M, x) \models \mathcal{B}_{iv}^\alpha$ and by Lemma 9.2.4 and $(M, x) \models \mathcal{B}_{iu}^\alpha$. By KB1 of K_i and B_i, $(u, x) \in B_i$. It follows by definition that $(u', x') \in B_i^{\alpha 1}$ (Equation 4.20), which contradicts the assumption that $(u', v') \in B_i^\alpha \setminus B_i^{\alpha 2}$ and u' is nonserial in $B_i^{\alpha 2}$ and in $B_i^{\alpha 1}$. If $(v', x') \in B_i^{\alpha 2}$, then $(v', x') \in K_i^\alpha$ and, by transitivity of K_i^α, $(u', x') \in K_i^\alpha$. It follows by definition that $(u', x') \in B_i^{\alpha 2}$ (Equation 4.21). Thus, in either case, we must reject our assumption that $(v', x') \in B_i^{\alpha 2}$ and conclude that v' is nonserial in $B_i^{\alpha 2}$. Since $(v', w') \in K_i^\alpha$ by Euclideanness of K_i^α, it follows by definition that $(v', w') \in B_i^\alpha$ (Equation 4.22). □

Theorem 4.4.2. *B' satisfies $KD45_n$: for all $i \in \mathcal{G}$, B_i' is serial, transitive, and Euclidean.*

Proof.

Seriality Consider $i \in \mathcal{G}$, $u' \in W^\alpha, u = m(u'), \alpha = u'[\alpha]$. By definition $u' \in W^\alpha$ (Equation 4.16). By Lemma 9.2.6 $\exists v' \in W^\alpha.(u', v') \in B_i^\alpha$. Either $i \in F(\alpha, u) \cup P(\alpha, u)$, or $i \in N(\alpha, u)$. If $i \in F(\alpha, u) \cup P(\alpha, u)$, then $(u', v') \in B_{iF}' \subseteq B_i'$ (Equation 4.33). If $i \in N(u'[\alpha], u)$, then since *No-op* has no preconditions, for every world $x \in W$ there is a world $x' \in M^0[W]$ such that $m(x') = x$. Thus, there exsits some y' in $M^0[W]$ such that $m(y') = m(v')$, and if $M^\alpha \in H_i$ then by definition $(u', y') \in B_{iN}' \subseteq B_i'$ (Equation 4.34). If $M^\alpha \notin H_i$, then $M^\alpha = M^{v'[\alpha]}$ and by definition $(u', v') \in B_{iN}' \subseteq B_i'$ (Equation 4.34).

Transitivity Consider $i \in \mathcal{G}, u', v', w' \in W', (u', v'), (v', w') \in B_i', m(u') = u, m(v') = v, m(w') = w,$ and $u, v, w \in W$. Let $\alpha = u'[\alpha]$; by definition $u' \in W^\alpha$ (Equation 4.16).

Either $(u', v') \in B'_{iF}$ or $(u', v') \in B'_{iN}$ (Equation 4.35).

If $(u', v') \in B'_{iF}$, then $i \in F(\alpha, u) \cup P(\alpha, u)$ (Equation 4.33), $(u', v') \in B_i^\alpha$ (Equation 4.33), $(u', v') \in K_i^\alpha \subseteq K'_i$ (Equations 4.20, 4.21, 4.22), and $u'[\alpha] = v'[\alpha]$. By Lemma 9.2.2 $i \in F(\alpha, v) \cup P(\alpha, v)$, and it follows that $(v', w') \in B'_{iF}$ so $(v', w') \in B_i^\alpha$. By Lemma 9.2.6 $(u', w') \in B_i^\alpha \subseteq B'_{iF} \subseteq B'_i$.

If $(u', v') \in B'_{iN}$, then $i \in N(\alpha, u)$ (Equation 4.34) and $(\exists (x, y) \in M^0[B_i].m(x) = u \wedge m(y) = v)$. Either $(M^\alpha \in H_i \wedge M^{v'[\alpha]} = M^0)$ or $(M^\alpha = M^{v'[\alpha]} \notin H_i)$.

If $(M^\alpha \in H_i \wedge M^{v'[\alpha]} = M^0)$, then $v'[\alpha] =$ *No-op*, $v' \in M^0[W]$, $y = v'$, and, since *No-op* has only oblivious agents, $i \in N(v'[\alpha], v)$. It follows that $(v', w') \in B_i^{\textit{No-op}}$ and $w' \in M^0[W]$ (Equation 4.34). By transitivty of $M^0[B_i]$ (Lemma 9.2.6), $(x, w') \in B^{\textit{No-op}}$. It follows by definition that $(u', w') \in B'_{iN} \subseteq B'_i$ (Equation 4.34).

If $(M^\alpha = M^{v'[\alpha]} \notin H_i)$, then $u'[\alpha] = v'[\alpha]$, $i \in N(v'[\alpha], v)$ (Lemma 9.2.2), and $(v', w') \in B'_{iN}$ (Equation 4.34). It follows that $\exists (x, y), (y, z) \in M^0[B_i].m(x) = u, m(y) = v, m(z) = w$ and, by Lemma 9.2.6, $(x, z) \in M^0[B_i]$. Since $M^{v'[\alpha]} \notin H_i$, $M^{w'[\alpha]} \notin H_i$ and $M^{u'[\alpha]} = M^{v'[\alpha]} = M^{w'[\alpha]}$. It follows by definition that $(u', v') \in B'_{iN} \subseteq B'_i$ (Equation 4.34).

Euclideanness Consider $i \in \mathcal{G}$, $u', v', w' \in W^\alpha, (u', v'), (u', w') \in B'_i, m(u') = u, m(v') = v, m(w') = w,$ and $u, v, w \in W$. Let $\alpha = u'[\alpha]$; by definition $u' \in W^\alpha$ (Equation 4.16). Either $(u', v') \in B'_{iF}$ or $(u', v') \in B'_{iN}$ (Equation 4.35).

If $(u', v') \in B'_{iF}$, then $i \in F(\alpha, u) \cup P(\alpha, u)$, $(u', v') \in B_i^\alpha$, $(u', w') \in B'_{iF}$, and $(u', w') \in B_i^\alpha$ (Equation 4.33). By Lemma 9.2.6, $(v', w') \in B_i^\alpha \subseteq B'_{iF} \subseteq B'_i$.

If $(u', v') \in B'_{iN}$, then $i \in N(u'[\alpha], u)$ and $(u', w') \in B'_{iN}$ (Equation 4.34). It follows that $\exists (x, y), (x, z) \in M^0[B_i].m(x) = u, m(y) = v, m(z) = w$ and, by Lemma 9.2.6, $(y, z) \in M^0[B_i]$. Either $(M^{u'[\alpha]} \in H_i \wedge M^{v'[\alpha]} = M^0)$ or $(M^{u'[\alpha]} = M^{v'[\alpha]} \notin H_i)$. If $(M^{u'[\alpha]} \in H_i \wedge M^{v'[\alpha]} = M^0)$, then $M^{w'[\alpha]} = M^0$ and it follows by definition that $(v', w') \in B'_{iN} \subseteq B'_i$ (Equation 4.34). If $(M^{u'[\alpha]} = M^{v'[\alpha]} \notin H_i)$, then $(M^{u'[\alpha]} = M^{v'[\alpha]} = M^{w'[\alpha]} \notin H_i)$ and it follows by definition that $(v', w') \in B'_{iN} \subseteq B'_i$ (Equation 4.34). □

9.2.2 Proofs of Action Semantics

This section will prove several propertes pertaining to state $\langle M, d\rangle$, action a defined by action theory $\mathcal{T}$, agents i and j, and the result of applying the action $(M', d') \coloneqq \langle M, d\rangle \times a$, where $M = \langle W, V, K_{i\in A}, B_{i\in A}\rangle$ and $M' = \langle W', V', K'_{i\in A}, B'_{i\in A}\rangle$.

Lemma 9.2.7. *In worlds within the local equivalence class of some world w and action α, full and partial observers come to know whether formulae entailed by $\mathcal{K}^{\alpha}_{iw}$are true: given $u' \in W'$, $u[a] = \alpha$, $u[c_i] = C^{\alpha}_{iw}$, $\psi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, and $C^{\alpha}_{iw} \models \psi$, it follows that $(M', u') \models \boldsymbol{K}_i\psi \vee \boldsymbol{K}_i\neg\psi$.*

Proof. Consider $u' \in W'$, $m(u') = u$, $i \in F(\alpha, u) \cup P(\alpha, u)$, $u'[a] = \alpha$, $u[c_i] = C^{\alpha}_{iw}$, $\psi \in \mathcal{L}^{\mathcal{P}}_{\mathcal{G}B}$, $C^{\alpha}_{iw} \models \psi$. It follows that $(u', v') \in K^{\alpha}_i$ and $v'[a] = \alpha$ (Equation 4.30), $v'[c_i] = C^{\alpha}_{iw}$ (Equation 4.18), and $\langle M, v\rangle \models \mathcal{K}^{\alpha}_{iw}$ (Equation 4.12), and therefore that $v \models \psi$. Either $\exists p \in \mathcal{P}.\exists$"$a$ causes $l_0, \ldots, l_n$" $\in \mathcal{T}.\mathcal{K}^{\alpha}_{iw} \models \varphi$ and $(\psi \models p) \neq l$, or not. If so, then by Equation (4.17) $v \models \varphi$ and $V^{\alpha}(p, v') = l$, i.e. $(v' \models p) = l$, so $(v' \models p) \neq (\psi \models p)$ and $v' \models \neg\psi$. If not, then $\forall p \in \mathcal{P}$. if $\exists$"a causes $l_0, \ldots, l_n$" $\in \mathcal{T}$ then either $(\psi \models p) = l$ or $\mathcal{K}^{\alpha}_{iw} \not\models \varphi$ and thus $v \not\models \varphi$. If $(\psi \models p) = l$ then since $v \models \psi$ it follows that $V(p, v) = l$ and since by Equation (4.17) $V^{\alpha}(p, v') = l$ it follows that $V^{\alpha}(p, v') = V(p, v)$. Alternatively, if $\mathcal{K}^{\alpha}_{iw} \not\models \varphi$ then $v \not\models \varphi$ and it follows directly from Equation (4.17) that $V^{\alpha}(p, v') = V(p, v)$. Either way, since $v \models \psi$ we have that $v' \models \psi$. □

Proposition 4.4.5. *Ontic effects alter the state if their conditions are met: if "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$ and $u \models \varphi$ then $p \in V'(u')$ iff $l = true$.*

Proof. By Equation (4.25) $m(d') = d$. By Equation (4.17), $V^a(f, d') = l$, and by Equation (4.29) $V'(f, d') = l$. □

Proposition 4.4.6. *Full and partial observers learn the effects of ontic actions: for every $i \in F(a, d) \cup P(a, d)$ and "a causes $l_0, \ldots, l_n$" $\in \mathcal{T}$, $\langle M', u'\rangle \models \boldsymbol{K}_ip \vee \boldsymbol{K}_i\neg p$.*

Proof. Consider "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$. By Equation (4.25), $m(d') = d, d'[a] = a, d[c_i] = C^a_{id}$. If $d \models f$ then by Equations (4.7) and (4.11) $\mathcal{K}^a_{id} \models p$, and if $d \not\models p$ then $\mathcal{K}^a_{id} \models \neg f$. Either way, by Lemma 9.2.7 $(M', d') \models \mathbf{K}_ip \vee \mathbf{K}_i\neg p$. □

Proposition 4.4.7. *Full and partial observers learn that full and partial observers learn ontic effects: If "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$ and $u \models \varphi$ and $i \in F(a,u) \cup P(a,u)$ and $(\forall v \in M[W]$. if $\langle u, v\rangle \in M[B_i]$ then $j \in F(a,u) \cup P(a,u))$, then $\langle M', u'\rangle \models \boldsymbol{B}_i(\boldsymbol{K}_j p \vee \boldsymbol{K}_j \neg p)$.*

Proof. Consider "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$, $i \in F(a,d) \cup P(a,d)$, $\forall (d,u) \in M[B_i].j \in F(a,u) \cup P(a,u)$, and $(d', v') \in B_i'$. Let $v[c_j] = C^a_{jw}$ where $w \in W$. If $w \models p$ then by Equations (4.7) and (4.11) $\mathcal{K}^a_{jw} \models p$. Simmilarly if $w \not\models p$ then $\mathcal{K}^a_{jw} \models \neg p$. In either case, it follows from Lemma 9.2.7 that $(M', v') \models \mathbf{K}_j p \vee \mathbf{K}_j \neg p$. □

Proposition 4.4.8. *If one or more indicated effect conditions condition an effect, full and partial observers learn whether at least one of those conditions holds: $\forall p \in \mathcal{P}.\langle M', u'\rangle \models \boldsymbol{K}_i \phi_{uap} \vee \boldsymbol{K}_i \neg \phi_{uap}$.*

Proof. Given $i \in F(a,d) \cup P(a,d)$ and since either $\mathcal{K}^a_{id} \models \phi_{dap}$ or $\mathcal{K}^a_{id} \models \neg \phi_{dap}$ (Equations (4.9) and (4.11)) it follows from Lemma 9.2.7 that $(M', d') \models \mathbf{K}_i \phi_{dap} \vee \mathbf{K}_i \neg \phi_{dap}$. □

Proposition 4.4.9. *Full and partial observers know that full and partial observers learn whether indicated effect conditions hold: If $i \in F(a,u) \cup P(a,u)$ and $\forall (u,v) \in M[B_i].j \in F(a,v) \cup P(a,v)$, then $\langle M', u'\rangle \models \boldsymbol{B}_i(\boldsymbol{K}_j \phi_{uap} \vee \boldsymbol{K}_j \neg \phi_{uap})$.*

Proof. Consider $i \in F(a,d) \cup P(a,d)$, $\forall (d,u) \in M[B_i].j \in F(a,u) \cup P(a,u)$, and $(d', v') \in B_i'$. By Theorem 4.4.3, $(d', v') \in K_i'$, and by Equations (4.18), (4.30), and (4.33) $d'[c_j] = v'[c_j]$. Thus by Equation (4.25) $v[c_j] = C^a_{jd}$. Since either $\mathcal{K}^a_{jd} \models \phi_{dap}$ or $\mathcal{K}^a_{jd} \models \neg \phi_{wap}$ (Equations (4.9) and (4.11)) it follows from Lemma 9.2.7 that $(M', v') \models \mathbf{K}_j \phi_{dap} \vee \mathbf{K}_j \neg \phi_{dap}$. □

Proposition 4.4.10. *Full observers learn the value of sensed propositions: $\forall$ "a determines whether φ" $\in \mathcal{T}.u' \models \boldsymbol{K}_i \varphi \vee \boldsymbol{K}_i \neg \varphi$.*

Proof. Consider $i \in F(a,d)$ and "a determines whether φ"$\in \mathcal{T}$. By Equation (4.6) either $\overset{det}{\mathcal{K}}{}^a_d \models \varphi$ or $\overset{det}{\mathcal{K}}{}^a_d \models \neg \varphi$. Either way, since $\mathcal{K}^a_{id} \models \overset{det}{\mathcal{K}}{}^a_d$ it follows from Lemma (9.2.7) that $(M', d') \models \mathbf{K}_i \varphi \vee \mathbf{K}_i \neg \varphi$. □

Proposition 4.4.11. *Full and partial observers learn that full observers learn sensed propositions. If "a determines whether φ"$\in \mathcal{T}$ and $i \in F(a,u) \cup P(a,u)$ and $\forall (u,v) \in M[B_i].j \in F(a,u)$, then $\langle M', u' \rangle \models \boldsymbol{B}_i(\boldsymbol{K}_j\varphi \vee \boldsymbol{K}_j\neg\varphi)$.*

Proof. Consider $i \in F(a,d) \cup P(a,d)$, $\forall (d,u) \in M[B_i].j \in F(a,u)$, $(d',v') \in B'_i$, and "a determines whether φ"$\in \mathcal{T}$. By Theorem 4.4.3, $(d',v') \in K'_i$, and by Equations (4.18), (4.30), and (4.33) $d'[c_j] = v'[c_j]$. Thus by Equation (4.25) $v[c_j] = C^a_{jd}$. By Equation (4.6) either $\overset{det}{\mathcal{K}}{}^a_v \models \varphi$ or $\overset{det}{\mathcal{K}}{}^a_v \models \neg\varphi$. Either way, since $\mathcal{K}^\alpha_{iv} \models \overset{det}{\mathcal{K}}{}^a_v$ it follows from Lemma (9.2.7) that $(M', v') \models \mathbf{K}_j\varphi \vee \mathbf{K}_j\neg\varphi$. □

Proposition 4.4.12. *Full observers believe that the conjunction of all announced propositions* was *true if it is not self-contradictory and does not contradict their prior knowledge or current observations. If i is a full observer then in any world $v' \in W'$ such that $\langle u', v' \rangle \in B'_i$, it was the case before the action occurred that announcements were true: $\langle M, m(v') \rangle \models \mathcal{B}^\alpha_{iu}$.*

Proof. Consider $i \in F(a,u)$, $\mathcal{B}^\alpha_{iu} \not\models \bot$, $\langle M,u \rangle \not\models \mathbf{K}_i\neg\mathcal{B}^\alpha_{iu}$, $\mathcal{K}^\alpha_{iu} \not\models \neg\mathcal{B}^\alpha_{iu}$. By Equations (4.20) and (4.21) $\forall (u',v') \in B^{\alpha 2}_i.(M, m(v')) \models \mathcal{B}^\alpha_{iu}$. Since $\langle M,u \rangle \not\models \mathbf{K}_i\neg\mathcal{B}^\alpha_{iu}$, $\exists (u,w) \in K_i.(M,w) \models \mathcal{B}^\alpha_{iu}$. Thus by Equation (4.21) $B^{\alpha 2}_i$ is not empty and by Equations (4.22), (4.33), and (4.35), $B^{\alpha 2}_i = B^\alpha_i = B'_{iF} = B'_i$, so $\forall (u',v') \in B'_i.(M, m(v')) \models \mathcal{B}^\alpha_{iu}$. □

Proposition 4.4.13. *Full and partial observers know that full observers believe that the conjunction of all announced propositions* was *true if it is not contradictory and does not contradict their prior knowledge or current observations. If i is a full or partial observer and believes that j is a full observer then $\forall \langle u',v' \rangle \in B'_i.\forall \langle v',w' \rangle \in B'_j.\langle M, m(w') \rangle \models \mathcal{B}^\alpha_{jm(v')}$.*

Proof. Consider $i \in F(a,d) \cup P(a,d)$, $(d',u') \in B'_i$, $u = m(u')$, $j \in F(a,u)$, $\langle M,u \rangle \not\models \mathbf{K}_j\neg\mathcal{B}^\alpha_{ju}$, and $\mathcal{K}^a_{ju} \not\models \neg\mathcal{B}^\alpha_{ju}$. By Equations (4.22), (4.33) and (4.35), $u'[a] = d'[a] = a$. By Equations (4.20) and (4.21) $\forall (u',v') \in B^{a2}_j.(M, m(v')) \in \mathcal{B}^\alpha_{ju}$. Since $\langle M,u \rangle \not\models \mathbf{K}_j\neg\mathcal{B}^\alpha_{ju}$, $\exists (u,w) \in K_j.(M,w) \models \mathcal{B}^\alpha_{ju}$. Thus by Equation (4.21) $B^{\alpha 2}_i$is not empty and by Equations (4.22), (4.33), and (4.35), $B^{a2}_j = B^a_i = B'_{jF} = B'_j$, so $\forall (u',v') \in B'_j.(M, m(v')) \models \mathcal{B}^\alpha_{ju}$. □

Lemma 9.2.8. $\langle M, d \rangle \bar{\times}$No-op *is identical to M except for the names of worlds.*

Proof. Since no statement in $\mathcal{T}$ contains *No-op*, $\forall i \in \mathcal{G}.\forall u \in W, \mathcal{K}_{iu}^{\textit{No-op}} = \top$ (Equation (4.11)). It follows that for all agents $i \in \mathcal{G}$, $C_i^{\textit{No-op}}$ is a partition of W (Equation (4.13)), for all worlds $u \in W$ $G_u^{\textit{No-op}}$ is a singleton (Equation (4.15)), and that $\forall u \in W.\exists u' \in W'$ and $\forall u' \in W'.\exists u \in W$ with $u = m(u')$ and $\forall p \in \mathcal{P}.V(p,u) = V'(p,u')$. Furthermore, if $u = m(u')$ and $v = m(v')$, then by Equations (4.20), (4.21), (4.22), (4.33), (4.34), and (4.35), that $(u,v) \in B_i \leftrightarrow (u',v') \in B_i^{\textit{No-op}}$, and by Equations (4.12) and (4.18) $(u,v) \in K_i \leftrightarrow (u',v') \in K_i^{\textit{No-op}}$. □

Proposition 4.4.14. *Actions do not alter the beliefs of non-observant agents, except for beliefs about knowledge (e.g. they do come to believe that they know that the action could have occurred). Let $\mathcal{L}_{\mathcal{G}B}^{\mathcal{P}}$-$K \subset \mathcal{L}_{\mathcal{G}B}^{\mathcal{P}}$ be the set of formulas that do not contain any subformula $\boldsymbol{K}_j\varphi$ for any $j \in \mathcal{G}$ and $\varphi \in \mathcal{L}_{\mathcal{G}B}^{\mathcal{P}}$. If $i \in N(a,u)$ then for all $\psi \in \mathcal{L}_{\mathcal{G}B}^{\mathcal{P}}$-$K$, $\langle M,u\rangle \models \boldsymbol{B}_i\psi$ iff $\langle M',u'\rangle \models \boldsymbol{B}_i\psi$.*

Proof. Assume $i \in N(a,d)$ and consider $(d,u) \in B_i$. By Lemma 9.2.8 $(d^0,u^0) \in M^0[B_i]$ where $m(d^0) = d$ and $m(u^0) = u$. By Equations (4.24) and (4.27) $M^{d'[a]} = M^a \in H_i$ and $M^{u[a]} = M^0$. It follows by Equations (4.34) and (4.35) that $(d',u^0) \in B_i'$. Now consider $(d',v') \in B_i', m(v') = v$. By Equations (4.24) and (4.27) $M^{d'[a]} \in H_i$ and it follows by Equations (4.34) and (4.35) that $M^{v'[a]} = M^0$, i.e. $v' \in M^0[W]$. Also by Equation (4.34) $\exists(d^0,u^0) \in M^0[B_i].m(d^0) = d, m(v^0) = v$, and it follows from Lemma 9.2.8 that $(d,v) \in B_i$. Thus, for every $(d,u) \in B_i$ there is $(d',u') \in B_i'$ with $u' \in M^0[W]$ and $m(u') = u$, and for every $(d',u') \in B_i'$ it is the case that $u' \in M^0[W]$ and there is $(d,m(u')) \in B_i$, and the theorem follows from Lemm 9.2.8. □

9.3 m$\mathcal{B}$ Proofs

In this section we prove several propositions regarding the semantics of m$\mathcal{B}$, where $\mathcal{T}$ is an action theory and $\langle M,d\rangle \times a = \langle M',d'\rangle$ where $M = \langle W,R,V\rangle$, $d \in W$, $M' = \langle W',R',V'\rangle$, and $d' \in W'$.

Proposition 5.2.1. *Full observers come to believe announcements that do not contradict prior knowledge or the action's preconditions. If "if ψ then i observes a", "a announces*

φ"$\in \mathcal{T}$, *and* $\langle M, d\rangle \models \psi$, *and* $\langle M, d\rangle \not\models \boldsymbol{K}_i\neg(a_{pre} \wedge \varphi)$, *then* $\langle M', d'\rangle \models \boldsymbol{B}_i\varphi$.

By Definitions 2 and 3 every $u' \in \sim_i^{d'}$ is either $\langle u, e^{\varphi}\rangle$ or $\langle u, e^{\neg\varphi}\rangle$ for some u in $\sim_i^d$ (since $e^{\top}$ is epistemically disconnected from designated events e^{φ} and $e^{\neg\varphi}$ for full observers). Assume that there exists some $u' \in \rightarrow_i^{d'}$ such that $\langle M', u'\rangle \not\models \varphi$. According to the event preconditions given in Definition 3, and since add and del are $\varnothing$, $u' = \langle u, e^{\neg\varphi}\rangle$ for some u in $\sim_i^d$. Since $\langle M, d\rangle \not\models \mathbf{K}_i\neg(a_{pre} \wedge \varphi)$, there exists some $v \in \sim_i^d$ such that $\langle M, v\rangle \models a_{pre} \wedge \varphi$. It follows by Definitions 2 and 3 that there exists $\langle v, e^{\varphi}\rangle \in \sim_i^{d'}$. However, since $Q(e^{\varphi}, e^{\neg\varphi})(i) = \bot$, it follows that $\langle\langle v, e^{\varphi}\rangle, \langle u, e^{\neg\varphi}\rangle\rangle \notin R_i'$, contradicting the assumption that $u' \in \rightarrow_i^{d'}$. Rejecting the assumtion, it follows that $\langle M', d'\rangle \models \mathbf{B}_i\varphi$.

Proposition 5.2.2. *Full and partial observers know that full observers come to believe announcements unless they contradict prior knowledge or the action's preconditions. If either "if* ω *then* j *observes* a" $\in \mathcal{T}$ *or "if* ω *then* j *is aware of* a" $\in \mathcal{T}$, *and "if* ψ *then* i *observes* a", "a *announces* φ"$\in \mathcal{T}$, *and* $\langle \mathcal{M}, d\rangle \models \omega$, *and if for every* $u \in \sim_j^d \langle M, u\rangle \models \psi$ *and* $\langle M, u\rangle \not\models \boldsymbol{K}_i\neg(a_{pre} \wedge \varphi)$, *then* $\langle M', d'\rangle \models \boldsymbol{K}_j\boldsymbol{B}_i\varphi$.

For every $\langle u, e\rangle \in W'$ such that $d' \sim_j' \langle u, e\rangle$, it follows from Definition 2 that $d \sim_j u$, i.e. $u \in \sim_j^d$. Since $\langle M, u\rangle \models \psi$ and $\langle M, u\rangle \models \mathbf{K}_i\neg(a_{pre} \wedge \varphi)$, it follows from Proposition 5.2.1 that $\langle M', \langle u, e\rangle\rangle \models \mathbf{B}_i\varphi$, and thus $\langle M', d'\rangle \models \mathbf{K}_j\mathbf{B}_i\varphi$.

Proposition 5.2.3. *Full observers learn the value of sensed propositions. If "if* ψ *then* i *observes* a", "a *determines whether* φ"$\in \mathcal{T}$ *and* $\langle \mathcal{M}, d\rangle \models \psi$, *then* $\langle \mathcal{M}', d'\rangle \models \boldsymbol{K}_i\varphi \vee \boldsymbol{K}_i\neg\varphi$.

If $d' \models \varphi$ then $d \models \varphi$ (there are no ontic effects) and $d' = \langle d, e^{\varphi}\rangle$ since pre$(e^{\neg\varphi}) = \neg\varphi$ and $e^{\top} \notin \Gamma$. Consider some $u' \in \sim_i^{d'}$. Since $Q(e, f)(i) = \bot$ for every $e, f \in E$ except for $e = f = e^{\varphi}$, it follows that $u' = \langle u, e^{\varphi}\rangle$ for some $u \in W$. Since pre$(e^{\varphi}) = \varphi$, $u \models \varphi$, and since there are no ontic effects, $u' \models \varphi$. If $d' \models \neg\varphi$ then for every $u' \in \sim_i^{d'}$, $u' \models \neg\varphi$ by the same reasonging.

Proposition 5.2.4. *Full and partial observers learn that full observers learn sensed propositions. If either "if* ω *then* j *observes* a" $\in \mathcal{T}$ *or "if* ω *then* j *is aware of* a" $\in \mathcal{T}$, *and "if* ψ *then* i *observes* a", "a *determines whether* φ"$\in \mathcal{T}$, *and if for every* $u \in \sim_j^d$, $\langle M, u\rangle \models \psi$, *then* $\langle M', d'\rangle \models \boldsymbol{K}_j(\boldsymbol{K}_i\varphi \vee \boldsymbol{K}_i\neg\varphi)$.

Since $e^{\top}$ is not connected to e^{φ} or $e^{\neg\varphi}$ for full and partial observers, it follows from Definition 2 that every $u' \in \sim_j^{d'}$ is either $\langle u, e^{\varphi}\rangle$ or $\langle u, e^{\neg\varphi}\rangle$ for some $u \in \sim_j^{d}$. Since $\langle M, u\rangle \models \psi$, it follows from Proposition 5.2.3 that $\langle M', u'\rangle \models \mathbf{K}_i\varphi \vee \mathbf{K}_i\neg\varphi$, and thus $\langle M', d'\rangle \models \mathbf{K}_j(\mathbf{K}_i\varphi \vee \mathbf{K}_i\neg\varphi)$.

Proposition 5.2.5. *Ontic effects update the state: if "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$ then for every k in $\{0, \ldots, n\}$, $\langle M', d'\rangle \models l_k$.*

Since $\Gamma = \{e^c\}$, by Definition 2 $d' = \langle d, e^c\rangle$ and by Definition 4, $V'(d')$ contains p if $l_j = p$ and does not contain p if $l_j = \neg p$.

Proposition 5.2.6. *Full and partial observers learn the effects of ontic actions: if "a causes $l_0, \ldots, l_n$"$\in \mathcal{T}$ and either "if ψ then i observes a"$\in \mathcal{T}$ or "if ψ then i is aware of a"$\in \mathcal{T}$ and $\langle M, d\rangle \models \psi$ then for every k in $\{0, \ldots, n\}$, $\langle M', d'\rangle \models \boldsymbol{K}_i l_k$.*

Since i is a full or partial observer, by Definitions 4 and 2 every $u' \in \sim_i^{d'}$ is $\langle u, e^c\rangle$ for some $u \in W$, and it folloss from Proposition 5.2.5 that $\langle M', u'\rangle \models l_k$, and thus $lM', d'\rangle \models \mathbf{K}_i l_k$.

Proposition 5.2.7. *Full and partial observers learn that full and partial observers learn ontic effects: If either "if ψ then j observes a"$\in \mathcal{T}$, or "if ψ then j is aware of a"$\in \mathcal{T}$, and either "if ω then i observes a" $\in \mathcal{T}$ or "if ω then i is aware of a" $\in \mathcal{T}$, and $\langle M, d\rangle \models \psi$ and for every $u \in \sim_j^{d}$, $\langle M, u\rangle \models \omega$, and "$a$ causes $l_0, \ldots, l_n$"$\in \mathcal{T}$, then for every k in $\{0, \ldots, n\}$, $\langle M', d'\rangle \models \boldsymbol{K}_j\boldsymbol{K}_i l_k$.*

Since j is a full or partial observer, by Definitions 4 and 2 every $u' \in \sim_j^{d'}$ is $\langle u, e^c\rangle$ for some $u \in W$, so it follows form Proposition 5.2.6 that $\langle M', u'\rangle \models \mathbf{K}_i l_k$ and thus $\langle M', d'\rangle \models \mathbf{K}_j\mathbf{K}_i l_k$..

Proposition 5.2.8. *Actions do not alter the beliefs of non-observant agents, except for beliefs about knowledge (e.g. they do come to believe that they know that the action could have occurred). Let $\mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$-$K \subset \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$ be the set of formulas that do not contain any subformula $\boldsymbol{K}_i\varphi$ for any $i \in \mathcal{G}$ and $\varphi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$. If for every "if ψ then i observes a" or "if ψ then i is aware of a" in $\mathcal{T}$, $\langle \mathcal{M}, d\rangle \not\models \psi$, then for every $\varphi \in \mathcal{L}_{\mathcal{G}}^{\mathcal{P}}$-$K$, $\langle \mathcal{M}', d'\rangle \models \boldsymbol{B}_i\varphi$ iff $\langle \mathcal{M}, d\rangle \models \boldsymbol{B}_i\varphi$.*

Since every action $e^\top$ has precondition $\top$ and no ontic effects, and every world has a reflexive edge for evey agent, M' restricted to worlds $\{\langle u, e^\top\rangle \mid u \in W\}$ is isomporhic, and hence bisimilar, to M. Since for every action and events e^* and $e^\top$ where $e^* \in \{e^\varphi, e^{\neg\varphi}, e^c\}$, $\langle\langle e^*, e^\top\rangle, N\rangle \in Q$ but $\langle\langle e^\top, e^*\rangle, N\rangle \notin Q$, it follows that there are no $\langle\langle u, e^\top\rangle, \langle v, e^*\rangle\rangle$ in R'_j for any $j \in \mathcal{G}$, although for every $\langle u, v\rangle \in R_i$ there is $\langle\langle u, e^*\rangle, \langle u, e^\top\rangle\rangle \in R'_i$.

9.4 Cooperation-Agnostic Search Proofs

Definition 5. Given a domain, a policy Π *solves* (is a solution for) a g-node $n = \langle s, n'\rangle$ for a goal φ, written $Solves(\Pi, n, \varphi)$, iff n is not cyclic and any of the following hold, with $t = d(n)$ and $i = \mathcal{G}[t]$:

- $s \models \varphi$, or
- $i \in \mathcal{G}_s$ and $\Pi(s^i, t)$ is defined and $\forall n'' \in T(n, \Pi(s^i, t)).Solves(\Pi, n'', \varphi)$, or
- $i \in \mathcal{G}_e$ and $\forall a \in \beta(s, i).\forall n'' \in T(n, a).Solves(\Pi, n'', \varphi)$.

Policy Π solves a set of g-nodes N for goal φ, written $Solves(\Pi, N, \varphi)$, if $\forall n \in N.Solves(\Pi, n, \varphi)$. Policy Π solves a g-state s for goal φ, written $Solves(\Pi, s, \varphi)$, if $Solves(\Pi, \langle s, Null\rangle, \varphi)$. Policy Π solves a set of g-states S for goal φ, written $Solves(\Pi, S, \varphi)$, if $\forall s \in S.Solves(\Pi, s, \varphi)$. Policy Π solves a p-node $\hat{n}$ for goal φ, written $Solves(\Pi, \hat{n}, \varphi)$, if $Solves(\Pi, \hat{n}[N], \varphi)$. Policy Π solves a set of p-nodes $\widehat{N}$ for goal φ, written $Solves(\Pi, \widehat{N}, \varphi)$, if $\forall \hat{n} \in \widehat{N}.Solves(\Pi, \hat{n}, \varphi)$.

We distinguish between the size of a policy (number of entries), its g-length (maximum number of g-states visited during execution), and its p-length (maximum number of times the policy is queried during execution).

Definition 6. The *size* of policy Π, written $|\Pi|$, is the number of entries it contains.

Definition 7. The *g-length* of policy Π with respect to a g-node n that it solves, is denoted $G - length(\Pi, n)$. If $Solves(\Pi, n = \langle s, n'\rangle, \varphi)$, with $t = d(n)$ and $i = \mathcal{G}[t]$, then $G - length(\Pi, n)$ is

- 1 if $s \models \varphi$,
- $1 + max_{n'' \in T(n,\Pi(s^i,t))}(G - length(\Pi, n''))$ if $s \not\models \varphi$ and $i \in \mathcal{G}_s$,
- $1 + max_{a \in \beta(s,i)}(max_{n'' \in T(n,a)}(G - length(\Pi, n''))$ if $s \not\models \varphi$ and $i \in \mathcal{G}_e$.

The g-length of Π with respect to a set of g-nodes N, written $Length(\Pi, N)$, is $max_{n \in N}(g - length(\Pi, n))$. The g-length of Π with respect to a set of g-states S, written $Length(\Pi, S)$, is $g - length(\Pi, \{\langle s, Null\rangle \mid s \in S\})$. The g-length of Π with respect to a p-node $\hat{n}$, written $Length(\Pi, \hat{n})$, is $g - length(\Pi, \hat{n}[N])$. The g-length of Π with respect to a set of p-nodes $\widehat{N}$, written $Length(\Pi, \hat{n})$, is $(g - length(\Pi, \hat{n}[N]).\ max_{\hat{n} \in \widehat{N}}(g - length(\Pi, \hat{n}))$. The g-length of a policy, denoted $G - length(\Pi)$, is its g-length with respect to the start states S_0 passed to IDFS.

Definition 8. The *p-length* of policy Π is defined with respect to a g-node n that it solves, and is denoted $P - length(\Pi, n)$. If $Solves(\Pi, \hat{n} = \langle \hat{s}, N\rangle, \varphi)$, with $t = d(n)$ and $i = \mathcal{G}[t]$, then $P - length(\Pi, n)$ is

- 1 if $s \models \varphi$,
- $1 + max_{n'' \in T(n,\Pi(s^i,t))}(P - length(\Pi, n''))$ if $s \not\models \varphi$ and $i \in \mathcal{G}_s$,
- $max_{a \in \beta(s,i)}(max_{n'' \in T(n,a)}(P - length(\Pi, n''))$ if $s \not\models \varphi$ and $i \in \mathcal{G}_e$.

The p-lengths of a policy with respect to a set of g-nodes, a p-node, and set of p-nodes are defined similarly as for g-length.

Definition 9. Given domain D, a *g-path* is a sequence of g-nodes $(n_0, \ldots, n_m)$ such that each of the following hold:

- Either $n_m[s] \models \varphi_{goal}$ or $\mathcal{G}[d(n_m)] \in \mathcal{G}_s$ or n_m is cyclic.
- For $i < m$, $n_i[s] \not\models \varphi_{goal}$ and $\mathcal{G}[d(n_i)] \in \mathcal{G}_e$ and n_i is not cyclic.
- For $i < m$, $n_{i+1} \in T(n_i, a)$ for some $a \in \beta(n_i[s], \mathcal{G}[d(n_i])$.

Given g-node n, an n-path is a g-path $(n_0, \ldots, n_m)$ with $n_0 = n$. Given the set of g-nodes N, an N-path is a g-path $(n_0, \ldots, n_m)$ with $n_0 = N$.

Lemma 9.4.1. *If $\hat{n}$ has a solution of g-length $1 \leq l \leq L$, then algApply($\hat{n}, a, \varphi, L$) returns $\widehat{N}''$ and which has a solution of g-length $l - 1$.*

Proof. If $l = 1$ then DESCEND returns $\varnothing$, since recursive callse to DESCEND will bottom out by returning $\varnothing$ on Line 9 or 17, so APPLY returns $\varnothing$ which is solved by the empty policy. Otherwise, the recursive calls to DESCEND don't bottom out on Lines 3 or 6 or 15 because there is a solution. Instead recursive calls to DESCEND bottom out on Line 8, i.e. at the next system agent, and DESCEND returns a set of g-nodes corresponding to the next system agent timestep. Since each recursive call to DESCEND transitions g-nodes according to β (Lines 12 and 13) the states of the g-nodes returned by DESCEND are those resulting from applying the actions given by β according to Definition 5. Thus the solution to the p-nodes returned by APPLY is the same as the solution to the $\hat{n}$ without the entry for $\hat{n}$. □

Lemma 9.4.2. *If for some action a, for every $n \in N$ and some a, for every $n \in T(n, a)$, $d(n) \leq L$, and an empty policy solves n, then* APPLY*($\langle \hat{s}, N \rangle, a, \varphi, L$) returns $\varnothing$.*

Proof. It follows from Definition 5 that $gnodeisnotcyclic$ and $n[s] \models \varphi$. Thus DESCEND(N, φ, L) returns $\varnothing$ (either on Line 9 or Line 17, possibly after recursive calls to DESCEND that also return $\varnothing$), and so PERSPECTIVE(N") returns $\varnothing$ and this APPLY returns $\varnothing$. □

Lemma 9.4.3. *If there is a solution for $\hat{n}$ with g-length less than or equal to L then there is some action a such that* APPLY*($\hat{n}, a, \varphi, L$) returns not Null.*

Proof. If the call to DESCNED on Line 3 returns Null, then either DESCEND returns on Line 3 or 6 or a recursive call to DESCEND returns on Line 3 or 6. However, since some solution has g-length less than L, the recursive calls to DESCEND will bottom out with returns on Line 9 or 17, not 3 or 6. □

Lemma 9.4.4. *Let i be the smallest integer greater than $d(\langle \hat{s}, N \rangle)$ such that $\mathcal{G}[i] \in \mathcal{G}_s$, i.e. the index of the next s-agent.* APPLY*($\hat{n}, a, \varphi, L$) returns Null if L is less than i and L is less than the g-length of every policy that solves $\langle \hat{s}, N \rangle$.*

Proof. Since DESCEND will recursively reach L before reacing solutions (i.e. nodes whose states satisfy φ), the call to DESCEND on (APPLY Line 3) and any recursive calls within DESCEND (DESCEND Line 13) return Null, with recursion bottoming out with DESCEND returning on Line 3 or Line 6. □

Lemma 9.4.5. *If* EVALUATE*($\widehat{N}, \varphi, L$) returns policy Π, then Π solves $\widehat{N}$ for φ.*

Proof. As a base coase, consider that Π has a p-length of 0 (i.e. is empty). It follows that $\widehat{N} = \varnothing$ and thus Π is a solution. As an inductive step, consider that Π has p-length $l > 0$ and assume that the lemma holds for $l - 1$. For every $\hat{n} \in \widehat{N}$, it folows from Lemma 9.4.1 and from the inductive hypothesis that that $\pi(a)$ (as defined on Line 7) is a solution for the p-nodes returned by APPLY($\hat{n}, a, \varphi, L$). Thus $\{\langle\langle \hat{n}[\hat{s}], d(\hat{n}[N])\rangle, a\rangle\} \cup \pi(a)$ solves $\hat{n}$ and Π (returned on Line 12 solves $\widehat{N}$. □

Lemma 9.4.6. *If* DESCEND*(N, φ, L) returns $\varnothing$ then the empty policy solves N.*

Proof. If DESCEND returns $\varnothing$ on Line 9, then clearly for every $\langle s, n'\rangle \in N$, $s \models \varphi$. If DESCEND returns $\varnothing$ on Line 17, then for any $\langle s, n'\rangle \in n$ such that $s \not\models \varphi$, the recursive call to DESCEND on Line 13 returns $\varnothing$. It follows that for each recursive call to DESCEND, for every action considered on Line 12, the recursion bottoms out with DESCEND returning $\varnothing$ at a goal, i.e. returning at Line 9 or Line 17, and the recursion does not bottom out on Line 3 (reaching depth L) or Line 6 (reaching a cycle), or returning a non-empty set on Line 17 (at the next s-agent's turn). Thus every N-pathend at a goal and it follows that $\varnothing$ solves N. □

Lemma 9.4.7. *If there exists a non-empty policy with g-length greater than or equal to L that solves $\widehat{N}$ for φ then* EVALUATE*($\widehat{N}, \varphi, L$) returns a solution.*

Proof. By Lemma 9.4.3 there is some action a such that $\pi(a) \neq Null$. As a base case, suppose that the a selected on Line 10 is such that for every $n \in \hat{n}[N]$, $T(n, a)$ has a solution of p-length 0 (i.e. the solution $\varnothing$). Then by Lemma 9.4.2, APPLY returns $\varnothing$ and the recursive call to EVALUATE returns $\varnothing$, and the policy returned on Line 12 is $\{\langle\langle \hat{n}[\hat{s}], d(\hat{n}[N])\rangle, a\rangle\}$, which solves $\widehat{N}$. As an inductive step, suppose that the a selected on Line 10 is such that for every $n \in \hat{n}[N]$, $T(n, a)$ has a solution of p-length $l > 0$, and assume that the Lemma

holds for $l-1$. By Lemma 9.4.1 when APPLY is called on Line 11 it returns p-nodes for which there is a solution with p-length $l-1$. It follows from the inductive assumption that the recursive call to EVALUATE on Line 11 returns a policy that solves the p-nodes returned by APPLY, and $\{\langle\langle\hat{n}[\hat{s}], d(\hat{n}[N])\rangle, a\rangle\} \cup \pi(a)$ solves $\hat{s}$. □

Lemma 9.4.8. *If some policy* Π *solves* $\widehat{N}$ *and the g-length of* Π *is greater than or equal to* L *then* EVALUATE*(*$\widehat{N}, \varphi, L$*) returns Null.*

Proof. Let i be the smallest integer greather than $d(\widehat{N})$ such that $\mathcal{G}[i] \in \mathcal{G}_s$. If $i \geq L$ then by Lemma 9.4.4, for any action a, APPLY($\hat{n}, a, \varphi, L$) returns Null. Otherwise, the recursive call to Evaluate to EVALUATE on Line 11 increases i (since APPLY advances i via DESCEND while preserving L), and eventually APPLY returns Null. When APPLY returns Null, the recursive call to Evaluate on Line 8 returns Null (on Line 3) so Evaluate will return Null on Line 9. □

Lemma 9.4.9. *If every* $n{-}path = (n_0, \ldots, n_m)$ *reaches a goal before the next s-agent turn or the depth limit, i.e.* $n_m[s] \models \varphi_{goal}$ *and* $\mathcal{G}[depth(n_m)] \in \mathcal{G}_e$ *and* $depth(n_m \leq L)$*, then* DESCEND*(*N, φ, L*) return* $\varnothing$*.*

Proof. It follows for the assumptions that DESCEND does not return on Line 3, 6, or 9. Let $(n, \ldots, n_m)$ be the shortest N-path. As a base condition suppose that $m = 0$. Thus for every $n \in N$, $n[s] \models \varphi$, and DESCEND will never enter the loop starting on Line 11 and will return $\varnothing$ on Line 17. As an inductive step, assume that the lemma holds for $m-1$ and suppose $m > 0$. For each $n \in n$ such that $n[s] \not\models \varphi$, and for each $a \in \beta(n[s], \mathcal{G}[d(n)])$, $T(n, a)$ is an element of a n-path, so by induction N'' gets $\varnothing$ on Line 13 so DESCEND returns $\varnothing$ on Line 17. □

Proposition 6.3.1. *If a policy exists that solves* S_0 *for* φ*, then* IDFS(S_0, φ) *returns a solution.*

Proof. Let l be the shortest g-length among the policies that solve $\langle D, \mathcal{G}, S_0, \varphi_{goal}\rangle$. If $l = 0$, then by Lemma 9.4.9 the call to DESCEND on Line 6 will return $\varnothing$ and IDFS will return $\varnothing$ on Line 10. Otherwise, while $L < l$, EVALUATE will return Null by Lemma 9.4.8,

and when $L = l$, by Lemma 9.4.7 EVALUATE returns a solution, which IDFS returns on Line 16. □

Proposition 6.3.2. *If* $\text{IDFS}(S_0, \varphi_{goal})$ *returns policy* Π*, then* Π *is a solution.*

Proof. If IDFS returns on line 10 then by Lemma 9.4.6 the returned empty policy $\varnothing$ is a solution. If IDFS returns on line 16 then by Lemma 9.4.5 the returned policy Π is a solution. □