NORTHWESTERN UNIVERSITY

Analogical Theory of Mind: Computational Model and Applications

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Irina Rabkina

EVANSTON, ILLINOIS

September 2020

ABSTRACT

Analogical Theory of Mind: Computational Model and Applications

Irina Rabkina

Theory of mind (ToM) reasoning is defined as the ability to reason about another's internal states, such as beliefs, goals, and desires. It is a major aspect of human social interaction and is mastered by most typically developing children by age five. On the other hand, simulated agents generally lack such reasoning abilities—even in situations when they must interact with others. This dissertation shows that human-like ToM reasoning improves simulated agents' decision-making in complex multi-agent environments.

The major contribution of this work is the Analogical Theory of Mind (AToM) model, an implemented computational cognitive model of human ToM reasoning and development. AToM claims that human ToM reasoning and development occur via analogical processes (i.e., Gentner's Structure-mapping Theory). This claim is tested through simulations of three related phenomena: (1) children learning ToM from structured stories; (2) children learning ToM as a side effect of learning a complex grammatical structure; and (3) children's failures in pretend play. AToM successfully models the children's performance in each and makes testable predictions.

The model is then applied to simulated agents reasoning in two complex multi-agent environments. First, it is used for goal recognition in the Minecraft game. AToM slightly underperforms a state-of-the-art goal recognition system under standard goal recognition conditions (i.e., when reasoning from planner outputs), and significantly outperforms it when reasoning from only external observations. Next, AToM is used to recognize intent to cooperate among simulated players in stag-hunt, a prisoner's dilemma-style game. AToM's performance does not differ from a Bayesian model or human

performance on a limited dataset from the literature and performs well when the dataset is extended in size and complexity. These results suggest that using AToM during reasoning can improve multi-agent interaction.

ACKNOWLEDGEMENTS

This dissertation would not be possible without the support of many. Financial support for this work was provided by Socio-Cognitive Architectures for Adaptable Autonomous Systems Program of the Office of Naval Research [N00014-13-1-0470]. Summer 2018 internship research was funded by the Naval Research Enterprise Intern Program (NREIP), and dissertation year funding was partially provided by anonymous donors to Northwestern University's Terminal Year Fellowship.

I am extremely grateful to my advisor, Ken Forbus, for his unwavering support throughout the last five years. Ken has encouraged me to pursue my good ideas and cautioned me from pursuing the bad ones. Ken treats his students like colleagues from day one. His high expectations and steadfast mentorship make us all better.

I would also like to thank Dedre Gentner. I aspire to be half as conversant in the cognitive science literature as Dedre. Our group's joint meetings with her lab have always been my favorite for their fruitful discussions and friendly banter. Whether in lab meetings, over email, or in class, I always walk away from a conversation with Dedre having learned something new.

Thank you also to Laura Hiatt, who took her role as the doctoral consortium mentor of a first-year grad student seriously and has been in my corner ever since. Laura took the scattered ideas I brought into that consortium and pushed me to develop them into the first glimpse of what is now a dissertation. I am grateful for Laura's feedback on my work over the years and for our collaborations. I look forward to continuing to collaborate on future projects.

I am grateful for Ian Horswill's input into this dissertation. Ian, Laura, Dedre and Ken have been an incredible committee to work with. Their guidance and expertise have been invaluable to the work in this dissertation and my development as a researcher. Thank you all.

A huge thanks also to Sara Sood. Without her I would not be a computer scientist. Literally.

To my collaborators and coconspirators in the Qualitative Reasoning Group—thank you. Dave Barbella, Maria Chang, Subu Kandaswamy, Matt McLure, CJ McFate, and Chen Liang—thank you for paving the way. Willie Wilson, Joe Blass, Max Crouse, Constantine Nakos, Kezhen Chen, Will Hancock, Sam Hill, Danilo Ribeiro, Taylor Olson, and Cathy Lin—thank you for walking beside me. Tom Hinrichs and Maddy Usher—thank you for sharing your expertise and institutional knowledge. We all know that you two are QRG's duct tape and WD-40.

Thank you to my collaborators outside of QRG, Christian Hoyos, Mak Roberts, and Pavan Kathnaraju.

Thank you to my many friends and colleagues in the NU CS department. Thank you especially to Jamie Gorson and Spencer Florence, who provided the best distractions from my computer screen—but were always respectful of, "I'm on a deadline."

Finally, and most importantly, a huge thank you to my family. Thank you to my parents, Alex and Svetlana, for always being my number one fans. Thank you to my sister, Liya, for always being my number one confidant (and my number one thorn in the side). And thank you to my husband, Konstantin, for always being my number one. I would not be who I am without you all, and I would not have gotten through grad school without your support. Thank you.

To the dozens of K-12 public school teachers,

who gave me the foundation on which to build this dissertation.

And to my parents,

who taught me everything else.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 Introduction

Theory of mind (ToM) reasoning is a major aspect of human social cognition. It allows us to consider and predict the internal states (i.e., knowledge, beliefs, goals, desires, etc.) of others. Implicitly or explicitly, ToM informs all of our interactions. The processes by which people learn and perform ToM reasoning have been hotly debated by cognitive scientists for decades. On the other hand, there have been few attempts to give simulated agents the ability to reason about others in the same way. The goal of this dissertation is two-fold: to improve our understanding of ToM reasoning in humans and to apply that understanding to simulated agents, thereby giving them the ability to reason about other agents' internal states in multiagent environments.

In this dissertation, I develop the Analogical Theory of Mind (AToM) model. As a model of human ToM reasoning and development, AToM suggests a process-level mechanism by which people continuously improve their ability to reason about others. As a component of simulated agents' reasoning, it gives simulated agents the ability to do the same.

1.1 Motivation

AToM proposes structure-mapping (Gentner, 1983) as the process underlying ToM reasoning and development. This argument is inspired by Bach's (2011) similar theoretical proposal and motivated by Hoyos and colleagues' (Hoyos, Horton & Gentner, 2015; Hoyos, Horton, Simms & Gentner, under review) empirical finding that structural alignment aids children's ToM learning. More broadly, Structure-mapping Theory (SMT; Gentner, 1983) describes a process of higher order reasoning by analogy. SMT has inspired computational cognitive models of many cognitive processes, including visual problem-solving (Lovett & Forbus, 2017), moral decision-making (Dehghani et al., 2008; Blass & Forbus, 2015), and language learning (McFate & Forbus, 2016). These are all implemented within the

Companion cognitive architecture (Forbus & Hinrichs, 2017), which aims to reach human-level AI through strong architecture-level commitments to reasoning and representation[1]. The breadth of cognitive processes modeled by structure-mapping within Companions makes them an attractive candidate for modeling ToM reasoning, as well.

Using the cognitive model as part of simulated agents' reasoning is motivated by the agents' need to reason about other agents. Whether cooperating, competing, or simply coexisting in an environment, being able to reason about other agents is likely to benefit simulated agents[2]. This is especially true when internal states (i.e., their plans, policies, and goals) are opaque—such as when the agent is interacting with a person. ToM reasoning gives people the ability to reason about opaque others. In fact, typically developing adults are at ceiling at most ToM tasks (see Happe, 1994; Baron-Cohen, Jolliffe, Mortimore & Robertson, 1997). By using a computational cognitive model of human ToM reasoning to reason about other agents, simulated agents should be able to achieve similar performance in a wide variety of domains—without the need for large amounts of domain-specific training data or task-specific model architectures (cf. statistical and deep learning approaches).

1.2 Claims and Contributions

This dissertation is based around two central claims:

1) Human ToM reasoning and development occur via analogical processes.

2) The same processes can be used by simulated agents to improve their ToM reasoning.

Thus, the main contribution of the work is the Analogical Theory of Mind (AToM) model. AToM is a

[1] In the case of Companions, analogical processes and qualitative representations specifically.
[2] We discuss the use cases for ToM reasoning in these situations in detail in Rabkina, Nakos & Forbus (2019a).

theoretical and computational cognitive model of ToM reasoning and development which is developed and tested throughout this dissertation in support of these claims.

In support of Claim 1, AToM provides a process-level model of how children learn ToM. The theory driving the model makes the following additional claims:

3) Human ToM reasoning occurs specifically via structure-mapping processes.

4) Human ToM reasoning occurs in working memory when possible. Retrieval from long-term memory occurs when triggered by the environment.

5) Human ToM reasoning is driven by analogical inferences.

These additional claims are supported by computational modeling experiments. Specifically, computational modeling experiments show that AToM matches children's performance when learning ToM from structured false belief stories and as a side effect of learning a complex grammatical rule. Furthermore, because it has been hypothesized (see Weisberg, 2015) that the processes and representations for ToM reasoning are learned, in part, through pretend play, this dissertation makes the additional claim that:

6) Successful pretend play requires the ability to reason analogically, including generating and accepting appropriate candidate inferences.

This claim is supported by a computational cognitive modeling experiment, which shows that failures in generating and accepting analogical inferences can explain young children's failures in pretend play.

Claim 2 is supported by two experiments in which AToM is used to reason about simulated agents. This also leads to the additional claim that:

7) ToM reasoning, specifically via AToM, allows simulated agents to reason about the

internal states of others even when those internal states are not inspectable.

In the experiments supporting these claims, AToM's goal and intent recognition abilities are compared to those of a state-of-the art goal recognition system and a Bayesian model. To compare against the goal recognition system, a novel dataset that allows reasoning from several levels of access to internal states (i.e., agent planner outputs to purely external observations) is created. To compare against the Bayesian model, a dataset from the literature is used. This dataset is then extended in size and complexity in order to further test AToM's ability to reason about agents' internal states.

1.3 Overview

This dissertation is comprised of two major components, corresponding to its major claims: computational modeling of ToM reasoning and applications to simulated agents in multiagent environments. Related work and future directions are also discussed.

Chapter 2 reviews prior work on ToM modeling from the points of view of psychology, cognitive modeling, and artificial intelligence. First, the preeminent theories of ToM reasoning and development, including psychological evidence that supports and contradicts each, are discussed. Then, the cognitive models of ToM reasoning and development that preceded AToM are described, as are applications of ToM reasoning by simulated agents.

Chapter 3 describes the Analogical Theory of Mind (AToM) model that is used throughout the rest of the work. This includes the computational analogy stack that it is built upon.

Chapters 4, 5, and 6 make up the computational cognitive modeling component of the dissertation. In chapters 4 and 5, AToM is used to model training studies in which children improved ToM reasoning through explicit false belief training (chapter 4) and through training on a complex grammatical structure (chapter 5). In chapter 6, pretend play, a precursor to ToM reasoning, is modeled as an

analogical process.

Chapters 7 and 8 demonstrate applications of AToM to reasoning about simulated agents. In chapter 7, AToM is compared to a state-of-the-art goal recognition system. AToM performs comparably to the goal recognition system when internal states of the agent whose goals are being recognized are available, and significantly outperforms it when information about internal states is abstracted away. Chapter 8 builds on this, by showing that AToM performs comparably to a Bayesian model and humans when attempting to recognize cooperation between agents in a simple prisoner's dilemma-style game, stag-hunt. AToM's performance is then tested on an expanded version of the stag-hunt dataset. AToM is extended to second-order ToM reasoning, although testing on the expanded dataset does not support the hypothesis that second-order ToM improves predictive performance.

Finally, the work is summarized, claims are revisited, and future directions are proposed in chapter 9.

2 Related Work: Theory of Mind Reasoning in Humans and Simulated Agents

Theory of Mind (ToM) has been well-studied in psychology and has recently gained more attention in artificial intelligence. This chapter describes the prior work that inspires and informs the Analogical Theory of Mind (AToM) model. Because AToM proposes a process by which ToM reasoning and development occur, current psychological theories of ToM reasoning and development are discussed first. Next, computational cognitive models of ToM are discussed. Finally, because AToM is applied to simulated agents reasoning about other agents, I discuss the ways in which other agents' internal states have been integrated into software agents' reasoning in the past.

2.1 Human Theory of Mind Reasoning and Development

2.1.1 Theory Theory. Several accounts of ToM reasoning and development fall under the umbrella of Theory Theory, all unified by the proposal that ToM reasoning occurs with respect to a theory, or a set of rules that explains how the beliefs, desires, and mental states of others can be predicted (Gopnik & Wellman, 1994). The most of popular of these is the child-scientist view (Leslie, 1994), which proposes a process by which the theory that defines one's ToM is learned. In this section, the child-scientist view will be used as the default description of Theory Theory.

Per the child-scientist view, a child starts with a naïve theory for reasoning about others and adjusts it as new evidence is encountered. This process is analogous to a scientist starting with a working hypothesis or theory, performing experiments to find evidence in support of (or in contradiction to) the theory, and adjusting the theory based on experimental findings. In the child's case, the working theory may be as simple as, "everyone has the same mental states as I do," experiments might be interactions with others, and findings may be the result of the interaction (i.e., whether the child's predictions were correct). While each child's experiences will vary, most children converge to a similar theory of ToM by

about age five (Wellman & Liu, 2004).

Evidence suggests that the theory converges not only on a complete ToM, but that there are earlier developmental milestones. In a meta-analysis of several studies, Wellman and Liu (2004) found that ToM follows a consistent trajectory: children first learn about others' differing desire states (i.e., that two people might want different things), then differing belief states (i.e., that two people may have beliefs that differ from each other[3]), then knowledge states (i.e., that a person's beliefs may differ from reality), and finally the interaction between them. Wellman and Liu confirmed this trajectory in an empirical study. Children of varying ages were asked to complete a set of ToM tasks. Children's performance showed a pattern that is consistent with the meta-analysis: children who could successfully complete tasks that require belief state knowledge could also complete tasks that required an understanding of desire states (but not vice versa), children who could complete tasks that test knowledge states could complete belief state and desire state tasks (but not vice versa), etc. Bartsch and Wellman (1995) argue that such a pattern is consistent with the Theory Theory account of ToM. Desire states (e.g., "Do you want a cookie?") are discussed with children earlier than are belief states (e.g., "What do you think is in the box?") and knowledge states (e.g., "Do you know what this is?"), giving them evidence for theory adaptation in the order described.

Goldman (2012), however, argues that this convergence is itself evidence against Theory Theory. It is rare for multiple scientists to converge on the same theory, especially when that theory is only partially developed. Rather, arguments over how to interpret findings and which theories are superior to others are an important factor for driving science. It seems unlikely, then, that nearly all children within

[3] Note that, in this case, the children do not know whether either person's belief is correct (cf. knowledge states)

and across generations would naturally converge to the same theory of ToM. Instead, Goldman argues for Simulation Theory—an account of ToM which is discussed next.

2.1.2 Simulation Theory Most Simulation Theory accounts define ToM reasoning as direct simulation of events. That is, ToM reasoning occurs by mentally simulating events as if the reasoner were the person being reasoned about (Goldman, 2006). Colloquially, this is reasoning by putting yourself into somebody else's shoes.

Simulation Theory is consistent with a number of other empirical findings. In particular, cognitive neuroscientists (e.g., Rizzolatti, Fogassi & Gallese, 2001) have found that a group of neurons, called mirror neurons, show similar patterns of activation when people watch others perform an action as when they perform that action themselves. Such patterns of activation suggest that people mentally simulate doing what a compatriot is doing, perhaps to better understand their mental states (Gallese & Goldman, 1998). Yet, few studies test the role of mirror neurons in ToM reasoning directly. Rather, the majority of evidence of mirror neuron activation is with respect to motor tasks (e.g., grasping an object), perhaps because most mirror neurons are located in the premotor and motor cortices (e.g., di Pellegrino et al., 1992; Cisek & Kalaska, 2004; Vigneswaran, Philipp, Lemon & Kraskov, 2013; cf. Bonini, 2017).

Simulation Theory is also consistent with accounts of egocentric bias in ToM and perspective taking. Keysar and colleagues (e.g., Keysar, Lina & Barr, 2003; Epley, Morewedge & Keysar, 2004; Epley, Keysar, van Boven & Gilovich, 2004) have found that people tend to make initial mental state judgements that are more accurate relative to their own mental states than others', and correct as needed. If the basis of ToM reasoning is projections of one's own mental states, it follows that mistakes in ToM reasoning tend to match one's own desires, knowledge, and beliefs (Goldman & Sebanz, 2005). Thus, the egocentric bias follows from Simulation Theory.

On the other hand, Simulation Theory is not consistent with developmental accounts of ToM (Perner & Howes, 1992). While some changes in ToM can be attributed to improvement in one's simulating abilities (Flavell, 2004), Simulation Theory does not account for the developmental trajectory described by Wellman and Liu (2004). Furthermore, Saxe (2005) argues that Simulation Theory cannot account for other types of systematic errors made in ToM reasoning, both by children and adults.

Saxe points to evidence like Kruger and Gilovich's (1999) finding that adults tend to overestimate others' self-serving bias. Kruger and Gilovich asked participants to rate how frequently they were responsible for various events in their marriages, and to predict their spouses' rankings of the same events. Overall, participants took equivalent responsibility for positive and negative events. However, they tended to predict that their spouse would take more responsibility for positive events, and less responsibility for negative events. Saxe argues that such findings are consistent with having strong beliefs about how minds work (i.e., that people tend to be self-serving). On the other hand, she argues, they are not consistent with simulation.

Responses to Saxe (e.g., Goldman & Sebanz, 2005; Mitchell, 2005) argue that her description of Simulation Theory is too narrow. Rather, simulation should be viewed as one component of ToM, perhaps in conjunction with theory-based accounts. Such hybrid theories of ToM reasoning are discussed next.

2.1.3 Hybrid theories. Hybrid theories of ToM combine aspects of Theory Theory and Simulation Theory in order to account for findings consistent with one, the other, or neither. Bach (2011) separates hybrid theories into two types, *divided hybrid theories* and *dynamic hybrid theories*. This distinction is followed here. Divided hybrid theories divide ToM into a set of processes and categorize these processes into those handled by Theory Theory style reasoning (i.e., with reference to a

developing theory of mental states) those handled by Simulation Theory style reasoning (i.e., by mentally simulating the person being reasoned about). Dynamic hybrid theories, on the other hand, describe a more fluid interplay between the two types of reasoning. Thus, dynamic hybrid theories unify Theory Theory and Simulation Theory, while divided hybrid theories acknowledge evidence for both, while ultimately keeping them separate.

Often, divided hybrid theories separate between theory and simulation at the task level. For example, Nichols and Stich (2003) argue that inference prediction occurs via simulation, while belief and desire attribution occur via theory. On the other hand, Heal (1996) and Perner (1994; 1996) make a distinction between content and non-content aspects of ToM reasoning, and proposes that reasoning surrounding the representational content of mental states (e.g., the state of the world) should be assigned to simulation, while non-content reasoning (e.g., perception) should be assigned to theory. Unlike Nichols and Stich's (2003) approach, the content-based hybrid theory does not require classifying every ToM task individually. However, it can be difficult to differentiate between content and non-content in a given context, and therefore predict whether simulation or theory-based based reasoning is required, a priori.

Further, most divided hybrid theories explicitly do not address developmental trajectories (e.g., Heal, 1996; see also Bach, 2011). Yet, they also do not claim that ToM is innate. Rather, development of ToM reasoning is discussed with respect to the development of other capabilities (e.g., developing concepts or necessary theoretical categories, like inference; Heal, 1995) or is not addressed at all.

Unlike divided hybrid theories, dynamic hybrid theories describe a unified ToM process, which is itself a combination of theory-style and simulation-style processing. That is, per Bach's (2011) definition, dynamic hybrid theories describe an interaction between Theory Theory and Simulation Theory that

tells a cohesive story of ToM reasoning and development. Such an approach is consistent with Ball et al.'s (2013) finding that some people report using theory-based reasoning in situations where others report using a strategy more consistent with simulation. These differences largely depend on life experiences, such as number of siblings (Ball et al., 2013). Similarly, Kuhberger and Luger-Bazinger (2016) found that people report using simulation more frequently when reasoning about the decision-making of unknown rather than known others[4]. To account for these findings (in addition to canonical findings on ToM use and development) a dynamic hybrid theory must describe how individual experience affects whether simulation or theory-style processing is used for a given task.

Bach (2011; 2014) proposes one such theory. He argues that structure-mapping (Gentner, 1983) provides the processes necessary to describe ToM development and reasoning, including transitions between theory and simulation. The Analogical Theory of Mind (AToM) model, which is proposed, implemented, and tested in this dissertation, is inspired by Bach's proposal.

Structure-mapping Theory[5] (SMT; Gentner, 1983) is a theory of analogy and similarity that emphasizes structural and relational similarity over similarity based on features alone. Gentner argues that this emphasis on structure is a key element of higher order cognition and is domain independent. Skorstad, Gentner, and Medin (1988) extend SMT to include analogical generalizations, which are created based on the alignment of two or more structurally similar cases. As a generalization grows to include more and more examples, it becomes more general, and eventually becomes rule-like (Gentner & Medina, 1998).

---

[4] Note that neither simulation nor theory-style reasoning was reported exclusively for either task.
[5] See section 3.3 for a detailed discussion of SMT.

Bach argues that SMT is the underlying process behind ToM reasoning. Generalizations and individual cases for comparison are stored in memory. Analogical comparison to a generalization is like Theory Theory processing. The generalization can be viewed as a rule, which is consistent with Theory Theory accounts. Development of a theory, then, is the formation of generalizations. This falls naturally out of structure-mapping. Each time a new example of a case where ToM reasoning is required is encountered, it is assimilated into the generalization it was compared against. This changes the contents of the generalization: when the alignment is high, it makes the generalization more abstract; when alignment is lower, it adds new information to the generalization. Through this process, Bach argues, ToM reasoning abilities develop in childhood and continue to change throughout adulthood.

If a fitting generalization is not present in memory, Bach argues that simulation occurs. First, a new case is created, which includes all the information in the original re-represented to the first person. Additional information is added to the case via simulation and autobiographical memory (e.g., when reasoning about another person's missed flight, the fact that I would be upset if I missed a flight is added[6]). This new case is then aligned with the original case via structure-mapping, and projections are made (e.g., the person is upset that they missed their flight). These projections are used for reasoning. The two cases are combined into a new generalization, which is available for future ToM reasoning.

Thus, the processes of structure-mapping unify theory and simulation-style reasoning under Bach's account. They describe development as the formulation of increasingly more abstract rules as a person gains experience, and simulation as re-representation and analogical projection. Furthermore, individual differences in reasoning style can be attributed to the development of generalizations with experience.

---

[6] Adapted from Bach (2011), Kahneman & Tversky (1982)

That is, people who have a strong generalization built up for a reasoning task should prefer theory-style reasoning for that task, whereas those who do not have a strong generalization should prefer simulation-style reasoning.

2.1.4 Innate accounts. While the majority of theories about ToM reasoning and/or development fall into the categories described above, other theories have argued for the innateness of ToM—in other words, the extent to which people are born with a capacity for ToM reasoning. For example, Baillargeon and colleagues (e.g., Onishi & Baillargeon, 2005; He, Bolz & Baillargeon, 2011; Scott, Richman & Baillargeon, 2015; see Scott & Baillargeon, 2017 for a review) have argued that ToM is an innate characteristic of human reasoning. That is, they argue that babies are born with full ToM reasoning capabilities, and any observed developmental changes can be attributed to development of other skills, such as language. This theory has been tested through adaptation of standard false belief tasks to avoid reliance on verbal skills. Researchers have demonstrated that children as young as 13 months succeed at such adapted tasks (e.g., Surian, Caldi & Sperber, 2007).

However, these findings—and their interpretation—have been disputed. Heyes (2014), for example, argues that infants' performance on adapted tasks can be attributed entirely to perceptual novelty[7]. Furthermore, other studies have failed to replicate young children's performance on these tasks (e.g., Yott & Poulin-Dubois, 2016; Dorrenberg, Rakoczy & Liszkowski, 2018; Kulke, Reiß, Krist & Rakoczy, 2018; Powell, Hobbs, Bardis, Carey & Saxe, 2018). While Baillargeon, Buttelman, and Southgate (2018) argue that other researchers' failure to replicate results can be attributed to procedural differences or differences in participant motivation and attention, Poulin-Dubois et al. (2018) point out that the fact

[7] See Scott & Baillargeon (2014) for a response to this critique.

that infant ToM studies have only been successful in a small number of labs suggests a potential replicability crisis. They argue that a significant amount of work is necessary to confirm the validity of such findings.

Another account arguing for the innateness of ToM reasoning suggests that ToM is made up of one or more innate, domain-specific cognitive modules (e.g., Leslie, 1987; Baron-Cohen, 1994; Scholl & Leslie, 1999). According to this hypothesis, these modules come online at various points during early childhood—a progression that explains developmental ToM findings (see Scholl & Leslie, 1999). Baron-Cohen (1994) suggests that a missing ToM module may be responsible for the social deficits observed in autism. While significant debate has surrounded this modularity hypothesis (e.g., Baldwin & Moses, 1994; Stone & Gerrans, 2006), this debate is ultimately orthogonal to the Theory Theory-Simulation Theory debate (cf. Scholl & Leslie, 1999). Although individual accounts of modularity may assign modules to theory and/or simulation, the existence (or lack thereof) of ToM modules does not preclude either type of reasoning.

2.2 Computational Cognitive Models of Theory of Mind Reasoning and Development

Computational cognitive models of ToM reasoning vary in their adherence to Theory Theory, Simulation Theory and/or hybrid theories. All, however, are tested primarily on their ability to model children's performance on false belief tasks (Figure 1) at various stages of ToM development.

2.2.1 Models in cognitive architectures. Several models of children's ToM have been implemented within cognitive architectures. Bello and Cassimatis (2006), for example, modeled ToM reasoning in Polyscheme (Cassimatis, 2005). Polyscheme models reasoning as the interaction of a sequence of modules, each of which is specialized to make inferences about a particular aspect of the world. The world is represented explicitly, and alternate worlds can be defined for reasoning. The

Polyscheme model of a four-year-old child's ToM involves a module for rule-based inference (the *rule specialist*) and a module for tracking when items in the world have been seen and when (the *temporal-perception specialist*). It also includes an explicit representation of the minds of others via the `MindOf` predicate, which links a person and a possible world. The possible world includes information about the real world from the person's point of view (i.e., the person's beliefs about the world). A three-year-old's undeveloped ToM is modeled by removing the `MindOf` predicate and associated facts.

Bello and Cassimatis (2006) tested their model on two tasks: a change of location task (Wimmer & Perner, 1983; see Figure 1) and an alternative location task (Wellman & Bartsch, 1988; see Figure 1). Just as children younger than four years old were able to perform the alternative location task, Bello and Cassimatis's model made the correct inference both with and without the `MindOf` predicate in this task. On the other hand, only the four-year-old version of their model (i.e., with the `MindOf` predicate) successfully performed the change of location task, matching children's performance. These findings suggest that a shift in how a child thinks about others' minds is sufficient to explain differences between three- and four-year-old children's ToM reasoning. However, Bello and Cassimatis do not propose a process by which this shift occurs.

Other models of ToM reasoning and development have focused on the shift in children's ToM reasoning abilities. Hiatt and Trafton (2010) model ToM development in the ACT-R cognitive architecture. They argue that changes in children's performance on false belief tasks can be attributed to a combination of learning and maturation. They model learning as a shift in choice of production rules over time. In particular, the model begins with a rule that retrieves the most highly activated belief chunk in response to false belief questions. This chunk typically corresponds to the model's own beliefs about the world. The model learns to prefer a competing production rule, which checks whether a

Appearance-Reality Task (AR): The child is presented with an object. The child is asked what she thinks the object is (e.g., a rock). The experimenter then shows the child that the object is, in fact, something else (e.g., a sponge). After determining the true identity of the object, the child is asked an appearance question (i.e., "What does this look like?") and a reality question (i.e., "What is this truly?"). In some cases, a question asking what a third person would think the object is, is asked.

Change of Location Task (COL): The child watches as a character places an object—often a toy—in a specific location (e.g., a toy box). The character then leaves, and another character moves the object elsewhere (e.g., from the toy box to the closet). The original character then returns, and the child is asked where the character will look for the object, and whether the character knows where the object actually is.

Alternative Location Task (AL): The child watches as a character places an object (e.g., a black pen) in a location (e.g., on a table). The character leaves, and the child is shown that a duplicate of the object (e.g., another black pen) is in a different location (e.g., in a drawer). The character returns, and the child is asked where the character will look for the object.

Unexpected Contents Task (UC): The child is presented with a familiar container whose contents are generally known (e.g., a crayon box). The child is asked what they think is inside the container. The contents of the container are then revealed to be different from the expected prototypical contents (e.g., rocks instead of crayons). After the contents are put away, children are asked 1) what they thought was in the box before being shown the contents and 2) what somebody who has not seen the box (e.g., Mom, a friend, a puppet) will think is inside.

Verbal False Belief Task (VFB): The child is shown a scene and told that the character in the scene holds a false belief (e.g., "Billy thinks that his sweater is in the closet, but really it's in his backpack."). The child is then asked questions about the character's future actions surrounding the false belief (e.g., "Where will Billy look for his sweater?").

Figure 1. Classic false belief tasks.

retrieved chunk is known by the person in the question. The model is rewarded when it retrieves the correct chunk and punished otherwise.

Hiatt and Trafton (2010) model maturation via an explicit maturation parameter. This parameter affects which production rules are available to the model. A lower maturation parameter value corresponds to a younger age. Effectively, it is a probability: as the maturation parameter increases, so does the likelihood that the model will fire the correct production rule. The combination of maturation

and learning allows for modeling children at various stages of development. Indeed, Hiatt and Trafton show a progression that correlates with children's performance as described in a meta-analysis of ToM development by Wellman et al. (2001). The model's performance was measured as it learned the change of location task (Figure 1). It was later extended to account for children's second-order ToM learning[8] (Hiatt & Trafton, 2015).

Arslan, Taatgen, and Verbrugge (2013) propose an alternative model of ToM development in ACT-R. In this model, ToM development is not a change in preference of production rules, but rather in activation of declarative knowledge. Chunks represent reasoning strategy (e.g., the chunk for reasoning without consideration of the other's mental state corresponds with answering with one's own knowledge). The model learns to activate appropriate chunks via feedback from the experimenter, similar to the process by which Hiatt and Trafton's (2010) model learns to switch between production rules. Arslan, Taatgen, and Verbrugge tested their model on a second-order change of location task (i.e., "where does the person who moved the item think the other person will look for the item?"). The model showed a similar pattern of learning to the pattern observed in children (Arslan, Taatgen & Verbrugge, 2017).

The two ACT-R models of ToM development propose competing accounts of children's transition between naïve and full ToM reasoning. However, both the production rules used by Hiatt and Trafton's model and the declarative chunks used by Arslan, Taatgen and Verbrugge's model were already known to the model. That is, the models learned to transition between strategies, but did not learn the

[8] Second-order ToM is the ability to reason about another person's beliefs about another person's internal states. For example, "I think she thinks that he knows…"

strategies themselves. The Analogical Theory of Mind (AToM) model described in this dissertation learns both.

2.2.2 Statistical model. Others have modeled ToM reasoning as a statistical process. The most popular of these is Bayesian Theory of Mind (BToM; Baker, Saxe & Tenenbaum, 2011), which implements ToM reasoning and development as a theory-based Bayesian framework (Tenenbaum, Griffiths & Kemp, 2006). In BToM, ToM reasoning is inference over a partially observable Markov decision process (POMDP) with a stochastic policy. Given an agent's behavior, BToM generates hypotheses about its beliefs and desires. The POMDP can be defined over any combination of action and state spaces, making BToM a domain-general model of ToM. Versions of BToM have been used to model children's ToM reasoning (Goodman et al., 2006), adults' plan and intent recognition (Baker & Tenenbaum, 2014; Shum, Kleiman-Weiner, Littman & Tenenbaum, 2019), and to predict a basic interaction type (e.g., chasing, fleeing, etc.) between simulated agents (Baker, Goodman & Tenenbaum, 2008). Unlike the other computational cognitive models presented here, BToM is a computational-level model (Marr, 1982). That is, while it models the behavior and learning trajectory of human ToM, BToM does not attempt to explain the process that underlies human ToM reasoning.

2.3 Theory of Mind Reasoning by Simulated Agents

Several groups have considered the role that ToM reasoning plays in interactions among simulated agents[9]. These can be separated into deep learning approaches, goal recognition approaches, and applied approaches (i.e., in robots reasoning about human teammates). Note that only the approaches in robots reference cognitive models of human ToM.

[9] For a detailed review of agents modeling other agents see Albrecht & Stone (2018).

2.3.1 Deep learning. A few groups have recently used deep learning approaches for simulated agents' ToM reasoning. For example, Rabinowitz et al. (2018) demonstrate a neural network, ToMnet, which learns to model other agents based on limited prior knowledge. Their model differentiates between a general ToM—overall weights in the network—and an agent-specific ToM—the learned embeddings for a specific agent.

ToMnet was tested in a fully observable grid world environment, with a map size of 11x11. It was tested on its ability to predict the behavior (i.e., policy) of three agent types: random agents, reward-seeking agents, and deep reinforcement-learning agents. ToMnet was trained individually on each agent type. In fact, in the random agent condition, a different ToMnet observer was trained for every species of agent (defined by the sparsity of the vector describing their policy). In the deep reinforcement-learning condition, ToMnet included an additional character neural network and mental state neural network to account for additional complexity in the task.

While ToMnet performed well on the tasks it was given, it is notable that the architecture needed to change as task complexity increased. This suggests that increasing task complexity further (e.g., by switching to a more complex environment or increasing the size of the grid world) may, once again, necessitate a new architecture. Thus, scaling ToMnet to more realistic tasks may prove challenging.

Similarly, Raileanu et al. (2018) trained a neural network to predict another player's goals and behaviors in a simple game. Unlike ToMnet, their network used its own policy as a baseline. The network was tested in three different game settings. Each setting had a different agent type—cooperative with symmetric roles, adversarial with symmetric roles, and cooperative with asymmetric roles. The network learned to recognize and predict behavior for each agent type. Note, however, that it was trained from scratch for each; that is, the agent types were not modeled simultaneously, and the

model did not learn to differentiate between them.

2.3.2 Goal recognition. Another approach to recognizing and predicting the behavior of other agents is the field of plan and goal recognition (Schmidt, Sridharan, & Goodson, 1978; Kautz, 1985; E-Martin, R-Moreno & Smith, 2015). Many goal recognition techniques[10] are based in logic and chaining through goal-action relations from known plans (Carberry, 2001; see also Sukthankar, Goldman, Geib, Pynadath, & Bui, 2014). Other approaches include mapping goal recognition to a planning task (e.g., Ramirez & Geffner, 2009; Geib & Goldman, 2011; Holler et al., 2018) or recognizing goals from prior observations via case-based reasoning (e.g., Kerkez & Cox, 2003; Vattam, Aha & Floyd, 2014). While some goal recognition techniques draw inspiration from human reasoning (e.g., Vered, Kaminka & Biham, 2016), goal recognition systems do not model human ToM reasoning or consider the internal states of the agents whose goals are being predicted. Thus, they typically take as inputs the agent's planner outputs. That is, goal recognition is based on the primitive actions sent to an agent by its planner. It is not clear how such systems perform when planner outputs are not available—such as in the case of adversarial agents (which are unlikely to make their plans/planners available) or humans.

On the other hand, Belief-Desire-Intention (BDI) frameworks (Bratman, 1987; Rao & Georgeff, 1991) provide a rich set of representations for reasoning about an agent's internal states. BDI has largely been applied as a self-model, rather than being used as a formalism for reasoning about other agents. While Rao and Murray (1994) modify the BDI framework to allow for recognition of another agent's beliefs, desires, and intentions, the recognition component of their framework is not substantially different

[10] The term *goal recognition* here refers to goal, plan, intent, and task recognition, since these tasks all involve recognizing the objective of an observed trace (i.e., the goal).

from the goal recognition techniques described above. Rather, their approach integrates goal recognition as a potential desire for BDI agents and allows for the agents' plans to have the inferred states of other agents as preconditions. Similarly, Jennings (1993) proposes the addition of joint intention for group problem solving in BDI agents, but suggests that this mental state be explicitly shared, rather than inferred. On the other hand, other work on collaborative planning and building common ground (e.g., Allen et al., 1995; Rich & Sidner, 1998; Grosz & Kraus, 1999) has involved inferring task-specific mental states but has not attempted ToM reasoning in a broader sense.

2.3.3 Human-robot interaction. One application of ToM reasoning is as part of the reasoning process of robots that interact with humans directly, as in assistive robotics or human-robot teams[11]. In both cases, the robot is tasked with helping a human achieve some known goal. The role of ToM, then, is to help the robot predict or explain the human's actions, especially when they deviate from expectation, and determine whether and how to help the human[12].

Devin and Alami (2016) use a ToM module in combination with a task planner and a motion planner to manage shared plan execution. Specifically, the ToM module analyzes symbolic representations of the world state and information about goal, plan, and action execution. It outputs an estimate of the person's knowledge about the world and beliefs about execution states. A Supervisor module combines this information with outputs from the planner modules to decide on future actions and dialog. Devin and Alami evaluate their architecture on two tasks, in which a simulated human and robot cooperate on a shared plan. They show that this architecture reduces the number of communication acts sent by the

[11] Note that perceptual ToM abilities in robots (e.g., Scassellati, 2002) are out of scope of this dissertation, and thus not discussed here.

[12] See Bianco & Ognibene, 2019 for a discussion of other possible functions of ToM in robots.

robot in both tasks. The authors argue that the robot is communicating less unnecessary information to the human, and thus is helping more efficiently.

Devin and Alami (2016) assume that the human and robot are completing a known shared plan. On the other hand, Görür, Rosman, Hoffman, and Albayrak (2017) propose an architecture that predicts the shared plan that is intended by the human. They combine a belief distribution over the human's possible action states (ready, not ready, in progress, help needed, aborted, done) and inferred emotional state (positive, negative, neutral; based on reactions to the robot's actions) as input to a POMDP to stochastically determine which plan the human is attempting, and whether they need help completing any actions in that plan.

Görür et al. (2017) leave evaluation of the efficacy of their architecture to future work. However, Brooks and Szafir (2019) show that another POMDP based architecture can infer people's mental models of a robot's actions in a simple grid world task. While Brooks and Szafir do not test whether a robot equipped with such second-order ToM is more helpful than other robot types, their findings suggest that POMDP-based architectures may be effective at modeling ToM for some tasks.

The experiments discussed above propose architectures for robots' ToM reasoning when helping humans, but their metrics focus on predictive accuracy or communication efficacy, rather than whether humans actually found them more helpful. Hiatt, Harrison, and Trafton (2011), on the other hand, tested whether people actually prefer interacting with robots that incorporate ToM reasoning when giving assistance. Specifically, the robots in Hiatt et al.'s (2011) experiments simulated the humans' reasoning using an ACT-R based cognitive model. When a human performed an unexpected action, the robot used the model to simulate the possible states that could lead to the observed behavior. If a simulation sufficiently explained the behavior, the robot communicated its understanding to the person; otherwise,

the robot asked the person for clarification. Participants judged these robots to be more intelligent and natural than robots that either always corrected humans' unexpected behavior, or never commented on it at all. That is, they preferred interacting with robots that incorporated ToM.

2.4 Conclusion

In this chapter, I have discussed psychological theories, computational cognitive models, and simulated agent applications of ToM reasoning and developed. The Analogical Theory of Mind (AToM) model attempts to bridge these categories: it describes and implements a process by which people learn and perform ToM reasoning and proposes that the same process improve simulated agents' ability to reason about others. In terms of psychological theories, AToM is a hybrid between Simulation Theory and Theory Theory. As a computational cognitive model, it provides not only a process by which children transition from naïve to adult ToM, but also a process by which children learn the rules that allow them to do so. Finally, for simulated agents, it provides a domain-general model for a variety of ToM tasks.

3 The Analogical Theory of Mind Model

3.1 Introduction

This chapter presents the Analogical Theory of Mind (AToM) model, a theoretical and computational model of theory of mind (ToM) reasoning and development. AToM is a process-level hybrid model of ToM, in that it provides and implements a process by which both Theory Theory and Simulation Theory-style ToM reasoning and development can occur. Unlike other computational cognitive models of ToM, AToM can be used directly by simulated agents that need to reason about others. This chapter describes the theoretical and computational underpinnings of AToM.

3.2 Model Description

The Analogical Theory of Mind (AToM) model is inspired by Bach's (2011, 2014) theoretical model of ToM reasoning and by Hoyos and colleagues' (Hoyos et al., 2015; Hoyos et al., under review) finding that structural similarity aids ToM development. Bach's proposed model is explained in section 2.1.3 of this work; Hoyos et al.'s experiments are detailed in section 4.2. This section focuses instead on the specific process-level claims made by AToM. Implementation details are provided in the following section.

AToM's main claim is that ToM reasoning occurs via analogical comparison. Specifically, when a person encounters a situation in which ToM reasoning is necessary, they retrieve a structurally similar memory. Because everyday interaction often involves reasoning about the same person (i.e., an interlocutor) repeatedly, immediate memories are preferred. This means that retrieval begins in working memory (WM); long term memory (LTM) retrieval is triggered by specific circumstances. For example, in the cognitive model in chapter 4 of this work, LTM retrieval is triggered by surprise. Note that in chapters 7 and 8, in which AToM is used to reason about simulated agents, LTM is used exclusively, because of the much longer timescale over which learning and reasoning occur.

After an appropriate memory is retrieved, any inferences made by the analogical comparison are analyzed with respect to the situation at hand[13]. Because ToM reasoning is often goal-driven, this analysis is relatively straight-forward. That is, an inference is accepted if it proposes a solution to the question driving the reasoning (e.g., "Where will she look for her sweater?") and no evidence to the contrary has been observed.

In interactive scenarios, feedback to one's reasoning is often immediately available. This may be explicit (e.g., a correction) or implicit (e.g., continued dialogue). This feedback is incorporated in the memory of the current encounter. If the feedback was positive, the memory constructed for the current situation may then also be integrated with the retrieved one. This allows continuous learning and improvement to one's ToM—and suggests that ToM reasoning and development occur via the same processes.

Note that, much like Bach's theory, AToM claims that retrieved memories can be either individual episodic memories or schemas. Retrieval of individual memories is similar to simulation. Unlike Bach's proposal, however, AToM does not re-represent cases to the first person or add autobiographical information when reasoning about individual memories. Instead, candidate inferences project what the person doing the reasoning did in a similar situation—a simulation in the form of "what *did* I do?" rather than "what *would* I do?" When a schema is retrieved, on the other hand, candidate inferences act as the consequents of Theory Theory-style rules. Thus, AToM is a hybrid, process-level account of ToM reasoning and development.

3.3 Implementation

[13] Note that inferences made by analogy are not guaranteed to be correct (cf. logical inference).

AToM is implemented within the Companion cognitive architecture (Forbus & Hinrichs, 2017), which posits that analogical reasoning is central to higher order cognition. The algorithms for analogical mapping, generalization, and retrieval, along with memory models, used within Companions are described here. Note that each is an independently validated computational cognitive model of its respective process[14]. Finally, the role of these algorithms in AToM is discussed.

3.3.1 Analogical mapping. The approach in AToM is based on the Structure-mapping Theory of analogy and similarity (SMT; Gentner, 1983). SMT views analogy and similarity as the process of aligning two structured, relational representations, and claims that much of human reasoning relies on this process. Representations can include object attributes as well as relationships between objects. Attributes can be perceptual (e.g., color), category information (e.g., Horse), or functional. Similarly, relationships can be perceptual (e.g., above), causal, functional, or evidential. Each set of representations being compared can be referred to as a case.

The alignment process constructs a set of correspondences between the entities and statements in the two cases being compared. Based on these correspondences, *candidate inferences* consisting of information that can be projected from one case to another are proposed. Analogy constructs candidate inferences, but their evaluation is left to processes outside the matching process itself. Alignments follow a set of constraints defined by SMT, which have received considerable psychological support (e.g., Gentner & Clement, 1988; Markman, 1997). These constraints are: (1) *one-to-one mapping*, which states that each entity and statement in one case can match to at most one entity or statement

[14] Forbus (2001) argues that the use of component models in larger scale simulations is an important step toward human-level AI, as they serve to verify assumptions and modeling constraints.

(respectively) in the other; (2) *parallel connectivity*, which states that two expressions can only be aligned if their arguments are aligned; (3) *tiered identicality*, which states a preference for matches between identical relations, or semantically similar relations when supported by higher-order structure; and (4) *systematicity*, which states a preference for alignments which include overlapping higher-order structure. The Structure Mapping Engine (SME; Falkenheiner, Forbus & Gentner, 1986; Forbus, Ferguson, Lovett, & Gentner, 2016) is a computational model of the structure-mapping process. SME takes two structured cases as inputs. These cases are called the base and target. It returns up to three *mappings* between the base and the target. Each mapping includes a set of correspondences, a set of candidate inferences, and a numerical *similarity score*. SME is a greedy algorithm. To compute mappings, it first finds all potential correspondences between identical relations in the base and target. Per parallel connectivity, correspondences between the arguments of these relations are also proposed. These *match hypotheses* are made without regard for structural consistency.

Next, SME gathers locally consistent match hypotheses into structures called *kernels*. First, match hypotheses that are inconsistent (i.e., violate parallel connectivity or one-to-one mapping) are marked. Kernels are then created from connected sets of consistent match hypotheses. A structural evaluation score is calculated for each kernel. Every match hypothesis is assigned an initial score. Per the systematicity principle, initial scores are propagated to the arguments of matching statements using a trickle-down approach. In this way, highly structured kernels receive higher structural evaluation scores.

Finally, mappings are constructed via greedy merge. That is, starting with the kernel with the highest structural evaluation score, structurally consistent kernels are added in order of their structural evaluation score. This set of kernels makes up the correspondences in a mapping. Structures in the base that do not have a corresponding structure in the target, and vice versa, are then projected as candidate

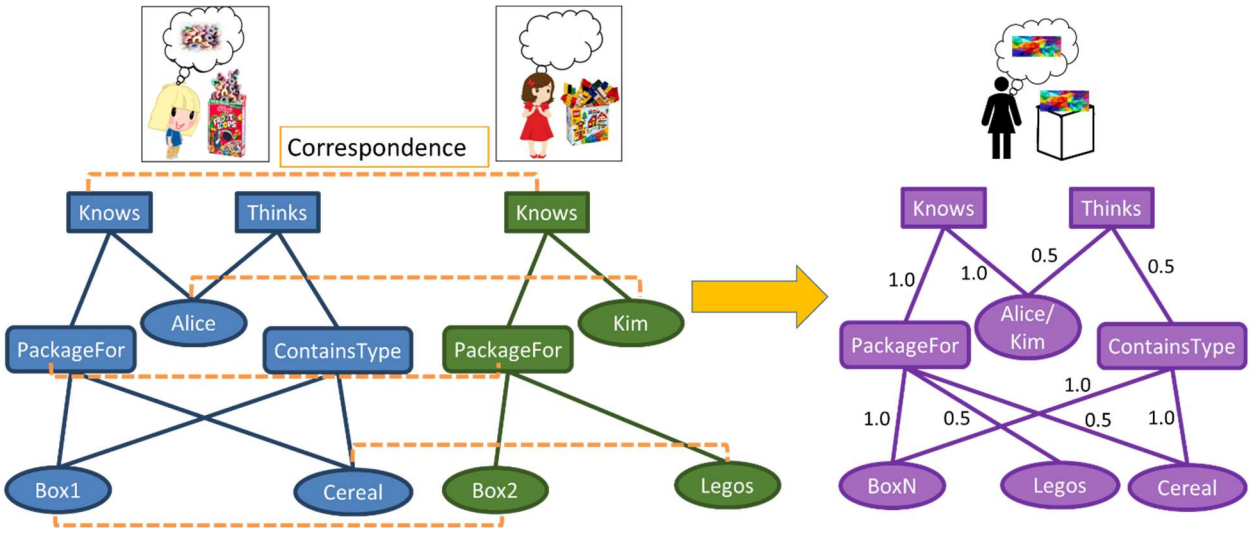

Figure 2. An example of structural alignment between two cases leading to a generalization.

inferences. If a candidate inferences includes an entity that does not have a correspondence, it is projected as an *analogy skolem*.

Typically, SME returns a single mapping. However, up to three mappings may be produced. The overall structural similarity score of any mapping that is returned by SME must be within 20% of the highest score produced. In this work, the mapping with the highest score is always used.

3.3.2 Analogical generalization. Learning in AToM takes place via analogical generalization and retrieval. Analogical *generalizations* are created via the Sequential Analogical Generalization Engine algorithm (SAGE; McLure et al., 2015). Generalizations are composed of two or more structurally aligned cases. A generalization includes a frequency distribution over the contents of its constituent cases. That is, each statement in a generalization is assigned a probability based on the proportion of times it has corresponded to a statement in a constituent case. Entities in such statements are converted into generalized entities.

Consider the cases in Figure 2. The case on the left (in blue) can be interpreted as, “Alice knows that box1 is a package for cereal. Alice thinks that box1 contains cereal”. The case in the middle (in green)

can be interpreted as, “Kim knows that box2 is a package for Legos.” Note that both contain a statement of the form `(knows person statement)`, but only the case on the left contains a statement of the form `(thinks person statement)`. Thus, the generalization (in purple), gives `(knows person statement)` a probability of 1.0 and `(thinks person statement)` a probability of 0.5. It also gives a probability of 1.0 to the `(packageFor box type)` statement, which appears in both constituent cases, but that type being `Cereal` a probability of 0.5. Similarly, `person` and `box` are each generalized entities in the generalization, as they have referred to different entities in the constituent cases.

A generalization is itself a case and can be used in analogical mappings and for further generalization. As additional constituent cases are added to a generalization, probabilities are updated. If the probability of a statement ever falls below a preset threshold[15], that statement is removed from the generalization. Thus, over time, generalizations become schemas that contain structures representative of the concept, while filtering out noise. Causal relationships in such schemas, when projected via candidate inference, can be treated as rules (Gentner & Medina, 1998).

3.3.3 Analogical retrieval in Working Memory. The AToM model uses two types of memory, working memory (WM) and long-term memory (LTM). WM stores a small number of cases for a limited amount of time. Because psychological accounts of human WM differ on both number and length of time (see Baddeley, 2007; Cowan, 2015), no theoretical claims about either are made in this work. In practice, a single psychological experimental session is assumed to be short enough for encountered cases to be accessible to working memory. Similarly, the number of encountered cases is assumed to be

[15] A default probability cutoff of 0.2 is used in this work.

small enough to all be accessible.

Computationally, the SAGE-WM model of WM (Kandaswamy, Forbus & Gentner, 2014) is used in this work. SAGE-WM stores structured cases, including analogical generalizations. A small number of cases can be stored at one time, usually for within-task comparison. Cases are retrieved from SAGE-WM based on structural similarity, biased by recency. Given a *probe* case for which a similar case is to be retrieved and a *similarity threshold*[16], an analogical comparison is made between the probe and each stored generalization in reverse chronological order (i.e., starting with the most recently seen case). If a mapping with a similarity score above the threshold is encountered, it is retrieved and used for reasoning. The probe is then assimilated into the generalization, and the new generalization is stored as the most recent.

If no generalization above threshold is encountered, the probe is compared to each individual case in WM, again in reverse chronological order. If an individual case above threshold is encountered, it is used for reasoning and a new generalization (formed from the retrieved case and the probe) is stored. If no case above threshold is encountered, reasoning fails and the probe is stored as an individual case. Note that, theoretically, adding a new case in this way can cause an older case to be forgotten (i.e., because SAGE-WM storage capacity is exceeded). However, this does not occur in the present work.

3.3.4 Analogical retrieval in Long Term Memory. Unlike WM, LTM can store many cases over long periods of time (i.e., a human lifetime). In this work, LTM storage is modeled via SAGE (McLure et al., 2015) with MAC/FAC[17] retrieval (Forbus, Gentner & Law, 1995).

---

[16] Unless otherwise specified, a default similarity threshold of 0.8 (average normalized score) is used.
[17] "Many are Called/Few are Chosen"

Cases are stored in *generalization pools*. Each generalization pool represents a concept and holds the generalizations and individual examples of that concept that have been encountered. Generalization pools are built up incrementally. A new case can enter a generalization pool by being assimilated into a generalization already in the generalization pool, by forming a new generalization with an individual example already in the generalization pool, or as a new individual example.

Analogical retrieval takes place via MAC/FAC (Forbus et al., 1995) over the union of available generalization pools, called a *case library*. MAC/FAC consists of two steps: the computationally cheap and coarse MAC, and the computationally more expensive, but more accurate FAC. During the MAC stage, a *content vector* is created for the probe and each case in the case library. Each element of a content vector represents the number of times an element (i.e., a predicate or collection) occurs in a given case. Thus, a content vector is a flat representation of a structured case, and the dot product of two content vectors provides a coarse estimate of their structural similarity.

MAC computes the dot product of the probe and each case in the case library and returns up to three cases with the highest scores[18]. These cases are then passed to FAC. FAC computes the analogical similarity between the probe and each case via SME. It returns the case with the highest structural similarity and up to two others if their scores are sufficiently high[19]. In this work, only the case with the highest similarity score is used.

The case returned by MAC/FAC can be used for further reasoning. The probe can also be merged with the case to form a generalization (or assimilated into it if the retrieved case is itself a

---

[18] The case with the highest score is always returned. Additional cases are returned if their score is within a preset threshold of the highest. The default threshold of 10% is used here.

[19] The default threshold of within 10% of the highest similarity score is used here.

generalization). This is how learning occurs in SAGE. SAGE can be used for supervised learning when cases are labeled with their expected generalization pools, or unsupervised learning, with cases entering the generalization pool of their retrieved match. In this work, the method is determined on a by-experiment basis and is specified in each experiment individually.

3.3.5 Analogy in AToM. AToM is implemented using the models of analogical processes described above. Scenarios are represented in predicate calculus cases using the NextKB ontology (Forbus & Hinrichs, 2017). Because AToM is domain-general, the process for generating cases is described individually for each experimental domain. Similarly, representational choices are described per experiment.

Given a probe, AToM first searches its WM (implemented as SAGE-WM) for a sufficiently similar case. If such a case is found, an analogical mapping is computed via SME and candidate inferences are analyzed. Candidate inferences that are applicable to the given task (e.g., ones that answer the appropriate question) are tentatively accepted and used in a response. Feedback is then incorporated into the initial case, and the case is added to WM, either via generalization with the retrieved case or individually.

When appropriate, retrieval from LTM is triggered. In the cognitive modeling experiments presented here, this is usually a search for explanation due to surprise. However, this is not a specific assumption of AToM and other drivers for LTM retrieval are possible. LTM retrieval happens via MAC/FAC over generalization pools stored via SAGE. To model retrieval of human memories, SAGE is populated with a

small number of synthetic cases that are representative of memories one may have[20]. If a case is retrieved from LTM, the current probe is also added to LTM—again, either by generalization with the retrieved case or as an individual example, depending on the feedback received.

As mentioned previously, AToM begins with a search in WM because of the assumption that people most often perform ToM reasoning about interlocuters throughout the course of an interaction. Recent retrievals allow for specific information about the interlocuter to be incorporated into one's reasoning. Such an assumption does not hold when reasoning about individuals with whom one is not actively interacting. For example, in the experiments in chapters 7 and 8, an agent with new behaviors, goals, and preferences is reasoned about each time. In such situations, SAGE-WM retrieval is bypassed in favor of LTM retrieval via SAGE, as the contents of working memory would not be relevant. See Figure 3 for complete AToM algorithm[21].

3.4 Related Work: Analogical Reasoning

This work uses the Structure Mapping Engine (SME; Falkenheiner et al., 1986; Forbus et al., 2016) as

```
1. Given probe case, WM contents, LTM contents
2. Try WM retrieval via SAGE-WM
   2.1. If retrieval is successful, request feedback
3. When LTM retrieval is triggered
   3.1. Try LTM retrieval via MAC/FAC
      3.1.1. If retrieval is successful, request feedback
4. Learn from case and feedback
   4.1. Modify case with feedback
   4.2. If retrieved from LTM, assimilate modified case in LTM via SAGE
   4.3. Else, assimilate modified case in WM via SAGE-WM
```

Figure 3. Pseudocode for AToM algorithm.

[20] We have estimated that to start to model the full memory of an adult human, one may need over two million generalization pools comprised of over 45 million examples (Forbus, Liang & Rabkina, 2017).
[21] Note that step 4 was not included during testing trials.

its computational model of structural analogy. In addition to SME, a variety of computational models of analogy have been proposed (see Gentner & Forbus, 2011 for a review). Some of these models, and their suitability as an underlying model for AToM, are discussed here[22]. For the purposes of organization, I follow Gentner and Forbus's (2011) classification of symbolic, connectionist, and hybrid models.

3.4.1 Symbolic models. SME, CARL (Burstein, 1983), Cascade (VanLehn & Jones, 1993) and the Incremental Analogy Machine (IAM; Keane & Brayshaw, 1988) are symbolic computational models that draw on the structural constraints proposed by SMT. Because SME has already been discussed in detail (see section 3.3), I focus on CARL, Cascade, and IAM here.

CARL (Burstein, 1983) learned to manipulate assignment statements in the BASIC programming language from analogies commonly used by teachers. Unlike SME, which uses a middle-out approach, CARL used a top-down algorithm to build mappings between descriptions. It relaxed SMT's tiered identicality constraint, allowing for mappings between semantically related predicates at different levels of abstraction regardless of support from higher order structure. Although CARL could, in principle, be used outside of the programming language domain, it appears to never have been used as a domain-general algorithm.

Cascade (VanLehn & Jones, 1993) is another domain-specific model of students' problem solving, this time in the physics domain. VanLehn and Jones define analogical reasoning as referring to a previously seen written example when solving a similar problem[23]. In addition to problem solving via analogy (i.e., by direct comparison to previous examples), Cascade can reason via rules learned using an

[22] For clarity, discussion of computational models of analogy that do not model structural analogy (e.g., Greiner, 1988; Ramscar & Pain, 1996; Gust, Kühnberger & Schmid, 2006) is omitted.
[23] VanLehn and Jones (1993) argue that similar results would be expected when comparing to a mental example.

impasse-repair-reflect cycle (VanLehn, 1999). Results from the Cascade model suggest that better-performing students tend to rely on rules when problem solving, whereas students who perform more poorly tend to rely on previously seen examples (VanLehn & Jones, 1993). Note that Cascade is a model of students' classroom learning and is not intended to be applied to more general reasoning.

IAM (Keane & Brayshaw, 1988), on the other hand, is a domain-general model of analogy that has been used to model a variety of analogical reasoning tasks. Unlike SME, however, it builds up structurally consistent mappings incrementally and includes constraints on working memory and effects of background knowledge, in addition to SMT's structural constraints. Due to its incrementality, IAM's mappings are subject to ordering effects. That is, the order in which a domain is processed matters (Keane, 1995). Keane argues that this is consistent with how people reason by analogy on some tasks. Forbus, Ferguson & Gentner (1994) extended SME to process analogies incrementally, demonstrating comparable performance to IAM on such tasks. However, for most tasks, the traditional SME approach is sufficient (and computationally more efficient).

3.4.2 Connectionist models. A number of connectionist models of analogy also incorporate structural constraints. Among these is the Analogical Constraint Mapping Engine (ACME; Holyoak &Thagard, 1989), which builds mappings using constraint satisfaction in an artificial neural network. In addition to structural constraints, ACME considers semantic and pragmatic constraints. Unlike in SME, in which constraints are guiding principles that cannot be violated in output mappings, constraints in ACME are inhibitive and excitatory connections between nodes (which represent possible correspondences). When the network is run, these connections guide node activation. The network converges on a final mapping. Note that one or more constraints may be violated in this mapping. Furthermore, ACME cannot generate candidate inferences.

Drama (Eliasmith & Thagard, 2001) extends ACME in two ways. First, it uses holographic reduced representations (Plate, 1994)—a fully distributed representation scheme—of the base and target. These are converted to a localist representation during network construction. Second, it incorporates structural and semantic constraints during neural network initialization, forming initial hypotheses based on both structure and semantics simultaneously. ACME, on the other hand, considers structural constraints prior to integrating other constraints. Eliasmith and Thagard (2001) argue that these changes lead to behavior that is more psychologically plausible than ACME's. However, like ACME, Drama cannot generate candidate inferences.

The Connectionist Analogy Builder (CAB; Larkey & Love, 2003) is a connectionist model of analogy that iteratively builds mappings between nodes of the base and target. Nodes with identical labels are initialized with a positive mapping weight, consistent with the tiered identicality principle of SMT. At each iteration, mapping weights are updated in accordance with structural constraints via model parameters. Particular emphasis is place on one-to-one mapping, but parallel connectivity and systematicity are also observed. When weights stabilize, a mapping is output. Like ACME, CAB's mappings do not include candidate inferences.

Another connectionist model of analogy, Learning and Inference with Schemas and Analogies (LISA; Hummel & Holyoak, 1996; Hummel & Holyoak, 1997) does produce candidate inferences. LISA also introduces an additional constraint: working memory capacity. Hummel and Holyoak (1997) argue that the exhaustive match hypothesis forests created during mapping by SME, ACME, and other models is not cognitively plausible given people's working memory capacity. Instead, LISA uses temporal bindings to build connections between nodes, which limits the number of relations LISA considers at one time. Forbus et al. (2016) argue that the working memory limitations set by LISA are too restrictive, given the

kinds of analogies people perform regularly. While Forbus et al., (2016) allow that Hummel, Licato, and Bringsjord's (2014) extension of LISA addresses this criticism by allowing LISA to consider systems of relations via *group units*, they also point out a number of unaddressed limitations of this model, including high tailorability.

Discovery of Relations by Analogy (DORA; Doumas, Hummel & Sandhofer, 2008) is an extension and adaptation of LISA that models how children learn relations. Doumas et al. argue that, after learning relations, DORA can perform the same simulations as LISA with less tailorability. However, DORA's working memory capacity is even lower than that of the original LISA implementation, leaving it open to Forbus et al.'s (2016) original criticism.

3.4.3 Hybrid Models Several models of analogy combine computational approaches. Many of these models, such as Copycat (Mitchell, 1993; Hofstadter & Mitchell, 1995) and Tabletop (French & Hofstadter, 1992; French, 1995), are domain specific. That is, they demonstrate the role of analogy in a particular domain using processes specific to that domain. Copycat, for example, forms analogies between strings of characters, while Tabletop does the same for table place settings. Both of these models interleave encoding and mapping processes and operate stochastically. This means that they may produce different mappings for the same base and target objects at different times. Furthermore, the performance of these models does not correspond to human performance on the same tasks, suggesting that their psychological plausibility is limited (e.g., Mitchell, 1993).

Associative Memory Based Reasoning (AMBR; Kokinov, 1994; Kokinov & Petrov, 2001) is a general-purpose hybrid model of analogy. In AMBR, the processes of analogy (i.e., mapping, inference, evaluation, etc.) occur in parallel. The processes communicate via a shared memory. The memory is composed of nodes that contain symbolic representations of semantic knowledge. These nodes are also

part of a connectionist network, whose activations represent associative relevance of the symbolic description of the node, for a particular analogy. The interaction between symbolic and connectionist processing is tightly coupled, such that only nodes with high connectionist activations are available for symbolic processing. The mapping process incorporates structural, semantic, and pragmatic constraints. Hypotheses about correspondences are created as temporary nodes. Each correspondence node is connected to other correspondence nodes and to the items (in memory) it puts in correspondence. Constraints are applied as excitations and inhibitions on those connections, as suggested by symbolic processes over activated memory nodes. Inferences in AMBR are produced similarly to candidate inferences in SME: predicates and entities that do not have correspondences in a mapping are projected from the target to the base, and vice versa.

Of the discussed models, AMBR appears to be the most likely to be compatible with AToM. However, AMBR has not been used as broadly as SME, so there is less support for it as a psychologically plausible model. Furthermore, AToM is a hybrid approach to ToM reasoning. It requires the ability to both reason from individual examples and generalized schemas. To the best of my knowledge, AMBR does not have the capability to generate schemas, via generalization or otherwise.

3.4.4 Case Based Reasoning approaches. Case Based Reasoning (CBR; Kolodner, 1992; 2014; Leake, 1996) is an approach to reasoning that reasons about a new problem by retrieving a similar previously observed problem (called a *case*) and using the retrieved case to solve the new problem. The overall process of CBR involves four steps, called the 4 Rs: *retrieve*, *reuse*, *revise* (sometimes called *adapt*), and *retain*. That is, when a new problem is encountered, a CBR system *retrieves* a similar case from its set of previous observations. It *reuses* information from the retrieved case to apply to the new problem. After the suggested solution is applied, it is *revised* to better fit the situation if needed. Finally,

the new problem and its solution are *retained* for future use.

This four-step process is analogous to learning and reasoning within AToM. However, contrary to AToM, most CBR approaches are domain- or task-specific (e.g., see Holt et al., 2005 for a discussion of CBR in medicine and Perner, Holt, and Richter, 2005 for a discussion of CBR in image processing) and are not intended to model human cognition. Furthermore, current state of the art approaches in CBR define cases in terms of feature vectors and similarity as operations over them (see Plaza & McGinty, 2005). Thus, while CBR has been used for similar AI tasks to those presented in chapters 7 and 8 of this dissertation (e.g., Cox & Kerkez, 2006; Fagan & Cunningham, 2003), no individual modern CBR system can perform the range of tasks that AToM performs.

3.5 Conclusion

This chapter has described the AToM model of ToM reasoning and development, including its theoretical claims and implementational details. In the following chapters, AToM is applied to children's ToM development (chapters 4 and 5) and simulated agents' reasoning about each other (chapters 7 and 8). Chapter 6 provides additional evidence that the processes used by AToM are central to a phenomenon, pretend play, which has been implicated as being important to ToM development. Each chapter describes its specific application of AToM and any additional assumptions needed.

4 Computational Model: Theory of Mind Learning from Structured Stories

4.1 Introduction

In this chapter[24], AToM is used to model a training study by Hoyos et al. (2015), which showed that children can improve ToM reasoning after hearing just three stories over the course of a single training session. AToM provides a process that can explain how children learn in this experiment, including why their performance differs in the two experimental conditions. Furthermore, AToM makes several predictions for future human subjects studies.

4.2 Hoyos, Horton & Gentner (2015)

Recall that children typically develop ToM reasoning skills during preschool (approximately 3-5 years; Wellman & Liu, 2004). However, children's specific ToM development trajectories vary. Hoyos et al. (2015) recruited 80 four- and five-year-old children who were not at ceiling in their ToM performance, as measured by three pre-tests. The tests, unexpected contents (UC), verbal false belief (VFB), and change of location (COL), are described in Figure 1.

The children were split into two groups: *high alignment* and *low alignment*. Both groups were presented with three stories in the style of an UC task, in a repetition-break (Loewenstein & Heath, 2009) pattern: the main character in the first two stories held a true belief (i.e., repetition), while the character in the last held a false belief (i.e., break). Following training, all children were tested on the same three tasks (UC, VFB, COL) as before.

The stories heard by the children in the two groups differed. The children in the high alignment condition (Figure 4) heard stories that were very similar, both in terms of story structure and linguistic

[24] This chapter is an adaptation of Rabkina, McFate, Forbus & Hoyos (2017).

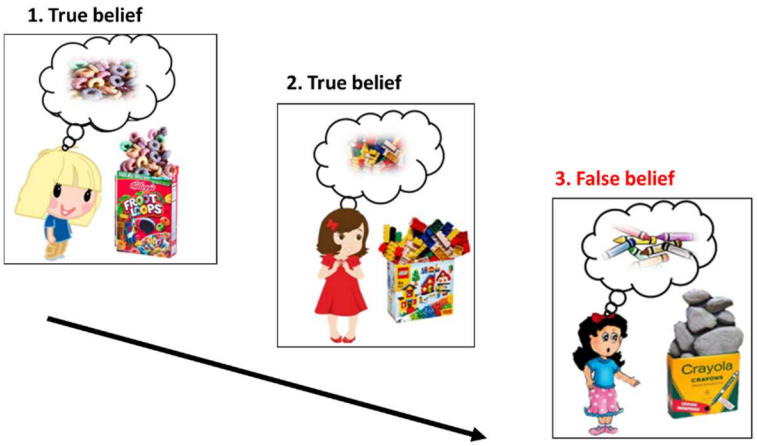

Figure 4. High alignment stories from Hoyos et al. (2015) experiment.

content. First, they heard a story about a little girl who was looking at a cereal box and correctly thought that there was cereal inside. Next, they heard a story about a little girl who was looking at a Legos box and correctly thought that there were Legos inside. Finally, they heard a story about a little girl who was looking at a crayon box and *incorrectly* thought there were crayons inside—instead, there were rocks. The accompanying illustrations (shown in Figure 4) were also structurally similar: there was always a little girl on the left side of the frame, looking at a box of small, colorful pieces to her right.

In the low alignment condition, on the other hand, the children heard stories that differed both linguistically[25] and structurally (Figure 5). The content of these stories, however, was similar to the stories heard by the children in the high alignment condition and still followed the repetition-break structure. In this case, the children first heard a story about a little girl who was looking at a shoebox and correctly thought there were shoes inside. Next, they heard a story about a little boy who was looking at a juice can, and correctly thought there was juice inside the can. Finally, they heard the same

[25] See Hoyos et al. (2015) for description of language used in both training conditions.

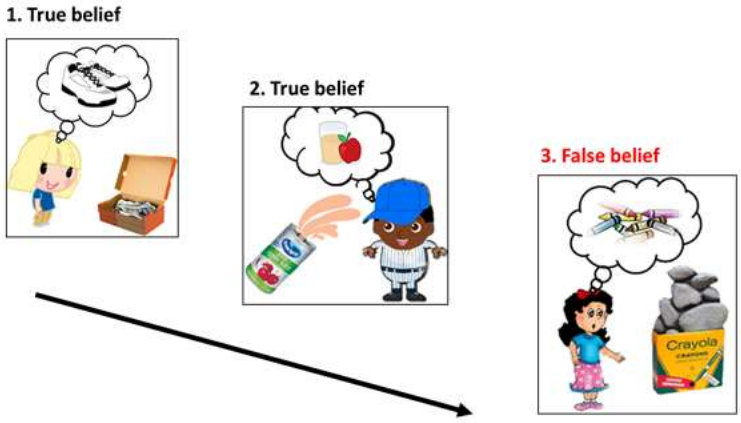

Figure 5. Low alignment stories from Hoyos et al. (2015) experiment.

false belief story as the children in the high alignment condition: a little girl was looking at a crayon box and incorrectly thought there were crayons inside, when there were really rocks. As shown in Figure 5, the pictures that accompanied the stories also differed structurally. The first showed a girl on the left looking at a plain box with white shoes to her right; the second showed a boy on the right looking at a can with liquid juice to his left; the last showed a girl on the left again, this time looking at a colorful box to her right. Due to these differences, Hoyos et al. (2015) hypothesized that the children in the low alignment condition would have more difficulty learning from the stories.

Hoyos et al. found that children in both conditions made significant improvements from pre- to post-test. Importantly, they found that the children in the high alignment condition made significantly higher gains than those in the low alignment condition. Hoyos et al. concluded that structural alignment aids false belief understanding. Furthermore, they, like Bach (2011, 2014) postulated that analogical comparison is "instrumental in children's understanding of mental states and their relation to the factual world." AToM provides a process by which structural alignment during learning can aid in false belief understanding.

```
(causes-Underspecified
 (not
   (and
     (inside-UnderspecifiedRegion see85 box1)
     (perceivedThings see85 (InsideOfSpaceRegionFn box1))
     (isa see85 VisualPerception)
     (doneBy see85 kim)))
 (opinions kim
  (and
    (containedObject contain84 cereal9)
    (containingObject contain84 box1)
    (isa cereal9 BreakfastCereal)
   (isa contain84 ContainingSomething))))
```

Figure 6. A partial representation of a true belief story. This statement represents the phrase "Kim thinks that the box contains cereal because Kim has never seen inside the box".

4.3 Simulation via AToM

A simplified English version of each training and testing example from Hoyos et al. (2015) was semi-automatically encoded using a natural language understanding system (EA NLU; Tomai & Forbus, 2009). Although syntax was simplified, overall structure and word choices were consistent with the original stories. Figure 6 shows a partial representation of a true belief story. Events are represented in the neo-Davidsonian style: a reified event with role relations connecting it to other constituents. The conjunction of statements about an event participates in causal relations. In English, Figure 6 states that because it is not the case that there is a seeing event in the box by Kim, Kim thinks that there is a containment event wherein the box contains cereal.

During training, the appropriate examples were passed into AToM in the order that the children in the corresponding condition saw them (true belief, true belief, false belief). The threshold for whether a probe was generalized was set to 0.01. If the incoming example matched to an example already in working memory with a score greater than 0.01, the model asked whether the match was correct. This corresponds to feedback in the Hoyos et al. (2015) experiment. When told it was correct, the model assimilated the examples into a generalization. Its behavior when told it was incorrect, on the other

hand, depended on its calculation of surprise. Surprise occurred when the model encountered an incorrect match whose similarity score was the same order of magnitude as the previous correct match. We propose that this comes out of the repetition break structure of the story order (Hoyos et al., 2015*;* Loewenstein & Heath, 2009): the high similarity to the interim generalization leads to a strong expectation of sameness, and the violation leads to a search for re-categorization. When surprised, the model probed LTM for an alternative case to align with[26].

4.3.1 Model walkthrough. AToM is trained following the algorithm described in Figure 3. In the high alignment condition, the first true belief story is stored in working memory. The second true belief story is then matched to the first, and an interim generalization is formed. When the false belief story comes in, it too matches to the generalization. Due to violated expectations, LTM is probed. LTM is a collection of generalized and specific cases that represent memories formed over time. If a case is retrieved, an interim generalization between the match and the false belief case is created and stored in working memory.

In the low alignment condition, on the other hand, no generalization is formed between the two true belief cases. This leads to them being stored as separate cases in working memory. When the false belief case comes in, it matches to the first true belief case, but no element of surprise is present when the model is corrected. For this reason, LTM is never probed, and working memory consists of only the three training examples.

Testing proceeded as follows: cases were again encoded semi-automatically using EA NLU. These

[26] This is consistent with findings that violation of expectations triggers a search for explanation and facilitates learning (e.g., Stahl & Feigenson, 2017).

cases were given to the model which retrieved the most similar case from working memory and generated candidate inferences by analogy. The candidate inferences corresponded to what the model predicts is missing from the test cases (e.g., what the agents will do). These candidate inferences were manually inspected to determine whether any could result in correctly answering the test questions.

4.3.2 Model results. In the high alignment condition, AToM generalized the true belief cases with a normalized match score of 0.075. It then matched the false belief to the generalization with a score of 0.066, which corresponds to the child incorrectly predicting that the character in the story knows what is in the box. The model was then informed that this match was incorrect. Because the similarity scores it had encountered were within the same order of magnitude, it searched LTM for another match. It then retrieved one of two memory cases that matched with a normalized score of 0.083 or 0.066 and created an interim generalization between it and the false belief case. We used stories intended to approximate a memory a child might have (e.g., thinking that a magician put a ball inside of a hat, only to find the hat empty) to model what might plausibly be retrieved.

Depending on the case retrieved, the model was then able to answer VFB or UL. Correctness was evaluated based on the candidate inferences generated from the best mapping between the test case and the contents of working memory. For example, to correctly answer “Where is Nora going to look for her ball?” (UL) the mapping must produce a candidate inference stating that there might be a looking event, in which Nora looks for her ball in the appropriate location.

In the low alignment condition, on the other hand, the second true belief case matched to the first with a very low similarity score of 0.0014, well below threshold. For this reason, the model did not form a generalization between them. When the false belief case was compared, it had a match score of 0.066 with the first true belief case. Similar to the high alignment condition, the model was informed that this

was not a correct match.

Because the previous match score was of a different order of magnitude, the model did not search LTM, and instead stored the false belief case alongside the two true belief cases. When the UC case came in, the false belief case was retrieved. The mapping generated a candidate inference that would allow the model to properly answer "What does she think is in the box?" This candidate inference stated that not having looked inside the cookie box would cause the agent to believe that it contained something analogous to crayons in the crayon box from the training example. That is, cookies.

Note that this retrieval is due to recency in working memory: the UC test case lacks the explanation present in the training cases about why a person holds a certain belief (e.g. "Kim thinks that the cereal box contains cereal because Kim has never looked inside the box."), so the first true belief case had the same match score. If that case had been retrieved, the model would not have been able to answer UC correctly.

4.4 Discussion and Predictions

AToM gives a process-level explanation for the results of the ToM training study presented in Hoyos et al. (2015). It also suggests that an important step in ToM development is generalizing belief-state cases in LTM. In training studies, understanding that the training cases can, and indeed should, be assimilated to LTM with belief-state interpretation cases is crucial. In other words, children may be accumulating experiences that require reasoning about belief states in LTM, but these memories remain inert until a surprising event—such the one experienced by the high alignment participants in the Hoyos et al. study—stimulates their retrieval and begins the process of creating schemas that can be used in future ToM reasoning. This predicts that children in the high alignment condition of Hoyos et al. (2015) are more likely to retain what they have learned than the children in the low alignment condition: the

children in the high alignment condition were more likely to access those experiences from LTM and form a generalization with them.

In addition, AToM predicts that reversing the order of training examples would cause children in both conditions to fail. In the low alignment condition, when the most recent training example is retrieved, children would match the UC task to a true belief scenario and answer incorrectly. Children in the high alignment condition would similarly fall back on retrieval of the most recent case, as they would not experience the surprise caused by the repetition break structure. Thus they, like the children in the low alignment condition, would match UC to the true belief scenario and answer incorrectly. Similarly, because they will not have accessed LTM, they would not be able to answer either of the other tests. Follow-up experiments confirmed the predictions made by AToM (C. Hoyos, personal communication, 2017).

Previous studies (e.g. Hale & Tager-Flusberg, 2003; Lohmann & Tomasello, 2003) have suggested that experience plays a role in ToM development. AToM provides a concrete explanation for how these experiences might lead to ToM and provides further suggestions for human subject experiments.

4.5 Conclusion

In this chapter, AToM was used to model a training study in which children improved their ToM capabilities after hearing only three stories. All stories were in style of the UC task. In one condition, however, the content and linguistic structure of the stories was formulated to be highly structurally alignable. In the other, the stories shared less overlapping structure. While children in the second group improved their ToM reasoning significantly, only children in the first group were able to transfer to other ToM tasks (i.e., VFB and/or UL).

AToM provides a process by which children in both conditions improved their ToM reasoning. It also

explains why children in the high alignment condition improved more; namely, because they accessed LTM during training. This difference leads to one of the predictions about the children's learning made by AToM: that only the children in the high alignment condition should retain their ToM improvement in the long term. AToM's other prediction comes from the proposed process itself. If Hoyos et al. (2015) had changed the order of stories told from true belief, true belief, false belief to false belief, true belief, true belief; children in neither condition would have shown improvement on the ToM tests.

This experiment provides evidence that AToM can explain children's ToM improvement during a training study. However, the original study was designed to illustrate the utility of structural alignment in learning ToM. Because structural alignment is central to AToM, it is important to demonstrate that AToM also models ToM learning when structural alignment is not a part of the initial experimental design. This is demonstrated in the following chapter.

5 Computational Model and Corpus Analysis: Theory of Mind Learning from Grammar

5.1 Introduction

There is considerable evidence that language acquisition affects Theory of Mind (ToM) development (Milligan, Astington, & Dack, 2007). However, debate has centered on the extent of the effects: some researchers report that the ability to understand more complex language simply gives children an ability to demonstrate pre-existing ToM reasoning skills (e.g. He, Bolz, & Baillargeon, 2011). Others suggest that, as children use language more frequently in conversation, they gain a vocabulary with which to mentalize about others' belief and desire states (Harris, 1996). Yet others find that learning certain grammatical structures is a necessary prerequisite for gaining ToM reasoning abilities, and that children bootstrap ToM from these grammatical structures (de Villiers & Pyers, 1997; de Villiers & Pyers, 2002; Hale & Tager-Flusberg, 2003; Lohmann & Tomasello, 2003; see Hofmann et al., 2016 for a review). Specifically, the sentential complement and the relative clause constructions have been proposed as playing an important role in ToM development.

This chapter[27] adds two pieces of support for the hypothesis that learning the sentential complement construction, but not the relative clause, leads to improved ToM capabilities. First, a corpus analysis is conducted. Its results suggest that children produce the sentential clause prior to gaining ToM competence. Next, AToM is used to model a training study in which children bootstrap ToM reasoning from learning the sentential complement grammatical form, but not the relative clause (Hale & Tager-Flusberg, 2003). The model supports the argument that the structure abstracted from the sentential complement leads to ToM improvement.

[27] This chapter is an adaptation of Rabkina, McFate & Forbus (2018) and Rabkina, Nakos & Forbus (2019b)

5.1.1 The sentential complement. The sentential complement is a complex grammatical structure, which contains an embedded clause in the verb phrase (e.g., "He saw that *the bird was yellow*."). De Villiers and colleagues have argued that learning the sentential complement bootstraps ToM development due to the nesting inherent in embedding a verb phrase (e.g., de Villiers & Pyers, 1997; de Villiers & Pyers, 2002; de Villiers & de Villiers, 2009). Contrast this with another complex grammatical structure, the relative clause. In the relative clause, the embedded clause is embedded in the noun phrase (e.g., "She wanted the shoe that *had blue laces*."). Because the relative clause has no inherent nesting, learning it should not lead to ToM improvements[28].

5.2 Corpus Analysis

If learning the sentential complement grammatical structure bootstraps the development of ToM reasoning skills, then this pattern should hold outside of the laboratory. That is, children's use of the sentential complement in everyday speech should anticipate the developmental trajectory of ToM. Because significant improvements in children's ToM occur between approximately 3 and 5 years of age (Wellman & Liu, 2004), we expect sentential complement use to reach a critical threshold immediately preceding this age range.

To test whether this relationship holds, we performed a corpus analysis of children's use of the sentential complement between 12 and 90 months of age. We also analyzed sentential complement use in child-directed speech (produced by mothers) during the same timeframe.

5.2.1 Approach. All data were extracted from the CHILDES project (MacWhinney, 2000),

[28] Note that Smith, Apperly & White (2003) do find an effect of learning the relative clause on improvement on ToM tasks. However, this finding is not consistent with other studies (e.g., Hale & Tager-Flusberg, 2003).

Table 1. Statistics for COMP and XCOMP tags (Sagae et al., 2007)

	Precision	*Recall*	*F-score*
COMP	0.83	0.86	0.84
XCOMP	0.86	0.87	0.87

which contains over 130 corpora of child-directed and child-produced speech. A corpus was included in our analysis if it contained speech by a typically developing North American English-speaking child between the ages of 12 months and 90 months. For consistency, only corpora with an available transcript and dependency parse data (Sagae et al., 2007) were included in the analysis. This resulted in a total of 32 corpora, leading to 3982 individual data points[29].

Each corpus included one or more conversations between a child and one or more adults. All conversation transcripts provided the child's age in months and relationship to the adult interlocutor(s) (i.e., mother and/or experimenter).

We extracted sentential complements from the children's speech using the "COMP" (finite verb complement) and "XCOMP" (other verb complement) dependency parse tags. Sagae et al. (2007) report overall parse accuracy for children's utterances between 72.7% and 92.3% on varying corpora within CHILDES. Table 1 shows reported precision, recall, and F-score for the "COMP" and "XCOMP" tags in the Eve corpus (Brown, 1973), one of the corpora included in the CHILDES project. Overall parse accuracy for the Eve corpus is 92.0%. Note that these analyses include both child and adult utterances.

Because a causal relationship between learning the sentential complement and developing ToM

[29] For longitudinal studies, a new data point was included for each recorded age in months.

reasoning abilities has been proposed (e.g., de Villier & Pyers, 1997), we expected children's use of the sentential complement to lead their ToM development. To examine this effect, we computed the average number of sentential complements produced per sentence at each age in months. If learning the sentential complement bootstraps ToM reasoning, then children should show an increase in sentential complement use leading into the ToM development period. Moreover, the increase should be specific to this timeframe; that is, children should achieve sentential complement proficiency prior to finishing ToM development.

5.2.2 Corpus analysis results. Our results indicate a concentrated growth period for children's sentential complement use that begins to plateau at the beginning of the ToM development period, suggesting a causal relationship between the two. Furthermore, this period of increasing sentential complement use coincides with a similar period found in parents' child-directed speech, which suggests a critical role for parents in children's acquisition of this grammatical structure.

Figure 7 shows the total number of sentences in our corpus of child-produced speech at each age in months along with the corresponding counts of sentential complement use. The corpus contains the most data in the range from 25 to 60 months. Note that this is an artifact of the data available and does

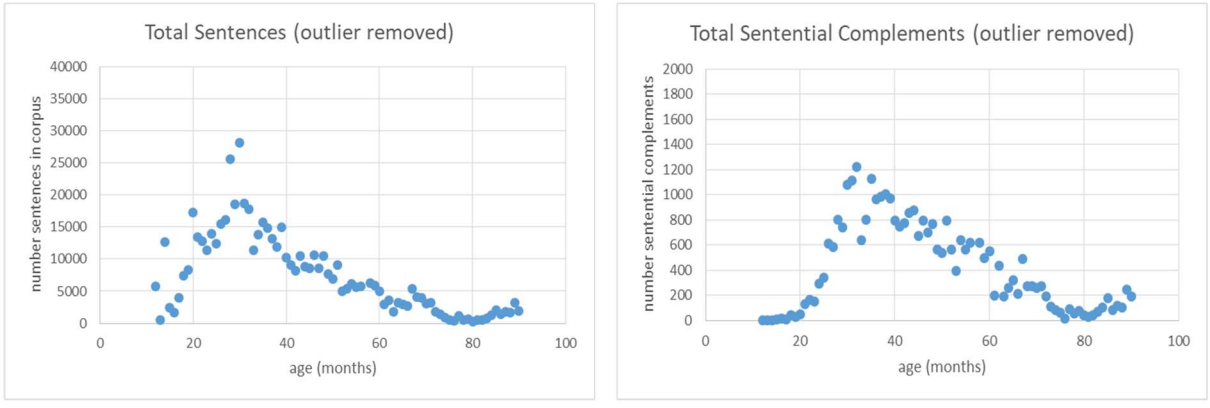

Figure 7. Counts for total sentences (left) and total sentential complements (right) in our corpus at each age in months. Note that one outlier (57 months) was removed from each graph.

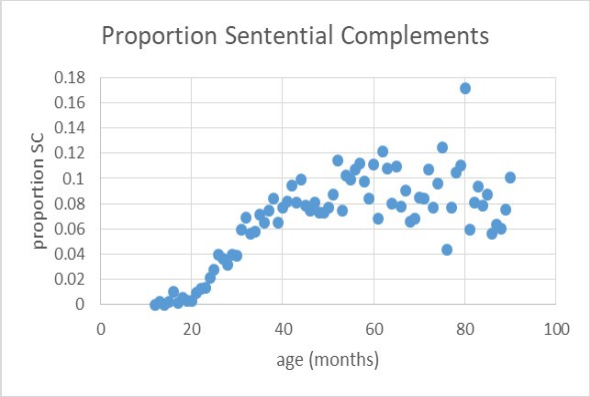

Figure 8. Average number of sentential complements per sentences produced by children at each age in months. No outliers were excluded.

not necessarily represent an increase in overall speech production during this age range.

Figure 8 shows children's sentential complement production as a proportion of overall sentences produced at a given age. The graph shows a linear increase from approximately 20 months to approximately 40 months of age, with a plateau beginning shortly thereafter. Once this baseline level of sentential complement production is reached, variance visibly increases. However, this variance is likely a byproduct of noise due to lower total sentence counts at later ages (Figure 7).

To determine the period of most concentrated sentential complement development, we isolated the interval with the strongest linear correlation between age and proportion of sentential complements (Figure 9, left). We fixed the starting point at 22 months, the first instance of appreciable sentential complement use (>1%). An endpoint of 38 months produced the strongest correlation, $r^2=0.9217$, $p<.001$. Beginning at 39 months, the distribution plateaus with a slope of approximately 0 (Figure 9, right).

Child-directed adult-produced speech follows a similar pattern (Figure 10). Following a period of linear

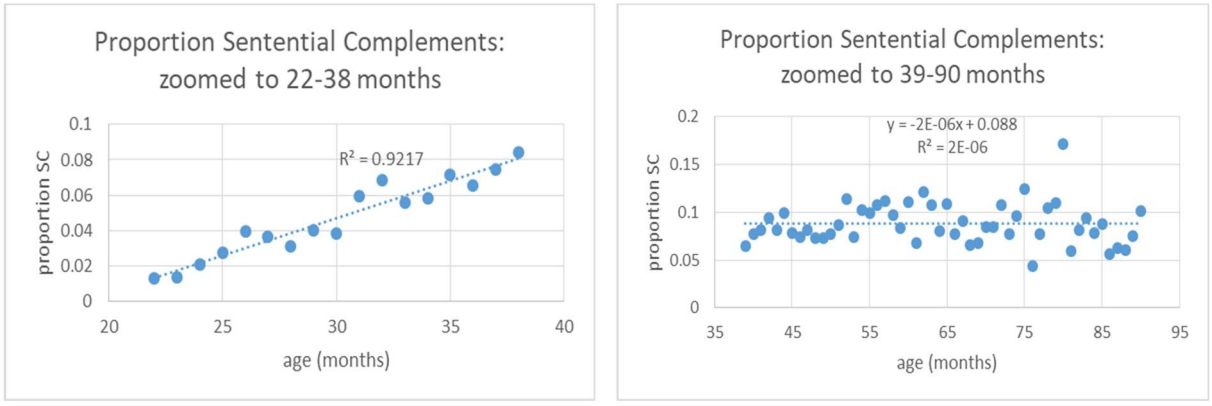

Figure 9. Proportion of sentential complement use by children at each age, zoomed to period of growth (left) and stabilization (right).

increase from child's age 12 months to 38 months (r^2=0.8603, p<.001), sentential complement use peaks and begins to gradually decline. Notably, the absolute proportion of sentential complements per sentence produced by adults is higher than the proportion produced by children at almost all ages.

As a potential contrast to the sentential complement, we also examined the use of another complex grammatical structure that has been argued to influence ToM acquisition, the relative clause (e.g., Smith, Apperly & White, 2003). However, we found negligible use of the relative clause in both child-produced and child-directed speech. This is consistent with a prior analysis of longitudinal data (Diessel

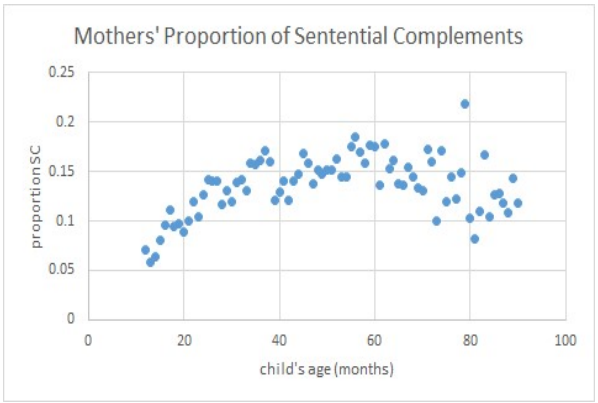

Figure 10. Average number of sentential complements per sentence produced by mothers at child's age in months. No outliers were excluded.

& Tomasello, 2000) which found that children use the relative clause in less than 0.5% of utterances. Absent a direct increase in the use of another such structure in child-produced speech during this period, the sentential complement stands out as the best candidate for a syntactic aid to ToM development.

5.2.3 Corpus analysis discussion. As predicted, children reach a critical threshold of sentential complement use prior to entering the major period of ToM development, typically regarded as 3 to 5 years of age. By 36 months children use sentential complements in an average of 6.5% of sentences (Figure 7). Their sentential complement use begins to plateau shortly thereafter, at 38 months and 8.4%.

It is important to note that both the ToM development period and the beginning of the observed plateau in sentential complement use are not hard boundaries. In fact, sentential complement use continues to increase after the onset of the plateau (between 39 and 58 months; $r^2=0.3581$, $p=.005$; Figure 9, right), albeit at a much reduced rate. However, weak correlation and high variance make it difficult to draw firm conclusions about trends within the plateau.

What is clear is that the most concentrated growth occurs before children make significant strides in their ToM development. Previous work has shown that training children to understand the sentential complement leads to improved ToM reasoning skills in a laboratory setting (Lohmann & Tomasello, 2003; Hale & Tager-Flusberg, 2003; Mo et al., 2014). Our results suggest that the same effect occurs outside of the laboratory. Taken together, these findings support the hypothesis that mastery of basic sentential complement use sparks ToM development.

Another finding of note is that child-directed sentential complement use shows a similar pattern of increase to child-produced sentential complement use. Specifically, adult sentential complement use increases from 7.0% at child's 12 months to 16.0% at child's 38 months. This period subsumes the

interval of greatest sentential complement development in children and gives way to a period of decline as children's use plateaus. Parents seem to adjust their sentential complement use according to the child's level of proficiency. Moreover, parents' sentential complement use seems to promote sentential complement production in children, as parents consistently overproduce compared to children at a given age.

Several explanations could account for the observed behavior. First, it is possible that parents mirror their children's speech patterns: as the child increases her sentential complement use, so does the parent. Under this hypothesis, other grammatical constructions should follow a similar trajectory. Alternatively, the causality could flow in the opposite direction, with children mirroring their parents. This explanation follows more directly from the present data, since the parents' sentential complement use precedes the children's, but it does not explain why the parents' use increases. Yet another explanation could be a mutual influence effect between children and their parents. As children begin to use the sentential complement, the parents increase their usage of the grammatical form, pacing their children's learning. Identifying the exact relationship at play will require data that can clarify the interaction between children's language use and their parents'.

Overall, our findings paint a picture of parental influence on children's sentential complement development, leading to children's acquisition of ToM. While the corpus analysis is not sufficient proof of a relationship between sentential complement proficiency and ToM development, it is consistent with prior laboratory evidence of a causal link between the two. This is a step toward showing that such a link exists in the wild.

5.2.4 Corpus analysis limitations. This analysis considered the relationship between children's sentential complement use and their ToM development. However, evidence exists that a more granular

view of the sentential complement might be appropriate. For example, Mo et al. (2014) found that, on ToM post-tests, children trained with sentential complements involving communication verbs outperformed children who were trained with mental state verbs. They note that this may be an artifact of the language used in the study, Mandarin, rather than a more general effect. On the other hand, Hale and Tager-Flusberg (2003) included only communication verbs in their training study of English-speaking children because of the potential confounding factor of the semantics carried by mental state verbs. A deeper analysis of the types of verbs used by children as they learn the sentential complement could shed some light on this question.

Because the effects of sentential complement training on ToM performance have been observed cross-linguistically, it is worth examining whether the patterns found in the present study are consistent across languages as well. Shatz et al. (2003) showed that 3- and 4-year-old speakers of languages with explicit false belief markings outperformed speakers of languages without such markings on some ToM tests. This suggests that other linguistic effects may be at play, and that the sentential complement may not be the sole way ToM is encoded in linguistic structure. For such languages, it is possible that the pattern of sentential complement use found in English may be less strong or entirely nonexistent.

Another question that merits further investigation is the nature of the plateau observed in Figure 8 and Figure 9 (right). A cursory analysis shows a period of continued increase from 39 months to 58 months before a period of mild decrease lasting through the end of the included data. The variance in the available data at this age range precludes a more concrete analysis, but the coincidence of the period of sustained increase in sentential complement use and the period of ToM development points to a tighter connection than can be shown at present.

Current data also do not fully illuminate the relationship between children's sentential complement

use and that of their parents. It is curious that the adult-produced speech so closely parallels the patterns observed in children's speech. However, identifying the exact mechanism by which this arises would require paired data to more closely track changes in sentential complement use.

Finally, the questions raised in this section tie into a broader debate about ToM acquisition as a whole. Although we provide evidence that is consistent with the hypothesis that sentential complement proficiency facilitates ToM development, strict causality has yet to be proven. The computational model described next proposes a process by which this causality may take place.

5.3 Modeling Study: Hale & Tager-Flusberg, 2003

We model a training study by Hale and Tager-Flusberg (2003), which showed that 4-year-old children who were given training on sentential complements (SC) also improved in their false belief reasoning. Children who were only given false belief (FB) training[30] did not improve in SC performance, and children who were trained on the relative clauses (RC) only improved their understanding of RC. Because children's ability to improve on false belief tasks from false belief stories was modeled in the previous chapter, here we focus on the SC and RC training conditions. The contrast between children's learning in these conditions suggests that linguistic bootstrapping for ToM reasoning is possible with some grammatical constructions (i.e., the sentential complement), but not others (i.e., the relative clause). Hale and Tager-Flusberg's (2003) training study is described next.

5.3.1 Sentential complements training.

During each of two training sessions, a child in the SC condition heard four stories about a boy's interaction with a *Sesame Street* character, which were also acted out by the experimenters using dolls. Each story contained a sentential complement structure

[30] No mental state language or sentential complement structure was used during FB training.

(e.g., “The boy said, ‘I kissed Grover.’”) which differed from the acted out reality (e.g., the boy kissing Big Bird). The child was then asked, “What did the boy say?” Regardless of whether the child answered correctly or not, the. experimenter emphasized the difference between the contents of the embedded clause and reality, (e.g. “That’s right/incorrect. The boy said, ‘I kissed Grover,’ but he really kissed Big Bird.”)

5.3.2 Relative clause training. Children in the RC condition were also trained using four stories during each of two training sessions. These stories all contained a relative clause structure (e.g., “Bert hugged the girl who jumped up and down.”). After hearing each story, the child was asked about the contents of the relative clause (e.g., “Who did Bert hug?”). The child was expected to use the relative clause structure in her answer, and the structure was emphasized in the experimenter’s response (e.g., “That’s right/incorrect. Bert hugged the girl who jumped up and down.”).

5.3.3 False belief tests. Three to five days after the conclusion of the second training session, each child was tested on SC, RC, and false belief (FB). Because the focus of this chapter is on linguistic bootstrapping of ToM, which is tested via FB tests, we focus on those here. Children were administered three false belief tests. These tests are described in Figure 1.

Scores on the FB post-test were calculated out of 6 points (2 per test; 1 per question). On average, children in the SC condition answered approximately 4.5 questions correctly[31], while children in the RC condition averaged approximately 1 correct answer. This corresponds to children in the SC condition improving on FB tests significantly more than children in the RC condition[32].

---

[31] This was not significantly different from the children in the FB training condition.

[32] Summary statistics show that children in the RC condition did not improve on FB from pretest to posttest. See Hale & Tager-Flusberg (2003).

5.4 Simulation via AToM

To model the Hale and Tager-Flusberg's (2006) findings, we combine a model of language acquisition (McFate, 2018) based on Construction Grammar (Goldberg, 2003) with AToM. In this section, we first discuss Construction Grammar and the language acquisition model. Then, we present the representations used to bootstrap from language to ToM. Finally, we walk through the combined model, and present results based on Hale and Tager-Flusberg.

5.4.1 Construction Grammar. The linguistic approach in this model is inspired by construction grammar (Goldberg, 2003). Construction grammar is an emerging paradigm in linguistics that proposes the fundamental unit of language to be pairings of form and meaning called *constructions*. Constructions are hierarchical and compositional, including morphemes, phrases and even fully grounded idioms. Under this approach, meaning arises not from a strict combination of words (lexical semantics) but rather from a unification of semantics provided by constructions at every level of interpretation.

It has been suggested that children acquire constructions by analogically aligning and generalizing over individual pairings of syntax and lexical semantics (Tomasello, 2009; McFate & Forbus, 2016). Here we specifically focus on argument structure constructions which define how phrases and clauses combine as arguments to form a sentence.

It has been argued that interpretation involves integrating the semantics associated with argument structure with the semantics of its arguments (e.g., verbal semantics; Goldberg, 1995). Following McFate and Forbus (2016), in the present work we model this integration as structural alignment (see McFate, 2018 for more detail). As a result, the nesting and implied semantics of a construction that combines clauses is applied to its arguments (i.e., the clauses themselves). Note that this approach is consistent with de Villiers and colleagues' description of the structure of the sentential clause (de Villiers & Pyers,

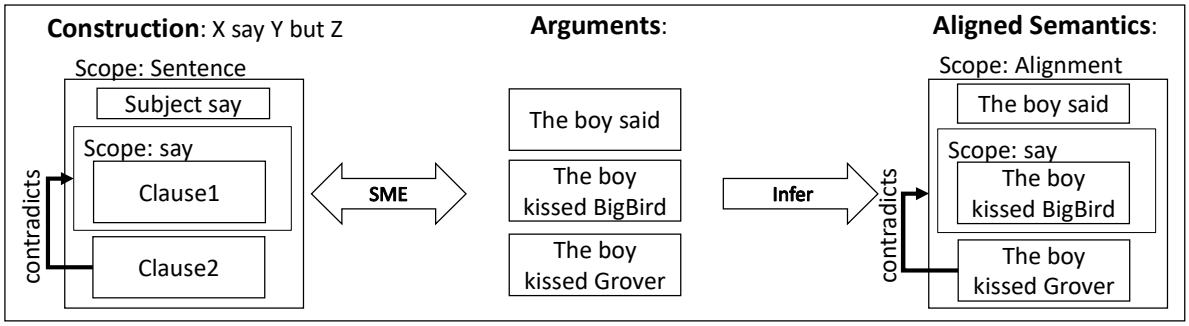

Figure 11. Example representations from a sentential complement training case, in which a boy said that he kissed Big Bird, but really he kissed Grover. The construction consists of the nesting of the first kiss verb phrase inside the say verb phrase, the un-nested second kiss verb phrase, and the contradiction between them. The arguments to the construction contain the semantics of the initial verb phrase and the two clauses. The aligned semantics (i.e. the candidate inference) state that the arguments to the un-nested verb phrase contradict the arguments to the inner, nested verb phrase.

1997; de Villiers & Pyers, 2002; de Villiers & de Villiers, 2009).

5.4.2 Representations. The first story in each training condition was semi-automatically encoded from the examples in Appendix B of Hale and Tager-Flusberg (2003). Because the text of the remaining stories was not available, we wrote new stories, including feedback, consistent with the examples provided in the original paper (see Appendix A). Testing cases were also semi-automatically encoded from the examples provided in Appendix C of Hale and Tager-Flusberg (2003). Semi-automatic encoding involved using the EA-NLU semantic parser (Tomai & Forbus, 2009) to generate initial lexical semantics which were then manually contextualized.

In the SC training condition, the key construction was of the form “X said Y, but really Z”. We represent this using a nested phrase structure representation: when an argument contains a finite clause, we maintain the verb’s scope over the clause (e.g. say “…”; Figure 11, left). Otherwise, the argument is collapsed into a phrase. The sentential complement construction takes as arguments a noun phrase subject (X), the verb, and two clauses (Y and Z). When unified by analogy, the first clause

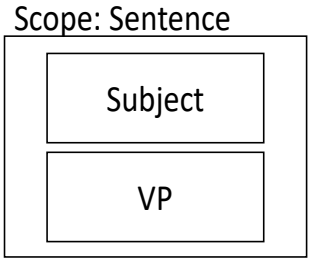

Figure 12: An example of the syntactic case of an RC training example. In this example, Bert hugs the girl who jumped.

becomes nested within the *say* verb phrase while the second remains at the same syntactic level. Both are within the scope of the completed clause. The construction combines the arguments, and, critically, it implies that the nested clause is contradicted by the external clause ("but really Z").

In the RC training condition, the feedback contained a relative clause "X verb the Y that Z." We use the same representation for this construction as for the SC condition (Figure 12). Because the relative clause modifies the noun, not the verb, there is no internal nesting structure. The construction takes a subject and a VP with a direct object, which in this case is a relative clause.

In each condition, we also represent the semantics of the arguments to the construction. The second element in Figure 11 shows the arguments to the construction in the SC training condition. Note that there are separate elements for each argument to the construction.

Following McFate & Forbus (2016), the construction and its arguments are unified by analogy. This results in the candidate inference shown on the right of Figure 11. What was *said* is now nested inside a separate context which is contained within the scope of the clause. Furthermore, the contents of the internal context are inferred to be contradictory from the external context. These inferences, called *aligned semantics*, are stored in AToM's working memory.

Because the language in the FB test cases did not involve sentential complements or relative clauses, we do not model interpretation of the grammatical forms used. Instead, we assume that an

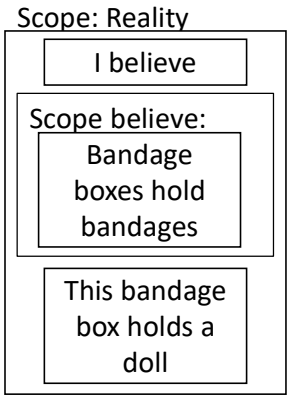

Figure 13. An example representation of a FB test case, Unexpected Contents.

appropriate representation can be extracted from the language and used EA-NLU to semi-automatically do so. We explicitly represent reality as a global scope. We also represent a belief held either by the child, or a character in the story, nested inside reality. While this presupposes that children understand that people have beliefs, it does not assume that they understand that these beliefs can differ between people or from reality. This is consistent with most verbal ToM tests, which often ask questions of the form, "What will X think?"

Figure 13 shows an example of an encoded test, Unexpected Contents. Here, the opinion that bandage boxes typically contain bandages is scoped inside reality. The belief is held by the child, and in reality, it is the case that the box contains a doll.

5.4.3 Model walkthrough. In each condition (SC and RC), AToM was trained on 8 stories, as in the original study. Training and testing cases were encoded as described above. For each incoming training example, AToM obtained the inferred semantics by analogy and passed them to working memory for retrieval and generalization using SAGE-WM. If a similar enough case was retrieved, the cases were generalized. Otherwise, the new case was added to the contents of WM ungeneralized. The generalization threshold was set to 0.01, consistent with Chapter 4.

During testing, each case entered AToM's working memory and a similar case was retrieved via SAGE-WM. When a case was retrieved, any candidate inferences that came out of the best mapping were examined. A test was considered correct if a candidate inference implied that the true belief condition contradicted the false belief condition. For example, in the Unexpected Contents test (Figure 13), the fact that there was a doll in the box should contradict the fact that bandage boxes usually contain bandages.

5.4.4 Model results. During each of the training trials, the inferred semantics from the construction alignment entered SAGE-WM. The first case entered ungeneralized and formed a generalization with subsequent examples. After SC training, because the training examples were all alignable, the working memory contained a single generalization. During testing, AToM had the generalization in working memory. AToM compared each test scenario to the contents of working memory. The nested structure within each false belief scenario aligned with the nested structure of the generalization and produced a single candidate inference. In each case, this candidate inference contained a contradiction between the true belief (e.g., there is a doll inside the bandage box) and the expected false belief (e.g., the box contains bandages). These candidate inferences predicted correct responses to all of the false belief questions.

During RC training, a similar pattern to that of SC training emerged: the inferred semantics from each RC case were accumulated into a single generalization within WM. However, during testing, AToM was unable to align the learned generalization with the false belief stimuli. Thus, it generated no correct inferences, and therefore no correct responses to the FB questions.

These results are consistent with the findings of Hale and Tager-Flusberg (2003): that sentential complement training bootstraps ToM, but relative clause training does not.

5.5 Discussion and Predictions

In this chapter, we have shown that the AToM model can explain bootstrapping from language in children's ToM development, when using representations that are inspired by construction grammar. We have modeled an empirical study by Hale and Tager-Flusberg (2003), which demonstrated that children's ToM reasoning abilities improve with sentential complement training.

One criticism of the original study is that the contents of the sentential complement are false (Lohmann & Tomasello, 2003). That is, the boy tells a lie. Our model's results suggest that this is important—the contradiction between the contents of the `say` and the `really` drives the subsequent inference that belief/observation and reality may differ. We view this as a feature, not a bug—after all, learning that beliefs may be inconsistent or incorrect is an important aspect of ToM development (de Villiers, Hobbs & Hollebrandse, 2014).

It is important to note, however, that the contradiction is not the only aspect of the SC training that leads to improved ToM reasoning in our model. The phrasal nesting structure of SC sentences allows for structural alignment between the learned construction and the test cases (e.g., I believe X, but really Y). It is this alignment that leads to a candidate inference about a potential contradiction. Without the sentential complement, this inference would not be made.

Yet, without the contradiction, it is not clear what would be learned from the alignment. Lohmann and Tomasello (2003) report that children can improve in ToM reasoning abilities by bootstrapping from sentential complements that do not contain such a contradiction. Their SC training, however, included mental state verbs. Others (e.g., Peskin & Astingon, 2004) have shown that children with more advanced mental state language tend to have more advanced ToM reasoning abilities. The question of how sentential complements might drive ToM development on their own deserves further research.

5.6 Conclusion

In this chapter, AToM was used to model children's ability to bootstrap ToM from linguistic constructions. It modeled Hale and Tager-Flusberg's (2003) experiment, in which children were trained on one of two grammatical structures, the sentential complement (SC) or the similarly-complex relative clause (RC). Children's performance on ToM tests was measured before and after training; Hale and Tager-Flusberg found that only children in the SC condition improved on these tests. In fact, children in this condition improved approximately as much as children explicitly trained on false belief tasks. Thus, children were able to bootstrap ToM from the SC, but not from the RC.

AToM, in conjunction with McFate's (2018) model of construction acquisition via structural alignment, proposes a process-level explanation of these findings. Specifically, the model suggests that learning the nested structure of the sentential complement facilitates the understanding that nested representations may signal inconsistency. Because belief and knowledge states are similarly nested, this leads to the ability to notice inconsistency between knowledge/belief and reality in tests of ToM. On the other hand, no such signal is learned from RC training. Thus, children's performance on ToM tests does not improve in that condition.

Unlike the experiment modeled in the previous chapter, the experimental conditions in Hale and Tager-Flusberg's (2003) experiment were not specific to structure mapping. That is, the experimenters told children stories intended to teach an unfamiliar grammatical structure. Structural alignment did not play a role in the experimental design. The fact that AToM successfully replicated the pattern of results in this study provides evidence of AToM as a general model of ToM development.

6 Computational Model: Failures in Pretend Play

6.1 Introduction

The previous chapter proposed a process by which children may learn ToM from language. It has also been argued that children gain ToM skills from pretend play, or pretense (e.g., Bach, 2014; Flavell, 1999; Lillard, 1994). While there is no consensus on the specific process by which children's pretend play improves their ToM reasoning, most conclude that the interaction is caused by shared representations and/or shared processes (e.g., Lillard, 1993; see also Weisberg, 2015). The AToM model suggests that analogy is the process underlying ToM reasoning. This chapter models pretense via analogy, as well, and shows that failures in analogical processes can explain young children's failures in pretend play.

6.2 Pretend Play and Analogy

Children begin to engage in some aspects of pretense at a very young age, and their ability to engage in pretense becomes increasingly sophisticated over time (see Weisberg, 2015; Thompson & Goldstein, 2019). Specifically, object substitution precedes use of imaginary objects, which precedes creation of imaginary friends. That is, a young child might use a toy cell phone to place a call to a person she knows, while an older child may hold a toy car—or nothing at all—to her ear to call an imaginary friend. More complicated pretense requires an increasingly more complicated understanding of the world, and the ability to make increasingly more difficult substitutions. Pretense is easier when there is structural and/or functional similarity between stand-in objects and the objects they represent: Bigham (2010) showed that children with Autism Spectrum Disorder "lack competence for some types of pretense." Specifically, they performed worse when objects used in pretense were not functionally or structurally similar to the objects they represented. It is easier to pretend to make a call from a toy cell phone than a toy car or a nonexistent handset.

Although a child who uses a toy phone to call Grandma is taking less of a leap than the child who calls via a toy car, both children are recreating a telephone conversation—an event that they have seen numerous times and which we can assume has become schematized. We propose that pretense involves making analogies between the situation in front of the child and schemas, treating the real objects as if they were the kinds of things found in the schemas. Thus, when the pretend objects are more similar to those in the schema, pretense should be easier, since there is more support for the mapping between the pretend event and the schema. Both the child who calls Grandma via a toy phone and the child who uses a toy car are aligning real-world objects with those in their schema, but in the second case, the alignment and inferences made by the child are more complex, due to the reduced object similarity[33].

In this chapter we argue that, at all levels of difficulty, pretend play recruits analogical processes—specifically, analogical projection to determine and accept inferences between pretend objects and events and their real-world counterparts. We also propose, but do not model, an interactive feedback loop: engaging in pretense leads to better analogical projection, and better analogical projection leads to more complex forms of pretense. That is, as a child pretends, she becomes better at analogical projection; as she becomes better at analogical projection, the complexity of pretend play that she participates in increases.

6.3 Psychological Studies

Recently, empirical research into pretend play has focused on establishing the role of pretense in development of other skills (e.g., self-regulation, Whitebread & O'Sullivan, 2012; emotional control,

[33] Younger children are especially likely to focus on object similarity (Christie, Gentner, Call, & Haun, 2016).

Goldstein & Lerner, 2018; language competence, Kizildere et al., 2020), determining children's preferences during play in general (e.g., Taggart, Heise & Lillard, 2018), or tracking the depth of children's pragmatic understanding of pretend play (e.g., Sobel & Letourneau, 2018). However, we are interested in the processes underlying the act of pretense in children—and, in particular, the causes of failure in these processes. For this reason, we model two studies (Fein, 1975; Onishi et al., 2007) that manipulate young children's ability to engage in pretense. Although we note that the pretense in these studies is prescribed and does not mimic the full depth of pretense in older children, we believe that the underlying processes (i.e., analogical projection) are the same. In this section, we explain the results of the Fein and Onishi et al. studies through the lens of analogy. In section 6.5 below, we show how our model's results support this view.

6.3.1 Fein (1975). Fein (1975) examined mental representations in childhood pretense. In her view, pretense occurs when a child uses analogy to mentally transform an object into something else—a seashell into a cup, for example, or a toy horse into a real one. Here, Fein tested children's ability to perform multiple transformations. First, children were presented with a highly prototypical toy horse, one that convincingly looked like the real thing, and a highly prototypical cup. The experimenter then pretended to feed the horse and asked the child to "… pretend he's still hungry. You give him something to drink." This was considered the baseline trial, and children who did not give the toy horse a drink from the cup were excluded from the experiment. This baseline also "anchored the analogy by explicitly proposing a highly prototypical reference point" (Fein, 1975). In other words, it told the children that the toy horse can be transformed into a real horse for the purposes of pretense and that the toy cup can be transformed in a similar way.

The experimental portion of the study was divided into three conditions (Table 2). In the first two, one

Table 2. Experimental conditions in Fein (1975). Prototypical objects are marked with (p).

	Horse	*Cup*
Anchor	toy horse (p)	cup (p)
Condition 1A	toy horse (p)	clam shell
Condition 1B	metal object	cup (p)
Condition 2	metal object	clam shell

of the highly prototypical objects was replaced with a less prototypical version. In Condition 1A, the cup was replaced with a clam shell; in Condition 1B the toy horse was replaced with a metal horse-shaped object. In the third condition, Condition 2, both substitutions occurred. Otherwise, the procedure mirrored the baseline trial.

Consistent with Fein's hypothesis, more children were able to "give [the horse] something to drink" in Conditions 1A and 1B, when only one item was replaced, than in Condition 2, when two items were replaced at once. Fein interpreted these results to suggest that "an easy transformation (toy animal to living animal) can support a more difficult one (empty shell to full cup)" and that such anchors are necessary for transformation in difficult pretense.

6.3.2 Onishi et al. (2007) This study examined the response of 15-month-olds to violations in pretense. Experiments were performed under three conditions[34] (Table 3). In the first, an experimenter presented a child with two empty cups and an empty pitcher. The experimenter pretended to pour from the pitcher into one of the cups. Children saw one of two events: in the expected event condition (A),

[34] Each condition corresponds to an experiment in the original study (i.e., our Experiment 2, Condition 1A is Onishi et al.'s Experiment 1A, etc.).

Table 3. Experimental conditions in Onishi et al. (2007). Conditions with a familiarization trial are marked with (F).

	Poured Into	*Drunk Out Of*
Condition 1A	cup1	cup1
Condition 1B	cup1	cup2
Condition 2A	shoe1	shoe1
Condition 2B	shoe1	shoe2
Condition 3A (f)	shoe1	shoe1
Condition 3B (f)	shoe1	shoe2

the experimenter pretended to drink out of the cup that she previously pretended to pour into; in the unexpected event condition (B), the experimenter pretended to drink out of the second cup. Children looked significantly longer in condition B than in condition A. In Condition 2, the same procedure was followed as in Condition 1, but with a single change: cups were replaced by shoes[35]. Under these conditions, the looking time differences disappeared. When a familiarization trial was introduced in Condition 3—that is, the experimenter pretended to drink from a shoe, then followed the protocol from the prior experiments—the look time differences reappeared. Specifically, children looked longer when the experimenter pretended to drink out of the shoe that she had not previously pretended to pour into, than when she pretended to drink out of the other shoe.

Onishi et al. interpreted these finding to suggest that children expect consistency in pretense. They

[35] For some children, the cups were replaced by tubes rather than shoes. Since the results between these conditions did not differ, only shoes are discussed here.

expect pretense actions to follow a script and are surprised (i.e. look longer) when pretend actions do not align with the script—in this case, when the experimenter drinks from something not typically used to drink liquid. Onishi et al. further suggest that the children in Condition 2 were "distracted by the novelty or incongruity of seeing the actor 'drink' from a shoe", and that Condition 3 shows that removing such novelty returns the child to an expectation of consistency to a script.

6.4 Modeling Pretense via Analogy

Our model combines aspects of the explanations proposed by Fein and Onishi et al., unifying them in terms of analogical processing. We agree with Fein that pretense takes place via analogy, and with Onishi et al.'s idea that pretense involves following a script. Specifically, we assume that the process of analogical generalization, as explained below, is used to construct schemas, and it is these schemas that are retrieved and mapped onto real-world objects during pretense. We view Fein's transformations as analogical projections via candidate inferences: an object that is mapped to a telephone in a "making a phone call" schema is assumed, for the purpose of pretense, to be a telephone. That some transformations are easier than others follows from the well-known object bias in analogical matching by young children: early in development, children tend to focus on surface-level properties, whereas structure-level properties become more important as children acquire more relational knowledge (Gentner & Rattermann, 1991; Christie et al., 2016). Accepting these transformations, even tentatively, is a form of analogical projection. We explain Fein's notion of anchoring in terms of doing an easier mapping first, and then doing a second mapping using the results of the first one, which serves as a scaffold. Our model suggests that the familiarization trial in Onishi et al.'s Condition 3 serves a similar anchoring function.

6.4.1 Model description. We propose that pretense takes place via a series of analogical

operations between the pretense event and a retrieved schema. When a pretense event is initiated—verbally, by watching someone else perform a pretend action, or by the presence of a toy—we believe that the child first retrieves a schema from long-term memory. In structure mapping terms, this schema would originally be created via analogical generalization. In the case of placing a pretend phone call, the schema would be based on telephone calls the child had witnessed previously. While we believe that retrieval occurs via structure mapping in many cases[36], our model is agnostic to the specific mechanisms of schema retrieval.

This retrieved schema is mapped onto the observed situation, which we call the *scenario*. Some of the overlaps between relationships in the scenario and the schema will suggest potential correspondences whose candidate inferences serve as suggestions for possible transformations. When an inference is generated and tentatively accepted for reasoning, analogical projection has occurred. For example, the telephone schema would suggest that the object the child is holding is a telephone. Each such inference has some associated probability (e.g., being a telephone might be 1.0, whereas its color might be black with 0.5, silver with 0.5). If the object is a toy telephone, it should both be easier to produce inferences and any inferences produced should be more compatible with the object—and so, the transformation should be easier to accept—whereas for a toy car, they should be less so. We are agnostic as to the process by which this evaluation takes place – it might be a recursive analogical match against a schema for the type of object involved, for example, or involve reasoning about conflicts between visible properties/prior knowledge with the projected properties, such as knowing telephones

[36] In the Onishi et al. study, for example, the experimenter's arm moving to her mouth may invoke an actual drinking event and retrieve the appropriate schema. In the Fein study, on the other hand, retrieval of the appropriate schema is likely aided by the verbal cue to "give [the horse] a drink."

do not have wheels. In any case, we assume that the more similar the pretend objects are to their real-world equivalents, the easier this process of evaluation is, and the more likely the child is to generate and accept the proposed mapping. If the mapping and corresponding inferences are accepted, then the pretense continues. If they fail, the pretense ends.

When an anchoring event is observed, it is stored together with the pre-existing schematic event in working memory (see Figure 14). This combination schema is represented via an interim generalization, which allows the pretense to continue more easily via direct mapping, since the candidate inference has already been accepted (i.e., the toy car is already viewed as a telephone, so the child will more likely be willing to use it to make a pretend call). We show how this works with the Fein (1975) and Onishi et al. (2007) studies next.

6.4.2 Model procedure. We assume that children are able to retrieve an appropriate schema out of long term memory, as failures in pretense tend to manifest as a lack of pretense rather than

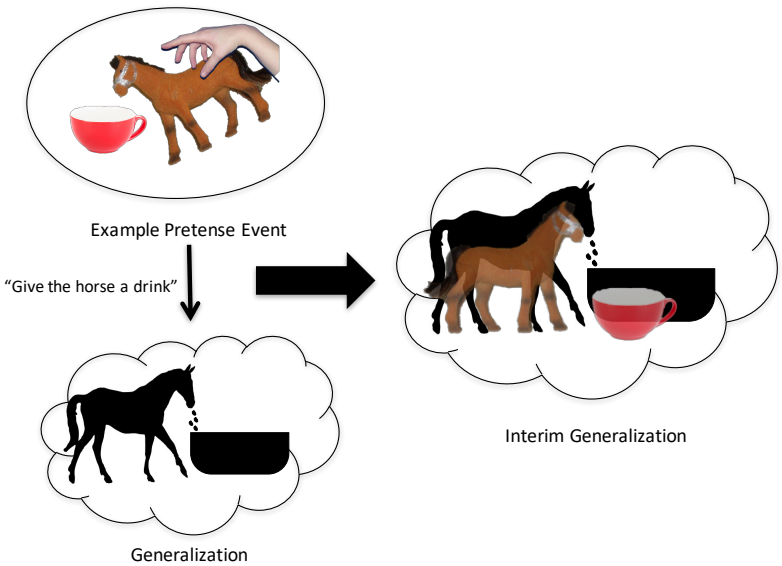

Figure 14. An Interim Generalization created from a generalization of a horse drinking and a single pretend event wherein a toy horse drinks from a cup.

unexpected pretense; Fein (1975) did not report any children pretending to make a phone call using the horse, for example. Furthermore, these studies were controlled such that unexpected pretense was unlikely. For this reason, and the impossibility of accurately modeling all of the generalizations in a child's long-term memory, we simply provide an appropriate schema to the model as one of its inputs. Since, again, modeling the sequence of experiences that a child might experience fully is impractical to impossible, we instead use *synthetic generalizations* to create input schemas. Synthetic generalizations are made by approximating the probability of facts in the schema based on plausible assumptions about the distribution of experiences that someone might see. For example, a child may have seen a telephone call being placed using a smartphone 50% of the time, an older cellular phone 20% of the time, and a wireless home telephone 30% of the time. Varying the distributions did not affect model outputs.

To model each study, we use synthetic generalizations to represent schematic events (e.g., taking a drink), interim generalizations between the synthetic generalization and the anchoring event to represent the combined schema in working memory (e.g., taking a drink from a shoe), and single events to represent individual expected pretense scenarios. In each case, the pretense scenario is the target, and the generalization or interim generalization is the base. We interpret candidate inferences relevant to schema satisfaction (i.e., the horse involved must be a real horse, but it does not necessarily need to be brown) suggested by SAGE as necessary in order for pretense to continue.

6.5 Simulations

We tested our model using the results of Fein (1975) and Onishi et al. (2007). In the model, transformations as proposed by Fein are candidate inferences that must be accepted for pretense to continue. Scripts, as suggested by Onishi et al. are represented as schemas produced via analogical generalizations from prior experiences. The process of anchoring is based on an initial comparison

between a schema and a situation, producing an interim generalization which will match better to subsequent pretense scenarios.

6.5.1 Experiment 1: Fein (1975). Recall that this study sought to elicit pretense in four-year-old children by providing them a horse stand-in and a cup stand-in and asking them to give the horse a drink. Following a baseline anchor trial with a highly prototypical horse and a highly prototypical cup, children were tested under one of three conditions: 1A, in which the horse stand-in remained highly prototypical, but the cup stand-in was replaced by a non-prototypical item; 1B, in which the cup stand-in remained highly prototypical, but the horse stand-in was replaced by a non-prototypical item; and 2, in which both the cup stand-in and the horse stand-in were replaced by their respective non-prototypical versions (see Table 2).

6.5.1.1 Model inputs. We model the experiments in this study as a schematic example of a horse drinking probed by the expected pretense of the horse stand-in drinking out of the cup stand-in. Since children who failed to pretend in the baseline trial were dismissed from the rest of the experiment, we assume that the toy horse and cup were part of the remaining children's interim generalization.

6.5.1.2 Model results. According to our model, pretense is possible in all the conditions tested by Fein. Condition 2, however, requires accepting more candidate inferences (CIs) than do Condition 1A and Condition 1B. The required inferences can be found in Table 4. Specifically, when a child is first presented with the toy horse and seashell in Condition 1A, he is able to match the horse as the drinking entity since this is how the horse was portrayed during the baseline trial. The remaining CI is of the seashell as a cup. Similarly, when a child is presented with a metal horse and toy cup in Condition 1B, she is able to match the toy cup to the item to be drunk out of but must now accept the CI of the metal

Table 4. Candidate Inferences needed for successful pretense in Fein (1975).

Condition	*Number of Category CIs*
1A	1
1B	1
2	2

object as a horse. In Condition 2, on the other hand, we see two CIs: that the metal object is a horse and that the seashell is a cup; the child must accept both. Failures happen when children are not able to accept these CIs, presumably because they find that two simultaneous inferences are implausible.

6.5.2 Experiment 2: Onishi et al. (2007). Recall that this study measured infant looking times when an experimenter pretended to drink out of cups (Condition 1) or shoes (Conditions 2 and 3). Each condition had a control and experimental trial. In the control, the experimenter pretended to pour water into the cup/shoe she then pretended to drink out of (A); in the experimental trial, the experimenter pretended to pour water into a different cup/shoe (B; see Table 3).

6.5.2.1 Model inputs. To model Conditions 1 and 2, we gave the model a schema for drinking with pretense scenarios corresponding to the control and experimental conditions (i.e., the object that is poured into is versus is not the object that is drunk out of). For Condition 3, an interim generalization—generated from the previous schema with the addition of drinking out of a shoe, based on the study's familiarization trial—was provided; pretense scenarios were reused from Condition 2.

6.5.2.2 Model results. Number of inferences required for successful pretense in each of Onishi et al. (2007) experiments can be found in Table 5.

In the control trial of Condition 1, pretense is easy: the only necessary CIs are that the pitcher is actually full, and that it causes the cup to become full after pouring. On the other hand, in the experimental condition there is an additional CI. The child must accept the that the cup being drunk out

Table 5. Candidate Inferences needed for successful pretense in Onishi et al. (2007). Conditions marked A correspond to control trials; conditions marked B correspond to experimental trials.

Condition	*Number Category CIs*	*Number Attribute CIs*
1A	0	2
1B	0	3
2A	1	2
2B	1	3
3A	0	2
3B	0	3

of had been poured into—or that the cup that had been poured into is being drunk out of[37].

This additional CI accounts for the looking time difference between the control and experimental conditions; both cases of pretense are plausible, one is just more difficult. Condition 2, however, requires accepting even more CIs: the child must additionally accept that a shoe can play the role of the object being drunk out of (i.e., that it can be a cup). This is much harder for the child to accept, as it requires changing the category of the item. For this reason, pretense fails. This is true of both the control and the experimental conditions. Finally, Condition 3 shows the importance of interim generalizations. Since an experimenter demonstrated the act of drinking from a shoe to these children, they were able to create the interim generalization that shoes can be drunk out of. Because of this, they did not have the additional CI as in Condition 2 and were able to accept the pretense again. While pretense is possible in all scenarios, it is substantially harder in Condition 2; so much so, that infants are

[37] SME returns two equivalent mappings here. We assume that the child entertains only one, although the looking time difference may be attributed to the multiple potential mappings, as well.

unable to participate in the pretense. As such, both looking time differences (in Condition 1 and Condition 3) and lack thereof (Condition 2) can be explained by the number and plausibility of candidate inferences.

6.6 Discussion and Predictions

Our model replicates the pattern of results for both the Fein (1975) and Onishi et al. (2007) studies. This provides evidence that analogical projection with judgment of candidate inference plausibility provides an explanation for children's failures and successes in pretense. Our results suggest that more advanced pretense reflects more advanced analogical projection abilities. We posit that this relationship is self-reinforcing—the more children play pretend, the better they become at analogical projection; the better children become at analogical projection, the more they are able to play more advanced forms of pretense.

The relationship between pretense and analogy leads to several predictions. First, we predict that progressive alignment (Kotovsky & Gentner, 1996) will bootstrap children's pretense. That is, by participating in a series of pretense scenarios wherein the objects that must be transformed become progressively more distant from their target, children will be able to participate in more complex pretense than they would otherwise. Fein's (1975) findings directly support this prediction—anchoring can be viewed as a short-term form of progressive alignment. Furthermore, progressive alignment has previously been modeled as online re-representation in interim generalizations (Kandaswamy et al., 2014). This suggests that similar mechanisms are involved in learning via progressive alignment and in pretend play.

We also predict that pretend play will be more difficult when the entities in the pretense scenario are cross-mapped, or when an entity in the pretense scenario is more similar to a different entity in the

real world scenario than the one it is intended to be mapped to (Gentner & Toupin, 1986). For example, a child playing making a family of stuffed animals should prefer to pretend that larger animals are the parents, while smaller animals are the children. In a more extreme example, if the experimenter in Fein's (1975) anchoring condition had fed the cup using the horse, rather than the other way around, we predict that the children would not have been able to carry on the pretend play by giving the cup a drink from the horse—and certainly would not have transferred the cross-mapping to the test conditions.

In analyzing our model, it is important to note that our representations are simplifications of the full pretense scenario. In addition to accepting candidate inferences to describe the objects involved in pretense, children must also accept candidate inferences relating to events. Such inferences, however, are common to all pretense, so we chose to omit them for clarity. Including more inferences and richer representations of events would not change the conclusions and predictions drawn from our model.

6.7 Conclusion

In this chapter, pretend play was modeled as an analogical process. Specifically, we showed that a failure to generate and accept appropriate candidate inferences can account for young children's failed pretense in two studies. It has previously been suggested that children learn ToM from pretend play, likely because the two share the same processes. According to AToM, analogy is the central process of ToM—and generating and accepting candidate inferences drives ToM reasoning. This chapter provides evidence that analogy is also the central process of pretense, suggesting that playing pretend helps children develop the ability to reason analogically between an imagined world and the real one. These skills may then transfer to reasoning analogically between people to allow for differing internal states (i.e., ToM reasoning).

7 Application: Using Theory of Mind for Goal Recognition

7.1 Introduction

The previous chapters have shown that AToM is a plausible model of human ToM reasoning (i.e., people reasoning about other people). In this chapter, AToM is applied to the task of goal recognition (E-Martin, R-Moreno & Smith, 2015), in which a virtual agent must infer another agent's goals from observations of that agent's actions[38].

Most state-of-the-art goal recognition systems make strong assumptions about the kind of information that is available during this task. They typically receive an observation trace of an agent's activities as a sequence of action-state pairs from the agent's planner, and reconcile these actions with a set of known or learned possible plans to infer the plan that the agent is performing, and thereby its top-level goal (Ramírez & Geffner, 2009). Alternatively, hierarchical plan recognition (Geib & Goldman, 2011; Holler et al., 2018) reconciles the observation trace using decomposition methods that aggregate the primitive actions into high-level tasks.

Since these recognition approaches access the same information about the observed agent's actions that the agent receives (i.e., the recognition algorithm observes the action-state pairs sent to the agent, including all parameters), the observation trace contains information about the internal state of the observed agent that cannot be gleaned from external observations alone. This type of internal information is unlikely to be available in many real-world scenarios (e.g., when an agent must reason about a person or an agent implemented by another organization whose internals are opaque). Instead, an agent must be able to reason based only on its own external observations.

---

[38] This chapter is an adaptation of Rabkina et al. (2020).

In this chapter, AToM's performance on goal recognition is compared to a state-of-the art goal recognition system (PANDA; Holler et al., 2018) under standard goal recognition conditions and as internal information is abstracted away to only the information available from external observations. A task in the open world Minecraft AI platform (Johnson et al., 2016) is used. When both systems have perfect internal knowledge, AToM is slightly worse than PANDA at recognizing an agent's goals. However, as knowledge is reduced, PANDA's performance drops substantially while AToM maintains performance.

7.2 Agent Simulation in Minecraft

7.2.1 Task Description We define a problem space in the open-world game Minecraft[39]. An agent, Alex, is placed in a flat Minecraft world with a small farm in the middle and items randomly distributed around the perimeter. These include crop seeds, bone meal, chickens, cows, buckets of milk, eggs, and sugar. After a period of exploring, Alex chooses a goal (to make a single food item) that will maximize its food points, given the items it has observed and its food preference (herbivore, omnivore, carnivore). We use Minecraft's internal food points system for value calculations, shown in the left column of Table 6. The goal recognition systems must recognize which goal Alex is pursuing. We assume that it pursues one goal at a time (i.e., no interleaved goals).

Many of Minecraft's crafting tasks have a natural hierarchical structure. For example, crafting bread requires three wheat, and wheat is grown and harvested using wheat seeds. Growth can additionally be sped up using an item called bone meal. Due to these natural hierarchies, the agent's

[39] See Roberts et al. (2016) for a description of the game and the supporting framework we leverage, and Johnson et al. (2016) for information on Minecraft's Malmo platform for AI experimentation.

Table 6. Minecraft model for planning with SHOP2

Top-Level Tasks (Food Point Values)	*Helper Task Categories*	*Action Categories*
Obtain Chicken (2)	Crafting Items	Movement
Obtain Beef (3)	Gathering Items	Look
Obtain Pumpkin Pie (8)	Growing Crops	Item Selection
Obtain Cake (14)	Using Inventory Item	Item Crafting
Obtain Carrot (3)		Item Gathering
Obtain Potato (1)		
Obtain Bread (5)		

behaviors are defined using Hierarchical Task Networks (HTNs; Erol, Hendler & Nau, 1994).

7.2.2 HTN Planning and Execution We use the HTN planner SHOP2 (Nau et al., 2003) to generate plans for an agent to execute in the Minecraft environment. Planning with SHOP2 requires two components of knowledge: a state model and an HTN planning model. The state model used by SHOP2 defines the Minecraft game state as a set of first-order predicates. These predicates can have both numerical and symbolic arguments. Specifically, the state model contains information about the inventory of the agent, such as items in the inventory and location of items in the hotbar. The model also contains information about entities and locations the agent has observed, and information about the agent itself, such as its current location and view location. An example of the state model is {`(entity_at cow123 loc123)`, `(inventory_count wheat_seeds 10)`}.

The HTN planning model contains primitive and compound tasks that the agent can do in the environment. We categorize primitive tasks in the HTN domain model into the following: movement, look, item selection, item crafting, and item gathering. We also categorize compound tasks into top-level tasks and helper tasks. Top-level tasks are objectives that the agent directly wants to pursue (such as making pumpkin pie and cake). Helper tasks are those that the agent does in order to complete top-level tasks. Helper tasks can be categorized into crafting items, gathering items, growing crops, and

consuming/using items in inventory. These categorizations are summarized Table 6. An example of a plan generated by the SHOP2 planner for obtaining beef would be: `{(move-near-entity cow123), (look-at-entity cow123) (select iron-sword) (attack cow123) (gather beef)}`.

Plans generated by the SHOP2 planner are used by Alex to construct executable actions in Minecraft. At a high level, a plan is expanded into an executable sequence of actions (i.e., an executable plan). This executable plan is then executed in the Minecraft environment to completion. For crafting-related task, an additional planning step may be added if Alex does not have necessary items in its inventory. Specifically, if the items have been observed in the environment, but are not in Alex's inventory, it constructs a plan to retrieve them. If items have not been observed in the environment, the entire plan is ignored. Once all items have been retrieved, Alex re-plans. Re-planning makes sense here because Alex may observe items for more important objectives while retrieving items for crafting. In such cases, during re-planning, Alex should execute the more important objective.

7.3 Approach

7.3.1 Analogical Theory of Mind for Plan Recognition Plan recognition is treated as a classification problem for AToM[40]. Because of the longer timescale of experimentation, LTM is always triggered, so AToM learns via SAGE (McLure et al., 2015). Cases are predicate calculus representations of a single trace of Alex performing a goal. Depending on experimental condition (see section 7.4), the trace consists of the output of the SHOP2 planner, a report of the agent's actual actions, or sensor-like observations of those actions.

[40] This allows candidate inferences to be used for further reasoning, such as making predictions about an agent's knowledge or projecting future actions.

During training, AToM learns a generalization pool for each goal by passing training cases to SAGE, one at a time. During testing, a case is retrieved from the union of learned generalization pools. The goal that corresponds to the retrieved case is returned as the predicted goal for the given test case.

7.3.2 Planning and Acting in a Network Decomposition Architecture (PANDA) We compare AToM's performance to an off-the-shelf goal recognition system. Following Holler et al.'s (2018) state-of-the-art approach, we use the Planning and Acting in a Network Decomposition Architecture (PANDA; Bercher, Keen, & Biundo, 2014) as an off-the-shelf HTN planner for goal recognition via plan construction. PANDA is a hybrid planning algorithm, which combines HTN planning concepts with partial-order causal link planning. We refer to HTN planning for plan and goal recognition using PANDA, as PANDA-REC.

PANDA-REC takes as input a sequence of actions and an HTN planning model and converts the planning model into a model for goal recognition. This model is then passed into PANDA, and the recognized goal (i.e., a single top-level task) is extracted. Note that, unlike AToM, PANDA-REC is given a model, rather than learning it.

7.4 Experiments

7.4.1 Overview The objective of our experiments is to compare AToM and PANDA-REC's ability to infer an agent's goals given different types of observed sequences of actions. Specifically, we extract sequences of actions (i.e., traces) from both planner output and agent action executions in Minecraft. We then compare AToM with PANDA-REC for goal recognition from these traces.

All traces used in our experiments were extracted from Minecraft play sessions logs. A play session corresponds to an agent being placed on a map and executing top-level tasks from Table 6 for a predefined amount of time. From this session, a single log is generated and consists of all planner output

and executed actions of an agent throughout the session.

We constructed a dataset of play session logs by having an agent play 10 pseudo-randomly generated maps 5 times, with simulations running for 180 seconds each. Once the log dataset was constructed, we randomly extracted datasets of 100 plan traces (i.e., planner output) and 100 action execution traces. Note that `obtain_carrot` did not appear in the action execution trace dataset.

Two random baselines were computed for each experiment. The first generated its interpretation of the agent's goal by sampling uniformly across goals that appear in the dataset. The second was biased, with each potential goal weighted by its prevalence in the dataset. All results are reported in Table 7. Where available, standard deviations are reported in parentheses. The highest accuracy for each test is bolded.

7.4.2 Experimental Conditions Goal recognition was tested under four conditions: (1) plan traces; (2) execution traces; (3) sensor-like external observation traces; (4) transfer from execution to external observation traces. The first condition includes all information expected by a standard goal recognition system (i.e., has full internal information), as it includes the exact trace sent from the SHOP planner to the agent. The second includes partial internal information, in that it contains information about what the agent did, but not what it intended to do. Condition three includes only information that can be seen by an external observer. Finally, the fourth condition uses a model for execution traces to test external observations. This tests whether each system can leverage partial internal information when reasoning about agents that can only be observed (i.e., to which it has no internal access). PANDA-REC was provided an appropriate HTN model for each condition, while AToM learned a model via 10-fold cross validation. Example traces of each condition can be found in Appendix B.

7.4.3 Results In the place traces condition, PANDA-REC was 100% accurate in recognizing

Table 7. Results for Goal Recognition Experiments

	PANDA-REC	*AToM*	*Uniform Baseline*	*Biased Baseline*
(1) Plan Traces	**1.0**	0.92 (0.075)	0.14	0.226
(2) Execution Traces	0.63	**0.90** (0.077)	0.167	0.237
(3) External Observations	0.63	**0.88** (0.098)	0.167	0.237
(4) Exec Model / Ext. Obs. Test	0.30	**0.90 (---)**	0.167	0.237

goals based on the SHOP2 planner's output. This fit our intuition, as PANDA-REC is given the HTN planning model used by the SHOP2 planner. AToM performed worse, with 92% accuracy. Both systems performed significantly better than the uniform and biased baselines ($p<.05$).

PANDA-REC's accuracy dropped substantially when working from agent actions but remained above both baselines. It performed at 63% accuracy. AToM's performance did not change substantially from when the planner trace outputs were used. It maintained 90% accuracy. Similarly, PANDA-REC performed at 63% accuracy when using external observations with an external observation HTN model, while AToM dipped slightly to 88% accuracy. In both conditions, a one sample non-parametric median test showed that AToM performed significantly better than PANDA-REC ($p <.05$).

When using an execution trace model to test recognition based on external observations, however, PANDA-REC's performance dropped again, to 30% accuracy, while AToM performed at 90% accuracy. Thus, AToM performed as well when transferring between knowledge conditions, as when trained and tested on the same condition.

7.5 Discussion

For these Minecraft recognition tests, AToM outperformed PANDA-REC on goal recognition conditions when given partial internal information or external information only. This is a hallmark of human ToM reasoning, which AToM models. Thus our results suggest that ToM reasoning in general,

and ToM reasoning via AToM in particular, can help agents reason about others.

The chief claim of AToM as a cognitive model is that ToM reasoning and development occur via analogical processes. Here, those same processes allow AToM to robustly reason about the internal states of agents, without direct knowledge of those states. Specifically, analogy allows AToM to make inferences based on its previous observations. For example, if it has learned that agents walk up to cows and chickens before slaughtering them (e.g., from agent action traces), it can infer that the object the agent was walking toward before slaughtering it (e.g., in an external observation trace) was also a cow or chicken. Furthermore, analogy's focus on structure makes retrieval with complete object uncertainty possible. That is, if all objects were removed from a trace, AToM would guess that throwing something at the ground and later harvesting something else is a planting task—perhaps mistaking `obtain_potato` for `obtain_carrot`, but not `obtain_beef`. It remains to be seen whether other ToM models can do similar reasoning.

From a practical standpoint, one disadvantage of AToM, as compared to PANDA-REC, is its need to be trained. When recognizing from planner output, PANDA-REC was able to use the planner. While the model did need to be modified further for the other conditions, training data was never necessary. On the other hand, PANDA-REC has the disadvantage of requiring a hand-crafted model.

Interestingly, the generalizations learned by AToM were often similar to the individual plans in PANDA-REC's model. This suggests that the models used by PANDA-REC, when converted to cases of a format similar to observation trace outputs, may be sufficient to populate AToM's case library. That is, explicit training may not be necessary. Alternatively, the AToM model might provide insights into learning, rather than hand crafting, the PANDA model. We will explore these possibilities in future work.

More generally, we would like to give agents the ability to not only recognize compatriots' goals, but

also to change their own behavior accordingly. This requires online goal recognition that is accurate while reasoning from partial data (i.e., before the compatriot finishes its task). PANDA-REC can be configured to make a recognition decision prior to seeing a complete plan trace (Holler et al., 2018). However, the computations for this can become too slow for online recognition. On the other hand, analogical retrieval allows AToM to be relatively fast. It remains to be seen whether AToM can maintain accuracy with partial traces. It is likely that other components of ToM reasoning (e.g., about knowledge and desire states) will need to be integrated in order to increase robustness of AToM's predictions from partial traces. We will explore applications of PANDA-REC and AToM to online goal recognition in future work.

7.6 Conclusion

This chapter has demonstrated that AToM can perform goal recognition, a classic task for virtual agents reasoning about other agents. During goal recognition from plan traces—which contain information about the observed agent's internal state—AToM performs comparably to a state-of-the-art goal recognition system (PANDA; Holler et al., 2018). However, when internal information is abstracted away, AToM maintains its performance at approximately 90% accuracy, while the state-of-the-art system dips from 100% in the full information condition to 63% in the partial information condition, to 33% when transferring from a partial internal information model to external-only test cases. Thus AToM, as a full ToM model, appears to be doing more robust reasoning than the state-of-the-art goal recognition model.

8 Application: Recognizing Cooperation in the Stag-Hunt Game

8.1 Introduction

The stag-hunt game (Skyrms, 2004), a simple prisoner's dilemma-style game, has recently gained popularity as a test of ToM reasoning for simulated agents. In this chapter[41], we show that AToM can recognize when players intend to cooperate in an observed stag-hunt game. Using a small pre-existing dataset (Shum et al., 2019) we show that AToM's predictive accuracy does not differ from a Bayesian model's or human performance (both Shum et al., 2019). We expand the dataset, increasing the complexity of both the grid and possible goals, and show that AToM performs well on the more complex dataset. Finally, we attempt to improve AToM's performance by extending it to second-order ToM reasoning (i.e., reasoning about agents' beliefs about other agents), but do not find evidence that second-order ToM is helpful for this task.

8.2 The Stag-hunt Game

Stag-hunt was first proposed as an alternative to the prisoner's dilemma set of cooperative/competitive games (Skyrms, 2004). Unlike the traditional prisoner's dilemma, cooperation in stag-hunt does not come with a penalty; if all parties choose to cooperate, each individual wins a greater reward than if they had chosen to compete. However, if an individual chooses to cooperate but their compatriots do not, then the individual receives no reward at all. In effect, stag-hunt is a test of one's ability to recognize others' intent to cooperate.

In a typical spatial stag-hunt scenario, a grid world map with hares and stags is generated. Hares are low-value targets that can be captured by a single hunter without the cooperation of others. Stags, on

[41] This chapter is, in part, an adaptation of Rabkina & Forbus (2019).

the other hand, are high value targets that must be captured by a team of (two or more) cooperating hunters. At each timestep, each hunter can take one step up, down, left, or right. Depending on the implementation, targets may be able to do the same. Capture occurs (and points are earned) when the necessary number of hunters occupies the same square as a target.

8.3 Experiment 1

In this experiment, we show that AToM can predict cooperation between agents on Shum et al.'s (2019) stag-hunt dataset. We compare AToM's predictive accuracy to a Bayesian model's and human subjects, both as reported by Shum et al. Note that, while Shum et al. show evidence that their model's results correlate well with people's predictions of cooperation between hunters, data about actual predictions made is limited to line graphs (i.e., no numerical values). Thus, our comparisons to their results are based on good faith approximations but may not be entirely accurate.

8.3.1 Dataset. We use Shum et al.'s (2019) stag-hunt simulations here. In their version of the game, three hunters, two hares, and two stags are placed on a 5x7 grid world. Some squares in the grid world are not traversable, creating a variety of spatial layouts across grids. Starting locations of hunters and targets also vary. A stag is considered caught when two or more hunters are in the same square as it at the same time. Similarly, a hare is considered caught when exactly one hunter is in the same square as it.

In their experiments, Shum et al. simulate three time-steps of nine different scenarios. At each timestep, each hunter moves zero or one squares up, down, left, or right. Hares cannot move, but stags can move to avoid capture. Thus, no target is ever captured before the third timestep, but at least one target is captured in each scenario. In four scenarios only a hare is captured, and no cooperation occurs (Figure 15 b, e, f, h). In three scenarios, a pair of hunters cooperates to capture a stag (Figure 15 a, c, d).

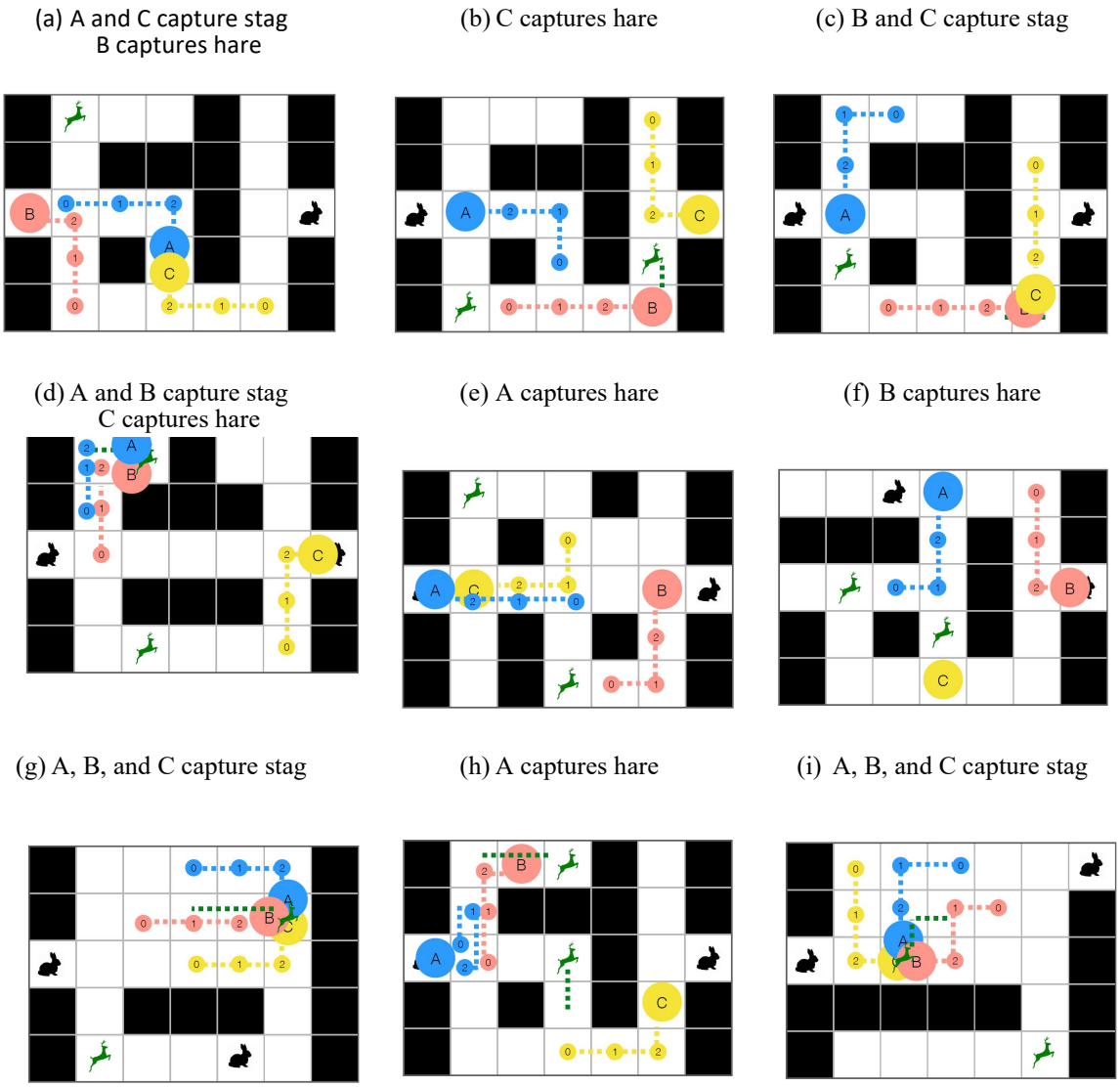

Figure 15. The nine stag-hunt scenarios from Shum et al. (2019).

In the remaining two scenarios, all three hunters cooperate to capture a stag (Figure 15 g, i). These scenarios are shown to an observer, which makes predictions about the agents involved.

8.3.1.1 Encoding and representations. All scenarios were encoded into predicate calculus from the images provided in Shum et al. (2019). Final spatial representations were based on QSRLib (Gatsoulis

et al., 2016), a library of qualitative spatial calculi, which are cognitively motivated relational representations of space, based on the qualitative representation literature. At each timestep, the system computed (1) whether each individual agent (i.e., hunter or target) is moving, per the Moving or Stationary (MOS) calculus, (2) whether each moving agent has moved closer to or farther from each other agent , per the Qualitative Distance Calculus (QDC; Clementini et al., 1997), (3) whether a pair of agents has overall moved closer to or farther from each other per the Qualitative Trajectory Calculus (QTC; Delafontaine, Cohn & van de Weghe, 2011; van de Weghe et al., 2005), and (4) whether two agents are qualitatively close, far, or located on the same square at all time points (i.e., before step 1, after step 1, after step 2, and after step 3; QDC). Because the agents are situated in a grid world and can move only up, down, left, or right, we used path distance for all distance measurements. Causal relationships between the relations generated in (1), (2), (3), and (4) were also computed (Figure 16).

Non-spatial events (i.e., capture of a target and ground truth cooperation between hunters) were manually encoded using the NextKB knowledge base (Forbus & Hinrichs, 2017), which integrates materials from several open source ontologies. When appropriate, causal relationships between capture events and cooperative events were also recorded.

8.3.1.2 Case generation. Recall that all scenarios in the stag-hunt domain proposed by Shum et al. (2019) include three timesteps. Our goal is to make predictions about hunters' cooperation and future movements at each step. Thus, we used the representations described above to generate a total of four structured cases for each scenario: three for testing (one per timestep) and one for training. The cases used for testing included all computed relations for that timestep and all previous timesteps. That is, the case for step 1 only had information about step 1 (i.e., reflected movement that happened between the start of the simulation and the end of the first step), but the case for step 2 had

```
a)
(causes-PropProp
 (and (holdsIn step1 (approaches hunterA hunterB))
      (holdsIn step1 (approaches hunterB hunterA)))
 (holdsIn step1 (closer hunterA hunterB)))
b)
(causes-PropProp
 (and (holdsIn step1 (distances hunterA stag1))
      (holdsIn step1 (stationary stag1)))
 (holdsIn step1 (farther hunterA stag1)))
```

Figure 16. Example qualitative spatial relations between agents in a stag-hunt step. In a), hunterA and hunterB move toward each other, resulting in the two hunters being closer together than in the previous timestep. In b), hunterA moves away from a stationary stag1, causing the two to be farther apart. These causal relationships were computed automatically.

representations for both step 1 and step 2 (i.e., reflected movement that happened between the start of the simulation and the end of the second step), etc. While capture information was included when capture events occurred, no cooperation events were included in these cases. The training case included information from all three timesteps (i.e., reflected all movement from the beginning to the end of the simulation), along with ground truth cooperation events and related causal relations.

8.3.2 Training and testing. After all cases were generated, we used a variant of leave-one-out cross validation to train and test AToM for each scenario at each timestep. This results in a total of 27 testing rounds (one per timestep per scenario). At each round, we trained using only complete cases of the other eight scenarios (including cooperation information) and tested using timestep-specific cases. As described above, training cases included information about movement at all three timesteps, along with ground truth cooperation events. Test cases included only movement information for the timestep being tested and any preceding timesteps.

During testing, the best overall match for the scenario at the given timestep was retrieved. Candidate inferences from the retrieved case to the test case were computed. This resulted in

a)
```
(and (isa (SkolemFn coop1) CooperationEvent)
     (cooperationParticipants (SkolemFn coop1) hunterA)
     (cooperationParticipants (SkolemFn coop1) hunterB))
```
b)
```
(and (isa (SkolemFn coop1) CooperationEvent)
     (cooperationParticipants (SkolemFn coop1) hunterA)
     (cooperationParticipants (SkolemFn coop1) hunterB)
     (cooperationParticipants (SkolemFn coop1) hunterC))
```

Figure 17. Two examples candidate inferences for cooperation recognition. a) predicts a cooperation event between hunterA and hunterB. It represents zero or one true positive inferences and up to two true negative inferences. b) predicts a cooperation event between all three hunters. It represents zero, one, or three correct true positive inferences. Representations are simplified for clarity.

inferences about cooperation events (Figure 17), future movements, and current and future causal relationships. Inferences about cooperation were automatically identified by their respective predicates. These were assumed to be AToM's cooperation predictions.

Consistent with Shum et al. (2019), we analyzed cooperation inferences in terms of hunter dyads. This means that, for each scenario at each timestep, up to three cooperation relationships could be inferred (i.e., hunterA and hunterB, huntertA and hunterC, hunterB and hunterC).

8.3.3 Results. We first compare our model's cooperation inferences to ground truth cooperation events: when two or more hunters captured a stag. Following Shum et al. (2019), we measure accuracy pairwise between hunters, for a total of three predictions for each scenario at each timestep (i.e., hunterA and hunterB, hunterA and hunterC, hunterB and hunterC). A true positive inference is one that predicts cooperation between two hunters that do, in fact, cooperate. A true negative, on the other hand, is the absence of an inference of cooperation between two hunters that do not cooperate in the full scenario. Example candidate inferences are shown in Figure 17.

Table 8 shows our model's overall accuracy. Accuracy is highest (96%) at timestep 3, where the model makes only one incorrect inference. At earlier timesteps, when less information about hunters'

Table 8. Accuracy of cooperation inferences made by AToM at each timestep.

Timestep	*Accuracy*
step 1	0.77
step 2	0.81
step 3	0.96
overall	0.85

behavior is available, accuracy is worse. The lowest accuracy is at timestep 1, when 77% of predictions are correct.

Shum et al. (2019) compare the inferences made by their model against human predictions about cooperation, rather than ground truth cooperative behavior. Both the model's inferences and human judgements are made on a continuous scale that represents degree of certainty that two agents are cooperating. To compare accuracy of our model's predictions to those made by Shum et al.'s model and human participants, we use a 0.5 cutoff. That is, a judgement that cooperation is at least 50% likely corresponds to a positive inference, while a judgement that cooperation is less than 50% likely corresponds to a negative inference. Note, that a lower threshold would correspond to a higher rate of false positives, while a higher threshold would correspond to a higher rate of false negatives.

A comparison of accuracy between human judgement, Shum et al.'s (2019) Bayesian model, and our analogical model is shown in Figure 18. Humans have the highest overall accuracy, tied with the analogical model at step 1 and the Bayesian model at step 2. At step 3, the humans reach 100% accuracy, while the analogical model slightly outperforms the Bayesian. However, none of these differences are statistically significant (all $p > 0.05$).

8.3.4 Discussion. AToM recognizes agents' intent to cooperate in the simple multi-player game stag-hunt no differently from a Bayesian model and humans. To evaluate the model, we assumed

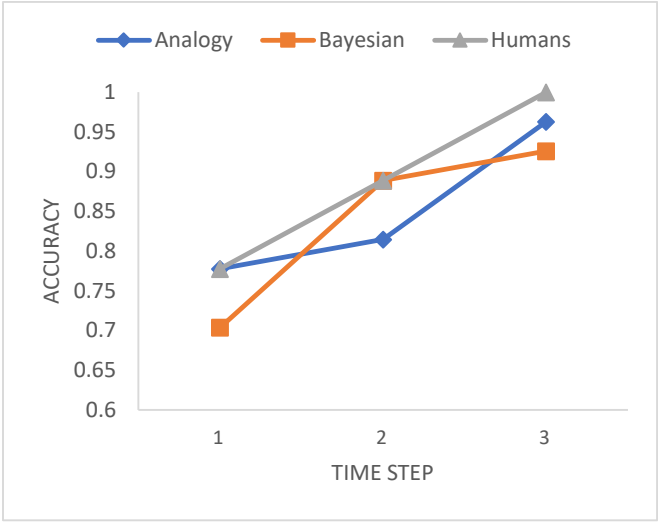

Figure 18. Accuracy of cooperation inferences made by the analogical model as compared to Shum et al.'s (2019) Bayesian model and human predictions (0.5 probability cutoff). Note that the y axis is shifted to the appropriate range.

that if, at the end of the scenario, two hunters cooperated to catch a stag, then they had intended to cooperate at each previous timestep. This is consistent with how humans made predictions about the agents' behavior in Shum et al.'s (2019) study: at the final timestep, humans inferred that exactly those agents that acted together to catch a stag were cooperating. They gave approximately 100% certainty to cooperation between those agents, and approximately 0% certainty to cooperation between all other agent pairs.

However, successful cooperation is not the only signal for intent to cooperate, and may, in fact, not be a reliable one at that. We have identified three situations within the stag-hunt game where the assumption that cooperation occurs if and only if it is successful does not hold. The simplest example is *unsuccessful cooperation*. That is, two hunters intend to capture a stag together, but the stag escapes before they are able to corner it. In this case, the intent to cooperate exists, even though the cooperation is not successful.

Similarly, *non-reciprocal cooperation* may occur when one hunter intends to cooperate with another, but its would-be partner has other plans. This example points to an inherent flaw in defining intent to cooperate as reciprocal; without access to other agents' internal states (or enough theory of mind reasoning capabilities to infer them), an agent can decide to cooperate with an unknowing partner. It would be incorrect to infer that the two agents intended to cooperate in this case. But it would also be incorrect to infer that no intent to cooperate took place at all. Instead, a unidirectional intent to cooperate should be inferred.

On the other hand, it is possible for two hunters to cooperate to capture a stag without intending to. *Non-intentional cooperation* occurs when two agents end up at the right place at the right time. For example, they are both pursuing the same hare and find themselves surrounding a stag. At this point, they might change their plans and decide to capture the stag. Alternatively, it may be that the stag is on both agents' path to the hare. It might be argued that there is an intent to cooperate in first case, albeit only at the last step. In the second case, however, there is no intent to cooperate whatsoever; the hunters capture the stag purely by happenstance. The assumption that cooperation is intended if and only if a cooperative event occurs would lead one to infer that there was an intent to cooperate in both of these cases, including when the agents were, in fact, individually pursuing a hare.

Whether the distinction between two agents cooperating and two agents intending to cooperate matters largely depends on the task at hand. In the present work, where an observer is making inferences about other agents, inferences made with this simplifying assumption may be sufficient. However, if an agent intends to act on its inferences, not considering unsuccessful or non-reciprocal cooperation to be cooperation at all can lead to suboptimal behavior, most likely in the form of missed opportunities to cooperate (and therefore to earn a high reward). Situated agents, then, should have a

broader definition of cooperation. Learning such a definition requires feedback based on more than agents' final behaviors. We explore such a definition next.

8.4 Experiment 2

In Experiment 1, we showed that AToM can recognize agents' intended cooperation in the stag-hunt game using a dataset presented by Shum et al. (2019) and compared AToM's performance to those of Shum et al.'s Bayesian model and human participants. However, the dataset used included only nine scenarios, used a very small (5x7) grid, and was guaranteed to include at least one capture at the third timestep. Furthermore, as discussed in section 8.3.4, testing the dataset required the assumption that all successful cooperation was intentional and that all intended cooperation was successful.

In this experiment, we create a new stag-hunt dataset, inspired by Shum et al. (2019). We relax several of Shum et al.'s assumptions and show that AToM can still recognize agents' intent to cooperate most of the time. Because the hunters in our dataset reason about their compatriots when deciding whether to cooperate, we explore the use of second-order ToM reasoning (i.e., reasoning about hunters' reasoning about others) to improve AToM's performance on the dataset, but find no significant differences.

8.4.1 Dataset. We created a new dataset of stag-hunt simulations (see Appendix C). Consistent with Shum et al. (2019), we simulated three timesteps on a grid world with three hunters, two stags, and two hares. Unlike Shum et al., we varied the size and density of the grid world (7x7 and 9x9; medium and low density). A total of 30 simulations were conducted for each size and density. Note that, because hunters are more spread out than in Shum et al.'s simulations, it is not guaranteed that a hare and/or stag will be captured at step 3 of each simulation. Capture is also not limited to the step 3; it is possible for a hare or stag to be captured earlier (see Table 9).

Table 9. Number of targets captured at each timestep across 30 simulations for each map type.

	Step1	*Step2*	*Step3*	*Total*
7x7/med density	Rabbits: 6 Stags: 0	Rabbits: 14 Stags: 1	Rabbits: 5 Stags: 2	Rabbits: 25 Stags: 3
7x7/ low density	Rabbits: 13 Stags: 0	Rabbits: 7 Stags: 3	Rabbits: 4 Stags: 3	Rabbits: 24 Stags: 6
9x9/med density	Rabbits: 8 Stags: 0	Rabbits: 7 Stags: 0	Rabbits: 10 Stags: 0	Rabbits: 25 Stags: 0
9x9/low density	Rabbits: 5 Stags: 1	Rabbits: 8 Stags: 0	Rabbits: 6 Stags: 2	Rabbits: 19 Stags: 3

Furthermore, our hunters reassess their goals before each timestep based on their beliefs about other hunters' goals. That is, a hunter's intent to cooperate changes, depending on whether or not it believes other hunters are cooperating with it. This means that a hunter's intent to cooperate can be unsuccessful because intentions are mismatched: if hunterA believes that hunterB is cooperating with it, but hunterB is actually acting alone, hunterA will not be successful in cooperating with hunterB. The hunters' decision-making process is described next.

8.4.1.1 Agent simulations. In order to make inferences about compatriots' plans to cooperate, each hunter is equipped with a naïve ToM. Before deciding on a movement, it simulates five scenarios: 1) hunterA and hunterB are cooperating; 2) hunterA and hunterC are cooperating; 3) hunterB and hunterC are cooperating; 4) all three hunters are cooperating; 5) none of the hunters are cooperating. In each scenario, it simulates the actions of all hunters given the goal in that scenario. It assumes that all hunters will try to capture the nearest target that meets their goal (i.e., a hunter acting alone will try to capture the closest hare, while a hunter cooperating with another hunter will try to capture the closest stag between them). It then chooses the scenario with the highest utility for itself (i.e., in which it earns

the most points), and assumes that the other hunters have the goals that correspond to that scenario[42].

8.4.1.2 Encoding and representations. All scenarios were automatically encoded from the simulations described above. The representation scheme did not differ from the representations in Experiment 1 (see section 8.3.1.1), except for intent to cooperate. Because cooperation is not guaranteed to be reciprocal in our dataset, an intent to cooperate (or not) was encoded with respect to each hunter at each timestep. The assumptions that led to that intent were encoded, as well. Note that, unlike in Experiment 1, cooperation is represented as a goal, rather than as a ground truth event.

8.4.1.3 Case generation. As in Experiment 1, four cases were generated for each simulation (see section 8.3.1.2): a testing case for each timestep, which includes only information about movement and capture, and a training case, which includes movement and capture information from all timesteps plus information about each agent's assumptions and intentions.

For second-order ToM reasoning, an additional case was generated for each hunter at each timestep. This case included only relations that involved that hunter (i.e., its movements and other agents' movements with respect to it). For example, hunterA's case would include that hunterA moved closer to hunterB and that hunterC moved away from hunterA, but not that hunterB and hunterC moved toward each other. How these cases were used for second-order ToM reasoning is described in the following section.

8.4.2 Training and testing. Training and testing proceeded using a variant of 10-fold cross validation, analogous to the variant of leave-one-out cross validation used in Experiment 1[43] (see section

[42] This further adds to the naivete of the hunters' ToM model, as they do not consider the utility of an assumption for other agents.

[43] Because Experiment 1 only contained nine examples, 10-fold cross-validation was not possible.

a)
```
(actualGoal hunterA (cooperateWith hunterA hunterB))
```
b)
```
(actualGoal hunterC (huntAlone hunterC))
```

Figure 19. Candidate inferences for intent to (a) cooperate and (b) work alone in Experiment 2. Unlike Experiment 1, hunters' goals are independent of each other.

8.3.2). As in Experiment 1, AToM was trained on complete cases and tested at each timestep, as described above. During testing, AToM's prediction for the intent of each hunter during that timestep was analyzed individually (cf. Experiment 1, where intent to cooperate was analyzed in dyads). Example candidate inferences are shown in Figure 19.

Prior evidence (e.g., de Weerd, Verbugge & Verheij, 2014) has shown that second-order ToM reasoning (i.e., reasoning about another agent's reasoning about others) improves simulated agents' performance on tasks in which reasoning about other agents' performance is required[44]. Because the hunters in our simulations have a naïve ToM, we also tested whether second-order ToM reasoning is beneficial for AToM on this intent recognition task. To model second-order ToM, the AToM algorithm was modified to include a round of agent-specific ToM reasoning (see Figure 20). That is, prior to reasoning about the goals of all hunters, AToM inferred the assumptions of each individual hunter. This was accomplished by creating a case of the scenario from each hunter's point of view (see section 8.4.1.3).

Each hunter-specific case was then used for retrieval from AToM's LTM. Candidate inferences about that hunter's assumptions (Figure 21) for each hunter-specific case were added to the original testing

[44] Human studies (e.g.,Meijering, Van Rijn, Taatgen & Verbrugge, 2011; Goodie, Doshi & Young, 2012) suggest that people also benefit from second-order ToM reasoning on some tasks.

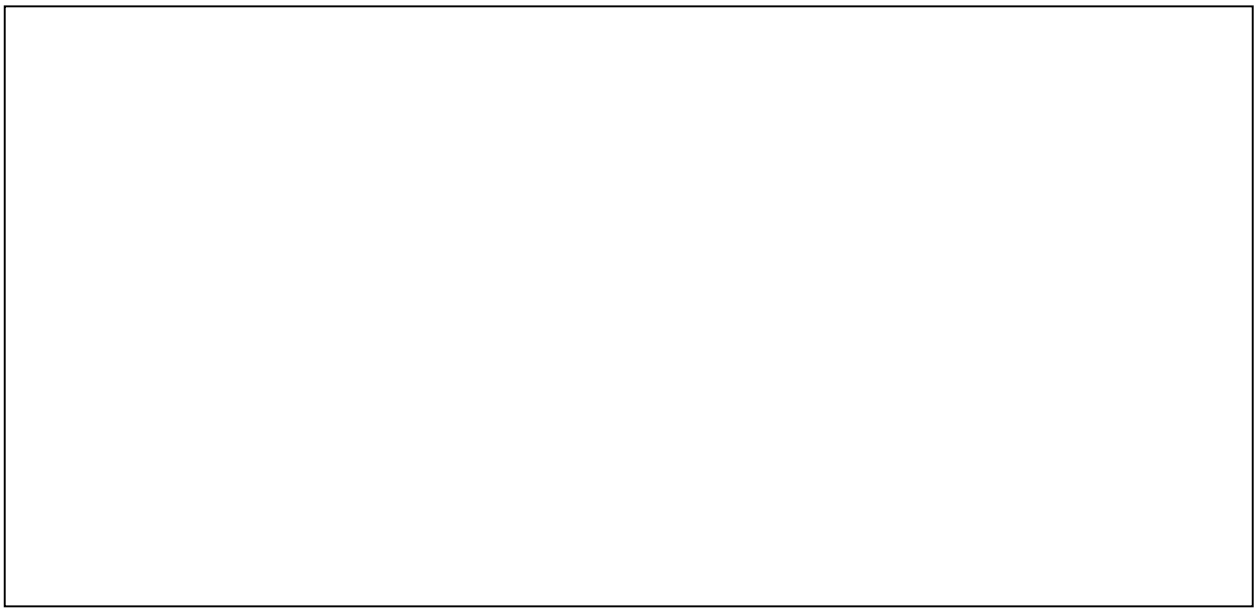

Figure 20. Pseudocode for modified (second order) AToM algorithm.

case. We then used these modified test cases (including information about all agents, plus hunters' inferred assumptions) to retrieve from AToM's LTM again. Candidate inferences about hunters' intents from the second retrieval were analyzed.

8.4.3 Results. Experiments were conducted independently for each of four map types: 7x7 medium density, 7x7 low density, 9x9 medium density, 9x9 low density. AToM's performance for each map type, using both first and second-order ToM reasoning is shown in Table 10. Note that baseline accuracy is 33%, as each hunter can have the goal to cooperate with each of the other two hunters or hunt alone. Using first-order ToM, AToM's accuracy was significantly higher than chance (one-sample

a)
```
(assumes hunterB
      (assumedGoal hunterA (cooperateWith hunterA hunterB)))
```
b)
```
(assumes hunterA
      (assumedGoal hunterC (huntAlone hunterC)))
```

Figure 21. Examples of assumption CIs used to modify probe for second order ToM reasoning.

Table 10. AToM accuracy at each timestep across 30 simulations for each map type. Mean accuracy and standard deviation are reported. Values not statistically above chance are marked *ns*.

	First Order ToM			*Second Order ToM*		
	Step1	*Step2*	*Step3*	*Step1*	*Step2*	*Step3*
7x7/med density	0.46(0.16)	0.64(0.21)	0.57(0.24)	0.46(0.16)	0.66(0.21)	0.51(0.27; *ns)*
7x7/ low density	0.44(0.17; *ns*)	0.52(0.21)	0.60(0.13)	0.44(0.17; *ns*)	0.52(0.21)	0.60(0.13)
9x9/med density	0.79(0.19)	0.64(0.22)	0.68(0.17)	0.80(0.19)	0.67(0.22)	0.67(0.17)
9x9/low density	0.66(0.20)	0.58(0.15)	0.63(0.19)	0.67(0.19)	0.58(0.15)	0.63(0.19)

two tailed t-test, p<.05) for all steps on all map types, except the first step on the 7x7 low density map (p=.07). Using second-order ToM, the third step of the 7x7 medium density map also fell below chance (p=.07).

A two-way ANOVA found a main effect of map type, but not step, in the first-order ToM experiment. Tukey's Honestly Significant Difference (HSD) comparison showed that AToM performed significantly better on the 9x9 medium density map than any of the other three maps. It performed better on the 7x7 medium density map than the 7x7 low density map, and better on the 9x9 low density map than the 7x7 low density map (all p<.05). There was no signficant difference in performance between the 7x7 medium density map and the 9x9 low density map.

There were no significant differences in performance between the first-order and second-order AToM models (two tailed paired samples t-test, p=.92).

8.4.4 Discussion. Using Shum et al.'s (2019) stag-hunt dataset, AToM's ability to recognize intent to cooperate between hunters did not differ from Shum et al.'s Bayesian model or human participants. In fact, at the third timestep, AToM made just one incorrect prediction. That dataset,

however, was small (only nine scenarios) and analysis required making several key assumptions. Among these: that any intended cooperation would be successfully completed within three timesteps, that cooperation is always reciprocal (i.e., that agents always know whether others are cooperating with them), and that once an agent decides to cooperate, it does not change its mind.

For this experiment, we relaxed these assumptions in a new, expanded stag-hunt dataset. The new dataset was intended to be more difficult across several dimensions. First, the maps were bigger and more complex. This meant that agents' movements were less restricted. Thus, successful capture was not guaranteed in three timesteps (but could happen earlier; see Table 9). Second, hunters' goals were set independently, rather than in groups, leading to the possibility of non-reciprocal or failed cooperation. Hunters also reevaluated their goals at each timestep, based on the behaviors of other agents. This led to goals changing between time steps in approximately 15% of all scenarios.

AToM's performance suggests that this dataset is, in fact, more difficult than Shum et al.'s (2019). The highest accuracy AToM achieved was 79% in the first step of the medium density 9x9 map[45]. This is counterintuitive on several fronts. We expected the larger 9x9 map to be more difficult than the smaller, 7x7 map. Furthermore, performance on the Shum et al. dataset suggested a trend toward better scores on later timesteps[46]. AToM's performance on our dataset suggests no such trend. However, it is unclear whether this is because our hunters could change their goals, because success was not guaranteed on our maps, or for another reason. Varying hunters' decision-making abilities may shed some light on this question.

---

[45] Recall that AToM averaged 88% accuracy across timesteps on the original dataset.
[46] No differences were significant on that dataset due, in part, to its small size.

The difference in performance between maps appears to be driven by the models of the agent behavior that AToM was able to build in each. Specifically, the number of scenarios with cooperation goals significantly differed between maps, $X^2(9,360) = 32.55$, $p<.001$. Tellingly, 69/90 scenarios on the 9x9 medium density map did not contain any cooperation goals (between 35 and 48 scenarios did not contain cooperation goals on the other map types). It remains to be seen whether the difference is a product of the map type or due to sampling bias. Similarly, it is not clear whether AToM's performance across map types would change given more uniform proportions of cooperation goals.

Finally, because the hunters in our dataset had a naïve ToM, we explored whether second-order ToM—reasoning about agents' beliefs about other agents' internal states—would improve AToM's ability to recognize intent to cooperate. We did not find evidence to support such a conclusion. We have identified three possibilities for why second-order ToM was not helpful here: 1) the second-order AToM model is a poor model of second-order ToM reasoning; 2) this version of the stag-hunt game does not benefit from second-order ToM reasoning (e.g., because the agents' ToM is too naïve or the task itself is too simple); 3) second-order ToM is not useful for multi-agent intent recognition more broadly.

Further research will be necessary to determine which of these possibilities is causing the lack of difference between first-order and second-order ToM predictions. For example, while some studies show that adult humans can use second-order ToM reasoning to improve performance on second-order ToM games, they need to be trained explicitly in order to do so (Meijering, Van Rijn, Taatgen, & Verbrugge, 2011). Without instruction, participants tended to rely on first-order ToM reasoning, and only shift toward second-order ToM after several rounds of play with an opponent who consistently applies ToM in her gameplay (Hedden & Zhang, 2002). This suggests that AToM may need to be trained for second-order ToM reasoning independently of first-order ToM reasoning. Alternatively, it suggests

that, when agents' ToM is too naïve, second-order ToM may be of limited benefit. Given how naïve our hunters' ToM is[47], it is possible that second-order ToM would not be beneficial at all. Testing people's predictions—and strategies—on our dataset would help answer this question.

8.5 Conclusion

In this chapter, we tested AToM's ability to recognize the intent to cooperate among hunters in the stag-hunt game. AToM performed comparably to a Bayesian model and humans on a simple version of the game. When assumptions were relaxed and the game made more complex, AToM continued to perform well. We tested whether second-order ToM reasoning would further improve performance on the more complex dataset but found no significant differences between the first-order and second-order models.

[47] Recall that hunters simulate forward at each timestep. For each simulation, they test assumptions about other hunters' goals to cooperate. However, they always choose the goal that corresponds with the simulation that results in the highest reward for themselves.

9 Conclusions and Future Work

In this dissertation, I have presented the Analogical Theory of Mind (AToM) model of theory of mind (ToM) reasoning and development. AToM was used as a model of children's ToM learning and development (chapters 4, 5, and 6) and as part of the reasoning processes of simulated agents (chapters 7 and 8). In this chapter, I discuss how the findings of these experiments relate to the claims of this dissertation, as well as open questions and directions for future work.

9.1 Claims Revisited

This dissertation began with two central claims:

1) Human ToM reasoning and development occur via analogical processes.

2) The same processes can be used by simulated agents to improve their ToM reasoning.

Each of these claims was tested via the AToM model. Specifically, with regard to claim 1, we considered the following related claims:

3) Human ToM reasoning occurs specifically via structure-mapping processes.

4) Human ToM reasoning occurs in working memory when possible. Retrieval from long-term memory occurs when triggered by the environment.

5) Human ToM reasoning is driven by analogical inferences.

In chapter 4, these claims were tested directly by modeling a study (Hoyos et al., 2015) in which children, who failed ToM pretests, were trained on three structurally similar stories via a repetition-break paradigm (i.e., true belief, true belief, false belief). Children who heard stories that shared more overlapping structure performed better on posttests than did children who heard stories that shared less structure.

Stories were represented as structured cases and were passed to AToM's WM one at a time. The

true belief stories that shared more structure formed a generalization in WM. When the false belief story, which shared structure with the first two but had a different conclusion, was presented to the model, AToM aligned it to the generalization and made a prediction consistent with the true belief cases, based on a candidate inference. Feedback that this was incorrect led to a search for explanation in LTM. This retrieval allowed the model to later answer several posttest questions.

On the other hand, no generalization was formed when the true belief cases shared less structure. Thus, there was no expectation that the false belief story should have the same conclusion as the true belief stories, and no search for explanation when it did not. This led the model to answer fewer posttest questions correctly.

The fact that AToM not only modeled the differences in children's performance in this experiment, but also made testable predictions about the children's learning, provides evidence that support claims 3, 4, and 5. These claims were further supported by chapter 5, in which AToM was used to model a training study (Hale & Tager-Flusberg, 2003). This study showed that children could improve ToM reasoning by learning a complex grammatical form, the sentential complement. Using the same processes as in chapter 4, AToM modeled children's performance here, too. This suggests that claims 3, 4, and 5 apply to ToM reasoning broadly (i.e., not only when learning from structured stories).

Chapter 6 expanded further on these claims, and provided evidence that pretend play—which has been linked with ToM learning in children (see Weisberg, 2015)—can be viewed as an analogical process, and that failures in pretense can be explained by failures in generating and accepting candidate inferences. Thus, through pretend play, children learn the processes necessary for successful ToM. Specifically, chapter 6 tested the claim that:

6) Successful pretend play requires the ability to reason analogically, including generating and

accepting appropriate candidate inferences.

Chapter 6 modeled two studies (Fein, 1975; Onishi et al., 2007) in which young children failed at pretense. Pretend play was modeled as analogy between a pretense scenario and a schema of the real-world event that the pretense mimicked. Anchoring/familiarization trials led to interim generalizations between the anchor and the schema. This interim generalization was used for pretense in trials that followed.

In both studies, failures could be explained as additional and/or more difficult candidate inferences between the pretense scenario and the schema. Interim generalizations following anchoring/familiarization trials lessened the number of candidate inferences, making pretense possible again. Thus, modeling pretense as analogy explains failures in pretense—and suggests that learning to accept and generate appropriate candidate inferences is the link between pretend play and ToM reasoning.

The second portion of this dissertation dealt with claim 2, that the processes of AToM can also improve simulated agents' reasoning. This was tested in conjunction with the following related claim:

7) ToM reasoning, specifically via AToM, allows simulated agents to reason about the internal states of others even when those internal states are not inspectable.

Claim 7 was tested in chapters 7 and 8. In chapter 7, AToM's ability to recognize the goals of agents in a Minecraft farming task was compared against a state-of-the-art goal recognition system (PANDA; Holler et al., 2018). On a standard goal recognition task—when the systems were tested using the same planner data made available to the agent—PANDA and AToM performed comparably. However, when tested on data from the agent's controller or external observations, AToM significantly outperformed PANDA. This suggests that AToM is a better choice for goal recognition in situations when planner data

is unavailable, such as when reasoning about an adversarial agent or a person. Thus, this set of experiments directly supports claim 7.

Chapter 8 further supports claim 7. In this chapter, AToM's ability to recognize agents' intent to cooperate in a stag-hunt game was tested. Stag-hunt is a prisoner's dilemma-style game, in which agents can choose to cooperate to attempt to catch a high-value target or act alone to attempt to catch a lower-value target. We first tested AToM's ability to recognize cooperation in this game using Shum et al.'s (2019) dataset of nine stag-hunt scenarios. When compared to the predictive accuracies of humans and a Bayesian model (both as reported by Shum et al., 2019), AToM's performance did not differ.

We then extended the dataset and relaxed several of its assumption. In the new dataset, maps were bigger and less dense (i.e., agents' movements were less limited) and agents chose whether or not to cooperate independently of each other. In most cases AToM continued to recognize agents' intent to cooperate significantly above chance but was well below ceiling. We tested whether extending AToM to second-order ToM would further improve its performance but did not find this to be the case. Further exploring the utility of second-order ToM and other directions for future work are discussed next.

9.2 Future Work

The findings presented in this dissertation point to several directions for future work. These range from expanding AToM as a cognitive model to using it more broadly in applied domains. I discuss several of these directions here.

9.2.1 Modeling second-order ToM. In chapter 8 of this dissertation, we expanded AToM to second-order ToM reasoning. However, the expanded model was based on theoretical accounts from the literature and our intuitions about how second-order ToM might work; it was not based on a tested cognitive model of second-order ToM. Developing and testing such a model is one clear direction for

future work.

There are several ways to test second-order AToM as a cognitive model. One is through modeling reasoning and behaviors in adversarial games, such as those reported by Hedden and Zhang (2002), Meijering et al. (2011), and Goodie et al. (2012). These studies consider different ToM reasoning strategies used by people against various types of opponents in strategic adversarial games. Importantly, the games used in these studies are relatively simple, zero-sum turn-based games. Difficulty comes from reasoning about the opponent's future actions, rather than from the game itself. Modeling differences in first-order and second-order ToM reasoning during such games may shed light on both when people use one versus the other, and on how performance in overall gameplay is affected. A second-order ToM model based on people's reasoning in such situations is also likely to help simulated agents in adversarial gameplay better than a first-order model alone.

However, gameplay is only one situation in which second-order ToM is used, and a somewhat limited one at that. More often, second-order ToM is used during communication. This has been studied in terms of deception, or lying. For example, Talwar, Gordon, and Lee (2007) found that children's ability to maintain a lie correlated with their second-order ToM reasoning skills. Similarly, Sullivan, Winner, and Hopfield (1995) found that children were only able to distinguish lies from jokes after gaining second-order ToM proficiency. In adults, deception has been studied in its own right (see Hyman, 1989), as has people's ability to recognize deception (e.g., Bond & DePaulo, 2006). Modeling these phenomena is another direction of future work.

9.2.2 Interaction between simulated agents. One of the claims of this dissertation was that AToM can aid simulated agents in reasoning about other agents. We tested this claim in terms of inference—can the model accurately predict agents' actions? However, a more convincing test might be

whether an agent that includes AToM as part of its general reasoning can make use of the inferences AToM provides. For example, in chapter 7, AToM was used for goal recognition of a Minecraft farmer. A natural extension of this task is placing a second farmer on the map and endowing one or both with AToM as part of its reasoning. If tasked with cooperation, Minecraft agents that use AToM to reason about each other's goals and beliefs should be more successful than those that act independently or use another approach to reason about each other. If tasked with competition, agents that use AToM should similarly be more successful than their counterparts. Because deception can play a role in competition, second-order ToM may prove useful here, too.

9.2.3 Interaction with people. Interacting with people poses challenges beyond those of interacting with other simulated agents. Because people rely on ToM reasoning when interacting amongst themselves, interaction without ToM often feels unnatural. Thus, giving simulated agents—whether they be assistive robots or smartphone virtual assistants—is likely to improve the user experience. Indeed, Hiatt, Harrison, and Trafton (2011) found that people preferred working with robot teammates that used ToM reasoning when giving instructions to working with those that did not have ToM. Because AToM is, first and foremost, a cognitive model of human ToM reasoning, using it for such agents' ToM is a promising direction for future work.

References

Albrecht, S. V., & Stone, P. (2018). Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, *258*, 66-95.

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., ... & Traum, D. R. (1995). The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, *7*(1), 7-48.

Arslan, B., Taatgen, N., & Verbrugge, R. (2013). Modeling developmental transitions in reasoning about false beliefs of others. *In Proceedings of the 12th International Conference on Cognitive Modeling*, Ottawa: Carleton University (pp. 77-82).

Arslan, B., Taatgen, N. A., & Verbrugge, R. (2017). Five-year-olds' systematic errors in second-order false belief tasks are due to first-order theory of mind strategy selection: a computational modeling study. *Frontiers in Psychology*, *8*, 275.

Bach, T. (2011). Structure-mapping: Directions from simulation to theory. *Philosophical Psychology*, *24*(1), 23-51.

Bach, T. (2014). A Unified Account of General Learning Mechanisms and Theory-of-Mind Development. *Mind & Language*, *29*(3), 351-381.

Baddeley, A. (2007). *Working memory, thought, and action* (Vol. 45). OUP Oxford.

Baillargeon, R., Buttelmann, D., & Southgate, V. (2018). Invited commentary: Interpreting failed replications of early false-belief findings: Methodological and theoretical considerations. *Cognitive Development*, 46, 112-124.

Baker, C. L., Goodman, N. D., & Tenenbaum, J. B. (2008). Theory-based social goal inference. In *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society* (pp. 1447-1452).

Baker, C., Saxe, R., & Tenenbaum, J. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 33, No. 33).

Baker, C. L., & Tenenbaum, J. B. (2014). Modeling human plan recognition using Bayesian theory of mind. In G. Sukthankar, G. Geib, C. Bui, H. H. Pynadath, & R. P. Goldman (Eds.) *Plan, activity, and intent recognition: Theory and practice*, 177-204.

Baldwin, D. A., & Moses, L. J. (1994). The mindreading engine: Evaluating the evidence for modularity. *Cahiers de Psychologie Cognitive/Current Psychology of Cognition,* 13(5), 553–560.

Ball, L. H., Besozzi, M., Ball, S. E., Anderson, L., & Geye, T. (2013). Strategies for deploying theory of mind in adults: theory vs simulation is not either/or. *The Irish Journal of Psychology*, *34*(2), 81-92.

Baron-Cohen, S. (1994). How to build a baby that can read minds: Cognitive mechanisms in mindreading. *Cahiers de Psychologie Cognitive/Current Psychology of Cognition, 13*(5), 513–552.

Baron-Cohen, S., Jolliffe, T., Mortimore, C., & Robertson, M. (1997). Another advanced test of theory of mind: Evidence from very high functioning adults with autism or Asperger syndrome. *Journal of Child Psychology and Psychiatry*, 38(7), 813-822.

Bartsch, K., & Wellman, H. M. (1995). *Children talk about the mind*. Oxford university press.

Bello, P., & Cassimatis, N. (2006). Developmental accounts of theory-of-mind acquisition: Achieving clarity via computational cognitive modeling. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 28, No. 28).

Bercher, P., Keen, S., & Biundo, S. (2014). Hybrid planning heuristics based on task decomposition graphs. In *Seventh Annual Symposium on Combinatorial Search*, 35-43.

Bianco, F., & Ognibene, D. (2019). Functional advantages of an adaptive Theory of Mind for robotics: a review of current architectures. In *Proceedings of 11th Computer Science and Electronic Engineering*

Conference (CEEC). IEEE.

Blass, J. and Forbus, K.D. (2015). Moral Decision-Making by Analogy: Generalizations vs. Exemplars. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin, Texas.

Bond Jr, C. F., & DePaulo, B. M. (2006). Accuracy of deception judgments. *Personality and Social Psychology Review*, 10(3), 214-234.

Bonini, L. (2017). The extended mirror neuron network: Anatomy, origin, and functions. *The Neuroscientist*, *23*(1), 56-67.

Bratman, M. (1987). *Intention, plans, and practical reason* (Vol. 10). Cambridge, MA: Harvard University Press.

Brooks, C., & Szafir, D. (2019). Building Second-Order Mental Models for Human-Robot Interaction. *arXiv preprint arXiv:1909.06508*.

Brown, R. (1973). *A first language: The early stages*. Harvard University Press.

Burstein, M. H. (1983). A Model of Learning by Incremental Analogical Reasoning and Debugging. In *Proceedings of AAAI*.

Carberry, S. (2001). Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, *11*(1-2), 31-48.

Cassimatis, N. (2005). Integrating cognitive models based on different computational methods. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Christie, S., Gentner, D., Call, J., & Haun, D. B. M. (2016). Sensitivity to relational similarity and object similarity in apes and children. *Current Biology*, 26(4), 531-535.

Cisek, P, Kalaska, J. F. 2004. Neural correlates of mental rehearsal in dorsal premotor cortex. *Nature*.

431:993–6.

Clementini, E.; Felice, P. D.; and Hernandez, D. (1997) Qualitative representation of positional information. *Artificial Intelligence* 95(2):317–356.

Cowan, N. (2015). George Miller's magical number of immediate memory in retrospect: Observations on the faltering progression of science. *Psychological Review*, 122(3), 536.

Cox, M. T., & Kerkez, B. (2006). Case-based plan recognition with novel input. *Control and Intelligent Systems,* 34(2), 96-104.

Dehghani, M., Tomai, E., Forbus, K., Iliev, R., Klenk, M. (2008). MoralDM: A Computational Modal of Moral Decision-Making. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*. Washington, D.C

Delafontaine, M.; Cohn, A. G.; and Van de Weghe, N. (2011) Implementing a qualitative calculus to analyse moving point objects*. Expert Systems with Applications* 38(5):5187–5196.

de Villiers, J. G., & de Villiers, P. A. (2009). Complements enable representation of the contents of false beliefs: The evolution of a theory of theory of mind. In *Language Acquisition* (pp. 169-195). Palgrave Macmillan, London.

de Villiers, J., Hobbs, K., & Hollebrandse, B. (2014). Recursive complements and propositional attitudes. In *Recursion: Complexity in cognition* (pp. 221-242). Springer, Cham.

de Villiers, J. G., & Pyers, J. (1997). Complementing cognition: The relationship between language and theory of mind. In *Proceedings of the 21st Annual Boston Conference on Language Development*.

de Villiers, J. G., & Pyers, J. E. (2002). Complements to cognition: A longitudinal study of the relationship between complex syntax and false-belief-understanding. *Cognitive Development*, *17*(1), 1037-1060.

Devin, S., & Alami, R. (2016). An implemented theory of mind to improve human-robot shared plans

execution. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE.

de Weerd, H., Verbrugge, R., & Verheij, B. (2014). Agent-based models for higher-order theory of mind. In *Advances in Social Simulation* (pp. 213-224). Springer, Berlin, Heidelberg.

Di Pellegrino, G., Fadiga, L., Fogassi, L., Gallese, V., & Rizzolatti, G. (1992). Understanding motor events: a neurophysiological study. *Experimental Brain Research*, 91(1), 176-180.

Diessel, H., & Tomasello, M. (2000). The development of relative clauses in spontaneous child speech. *Cognitive Linguistics*, 11(1/2), 131-152.

Dörrenberg, S., Rakoczy, H., & Liszkowski, U. (2018). How (not) to measure infant Theory of Mind: Testing the replicability and validity of four non-verbal measures. *Cognitive Development*, *46*, 12-30.

Doumas, L. A., Hummel, J. E., & Sandhofer, C. M. (2008). A theory of the discovery and predication of relational concepts. *Psychological Review*, 115(1), 1.

Eliasmith, C., & Thagard, P. (2001). Integrating structure and meaning: A distributed model of analogical mapping. *Cognitive Science*, 25(2), 245-286.

E-Martin, Y., R-Moreno, M. D., & Smith, D. E. (2015) A fast goal recognition technique based on interaction estimates. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 761-768.

Epley, N., Keysar, B., Van Boven, L., & Gilovich, T. (2004). Perspective taking as egocentric anchoring and adjustment. *Journal of Personality and Social Psychology*, *87*(3), 327.

Epley, N., Morewedge, C. K., & Keysar, B. (2004). Perspective taking in children and adults: Equivalent egocentrism but differential correction. *Journal of Experimental Social Psychology*, *40*(6), 760-768.

Erol, K., Hendler, J., & Nau, D. S. (1994). HTN planning: Complexity and expressivity. In *Proceedings of*

the 8th AAAI Conference on Artificial Intelligence, 1123-1128.

Fagan, M., & Cunningham, P. (2003). Case-based plan recognition in computer games. In *Proceedings of the International Conference on Case-Based Reasoning*. Berlin, Heidelberg.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1986). *The structure-mapping engine* (Vol. 1275). Department of Computer Science, University of Illinois at Urbana-Champaign.

Fein, G. G. (1975). A transformational analysis of pretending. *Developmental Psychology*, 11(3), 291.

Flavell, J. H. (1999). Cognitive development: Children's knowledge about the mind. *Annual Review of Psychology*, 50(1), 21-45.

Flavell, J. H. (2004). Theory-of-mind development: Retrospect and prospect. *Merrill-Palmer Quarterly (1982-)*, 274-290.

Forbus, K. (2001). Exploring analogy in the large. In Gentner, D., Holyoak, K., and Kokinov, B. (Eds.) *Analogy: Perspectives from Cognitive Science*. MIT Press.

Forbus, K., Ferguson, R., & Gentner, D. (1994). Incremental structure-mapping. In *Proceedings of the Cognitive Science Society*.

Forbus, K. D., Ferguson, R. W., Lovett, A., & Gentner, D. (2017). Extending SME to handle large-scale cognitive modeling. *Cognitive Science*, *41*(5), 1152-1201.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19(2), 141-205.

Forbus, K.D. & Hinrichs, T. (2017). Analogy and Qualitative Representations in the Companion Cognitive Architecture. *AI Magazine*.

Forbus, K. D., Liang, C., & Rabkina, I. (2017). Representation and computation in cognitive models. *Topics in Cognitive Science*, *9*(3), 694-718.

French, R. M. (1995). *The subtlety of sameness: A theory and computer model of analogy-making*. MIT press.

French, R., & Hofstadter, D. (1992). Tabletop: An emergent, stochastic model of analogy-making. In *Proceedings of the 13th Annual Conference of the Cognitive Science Society*.

Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, *2*(12), 493-501.

Gatsoulis, Y., Alomari, M., Burbridge, C., Dondrup, C., Duckworth, P., Lightbody, P., ... & Cohn, A. G. (2016). QSRlib: A software library for online acquisition of Qualitative Spatial Relations from Video.

Geib, C., & Goldman, R. (2011). Recognizing plans with loops represented in a lexicalized grammar. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*(2), 155-170.

Gentner, D., & Clement, C. (1988). Evidence for relational selectivity in the interpretation of analogy and metaphor. In *Psychology of Learning and Motivation* (Vol. 22, pp. 307-358). Academic Press.

Gentner, D., & Forbus, K. (2011). Computational models of analogy. *WIREs Cognitive Science*, 2. 266-276.

Gentner, D., & Medina, J. (1998). Similarity and the development of rules. *Cognition*, 65(2-3), 263-297.

Gentner, D. & Rattermann, M.J. (1991). Language and the career of similarity. In S.A. Gelman & J.P. Byrnes (Eds.), *Perspectives on thought and language: Interrelations in development*, (pp. 225-277). London: Cambridge University Press.

Gentner, D., & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. *Cognitive Science*, 10(3), 277-300.

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*.

University of Chicago Press.

Goldberg, A. E. (2003). Constructions: a new theoretical approach to language. *Trends in Cognitive Sciences*, 7(5), 219–224.

Goldman, A. I. (2006). *Simulating minds: The philosophy, psychology, and neuroscience of mindreading*. Oxford University Press.

Goldman, A. I. (2012). Theory of mind. *The Oxford handbook of philosophy of cognitive science*, *1*.

Goldman, A. I., & Sebanz, N. (2005). Simulation, mirroring, and a different argument from error. *Trends in Cognitive Sciences*, *9*(7), 320.

Goldstein, T. R., & Lerner, M. D. (2018). Dramatic pretend play games uniquely improve emotional control in young children. *Developmental Science*, *21*(4).

Goodie, A. S., Doshi, P., & Young, D. L. (2012). Levels of theory-of-mind reasoning in competitive games. *Journal of Behavioral Decision Making*, 25(1), 95-108.

Goodman, N. D., Baker, C. L., Bonawitz, E. B., Mansinghka, V. K., Gopnik, A., Wellman, H., ... & Tenenbaum, J. B. (2006). Intuitive theories of mind: A rational approach to false belief. In *Proceedings of the Twenty-eighth Annual Conference of the Cognitive Science Society* (Vol. 6). Vancouver: Cognitive Science Society.

Gopnik, A., & Wellman, H. M. (1994). The theory theory. *Mapping the mind: Domain specificity in cognition and culture*, 257.

Görür, O. C., Rosman, B. S., Hoffman, G., & Albayrak, S. (2017). Toward integrating Theory of Mind into adaptive decision-making of social robots to understand human intention.

Greiner, R. (1988). Learning by understanding analogies. *Artificial Intelligence*, 35(1), 81-125.

Grosz, B. J., & Kraus, S. (1999). The evolution of SharedPlans. In *Foundations of Rational Agency* (pp.

227-262). Springer, Dordrecht.

Gust, H., Kühnberger, K. U., & Schmid, U. (2006). Metaphors and heuristic-driven theory projection (HDTP). *Theoretical Computer Science*, *354*(1), 98-117.

Hale, C. M., & Tager-Flusberg, H. (2003). The influence of language on theory of mind: A training study. *Developmental Science*, *6*(3), 346-359.

Happé, F. G. (1994). An advanced test of theory of mind: Understanding of story characters' thoughts and feelings by able autistic, mentally handicapped, and normal children and adults. *Journal of Autism and Developmental Disorders*, 24(2), 129-154.

Harris, P. (1996). Desires, beliefs, and language. In P. Carruthers & P.K. Smith (Eds.) *Theories of theories of mind* (pp. 200-220). Cambridge University Press.

He, Z., Bolz, M., & Baillargeon, R. (2011). False-belief understanding in 2.5-year-olds: evidence from violation-of-expectation change-of-location and unexpected-contents tasks. *Developmental Science*, *14*(2), 292-305.

Hedden, T., & Zhang, J. (2002). What do you think I think you think?: Strategic reasoning in matrix games. *Cognition*, 85(1), 1-36.

Heal, J. (1995). How to think about thinking. In M. Davies & T. Stone (Eds.) *Mental simulation: Evaluations and applications* (pp. 33-52). Blackwell Publishers Ltd.

Heal, J. (1996). Simulation, theory, and content. In P. Carruthers & P.K. Smith (Eds.) *Theories of theories of mind* (pp. 75-89). Cambridge University Press.

Heyes, C. (2014). False belief in infancy: A fresh look. *Developmental Science*, *17*(5), 647-659.

Hiatt, L. M., Harrison, A. M., & Trafton, J. G. (2011). Accommodating human variability in human-robot teams through theory of mind. In *Proceedings of the Twenty-Second International Joint Conference*

on Artificial Intelligence.

Hiatt, L. M., & Trafton, J. G. (2010). A cognitive model of theory of mind. In *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 91-96). Philadelphia, PA: Drexel University.

Hiatt, L. M., & Trafton, J. G. (2015). Understanding second-order theory of mind. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts* (pp. 167-168).

Hofmann, S. G., Doan, S. N., Sprung, M., Wilson, A., Ebesutani, C., Andrews, L. A., ... & Harris, P. L. (2016). Training children's theory-of-mind: A meta-analysis of controlled studies. *Cognition*, *150*, 200-212.

Hofstadter, D. R., & Mitchell, M. (1995). The Copycat project: A model of mental fluidity and analogy-making. *Advances in Connectionist and Neural Computation Theory*, 2, 205-267.

Höller, D., Behnke, G., Bercher, P., & Biundo, S. (2018). Plan and goal recognition as HTN planning. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 466-473). IEEE.

Holt, A., Bichindaritz, I., Schmidt, R., & Perner, P. (2005). Medical applications in case-based reasoning. *The Knowledge Engineering Review*, 20(3), 289-292.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13(3), 295-355.

Hoyos, C., Horton, W., & Gentner, D. (2015). Analogical comparison aids false belief understanding in preschoolers. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society*.

Hoyos, C., Horton, W. S., Simms, N. K., & Gentner, D. (under review). Analogical comparison promotes theory-of-mind development.

Hummel, J. E., & Holyoak, K. J. (1996). LISA: A computational model of analogical inference and schema

induction. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society* (pp. 352-357). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104(3), 427.

Hummel, J. E., Licato, J., & Bringsjord, S. (2014). Analogy, explanation, and proof. *Frontiers in Human Neuroscience*, 8, 867.

Hyman, R. (1989). The psychology of deception. *Annual Review of Psychology*, 40(1), 133-154.

Jennings, N. R. (1993). Specification and implementation of a belief-desire-joint-intention architecture for collaborative problem solving. *International Journal of Intelligent and Cooperative Information Systems*, *2*(03), 289-318.

Johnson, M., Hofmann, K., Hutton, T., & Bignell, D. (2016). The Malmo platform for artificial intelligence experimentation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 4246-4247.

Kahneman, D & Tversky, A. (1982). The simulation heuristic. In D. Kahneman, P. Slovic & A. Tversky (Eds.) *Judgment under uncertainty: Heuristics and biases*, (pp. 201–208). Cambridge: Cambridge University Press.

Kandaswamy, S., Forbus, K., & Gentner, D. (2014). Modeling learning via progressive alignment using interim generalizations. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 36, No. 36).

Kautz, H. (1985). Toward a Theory of Plan Recognition (Technical Report 162). Department of Computer Science. University of Rochester.

Keane, M. T. (1995). On order effects in analogical mapping: Predicting human error using IAM. In

Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society (Vol. 17, p. 449). Psychology Press.

Kerkez, B., & Cox, M. T. (2003). Incremental case-based plan recognition with local predictions. *International Journal on Artificial Intelligence Tools*, 12(04), 413-463.

Keysar, B., Lin, S., & Barr, D. J. (2003). Limits on theory of mind use in adults. *Cognition*, 89(1), 25-41.

Kızıldere, E., Aktan-Erciyes, A., Tahiroğlu, D., & Göksun, T. (2020). A multidimensional investigation of pretend play and language competence: Concurrent and longitudinal relations in preschoolers. *Cognitive Development*, 54, 100870.

Kokinov, B. (1994). A hybrid model of reasoning by analogy. *Advances in Connectionist and Neural Computation Theory*, 2, 247-318.

Kokinov, B., & Petrov, A. (2001). Integrating memory and reasoning in analogy-making: The AMBR model. In D. Gentner, K. J. Holyoak & B. N. Kokinov (Eds.) *The analogical mind: Perspectives from cognitive science*, (pp. 59-124). Cambridge, MA: MIT Press.

Kolodner, J. (1992). An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1), 3-34.

Kolodner, J. (2014). *Case-based reasoning*. Morgan Kaufmann.

Kotovsky, L., & Gentner, D. (1996). Comparison and categorization in the development of relational similarity. *Child Development*, *67*(6), 2797-2822.

Kruger, J., & Gilovich, T. (1999). "Naive cynicism" in everyday theories of responsibility assessment: On biased assumptions of bias. *Journal of Personality and Social Psychology*, 76(5), 743.

Kühberger, A., & Luger-Bazinger, C. (2016). Predicting framed decisions: Simulation or theory?. *Psychology*, *7*(06), 941.

Kulke, L., Reiß, M., Krist, H., & Rakoczy, H. (2018). How robust are anticipatory looking measures of

Theory of Mind? Replication attempts across the life span. *Cognitive Development*, 46, 97-111.

Leake, D. B. (Ed.). (1996). *Case-based reasoning: Experiences, lessons & future directions* (pp. 1063-1069). Menlo Park: AAAI press.

Larkey, L. B., & Love, B. C. (2003). CAB: Connectionist analogy builder. *Cognitive Science*, 27(5), 781-794.

Leslie, A. M. (1987). Pretense and representation: The origins of" theory of mind.". *Psychological Review*, 94(4), 412.

Leslie, A. M. (1994). ToMM, ToBy, and Agency: Core architecture and domain specificity. *Mapping the Mind: Domain Specificity in Cognition and Culture*, 119-148.

Lillard, A. S. (1993). Pretend play skills and the child's theory of mind. *Child Development*, 64(2), 348-371.

Lillard, A. S. (1994). Making sense of pretence. *Children's Early Understanding of Mind: Origins and Development*, 211-234.

Lohmann, H., & Tomasello, M. (2003). The role of language in the development of false belief understanding: A training study. *Child Development*, 74(4), 1130-1144.

Loewenstein, J., & Heath, C. (2009). The Repetition-Break plot structure: A cognitive influence on selection in the marketplace of ideas. *Cognitive Science*, 33(1), 1-19.

Lovett, A. & Forbus, K.D. (2017). Modeling visual problem solving as analogical reasoning. *Psychological Review*, 124 (1): 60.

MacWhinney, B. (2014). *The CHILDES project: Tools for analyzing talk, Volume II: The database*. Psychology Press.

Markman, A. B. (1997). Constraints on analogical inference. *Cognitive Science*, 21(4), 373-418.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*, Henry Holt and Co. Inc., New York, NY, 2(4.2).

McFate, C. (2018) *An Analogical Account of Argument Structure Construction Acquisition and Application*. [Doctoral Dissertation]. Northwestern University, Department of Computer Science.

McFate, C., and Forbus, K. (2016). Analogical Generalization and Retrieval for Denominal Verb Interpretation. In *Proceedings of the 38th Annual Meeting of the Cognitive Science Society*, Philadelphia, PA.

McLure, M.D., Friedman S.E. and Forbus, K.D. (2015). Extending Analogical Generalization with Near-Misses. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas.

Meijering, B., Van Rijn, H., Taatgen, N., & Verbrugge, R. (2011). I do know what you think I think: Second-order theory of mind in strategic games is not that difficult. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Milligan, K., Astington, J. W., & Dack, L. A. (2007). Language and theory of mind: Meta-analysis of the relation between language ability and false-belief understanding. *Child Development*, 78(2), 622-646.

Mitchell, J. P. (2005). The false dichotomy between simulation and theory-theory: the argument's error. *Trends in Cognitive Sciences*, 9(8), 363-364.

Mitchell, M. (1993). *Analogy-making as perception: A computer model*. Cambridge, MA: MIT Press.

Mo, S., Su, Y., Sabbagh, M. A., & Jiaming, X. (2014). Sentential complements and false belief understanding in Chinese Mandarin-speaking preschoolers: A training study. *Cognitive Development*, *29*, 50-61.

Nau, D. S., Au, T. C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, *20*, 379-404.

Nichols, S., & Stich, S. P. (2003). *Mindreading: An integrated account of pretence, self-awareness, and*

understanding other minds. Clarendon Press/Oxford University Press.

Onishi, K. H., & Baillargeon, R. (2005). Do 15-month-old infants understand false beliefs?. *Science*, 308(5719), 255-258.

Onishi, K. H., Baillargeon, R., & Leslie, A. M. (2007). 15-month-old infants detect violations in pretend scenarios. *Acta Psychologica*, 124(1), 106-128.

Perner, J. (1994). The necessity and impossibility of simulation. *Proceedings of the British Academy*, 83, 129-144.

Perner, J. (1996). Simulation as explication of predication-implicit knowledge about the mind: arguments for a simulation-theory mix. In P. Carruthers & P.K. Smit (Eds.) *Theories of theories of mind* (pp. 90-104). Cambridge University Press.

Perner, J., & Howes, D. (1992). "He thinks he knows": And more developmental evidence against the simulation (role taking) theory. *Mind & Language*.

Perner, P., Holt, A., & Richter, M. (2005). Image processing in case-based reasoning. *The Knowledge Engineering Review*, 20(3), 311.

Peskin, J., and Astington, J. W. (2004). The effects of adding metacognitive language to story texts. *Cognitive Development*, 19(2), 253-273.

Plate, T. A. (1994). *Distributed representations and nested compositional structure*. [Doctoral Dissertation]. University of Toronto, Department of Computer Science.

Plaza, E., & McGinty, L. (2005). Distributed case-based reasoning. *Knowledge Engineering Review*, 20(3), 261-266.

Poulin-Dubois, D., Rakoczy, H., Burnside, K., Crivello, C., Dörrenberg, S., Edwards, K., ... & Perner, J. (2018). Do infants understand false beliefs? We don't know yet–A commentary on Baillargeon,

Buttelmann and Southgate's commentary. *Cognitive Development*, 48, 302-315.

Powell, L. J., Hobbs, K., Bardis, A., Carey, S., & Saxe, R. (2018). Replications of implicit theory of mind tasks with varying representational demands. *Cognitive Development*, 46, 40-50.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., & Botvinick, M. (2018). Machine theory of mind. In *International Conference on Machine Learning* (pp. 4218-4227). International Machine Learning Society (IMLS).

Rabkina, I., & Forbus, K.D. (2019). Analogical Reasoning for Intent Recognition and Action Prediction in Multi-Agent Systems. In *Proceedings of the Seventh Annual Conference on Advances in Cognitive Systems*. Cambridge, MA.

Rabkina, I., Kathnaraju, P., Roberts, M., Wilson, J., Forbus, K., & Hiatt, L. (2020). Recognizing the Goals of Uninspectable Agents. In *Proceedings of the AAAI Workshop on Plan, Activity and Intent Recognition*.

Rabkina, I., McFate, C. J., & Forbus, K. D. (2018). Bootstrapping from language in the Analogical Theory of Mind model. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*.

Rabkina, I., McFate, C., Forbus, K. D., & Hoyos, C. (2017). Towards a computational analogical theory of mind. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*.

Rabkina, I., Nakos, C., & Forbus, K. D. (2019a). Anticipatory Thinking in Multi-Agent Environments: The Role of Theory of Mind. In *Proceedings of the AAAI Fall Symposium on Cognitive Systems for Anticipatory Thinking*. Arlington, VA.

Rabkina, I., Nakos, C., & Forbus, K. D. (2019b). Children's sentential complement use leads the Theory of Mind development period: Evidence from the CHILDES corpus. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society.*

Raileanu, R., Denton, E., Szlam, A., & Fergus, R. (2018). Modeling others using oneself in multi-agent

reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, 6779-6788. International Machine Learning Society (IMLS).

Ramírez, M., & Geffner, H. (2009). Plan recognition as planning. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*.

Ramscar, M., & Pain, H. (1996). Can a real distinction be made between cognitive theories of analogy and categorization? In *Proceedings of the 18th Annual Conference of the Cognitive Science Society.*

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning,* 473-484.

Rao, A. S., & Murray, G. (1994). Multi-agent mental-state recognition and its application to air-combat modelling. In *Proceedings of the Workshop on Distributed Artificial Intelligence (DAI-94)*.

Rich, C., & Sidner, C. L. (1998). COLLAGEN: A collaboration manager for software interface agents. In *Computational Models of Mixed-Initiative Interaction* (pp. 149-184). Springer, Dordrecht.

Rizzolatti, G., Fogassi, L., & Gallese, V. (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2(9), 661-670.

Sagae, K., Davis, E., Lavie, A., MacWhinney, B., & Wintner, S. (2007). High-accuracy annotation and parsing of CHILDES transcripts. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*. Association for Computational Linguistics.

Saxe, R. (2005). Against simulation: the argument from error. *Trends in Cognitive Sciences*, 9(4), 174-179.

Scassellati, B. (2002). Theory of mind for a humanoid robot. *Autonomous Robots*, 12(1), 13-24.

Scholl, B. J., & Leslie, A. M. (1999). Modularity, development and 'theory of mind'. *Mind & Language*, *14*(1), 131-153.

Schmidt, C. F., Sridharan, N. S., & Goodson, J. L. (1978). The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, *11*(1-2), 45-83.

Scott, R. M., & Baillargeon, R. (2014). How fresh a look? A reply to Heyes. *Developmental Science*, 17(5), 660.

Scott, R. M., & Baillargeon, R. (2017). Early false-belief understanding. *Trends in Cognitive Sciences*, 21(4), 237-249.

Scott, R. M., Richman, J. C., & Baillargeon, R. (2015). Infants understand deceptive intentions to implant false beliefs about identity: New evidence for early mentalistic reasoning. *Cognitive Psychology*, 82, 32-56.

Shatz, M., Diesendruck, G., Martinez-Beck, I., & Akar, D. (2003). The influence of language and socioeconomic status on children's understanding of false belief. *Developmental Psychology*, 39(4), 717.

Shum, M., Kleiman-Weiner, M., Littman, M. L., & Tenenbaum, J. B. (2019). Theory of minds: Understanding behavior in groups through inverse planning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 6163-6170).

Skorstad, J., Gentner, D., & Medin, D. (1988). Abstraction processes during concept learning: A structural view. In *Proceedings of the 10th Annual Conference of the Cognitive Science Society*.

Skyrms, B. (2004). *The stag hunt and the evolution of social structure*. Cambridge University Press.

Smith, M., Apperly, I., & White, V. (2003). False belief reasoning and the acquisition of relative clause sentences. *Child Development*, 74(6), 1709-1719.

Sobel, D. M., & Letourneau, S. M. (2018). Children's Developing Descriptions and Judgments of Pretending. *Child Development*.

Stahl, A. E., & Feigenson, L. (2017). Expectancy violations promote learning in young children. *Cognition*, 163, 1-14.

Stone, V. E., & Gerrans, P. (2006). What's domain-specific about theory of mind?. *Social Neuroscience*, *1*(3-4), 309-319.

Sukthankar, G., Geib, C., Bui, H. H., Pynadath, D., & Goldman, R. P. (Eds.). (2014). *Plan, activity, and intent recognition: Theory and practice*. Newnes.

Sullivan, K., Winner, E., & Hopfield, N. (1995). How children tell a lie from a joke: The role of second-order mental state attributions. *British Journal of Developmental Psychology*, 13(2), 191-204.

Surian, L., Caldi, S., & Sperber, D. (2007). Attribution of beliefs by 13-month-old infants. *Psychological Science*, 18(7), 580-586.

Taggart, J., Heise, M. J., & Lillard, A. S. (2018). The real thing: preschoolers prefer actual activities to pretend ones. *Developmental Science*, *21*(3).

Talwar, V., Gordon, H. M., & Lee, K. (2007). Lying in the elementary school years: verbal deception and its relation to second-order belief understanding. *Developmental Psychology*, 43(3), 804.

Tenenbaum, J. B., Griffiths, T. L., & Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, 10(7), 309-318.

Thompson, B. N., & Goldstein, T. R. (2019). Disentangling pretend play measurement: Defining the essential elements and developmental progression of pretense. *Developmental Review*, 52, 24-41.

Tomai, E., & Forbus, K. D. (2009). EA NLU: Practical language understanding for cognitive modeling. In *Twenty-Second International FLAIRS Conference*.

Tomasello, M. (2009). *Constructing a language: A usage-based theory of language acquisition*. Harvard University Press.

Van de Weghe, N.; Cohn, A.; De Tre, B.; & De Maeyer, P. (2005). A Qualitative Trajectory Calculus as a basis for representing moving objects in Geographical Information Systems. *Control and Cybernetics* 35(1):97–120.

VanLehn, K. (1999). Rule-learning events in the acquisition of a complex skill: An evaluation of CASCADE. *The Journal of the Learning Sciences*, 8(1), 71-125.

VanLehn, K., & Jones, R. M. (1993). Better Learners Use Analogical Problem Solving Sparingly. In *Proceedings of the International Conference on Machine Learning.*

Vattam, S. S., Aha, D. W., & Floyd, M. (2014). Case-based plan recognition using action sequence graphs. In *International Conference on Case-Based Reasoning* (pp. 495-510). Springer, Cham.

Vered, M., Kaminka, G. A., & Biham, S. (2016). Online goal recognition through mirroring: Humans and agents. In *The Fourth Annual Conference on Advances in Cognitive Systems*.

Vigneswaran, G, Philipp, R, Lemon, RN, Kraskov, A. (2013). M1 corticospinal mirror neurons and their role in movement suppression during action observation. *Current Biology,* 23:236–43.

Weisberg, D. S. (2015). Pretend play. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(3), 249-261.

Wellman, H.M., Bartsch, K. (1988). Young children's reasoning about beliefs. *Cognition*, 30(2), 239-277.

Wellman, H. M., & Liu, D. (2004). Scaling of theory-of-mind tasks. *Child Development*, *75*(2), 523-541.

Wellman, H. W., Cross, D., & Watson, J. (2001). Metaanalysis of theory-of-mind development: The truth about false belief. *Child Development*, 72(3), 655-684.

Whitebread, D., & O'Sullivan, L. (2012). Preschool children's social pretend play: Supporting the development of metacommunication, metacognition and self-regulation. *International Journal of Play*, 1(2), 197-213.

Wimmer, H., & Perner, J. (1983). Beliefs about beliefs: Representation and constraining function of

wrong beliefs in young children's understanding of deception. *Cognition*, *13*(1), 103-128.

Yott, J., & Poulin-Dubois, D. (2016). Are infants' theory-of-mind abilities well integrated? Implicit understanding of intentions, desires, and beliefs. *Journal of Cognition and Development*, *17*(5), 683-698.

Appendix A: Training and Testing Stories from Hale & Tager-Flusberg (2003) Model

This appendix includes the representations and their English interpretations of the stories used for training and testing AToM in chapter 5. Where possible, the examples given by Hale and Tager-Flusberg (2003) were used.

Sentential Complement Construction (see McFate, 2018):

```
(naryHoldsIn clause1
             (situationConstituents
              arg1 NP-Subject)
             (naryHoldsIn say-clause
                          (situationConstituents
                           arg2 S-Comp))
             (situationConstituents
              arg3 AVP-Clause))
(syntacticOrder arg1 arg2 arg3)

(contradictory-Underspecified S-Comp AVP-Clause)
```

Sentential Complement Story1:
The boy said "I kissed Grover," but really he kissed Big Bird. (Hale & Tager-Flusberg, 2003)

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn
  (communicatorOfInfo say92
                      c1)))

 (situationConstituents
  arg2
  (SituationSuchThatFn
   (objectActedOn kiss1 bigbird)
   (performedBy kiss1 c1)))

(situationConstituents
 arg3
 (SituationSuchThatFn
  (objectActedOn kiss2 grover)
  (performedBy kiss2 c1)))

(isa arg3 (TokenAtFn grover))
(isa arg3 (TokenAtFn c1))
(isa arg3 (TokenAtFn kiss368723))
```

```
(isa arg2 (TokenAtFn bigbird))
(isa arg2 (TokenAtFn c1))
(isa arg2 (TokenAtFn kiss368592))

(isa arg1 (TokenAtFn c1))
(isa arg1 (TokenAtFn say92))
```

Sentential Complement Story2:

AB1 said, "AB1 hit AB2." but really AB1 hit AB3.

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn (communicatorOfInfo say123 ab1)))
(situationConstituents
 arg2
 (SituationSuchThatFn (objectActedOn hit25484 ab2)
                      (performedBy hit25484 ab1)))
(situationConstituents
 arg3
 (SituationSuchThatFn (objectActedOn hit25783 ab3)
                      (performedBy hit25783 ab1)))
(isa arg3 (TokenAtFn ab3))
(isa arg3 (TokenAtFn ab1))
(isa arg3 (TokenAtFn hit25783))

(isa arg2 (TokenAtFn ab2))
(isa arg2 (TokenAtFn ab1))
(isa arg2 (TokenAtFn hit25484))

(isa arg1 (TokenAtFn ab1))
(isa arg1 (TokenAtFn say123))
```

Sentential Complement Story3:

BC1 said, "BC1 slapped BC2." but really BC1 slapped BC3.

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn (communicatorOfInfo say234 bc1)))
(situationConstituents
 arg2
 (SituationSuchThatFn (victim slap26135 bc2)
                      (performedBy slap26135 bc1)))
(situationConstituents
 arg3
 (SituationSuchThatFn (victim slap26267 bc3)
                      (performedBy slap26267 bc1)))
(isa arg3 (TokenAtFn bc3))
```

```
(isa arg3 (TokenAtFn bc1))
(isa arg3 (TokenAtFn slap26267))

(isa arg2 (TokenAtFn bc2))
(isa arg2 (TokenAtFn bc1))
(isa arg2 (TokenAtFn slap26135))
```

Sentential Complement Story4:
BC1 said, "CD1 pushed CD2." but really CD1 pushed

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn (communicatorOfInfo say346 cd1)))
(situationConstituents
 arg2
 (SituationSuchThatFn (objectActedOn push27205 cd2)
                      (performedBy push27205 cd1)))
(situationConstituents
 arg3
 (SituationSuchThatFn (objectActedOn push27074 cd3)
                      (performedBy push27074 cd1)))
(isa arg3 (TokenAtFn cd3))
(isa arg3 (TokenAtFn cd1))
(isa arg3 (TokenAtFn push27074))

(isa arg2 (TokenAtFn cd2))
(isa arg2 (TokenAtFn cd1))
(isa arg2 (TokenAtFn push27205))

(isa arg1 (TokenAtFn cd1))
(isa arg1 (TokenAtFn say346))
```

Sentential Complement Story5:
EF1 said, "EF1 tickled EF2." but really EF1 tickled EF3.

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn (communicatorOfInfo say567 ef1)))
(situationConstituents
 arg2
 (SituationSuchThatFn (objectActedOn tickle27363 ef2)
                      (performedBy tickle27363 ef1)))
(situationConstituents
 arg3
 (SituationSuchThatFn (objectActedOn tickle27511 ef3)
                      (performedBy tickle27511 ef1)))
(isa arg3 (TokenAtFn ef3))
```

```
(isa arg3 (TokenAtFn ef1))
(isa arg3 (TokenAtFn tickle27511))

(isa arg2 (TokenAtFn ef2))
(isa arg2 (TokenAtFn ef1))
(isa arg2 (TokenAtFn tickle27363))

(isa arg1 (TokenAtFn ef1))
(isa arg1 (TokenAtFn say567))
```

Sentential Complement Story6:
EF1 said, "FG1 pinched FG2." but really FG1 pinched FG3.

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn (communicatorOfInfo say678 fg1)))
(situationConstituents
 arg2
 (SituationSuchThatFn (objectActedOn pinch28015  fg2)
                      (performedBy pinch28015  fg1)))
(situationConstituents
 arg3
 (SituationSuchThatFn (objectActedOn pinch28147  fg3)
                      (performedBy pinch28147  fg1)))
(isa arg3 (TokenAtFn fg3))
(isa arg3 (TokenAtFn fg1))
(isa arg3 (TokenAtFn pinch28147))

(isa arg2 (TokenAtFn fg2))
(isa arg2 (TokenAtFn fg1))
(isa arg2 (TokenAtFn pinch28015))

(isa arg1 (TokenAtFn fg1))
(isa arg1 (TokenAtFn say678))
```

Sentential Complement Story7:
GH1 said, "GH1 kicked GH2." but really GH1 kicked GH3.

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn (communicatorOfInfo say789 gh1)))
(situationConstituents
 arg2
 (SituationSuchThatFn (victim kick28743 gh2)
                      (performedBy kick28743 gh1)))
(situationConstituents
 arg3
```

```
 (SituationSuchThatFn (victim kick28910 gh3)
                      (performedBy kick28910 gh1)))
(isa arg3 (TokenAtFn gh3))
(isa arg3 (TokenAtFn gh1))
(isa arg3 (TokenAtFn kick28910))

(isa arg2 (TokenAtFn gh3))
(isa arg2 (TokenAtFn gh1))
(isa arg2 (TokenAtFn kick28743))

(isa arg1 (TokenAtFn gh1))
(isa arg1 (TokenAtFn say789))
```

Sentential Complement Story8:
GH1 said, "GH1 licked GH2." but really GH1 licked GH3.

```
(syntacticOrder arg1 arg2 arg3)

(situationConstituents
 arg1
 (SituationSuchThatFn (communicatorOfInfo say891 hi1)))
(situationConstituents
 arg2
 (SituationSuchThatFn (objectActedOn lick30162 hi2)
                      (performedBy lick30162 hi1)))
(situationConstituents
 arg3
 (SituationSuchThatFn (objectActedOn lick30311 hi3)
                      (performedBy lick30311 hi1)))
(isa arg3 (TokenAtFn hi3))
(isa arg3 (TokenAtFn hi1))
(isa arg3 (TokenAtFn lick30311))

(isa arg2 (TokenAtFn hi3))
(isa arg2 (TokenAtFn hi1))
(isa arg2 (TokenAtFn lick30162))

(isa arg1 (TokenAtFn hi1))
(isa arg1 (TokenAtFn say891))
```

Relative Clause Construction (see McFate, 2018):

```
(naryHoldsIn clause1
             (situationConstituents
              arg1 NP-Subject)
             (situationConstituents
              arg2 VP-Trans))

(syntacticOrder arg1 arg2)
```

Relative Clause Story1:

Bert kissed the girl who jumped. (adapted from Hale & Tager-Flusberg, 2003)

```
(syntacticOrder arg1 arg2)

(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy kiss2646 c2)))
(situationConstituents
 arg2
 (SituationSuchThatFn
  (objectMoving jump2739 g1)
  (situationLocation jump2739 g1)
  (objectActedOn kiss2646 g1)))

(isa arg1 (TokenAtFn c2))
(isa arg1 (TokenAtFn kiss2646))

(isa arg2 (TokenAtFn g1))
(isa arg2 (TokenAtFn kiss2646))
(isa arg2 (TokenAtFn jump2739))
```

Relative Clause Story2:
B1 hit the girl who cried.

```
(syntacticOrder arg1 arg2)
(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy hit123 b1)))
(situationConstituents
 arg2
 (SituationSuchThatFn
  (performedBy cry123 g1)
  (victim hit123 g1)
  (objectActedOn hit123 g1)))

(isa arg1 (TokenAtFn b1))
(isa arg1 (TokenAtFn hit123))

(isa arg2 (TokenAtFn g1))
(isa arg2 (TokenAtFn hit123))
(isa arg2 (TokenAtFn cry123))
```

Relative Clause Story3:
D1 slapped the girl who screamed.

```
(syntacticOrder arg1 arg2)
(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy slap123 d1)))
(situationConstituents
```

```
 arg2
 (SituationSuchThatFn
  (fe_sound_source scream123 h1)
  (victim slap123 h1)
  (objectActedOn slap123 h1)))

(isa arg1 (TokenAtFn d1))
(isa arg1 (TokenAtFn slap123))

(isa arg2 (TokenAtFn h1))
(isa arg2 (TokenAtFn slap123))
(isa arg2 (TokenAtFn scream123))
```

Relative Clause Story4:
E1 kicked the girl who whistled.

```
(syntacticOrder arg1 arg2)
(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy kick123 e1)))
(situationConstituents
 arg2
 (SituationSuchThatFn
  (fe_sound_source whistle123 i1)
  (victim kick123 i1)
  (objectActedOn kick123 i1)))

(isa arg1 (TokenAtFn e1))
(isa arg1 (TokenAtFn kick123))

(isa arg2 (TokenAtFn i1))
(isa arg2 (TokenAtFn kick123))
(isa arg2 (TokenAtFn whistle123))
```

Relative Clause Story5:
F1 tickled the girl who laughed.

```
(syntacticOrder arg1 arg2)
(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy tickle123 f1)))
(situationConstituents
 arg2
 (SituationSuchThatFn
  (fe_sound_source laugh123 j1)
  (objectActedOn tickle123 j1)))

(isa arg1 (TokenAtFn f1))
(isa arg1 (TokenAtFn kick123))

(isa arg2 (TokenAtFn i1))
```

```
(isa arg2 (TokenAtFn laugh123))
(isa arg2 (TokenAtFn tickle123))
```

Relative Clause Story6:
A1 pinched the girl who slept.

```
(syntacticOrder arg1 arg2)

(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy pinch123 a1)))
(situationConstituents
 arg2
 (SituationSuchThatFn
  (performedBy sleep123 k1)
  (objectActedOn pinch123 k1)))

(isa arg1 (TokenAtFn a1))
(isa arg1 (TokenAtFn kick123))

(isa arg2 (TokenAtFn k1))
(isa arg2 (TokenAtFn sleep123))
(isa arg2 (TokenAtFn pinch123))
```

Relative Clause Story7:
B1 bumped the girl who smiled.

```
(syntacticOrder arg1 arg2)
(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy bump36125 b1)))
(situationConstituents
 arg2
 (SituationSuchThatFn
  (performedBy smile36206 l1)
  (objectActedOn bump36125 l1)
  (fe_cognate_event smile36206 l1)
  ))

(isa arg1 (TokenAtFn b1))
(isa arg1 (TokenAtFn bump36125))

(isa arg2 (TokenAtFn l1))
(isa arg2 (TokenAtFn smile36206))
(isa arg2 (TokenAtFn bump36125))
```

Relative Clause Story8:
C1 pushed the girl who moved.

```
(syntacticOrder arg1 arg2)
```

```
(situationConstituents
 arg1
 (SituationSuchThatFn (performedBy push123 c1)))
(situationConstituents
 arg2
 (SituationSuchThatFn
  (objectMoving move123 m1)
  (objectActedOn push123 m1)
  ))

(isa arg1 (TokenAtFn c1))
(isa arg1 (TokenAtFn push123))

(isa arg2 (TokenAtFn m1))
(isa arg2 (TokenAtFn push123))
(isa arg2 (TokenAtFn move123))
```

Location Change Post-Test:

Daniel and his mother put the cup in the dishwasher. Daniel went outside to play. While Daniel was out, his mother put the cup in the cupboard. (adapted from Hale & Tager-Flusberg, 2003)

```
(isa dishwasher390019 Dishwasher)
(isa cup389759 Cup)
(isa m1 Person)
(isa d1 Person)
(isa put389647 PuttingSomethingSomewhere)
(isa go392964 LeavingAPlace)
(isa cupboard391144 Cupboard)
(isa put390859 PuttingSomethingSomewhere)
(isa enter393205 ArrivingAtAPlace)

(naryHoldsIn reality908
    (situationConstituents believe123
       (SituationSuchThatFn (hasBelief d1 belief123)))
          (naryHoldsIn believed
            (situationConstituents put289647
             (SituationSuchThatFn
             (implies (SituationSuchThatFn (objectPlaced put389647 cup389759)
                      (to-Generic put389647 dishwasher390019)
                      (performedBy put389647 (ConjunctiveVar d1 m1)))

(objectFoundInLocation cup389759 dishwasher390019)))))
             (situationConstituents put390859
                                    (SituationSuchThatFn
                                     (objectPlaced put390859 cup389759)
                                     (to-Generic put390859 cupboard391144)
                                     (performedBy put390859 m1)
                                     (objectMoving go392964 d1)
                                     (objectMoving enter393205 d1)
                                     )
                                    ))
```

Appearance-Reality Post-Test:
The object looks like a rock but is actually a sponge. (adapted from Hale & Tager-Flusberg, 2003)

```
(naryHoldsIn reality789
          (situationConstituents believe123 (SituationSuchThatFn (hasBelief
i123 opinions123)))
          (naryHoldsIn opinions123
             (situationConstituents sees123
                   (SituationSuchThatFn
             (implies (SituationSuchThatFn
                  (seemsLikeType sponge123 StoneStuff))
                  (isa sponge123 StoneStuff)))))
              (situationConstituents touches123
               (SituationSuchThatFn (seemsLikeType sponge123 StoneStuff)
                                    (isa sponge123 Sponge-CleaningImplement))))
```

Unexpected Contents Reality Post-Test:
There is a doll in the bandage box. (adapted from Hale & Tager-Flusberg, 2003)

```
(isa contain92749 ContainingSomething)
(isa bandage92818 Bandage)
(isa box123 (ContainerContainingFn Box-Container Bandage))
(isa opinions123 Hypothesis)
(isa me123 Agent)


(naryHoldsIn reality123
            (situationConstituents opinions12
                  (SituationSuchThatFn (hasBelief me123 opinions123)))
            (naryHoldsIn opinion
                   (situationConstituents containment12
                         (SituationSuchThatFn
                               (implies
                                (SituationSuchThatFn
                         (isa box123
                               (ContainerContainingFn Box-Container Bandage)))
            (relationInstanceExists containedObject contain92749 Bandage)))))
            (situationConstituents containment34
                  (SituationSuchThatFn
                     (relationInstanceExists containedObject contain92749
                                            Doll))))
```

Appendix B: Example Traces from Minecraft Experiments

This appendix contains examples of the Minecraft traces used in the experiments in chapter 7. To better highlight the differences between plan traces, execution traces, and external observation traces, an `obtain_chicken` plan is used for each. Note that facts containing the predicate `actualTask` were only used when calculating accuracy (i.e., were not include in analogical operation).

Example of plan trace (SHOP planner output):

```
(in-microtheory (MinecraftShopRecMtFn obtain_chicken_meat-2-76.hddl))
(isa action_0 Action)
(holdsIn action_0 (move_near_entity chicken_0f7c7ec163))
(isa chicken_0f7c7ec163 Chicken)
(isa action_1 Action)
(holdsIn action_1 (look_at_entity chicken_0f7c7ec163))
(isa chicken_0f7c7ec163 Chicken)
(isa action_2 Action)
(holdsIn action_2 (select_mc iron_sword))
(isa iron_sword Iron)
(isa action_3 Action)
(holdsIn action_3 (attack_mc chicken_0f7c7ec163))
(isa chicken_0f7c7ec163 Chicken)
(isa action_4 Action)
(holdsIn action_4 (gather chicken_meat))
(isa chicken_meat Chicken)
(actualTask (obtain_chicken_meat))
```

Example of execution trace (actual tasks performed by agent):

```
(in-microtheory (MinecraftTRACERecMtFn obtain_chicken_meat-593.hddl))
(isa action_0 Action)
(holdsIn action_0 (move chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb))
(isa chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb Chicken)
(isa action_1 Action)
(holdsIn action_1 (look_at chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb))
(isa chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb Chicken)
(isa action_2 Action)
(holdsIn action_2 (attack_mc chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb))
(isa chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb Chicken)
(isa action_3 Action)
(holdsIn action_3 (gather chicken_meat))
(isa chicken_meat Chicken)
(actualTask (obtain_chicken_meat))
```

Example of external observation trace (anonymized tasks performed by agent):

```
(in-microtheory (MinecraftANONRecMtFn obtain_chicken_meat-593.hddl))
(isa action_0 Action)
(holdsIn action_0 (move obj_0))
(isa obj_0 Obj)
(isa action_1 Action)
(holdsIn action_1 (look_at obj_0))
(isa obj_0 Obj)
(isa action_2 Action)
(holdsIn action_2 (attack_mc chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb))
(isa chicken_06212a7e-d4c1-4180-a9ca-8cd800e48efb Chicken)
(isa action_3 Action)
(holdsIn action_3 (gather chicken_meat))
(isa chicken_meat Chicken)
(actualTask (obtain_chicken_meat))
```

Appendix C: Example Stag-Hunt Simulations

This appendix contains an example final representation of the stag-hunt simulations generated for Experiment 2 in chapter 8. This example is of a complete scenario, as was used for training. For cross-validation folds where this scenario was used for testing, cases that contained subsets of these facts (i.e., relevant steps, no goal/assumption information, etc.) were generated.

```
(in-microtheory (CombinedMtFn Train (StagHuntMt 1 1 1 18)))
(isa r1 Rabbit)
(isa r2 Rabbit)
(isa s1 Stag)
(isa h1 Hunter)
(isa h2 Hunter)
(isa h3 Hunter)
(isa s2 Stag)
(causes-PropProp
 (holdsIn step2 (assumes h3 (assumedGoal h2 (cooperateWith h2 h1))))
 (holdsIn step2 (actualGoal h3 (huntAlone h3))))
(causes-PropProp
 (holdsIn step2 (assumes h3 (assumedGoal h1 (cooperateWith h1 h2))))
 (holdsIn step2 (actualGoal h3 (huntAlone h3))))
(causes-PropProp
 (holdsIn step3 (assumes h3 (assumedGoal h2 (cooperateWith h2 h1))))
 (holdsIn step3 (actualGoal h3 (huntAlone h3))))
(causes-PropProp
 (holdsIn step3 (assumes h3 (assumedGoal h1 (cooperateWith h1 h2))))
 (holdsIn step3 (actualGoal h3 (huntAlone h3))))
(causes-PropProp
 (and (holdsIn step1 (distances h3 s1)) (holdsIn step1 (approaches s1 h3)))
 (holdsIn step1 (sameDistance h3 s1)))
(causes-PropProp
 (and (holdsIn step1 (approaches h2 s1)) (holdsIn step1 (distances s1 h2)))
 (holdsIn step1 (sameDistance h2 s1)))
(causes-PropProp
 (and (holdsIn step1 (distances h1 s1)) (holdsIn step1 (distances s1 h1)))
 (holdsIn step1 (farther h1 s1)))
(holdsIn step3 (assumes h3 (assumedGoal h2 (cooperateWith h2 h1))))
(holdsIn step3 (assumes h3 (assumedGoal h1 (cooperateWith h1 h2))))
(holdsIn step2 (assumes h3 (assumedGoal h2 (cooperateWith h2 h1))))
(holdsIn step2 (assumes h3 (assumedGoal h1 (cooperateWith h1 h2))))
(holdsIn step1 (approaches s1 h3))
(holdsIn step1 (distances s1 h2))
(holdsIn step1 (distances s1 h1))
(holdsAtStart step1 (close s1 h1))
(causes-PropProp
 (and (holdsIn step2 (approaches-Agent h1 h2))
      (holdsIn step2 (distances-Agent h2 h1)))
```

```
 (holdsIn step2 (sameDistance h1 h2)))
(holdsIn step2 (distances-Agent h2 h1))
(causes-PropProp
 (and (holdsIn step3 (approaches h2 s2)) (holdsIn step3 (distances s2 h2)))
 (holdsIn step3 (sameDistance h2 s2)))
(causes-PropProp
 (and (holdsIn step3 (approaches h1 s2)) (holdsIn step3 (distances s2 h1)))
 (holdsIn step3 (sameDistance h1 s2)))
(holdsIn step3 (distances s2 h1))
(holdsIn step3 (distances s2 h2))
(causes-PropProp
 (and (holdsIn step1 (approaches h2 s2)) (holdsIn step1 (stationary s2)))
 (holdsIn step1 (closer h2 s2)))
(causes-PropProp
 (and (holdsIn step3 (distances-Agent h2 h3))
      (holdsIn step3 (approaches-Agent h3 h2)))
 (holdsIn step3 (sameDistance h2 h3)))
(holdsAtEnd step1 (close s2 h2))
(holdsIn step1 (approaches h2 s2))
(holdsIn step1 (approaches h2 s1))
(holdsAtEnd step3 (close r1 h2))
(holdsIn step1 (moving s1))
(causes-PropProp (holdsIn step1 (assumes h1 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step1 (actualGoal h1 (huntAlone h1))))
(causes-PropProp (holdsIn step1 (assumes h1 (assumedGoal h2 (huntAlone h2))))
 (holdsIn step1 (actualGoal h1 (huntAlone h1))))
(causes-PropProp
 (and (holdsIn step2 (approaches h2 s2)) (holdsIn step2 (stationary s2)))
 (holdsIn step2 (closer h2 s2)))
(causes-PropProp
 (and (holdsIn step3 (approaches h3 r1)) (holdsIn step3 (stationary r1)))
 (holdsIn step3 (closer h3 r1)))
(holdsIn step1 (assumes h1 (assumedGoal h2 (huntAlone h2))))
(holdsIn step3 (approaches h3 r1))
(holdsAtEnd step3 (close r1 h3))
(holdsIn step3 (distances-Agent h2 h3))
(holdsIn step2 (approaches h2 s2))
(causes-PropProp
 (and (holdsIn step1 (approaches h2 r1)) (holdsIn step1 (stationary r1)))
 (holdsIn step1 (closer h2 r1)))
(causes-PropProp
 (and (holdsIn step2 (approaches h3 s2)) (holdsIn step2 (stationary s2)))
 (holdsIn step2 (closer h3 s2)))
(causes-PropProp
 (and (holdsIn step2 (approaches h2 r1)) (holdsIn step2 (stationary r1)))
 (holdsIn step2 (closer h2 r1)))
(causes-PropProp
 (and (holdsIn step2 (approaches-Agent h2 h3))
      (holdsIn step2 (approaches-Agent h3 h2)))
 (holdsIn step2 (closer h2 h3)))
(causes-PropProp
 (and (holdsIn step3 (distances h3 s1)) (holdsIn step3 (stationary s1)))
```

```
 (holdsIn step3 (farther h3 s1)))
(causes-PropProp
 (and (holdsIn step3 (distances h2 s1)) (holdsIn step3 (stationary s1)))
 (holdsIn step3 (farther h2 s1)))
(causes-PropProp
 (and (holdsIn step3 (approaches h2 r1)) (holdsIn step3 (stationary r1)))
 (holdsIn step3 (closer h2 r1)))
(causes-PropProp
 (and (holdsIn step3 (distances h1 s1)) (holdsIn step3 (stationary s1)))
 (holdsIn step3 (farther h1 s1)))
(causes-PropProp
 (and (holdsIn step3 (approaches h1 r1)) (holdsIn step3 (stationary r1)))
 (holdsIn step3 (closer h1 r1)))
(holdsAtEnd step3 (close s2 h2))
(holdsIn step3 (approaches h1 r1))
(holdsIn step3 (approaches h2 r1))
(holdsAtEnd step3 (far r1 h1))
(holdsIn step3 (distances h3 s1))
(holdsIn step3 (distances h2 s1))
(holdsIn step3 (distances h1 s1))
(holdsIn step3 (stationary r1))
(holdsAtEnd step2 (close s2 h2))
(holdsIn step2 (approaches-Agent h3 h2))
(holdsIn step2 (approaches h2 r1))
(holdsAtEnd step2 (far r1 h3))
(holdsIn step1 (approaches h2 r1))
(causes-PropProp (holdsIn step1 (assumes h1 (assumedGoal h1 (huntAlone h1))))
 (holdsIn step1 (actualGoal h1 (huntAlone h1))))
(causes-PropProp
 (holdsIn step1 (assumes h3 (assumedGoal h2 (cooperateWith h2 h1))))
 (holdsIn step1 (actualGoal h3 (huntAlone h3))))
(causes-PropProp
 (holdsIn step1 (assumes h3 (assumedGoal h1 (cooperateWith h1 h2))))
 (holdsIn step1 (actualGoal h3 (huntAlone h3))))
(causes-PropProp
 (and (holdsIn step1 (approaches-Agent h2 h3))
      (holdsIn step1 (approaches-Agent h3 h2)))
 (holdsIn step1 (closer h2 h3)))
(causes-PropProp
 (and (holdsIn step2 (distances h3 s1)) (holdsIn step2 (stationary s1)))
 (holdsIn step2 (farther h3 s1)))
(causes-PropProp
 (and (holdsIn step2 (approaches h1 s2)) (holdsIn step2 (stationary s2)))
 (holdsIn step2 (closer h1 s2)))
(holdsIn step1 (assumes h3 (assumedGoal h2 (cooperateWith h2 h1))))
(holdsIn step1 (assumes h3 (assumedGoal h1 (cooperateWith h1 h2))))
(holdsIn step1 (assumes h1 (assumedGoal h1 (huntAlone h1))))
(holdsIn step1 (actualGoal h1 (huntAlone h1)))
(holdsAtEnd step3 (far s2 h3))
(holdsAtEnd step3 (far-Agent h1 h2))
(holdsAtEnd step3 (far s1 h3))
(holdsAtEnd step2 (far s2 h3))
```

```
(holdsIn step2 (approaches-Agent h2 h3))
(holdsIn step2 (distances h3 s1))
(holdsAtEnd step2 (far-Agent h1 h2))
(holdsIn step2 (stationary s2))
(holdsAtEnd step1 (far s2 h3))
(holdsIn step1 (distances h1 s1))
(holdsIn step1 (approaches-Agent h3 h2))
(holdsIn step1 (distances h3 s1))
(holdsIn step1 (approaches-Agent h2 h3))
(causes-PropProp (holdsIn step1 (assumes h2 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step1 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step1 (assumes h2 (assumedGoal h2 (huntAlone h2))))
 (holdsIn step1 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step1 (assumes h2 (assumedGoal h1 (huntAlone h1))))
 (holdsIn step1 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step1 (assumes h3 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step1 (actualGoal h3 (huntAlone h3))))
(causes-PropProp (holdsIn step2 (assumes h1 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step2 (actualGoal h1 (huntAlone h1))))
(causes-PropProp (holdsIn step2 (assumes h1 (assumedGoal h2 (huntAlone h2))))
 (holdsIn step2 (actualGoal h1 (huntAlone h1))))
(causes-PropProp (holdsIn step2 (assumes h1 (assumedGoal h1 (huntAlone h1))))
 (holdsIn step2 (actualGoal h1 (huntAlone h1))))
(causes-PropProp (holdsIn step2 (assumes h2 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step2 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step2 (assumes h2 (assumedGoal h2 (huntAlone h2))))
 (holdsIn step2 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step2 (assumes h2 (assumedGoal h1 (huntAlone h1))))
 (holdsIn step2 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step2 (assumes h3 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step2 (actualGoal h3 (huntAlone h3))))
(causes-PropProp (holdsIn step3 (assumes h1 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step3 (actualGoal h1 (huntAlone h1))))
(causes-PropProp (holdsIn step3 (assumes h1 (assumedGoal h2 (huntAlone h2))))
 (holdsIn step3 (actualGoal h1 (huntAlone h1))))
(causes-PropProp (holdsIn step3 (assumes h1 (assumedGoal h1 (huntAlone h1))))
 (holdsIn step3 (actualGoal h1 (huntAlone h1))))
(causes-PropProp (holdsIn step3 (assumes h2 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step3 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step3 (assumes h2 (assumedGoal h2 (huntAlone h2))))
 (holdsIn step3 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step3 (assumes h2 (assumedGoal h1 (huntAlone h1))))
 (holdsIn step3 (actualGoal h2 (huntAlone h2))))
(causes-PropProp (holdsIn step3 (assumes h3 (assumedGoal h3 (huntAlone h3))))
 (holdsIn step3 (actualGoal h3 (huntAlone h3))))
(causes-PropProp
 (and (holdsIn step1 (approaches h3 s2)) (holdsIn step1 (stationary s2)))
 (holdsIn step1 (closer h3 s2)))
(causes-PropProp
 (and (holdsIn step1 (approaches h3 r1)) (holdsIn step1 (stationary r1)))
 (holdsIn step1 (closer h3 r1)))
(causes-PropProp
```

```
 (and (holdsIn step1 (distances h1 s2)) (holdsIn step1 (stationary s2)))
 (holdsIn step1 (farther h1 s2)))
(causes-PropProp
 (and (holdsIn step1 (distances h1 r1)) (holdsIn step1 (stationary r1)))
 (holdsIn step1 (farther h1 r1)))
(causes-PropProp
 (and (holdsIn step1 (distances-Agent h1 h3))
      (holdsIn step1 (distances-Agent h3 h1)))
 (holdsIn step1 (farther h1 h3)))
(causes-PropProp
 (and (holdsIn step1 (distances-Agent h1 h2))
      (holdsIn step1 (approaches-Agent h2 h1)))
 (holdsIn step1 (sameDistance h1 h2)))
(causes-PropProp
 (and (holdsIn step2 (approaches h3 r1)) (holdsIn step2 (stationary r1)))
 (holdsIn step2 (closer h3 r1)))
(causes-PropProp
 (and (holdsIn step2 (distances h2 s1)) (holdsIn step2 (stationary s1)))
 (holdsIn step2 (farther h2 s1)))
(causes-PropProp
 (and (holdsIn step2 (distances h1 s1)) (holdsIn step2 (stationary s1)))
 (holdsIn step2 (farther h1 s1)))
(causes-PropProp
 (and (holdsIn step2 (approaches h1 r1)) (holdsIn step2 (stationary r1)))
 (holdsIn step2 (closer h1 r1)))
(causes-PropProp
 (and (holdsIn step2 (approaches-Agent h1 h3))
      (holdsIn step2 (approaches-Agent h3 h1)))
 (holdsIn step2 (closer h1 h3)))
(causes-PropProp
 (and (holdsIn step3 (approaches h3 s2)) (holdsIn step3 (distances s2 h3)))
 (holdsIn step3 (sameDistance h3 s2)))
(causes-PropProp
 (and (holdsIn step3 (approaches-Agent h1 h3))
      (holdsIn step3 (approaches-Agent h3 h1)))
 (holdsIn step3 (closer h1 h3)))
(causes-PropProp
 (and (holdsIn step3 (approaches-Agent h1 h2))
      (holdsIn step3 (distances-Agent h2 h1)))
 (holdsIn step3 (sameDistance h1 h2)))
(holdsIn step3 (assumes h3 (assumedGoal h3 (huntAlone h3))))
(holdsIn step3 (actualGoal h3 (huntAlone h3)))
(holdsIn step3 (assumes h2 (assumedGoal h3 (huntAlone h3))))
(holdsIn step3 (assumes h2 (assumedGoal h2 (huntAlone h2))))
(holdsIn step3 (assumes h2 (assumedGoal h1 (huntAlone h1))))
(holdsIn step3 (actualGoal h2 (huntAlone h2)))
(holdsIn step3 (assumes h1 (assumedGoal h3 (huntAlone h3))))
(holdsIn step3 (assumes h1 (assumedGoal h2 (huntAlone h2))))
(holdsIn step3 (assumes h1 (assumedGoal h1 (huntAlone h1))))
(holdsIn step3 (actualGoal h1 (huntAlone h1)))
(holdsIn step2 (assumes h3 (assumedGoal h3 (huntAlone h3))))
(holdsIn step2 (actualGoal h3 (huntAlone h3)))
```

```
(holdsIn step2 (assumes h2 (assumedGoal h3 (huntAlone h3))))
(holdsIn step2 (assumes h2 (assumedGoal h2 (huntAlone h2))))
(holdsIn step2 (assumes h2 (assumedGoal h1 (huntAlone h1))))
(holdsIn step2 (actualGoal h2 (huntAlone h2)))
(holdsIn step2 (assumes h1 (assumedGoal h3 (huntAlone h3))))
(holdsIn step2 (assumes h1 (assumedGoal h2 (huntAlone h2))))
(holdsIn step2 (assumes h1 (assumedGoal h1 (huntAlone h1))))
(holdsIn step2 (actualGoal h1 (huntAlone h1)))
(holdsIn step1 (assumes h3 (assumedGoal h3 (huntAlone h3))))
(holdsIn step1 (actualGoal h3 (huntAlone h3)))
(holdsIn step1 (assumes h2 (assumedGoal h3 (huntAlone h3))))
(holdsIn step1 (assumes h2 (assumedGoal h2 (huntAlone h2))))
(holdsIn step1 (assumes h2 (assumedGoal h1 (huntAlone h1))))
(holdsIn step1 (actualGoal h2 (huntAlone h2)))
(holdsIn step1 (assumes h1 (assumedGoal h3 (huntAlone h3))))
(holdsIn step3 (distances s2 h3))
(holdsIn step3 (approaches h3 s2))
(holdsIn step3 (approaches h2 s2))
(holdsIn step3 (approaches h1 s2))
(holdsAtEnd step3 (far s2 h1))
(holdsIn step3 (approaches-Agent h3 h2))
(holdsIn step3 (approaches-Agent h3 h1))
(holdsAtEnd step3 (far-Agent h2 h3))
(holdsIn step3 (distances-Agent h2 h1))
(holdsIn step3 (approaches-Agent h1 h3))
(holdsIn step3 (approaches-Agent h1 h2))
(holdsAtEnd step3 (far-Agent h1 h3))
(holdsAtEnd step3 (far s1 h2))
(holdsAtEnd step3 (far s1 h1))
(holdsIn step3 (moving s2))
(holdsIn step3 (moving h3))
(holdsIn step3 (moving h2))
(holdsIn step3 (moving h1))
(holdsIn step3 (stationary s1))
(holdsIn step2 (approaches h3 s2))
(holdsIn step2 (approaches h1 s2))
(holdsAtEnd step2 (far s2 h1))
(holdsIn step2 (approaches h3 r1))
(holdsIn step2 (distances h1 s1))
(holdsIn step2 (approaches h1 r1))
(holdsAtEnd step2 (far-Agent h2 h3))
(holdsAtEnd step2 (far r1 h2))
(holdsAtEnd step2 (far r1 h1))
(holdsIn step2 (approaches-Agent h3 h1))
(holdsIn step2 (distances h2 s1))
(holdsIn step2 (approaches-Agent h1 h3))
(holdsIn step2 (approaches-Agent h1 h2))
(holdsAtEnd step2 (far-Agent h1 h3))
(holdsAtEnd step2 (far s1 h3))
(holdsAtEnd step2 (far s1 h2))
(holdsAtEnd step2 (far s1 h1))
(holdsIn step2 (stationary r1))
```

```
(holdsIn step2 (moving h3))
(holdsIn step2 (moving h2))
(holdsIn step2 (moving h1))
(holdsIn step2 (stationary s1))
(holdsIn step1 (distances h1 s2))
(holdsIn step1 (approaches h3 s2))
(holdsAtEnd step1 (far s2 h1))
(holdsIn step1 (approaches-Agent h2 h1))
(holdsIn step1 (approaches h3 r1))
(holdsIn step1 (distances-Agent h1 h3))
(holdsIn step1 (distances h1 r1))
(holdsAtEnd step1 (far-Agent h2 h3))
(holdsAtEnd step1 (far r1 h3))
(holdsAtEnd step1 (far r1 h2))
(holdsAtEnd step1 (far r1 h1))
(holdsIn step1 (distances-Agent h3 h1))
(holdsIn step1 (distances-Agent h1 h2))
(holdsAtEnd step1 (far-Agent h1 h3))
(holdsAtEnd step1 (far-Agent h1 h2))
(holdsAtEnd step1 (far s1 h3))
(holdsAtEnd step1 (far s1 h2))
(holdsAtEnd step1 (far s1 h1))
(holdsIn step1 (stationary s2))
(holdsIn step1 (stationary r1))
(holdsIn step1 (moving h3))
(holdsIn step1 (moving h2))
(holdsIn step1 (moving h1))
(holdsAtStart step1 (far s2 h3))
(holdsAtStart step1 (far s2 h2))
(holdsAtStart step1 (far s2 h1))
(holdsAtStart step1 (close r2 h2))
(holdsAtStart step1 (far s1 h3))
(holdsAtStart step1 (far r1 h3))
(holdsAtStart step1 (far r1 h2))
(holdsAtStart step1 (far-Agent h2 h3))
(holdsAtStart step1 (far-Agent h1 h3))
(holdsAtStart step1 (far-Agent h1 h2))
(holdsAtStart step1 (far s1 h2))
(holdsAtStart step1 (far r2 h3))
(holdsAtStart step1 (far r2 h1))
(holdsAtStart step1 (far r1 h1))
```