Int J Software Informatics, Vol.3, No.1, March 2009, pp. 47–65 E-mail: ijsi@iscas.ac.cn
International Journal of Software and Informatics, ISSN 1673-7288 http://www.ijsi.org
 Tel: +86-10-62661040

Propositional Attitudes and Causation*

Konstantine Arkoudas and Selmer Bringsjord

(Cognitive Science and Computer Science Departments, Rensselaer Polytechnic Institute, NY 12180-3590, USA, arkoudas@csail.mit.edu, brings@rpi.edu)

Abstract Predicting and explaining the behavior of others in terms of mental states is indispensable for everyday life. It will be equally important for artificial agents. We present an inference system for representing and reasoning about mental states, and use it to provide a formal analysis of the false-belief task. The system allows for the representation of information about events, causation, and perceptual, doxastic, and epistemic states (vision, belief, and knowledge), incorporating ideas from the event calculus and multi-agent epistemic logic. Unlike previous AI formalisms, our focus here is on mechanized proofs and proof programmability, not on metamathematical results. Reasoning is performed via relatively cognitively plausible inference rules, and a degree of automation is achieved by general-purpose inference methods and by a syntactic embedding of the system in first-order logic.

Key words: propositional attitudes; multi-agent epistemic logic; event calculus; reasoning; mental states; false-belief tasks; theory-theory

Arkoudas K, Bringsjord S. Propositional attitudes and causation. *Int J Software Informatics*, **2009, 3(1): 47–65. http://www.ijsi.org/1673-7288/3/47.htm**

1 Introduction

Interpreting the behavior of other people is indispensable for everyday life. It is something that we do constantly, on a daily basis, and it helps us not only to make sense of human behavior, but also to predict it and—to a certain extent—to control it. How exactly do we manage that? That is not currently known, but many have argued that the ability to ascribe mental states to others and to reason about such mental states is a key component of our capacity to understand human behavior. In particular, all social transactions, from engaging in commerce and negotiating to making jokes and empathizing with other people's pain or joy, appear to require at least a rudimentary grasp of common-sense psychology (CSP), i.e., a large body of truisms such as the following: When an agent a (1) wants to achieve a certain state of affairs p, and (2) believes that some action c can bring about p, and (3) a knows how to carry out c; then, ceteris paribus,[1)] a will carry out c; when a sees that p, a knows that p; when a fears that p and a discovers that p is the case, a is disappointed; and so on.

Artificial agents without a mastery of CSP would be severely handicapped in their interactions with humans. This could present problems not only for artificial agents

* Corresponding author: Konstantine Arkoudas; email: arkoudas@csail.mit.edu
Manuscript received 2009-03-31; revised 2009-06-29; accepted 2009-07-15.

1) Assuming that a is able to carry out c, that a has no conflicting desires that override his goal that p; and so on.

trying to interpret human behavior, but also for artificial agents trying to interpret the behavior of one another. When a system exhibits a complex but rational behavior, and detailed knowledge of its internal structure is not available, the best strategy for predicting and explaining its actions might be to analyze its behavior in intentional terms, i.e., in terms of mental states such as beliefs and desires (regardless of whether the system *actually* has genuine mental states; for the purposes of this work we take a thoroughly instrumentalist view of mental states). Mentalistic models are likely to be particularly apt for agents trying to manipulate the behavior of other agents.

Any computational treatment of CSP will have to integrate action and cognition. Agents must be able to reason about the causes and effects of various events, whether they are non-intentional physical events or intentional events brought about by their own agency. More importantly, they must be able to reason about what others believe or know about such events. To that end, we present a system which combines and adapts ideas drawn from the event calculus and from multi-agent epistemic logics. It is based on multi-sorted first-order logic extended with subsorting, epistemic operators for perception, belief, and knowledge, and mechanisms for reasoning about causation and action. Using subsorting, we formally model agent actions as types of events, which enables us to use the resources of the event calculus to represent and reason about agent actions. The usual axioms of the event calculus are encoded as common knowledge, suggesting that people have an understanding of the basic folk laws of causality (innate or acquired), and are indeed aware that others have such an understanding.

It is important to be clear about what we hope to accomplish through the present work. In general, any logical system or methodology capable of representing and reasoning about intentional notions such as knowledge can have at least three different uses. First, it can serve as a tool for the specification, analysis, and verification of rational agents. Second, in tandem with some appropriate reasoning mechanism, it can serve as a knowledge representation framework, i.e., it can be used *by* artificial agents to represent their own "mental states"—and those of other agents—and to deliberate and act in accordance with those states and their environment. Finally, it can be used to provide formal models of certain interesting cognitive phenomena. One intended contribution of our present work is of the third sort, namely, to provide a formal model of false-belief attributions, and, in particular, a description of the logical competence of an agent capable of passing a false-belief task. It addresses questions such as the following: What sort of principles is it plausible to assume that an agent has to deploy in order to be able to succeed on a false-belief task? What is the depth and complexity of the required reasoning? Can such reasoning be automated, and if so, how? These questions have not been taken up in detail in the relevant discussions in cognitive science and the philosophy of mind, which have been couched in overly abstract and rather vague terms. Formal computational models such as the one we present here can help to ground such discussions, to clarify conceptual issues, and to begin to answer important questions in a concrete setting.

Although the import of such a model is primarily scientific, there can be interesting engineering implications. For instance, if the formalism is sufficiently expressive and versatile, and the posited computational mechanisms can be automated with reasonable efficiency, then the system can make contributions to the first two areas

mentioned above. We believe that our system has such potential for two reasons. First, the combination of epistemic constructs such as common knowledge with the conceptual resources of the event calculus for dealing with causation appears to afford considerable expressive power, as demonstrated by the sample model we present in this paper. A key technical insight behind this combination is the modelling of agent actions as events via subsorting. Second, procedural abstraction mechanisms appear to hold significant promise for automation; we discuss this issue later in more detail.

The remainder of this paper is structured as follows. The next section gives the formal definition of our system. Section 3 represents the false-belief task in our system, and Section 4 presents a model of the reasoning that is required to succeed in such a task, carried out in a modular fashion by collaborating methods. Section 5 presents an encoding of the system in first-order logic with a view to making reasoning in the system amenable to ATP technology. Finally, section 6 discusses some related work and concludes.

2 A Calculus for Representing and Reasoning about Actions and Mental States

The syntactic and semantic problems that arise when one tries to use classical logic to represent and reason about intentional notions are well-known. Syntactically, modelling belief or knowledge relationally is problematic because one believes or knows arbitrarily complex propositions, whereas the arguments of relation symbols are terms built from constants, variables, and function symbols. (The objects of belief could be encoded by strings, but such representations are too low-level for most purposes.) Semantically, the main issue is the referential opacity (or intensionality) exhibited by propositional-attitude operators. In intensional contexts one cannot freely substitute one coreferential term for another. Broadly speaking, there are two ways of addressing these issues. One is to use a modal logic, with built-in syntactic operators for intentional notions. The other is to retain classical logic but distinguish between an object-language and a meta-language, representing intentional discourse at the object level. Each approach has its advantages and drawbacks. Retaining classical logic has the important advantage of efficiency, in that (semi-)automated deduction systems for classical logic, such as resolution provers—which have made impressive strides over the last decade—can be used for reasoning. This is the option we have chosen in some previous work[3]. One disadvantage of this approach is that when the object language is first-order (includes quantification), then notions such as substitutions and alphabetic equivalence must be explicitly encoded. Depending on the facilities provided by the meta-language, this does not need to be overly onerous, but it does require extra effort. The modal-logic approach has the advantage of solving the syntactic and referential-opacity problems directly, without the need to distinguish an object-language and a meta-language. In this work we combine both approaches. The system is formulated (and implemented) as a properly intensional calculus, with knowledge, belief, etc., represented as sentential operators. For purposes of modeling, specification, and verification, the system is used in this form. For using the system as a knowledge-representation framework *by* artificial agents trying to negotiate the behavior of other agents, we have encoded the system in first-order logic to make reasoning in it more amenable to automated theorem-proving methods.

$$
\begin{array}{rl}
S ::= & \texttt{Object} \mid \texttt{Agent} \mid \texttt{ActionType} \mid \texttt{Action} \sqsubseteq \texttt{Event} \mid \texttt{Moment} \mid \texttt{Boolean} \mid \texttt{Fluent} \\
f ::= & \begin{array}{l} \mathit{action} : \texttt{Agent} \times \texttt{ActionType} \to \texttt{Action} \\ \mathit{initially} : \texttt{Fluent} \to \texttt{Boolean} \\ \mathit{holds} : \texttt{Fluent} \times \texttt{Moment} \to \texttt{Boolean} \\ \mathit{happens} : \texttt{Event} \times \texttt{Moment} \to \texttt{Boolean} \\ \mathit{clipped} : \texttt{Moment} \times \texttt{Fluent} \times \texttt{Moment} \to \texttt{Boolean} \\ \mathit{initiates} : \texttt{Event} \times \texttt{Fluent} \times \texttt{Moment} \to \texttt{Boolean} \\ \mathit{terminates} : \texttt{Event} \times \texttt{Fluent} \times \texttt{Moment} \to \texttt{Boolean} \\ \mathit{prior} : \texttt{Moment} \times \texttt{Moment} \to \texttt{Boolean} \end{array} \\
t ::= & x : S \mid c : S \mid f(t_1, \ldots, t_n) \\
P ::= & t : \texttt{Boolean} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid P \Leftrightarrow Q \mid \\
& \forall x : S \;.\; P \mid \exists x : S \;.\; P \mid \mathbf{S}(a, P) \mid \mathbf{K}(a, P) \mid \mathbf{B}(a, P) \mid \mathbf{C}(P)
\end{array}
$$

Figure 1. The specification of sorts, function symbols, terms, and propositions

The specification of the syntax of our system appears in Fig. 1, which describes the various sorts of our universe (S), the signatures of certain built-in function symbols (f), and the abstract syntax of terms (t) and propositions (P). The symbol $\sqsubseteq$ denotes subsorting. Propositions of the form $\mathbf{S}(a, P)$, $\mathbf{B}(a, P)$, and $\mathbf{K}(a, P)$ should be understood as saying that agent a sees that P is the case, believes that P, and knows that P, respectively. Propositions of the form $\mathbf{C}(P)$ assert that P is commonly known. Sort annotations will generally be omitted, as they are easily deducible from the context. We write $P[x \mapsto t]$ for the proposition obtained from P by replacing every free occurrence of x by t, assuming that t is of a sort compatible with the sort of the free occurrences in question, and taking care to rename P as necessary to avoid variable capture. We use the infix notation $t_1 < t_2$ instead of $\mathit{prior}(t_1, t_2)$.

We express the following standard axioms of the event calculus as common knowledge:

$$[A_1] \qquad \mathbf{C}(\forall\, f, t \;.\; \mathit{initially}(f) \wedge \neg \mathit{clipped}(0, f, t) \Rightarrow \mathit{holds}(f, t))$$

$$[A_2] \qquad \mathbf{C}(\forall\, e, f, t_1, t_2 \;.\; \mathit{happens}(e, t_1) \wedge \mathit{initiates}(e, f, t_1) \wedge t_1 < t_2 \wedge \neg \mathit{clipped}(t_1, f, t_2) \Rightarrow \mathit{holds}(f, t_2))$$

$$[A_3] \qquad \mathbf{C}(\forall\, t_1, f, t_2 \;.\; \mathit{clipped}(t_1, f, t_2) \Leftrightarrow [\exists\, e, t \;.\; \mathit{happens}(e, t) \wedge t_1 < t < t_2 \wedge \mathit{terminates}(e, f, t)])$$

suggesting that people have a (possibly innate) understanding of basic causality principles, and are indeed aware that everybody has such an understanding. In addition to $[A_1]$—$[A_3]$, we postulate a few more axioms pertaining to what people know or believe about causality. First, agents know the events that they intentionally bring about themselves—that is part of what "action" means. In fact, this is common knowledge. The following axiom expresses this:

$$[A_4] \qquad \mathbf{C}(\forall\, a, d, t \;.\; \mathit{happens}(\mathit{action}(a, d), t) \Rightarrow \mathbf{K}(a, \mathit{happens}(\mathit{action}(a, d), t)))$$

The next axiom states that it is common knowledge that if an agent a believes that a certain fluent f holds at t and he does not believe that f has been clipped between t and t', then he will also believe that f holds at t':

$$[A_5] \quad \mathbf{C}(\forall\, a, f, t, t' \;.\; \mathbf{B}(a, \mathit{holds}(f, t)) \wedge \mathbf{B}(a, t < t') \wedge \neg\mathbf{B}(a, \mathit{clipped}(t, f, t')) \Rightarrow \mathbf{B}(a, \mathit{holds}(f, t'))))$$

The final axiom states that if a believes that b believes that f holds at t_1 and a believes that nothing has happened between t_1 and t_2 to change b's mind, then a will believe that b will not think that f has been clipped between t_1 and t_2:

$$[A_6] \;\; \forall\, a, b, t_1, t_2, f \;.\; [\mathbf{B}(a, \mathbf{B}(b, \mathit{holds}(f, t_1))) \wedge \mathbf{B}(a, \neg\exists\, e, t \;.\; \mathbf{B}(b, \mathit{happens}(e, t)) \wedge \mathbf{B}(b, t_1 < t < t_2) \wedge \mathbf{B}(b, \mathit{terminates}(e, f, t)))] \Rightarrow \mathbf{B}(a, \neg\mathbf{B}(b, \mathit{clipped}(t_1, f, t_2)))$$

This captures a form of closed-world reasoning, for it could well be the case that, in fact, b has come to believe that something has happened between t and t' that terminated f, and therefore no longer believes that f holds. But if a believes that there have been no such events, then it is reasonable for a to assume that b will not believe that f has been clipped.

In addition to the usual introduction and elimination rules for first-order predicate logic with equality, we will make use of the following inference rules:

$$\frac{}{\mathbf{C}(\mathbf{S}(a, P) \Rightarrow \mathbf{K}(a, P))}\;[R_1] \qquad \frac{}{\mathbf{C}(\mathbf{K}(a, P) \Rightarrow \mathbf{B}(a, P))}\;[R_2]$$

$$\frac{\mathbf{C}(P)}{\mathbf{K}(a_1, \mathbf{K}(a_2, \mathbf{K}(a_3, P)))}\;[R_3] \qquad \frac{\mathbf{K}(a, P)}{P}\;[R_4]$$

$[R_1]$ says that it is common knowledge that visual perception is a justified source of knowledge. In other words, it is commonly known that if I see that P, I know P.[2)] $[R_2]$ says that it is commonly known that knowledge requires belief, while $[R_3]$ captures an essential property of common knowledge. Usually common knowledge of a proposition P is taken to mean that everybody knows that P, everybody knows that everybody knows that P, and so on ad infinitum. This is captured by recursive rules that allow us to "unfold" the common-knowledge operator arbitrarily many times. However, this viewpoint is quite problematic for finite knowers of limited cognitive capacity. After three or four levels of nesting, iterated knowledge claims become unintelligible. Because in the present setting we are concerned with cognitive plausibility, we refrain from characterizing common knowledge in the customary strong form, imposing instead limit of three levels of iteration, as indicated in $[R_3]$.[3)] $[R_4]$ is a veracity rule for knowledge.

The following rules can now be readily derived:

[2)] We ignore here the issue of perceptual illusions.

[3)] Although there is not enough space here for a full discussion, we point out that third-order epistemic and doxastic states (as opposed to n-order for $n > 3$) are often held to be at a level of iteration sufficient for general accounts of human thinking, e.g., see Dennett (1978). This is not to say that fairly realistic scenarios involving iteration of 4 or even 5 levels cannot be devised, but in the present paper we have used 3 for the purpose of modeling the false-belief task.

$$\frac{\mathbf{C}(P)}{\mathbf{K}(a_1, \mathbf{K}(a_2, P))}\ [DR_1] \qquad \frac{\mathbf{C}(P)}{\mathbf{K}(a, P)}\ [DR_2]$$

$$\frac{\mathbf{C}(P)}{P}\ [DR_3] \qquad \frac{\mathbf{S}(a, P)}{\mathbf{K}(a, P)}\ [DR_4] \qquad \frac{\mathbf{K}(a, P)}{\mathbf{B}(a, P)}\ [DR_5]$$

We next have the following three rules:

$$\frac{}{\mathbf{C}(\mathbf{K}(a, P_1 \Rightarrow P_2) \Rightarrow \mathbf{K}(a, P_1) \Rightarrow \mathbf{K}(a, P_2))}\ [R_5]$$

$$\frac{}{\mathbf{C}(\mathbf{B}(a, P_1 \Rightarrow P_2) \Rightarrow \mathbf{B}(a, P_1) \Rightarrow \mathbf{B}(a, P_2))}\ [R_6]$$

$$\frac{}{\mathbf{C}(\mathbf{C}(P_1 \Rightarrow P_2) \Rightarrow \mathbf{C}(P_1) \Rightarrow \mathbf{C}(P_2))}\ [R_7]$$

From these we can easily derive the so-called Kripke ("K") rules for knowledge, belief, and common knowledge:

$$\frac{\mathbf{K}(a, P_1 \Rightarrow P_2) \quad \mathbf{K}(a, P_1)}{\mathbf{K}(a, P_2)}\ [DR_6]$$

We likewise have derived rules $[DR_7]$ and $[DR_8]$ for belief and common knowledge, respectively (omitted here). We also assume that a few straightforward tautologies are common knowledge, and the self-explanatory $[R_{11}]$:

$$\frac{}{\mathbf{C}((\forall\, x\ .\ P) \Rightarrow P[x \mapsto t])}\ [R_8] \qquad \frac{}{\mathbf{C}([P_1 \Leftrightarrow P_2] \Rightarrow \neg P_2 \Rightarrow \neg P_1)}\ [R_9]$$

$$\frac{}{\mathbf{C}([P_1 \wedge \cdots \wedge P_n \Rightarrow P] \Rightarrow [P_1 \Rightarrow \cdots \Rightarrow P_n \Rightarrow P])}\ [R_{10}]$$

$$\frac{\mathbf{B}(a, P_1) \quad \mathbf{B}(a, P_2)}{\mathbf{B}(a, P_1 \wedge P_2)}\ [R_{11}]$$

Note that usually it is postulated that *every* tautology is common knowledge. If we took that as a principle, the presentation of the system could be somewhat simplified. However, such a principle (and other "logical omniscience" principles like it) is wildly implausible, as has often been pointed out. Since we do not accept such unrestricted principles, we only posit certain specific tautologies that are intuitively deemed as obvious. While this is not a general solution, it nevertheless averts the cognitive implausibility of the unrestricted rules, and also serves to isolate the logical knowledge that we need to attribute to agents for a specific reasoning problem.

The following rules are now readily derived:[4)]

[4)] Derivation proofs are omitted, but can be obtained (along with the computer implementation of the system) by contacting the authors.

$$\frac{\mathbf{K}(a, \forall\, x\ .\ P)}{\mathbf{K}(a, P[x \mapsto t])}\ [DR_9] \qquad \frac{\mathbf{B}(a, \forall\, x\ .\ P)}{\mathbf{B}(a, P[x \mapsto t])}\ [DR_{10}]$$

$$\frac{\mathbf{C}(\forall\, x\ .\ P)}{\mathbf{C}(P[x \mapsto t])}\ [DR_{11}] \qquad \frac{\mathbf{B}(a_1, \mathbf{K}(a_2, P))}{\mathbf{B}(a_1, \mathbf{B}(a_2, P))}\ [DR_{12}]$$

$$\frac{\mathbf{K}(a_1, \mathbf{K}(a_2, P_1 \Rightarrow P_2)) \quad \mathbf{K}(a_1, \mathbf{K}(a_2, P_1))}{\mathbf{K}(a_1, \mathbf{K}(a_2, P_2))}\ [DR_{13}]$$

$$\frac{\mathbf{B}(a_1, \mathbf{B}(a_2, P_1 \Rightarrow P_2)) \quad \mathbf{B}(a_1, \mathbf{B}(a_2, P_1))}{\mathbf{B}(a_1, \mathbf{B}(a_2, P_2))}\ [DR_{14}]$$

$$\frac{\mathbf{K}(a_1, \mathbf{K}(a_2, P_1 \Leftrightarrow P_2)) \quad \mathbf{K}(a_1, \mathbf{K}(a_2, \neg P_2))}{\mathbf{K}(a_1, \mathbf{K}(a_2, \neg P_1))}\ [DR_{15}]$$

$$\frac{\mathbf{B}(a_1, \mathbf{B}(a_2, P_1 \Leftrightarrow P_2)) \quad \mathbf{B}(a_1, \mathbf{B}(a_2, \neg P_2))}{\mathbf{B}(a_1, \mathbf{B}(a_2, \neg P_1))}\ [DR_{16}]$$

$$\frac{\begin{array}{c}\mathbf{K}(a_1, \mathbf{K}(a_2, [P_1 \wedge \cdots \wedge P_n] \Rightarrow P)) \\ \mathbf{K}(a_1, \mathbf{K}(a_2, P_1)) \quad \cdots \quad \mathbf{K}(a_1, \mathbf{K}(a_2, P_n))\end{array}}{\mathbf{K}(a_1, \mathbf{K}(a_2, P))}\ [DR_{17}]$$

$$\frac{\begin{array}{c}\mathbf{B}(a_1, \mathbf{B}(a_2, [P_1 \wedge \cdots \wedge P_n] \Rightarrow P)) \\ \mathbf{B}(a_1, \mathbf{B}(a_2, P_1)) \quad \cdots \quad \mathbf{B}(a_1, \mathbf{B}(a_2, P_n))\end{array}}{\begin{array}{c}\mathbf{B}(a_1, \mathbf{B}(a_2, P)) \\ \mathbf{B}(a, P_1 \wedge P_2 \wedge P_3 \Rightarrow P_4)\end{array}}\ [DR_{18}]$$

$$\frac{\mathbf{B}(a, P_1) \quad \mathbf{B}(a, P_2) \quad \mathbf{B}(a, P_3)}{\mathbf{B}(a, P_4)}\ [DR_{19}]$$

The system presented in this section has been implemented in the form of a denotational proof language similar to the Athena system[1], but with the operators for belief, knowledge, etc., directly available as propositional constructors. Note that this system is altogether different from its encoding *in* Athena described in section 5.

3 Encoding the False-Belief Task

False-belief scenarios can be regarded as the drosophila of computational theories of mind. Experiments with false beliefs were first carried out by Wimmer and Perner[12]. In a typical scenario, a child (we will call her Alice) is presented with a story in which a character (we will call him Bob) places an object (say, a cookie) in a certain location l_1, say in a particular kitchen cabinet. Then Bob leaves, and during

his absence someone else (say, Charlie) removes the object from its original location l_1 and puts it in a different location l_2 (say, a kitchen drawer). Alice is then asked to predict where Bob will look for the object when he gets back, the right answer, of course, being the original location—the cabinet. In this section we show how to formalize this scenario in our calculus. In the next section we will present a formal explanation as to how Alice can come to acquire the correct belief about Bob's false belief.

We introduce the sort `Location` and the following function symbols specifically for reasoning about the false-belief task:

$$\mathit{places} : \texttt{Object} \times \texttt{Location} \rightarrow \texttt{ActionType}$$
$$\mathit{moves} : \texttt{Object} \times \texttt{Location} \times \texttt{Location} \rightarrow \texttt{ActionType}$$
$$\mathit{located} : \texttt{Object} \times \texttt{Location} \rightarrow \texttt{Fluent}$$

Intuitively, $\mathit{action}(a, \mathit{places}(o, l))$ signifies a's action of placing object o in location l, while $\mathit{action}(a, \mathit{moves}(o, l_1, l_2))$ is a's action of moving object o from location l_1 to location l_2. It is common knowledge that placing o in l initiates the fluent $\mathit{located}(o, l)$:

$$[D_1]\ \ \mathbf{C}(\forall\, a, t, o, l\ .\ \mathit{initiates}(\mathit{action}(a, \mathit{places}(o, l)), \mathit{located}(o, l), t))$$

It is likewise known that if an object o is located at l_1 at a time t, then the act of moving o from l_1 to l_2 results in o being located at l_2:

$$[D_2]\qquad \mathbf{C}(\forall\, a, t, o, l_1, l_2\ .\ \mathit{holds}(\mathit{located}(o, l_1), t) \Rightarrow \mathit{initiates}(\mathit{action}(a, \mathit{moves}(o, l_1, l_2)), \mathit{located}(o, l_2), t))$$

If, in addition, the new location is different from the old one, the move terminates the fluent $\mathit{located}(o, l_1)$:

$$[D_3]\qquad \mathbf{C}(\forall\, a, t, o, l_1, l_2\ .\ \mathit{holds}(\mathit{located}(o, l_1), t) \wedge l_1 \neq l_2 \Rightarrow \mathit{terminates}(\mathit{action}(a, \mathit{moves}(o, l_1, l_2)), \mathit{located}(o, l_1), t))$$

The following axiom captures the constraint that an object cannot be in more than one place at one time; this is also common knowledge:

$$[D_4]\qquad \mathbf{C}(\forall\, o, t, l_1, l_2\ .\ \mathit{holds}(\mathit{located}(o, l_1), t) \wedge \mathit{holds}(\mathit{located}(o, l_2), t) \Rightarrow l_1 = l_2)$$

We introduce three time moments that are central to the narrative of the false-belief task: *beginning*, *departure*, and *return*. The first signifies the time point when Bob places the cookie in the cabinet, while *departure* and *return* mark the points when he leaves and comes back, respectively. We assume that it's common knowledge that these three time points are linearly ordered in the obvious manner:

$$[D_5]\ \ \mathbf{C}(\mathit{beginning} < \mathit{departure} < \mathit{return}).$$

We also introduce two distinct locations, *cabinet* and *drawer*:

$$[D_6]\ \ \mathbf{C}(\mathit{cabinet} \neq \mathit{drawer}).$$

Finally, we introduce a domain `Cookie` as a subsort of `Object`, and declare a single element of it, *cookie*. It is a given premise that, in the beginning, Alice sees Bob place the cookie in the cabinet:

$$[D_7]\ \ \mathbf{S}(Alice, happens(action(Bob, places(cookie, cabinet)), beginning)).$$

4 Modeling the Reasoning Underlying False-Belief Tasks, and Automating It via Abstraction

At this point we have enough representational and reasoning machinery in place to infer the correct conclusion from a couple of obvious premises. However, a monolithic derivation of the conclusion from the premises would be unsatisfactory, as it would not give us a story about how such reasoning can be dynamically put together. Agents must be able to reason about the behavior of other agents efficiently. It is not at all obvious how efficiency can be achieved in the absence of mechanisms for abstraction, modularity, and reusability.

We can begin to address both issues by pursuing further the idea of derived inference rules, and by borrowing a page from classic work in cognitive science and production systems. Suppose that we had a mechanism which enabled the derivation of not only *schematic* inference rules, such as the ones that we presented in section 2, but derived inference rules allowing for arbitrary computation and search. We could then formulate *generic* inference rules, capable of being applied to an unbounded (potentially infinite) number of arbitrarily complex concrete situations.

Our system has a notion of *method* that allows for that type of abstraction and encapsulation. Methods are derived inference rules, not just of the schematic kind, but incorporating arbitrary computation and search. They are thus more general than the simple if-then rules of production systems, and more akin to the knowledge sources (or "demons") of blackboard systems[10]. They can be viewed as encapsulating specialized expertise in deriving certain types of conclusions from certain given information. They can be parameterized over any variables, e.g., arbitrary agents or time points.

A key role in our system is played by an associative data structure (shared by all methods) known as the *assumption base*, which is an efficiently indexed collection of propositions that represent the collective knowledge state at any given moment, including perceptual knowledge. The assumption base is capable of serving as a communication buffer for the various methods. Finally, the control executive is itself a method, which directs the reasoning process incrementally by invoking various methods triggered by the contents of the assumption base.

We describe below three general-purpose methods for reasoning in the calculus we have presented. With these methods, the reasoning for the false-belief task can be performed in a handful of lines—essentially with one invocation of each of these methods. We stress that these methods are not ad hoc or hardwired to false-belief tasks. They are generic, and can be reused in any context that requires reasoning about other minds and satisfies the relevant preconditions. In particular, the methods do not contain or require any information specific to false-belief tasks.

– *Method 1*: This method, which we call M_1, shows that when an agent a_1 sees an agent a_2 perform some action-type α at some time point t, a_1 knows that a_2 knows that a_2 has carried out α at t. M_1 is parameterized over a_1, a_2, α, and t:

1. The starting premise is that a_1 sees a_2 perform α at t:

$$\mathbf{S}(a_1, happens(action(a_2, \alpha), t)) \tag{1.1}$$

2. Therefore, a_1 knows that the corresponding event has occurred at t:

$$\mathbf{K}(a_1, happens(action(a_2, \alpha), t)) \tag{1.2}$$

This follows from the preceding premise and $[DR_4]$.

3. From $[A_4]$ and $[DR_2]$ we obtain:

$$\begin{aligned}\mathbf{K}(a_1, \forall\, a, \alpha, t\ .\ happens(action(a, \alpha), t) \Rightarrow \\ \mathbf{K}(a, happens(action(a, \alpha), t)))\end{aligned} \tag{1.3}$$

4. From (3) and $[DR_9]$ we get:

$$\begin{aligned}\mathbf{K}(a_1, happens(action(a_2, \alpha), t) \Rightarrow \\ \mathbf{K}(a_2, happens(action(a_2, \alpha), t)))\end{aligned} \tag{1.4}$$

5. From (4), (2), and $[DR_6]$ we get:

$$\mathbf{K}(a_1, \mathbf{K}(a_2, happens(action(a_2, \alpha), t))) \tag{1.5}$$

– *Method 2*: The second method, M_2, shows that when (1) it is common knowledge that a certain event e initiates a fluent f; (2) an agent a_1 knows that an agent a_2 knows that e has happened at a time t_1; (3) it is commonly known that $t_1 < t_2$; and (4) a_1 knows that a_2 knows that nothing happens between t_1 and t_2 to terminate the fluent f; then a_1 knows that a_2 knows that f holds at t_2. M_2 is parameterized over a_1, a_2, e, f, t_1, and t_2:

1. The starting premises are the following:

$$P_1\text{:}\quad \mathbf{C}(\forall\, t\ .\ initiates(e, f, t));$$

$$P_2\text{:}\quad \mathbf{K}(a_1, \mathbf{K}(a_2, happens(e, t_1)));$$

$$P_3\text{:}\quad \mathbf{C}(t_1 < t_2);$$

$$P_4\text{:}\quad \begin{aligned}&\mathbf{K}(a_1, \mathbf{K}(a_2, \neg\exists\, e, t\ .\ happens(e, t) \wedge \\ &\quad t_1 < t < t_2 \wedge terminates(e, f, t))).\end{aligned}$$

From P_1, $[DR_{11}]$, and $[DR_1]$, we get:

$$\mathbf{K}(a_1, \mathbf{K}(a_2, initiates(e, f, t_1))) \tag{1.6}$$

2. From P_3 and $[DR_1]$ we get:

$$\mathbf{K}(a_1, \mathbf{K}(a_2, t_1 < t_2)) \tag{1.7}$$

3. From $[A_3]$, $[DR_{11}]$, and $[DR_1]$ we get:

$$\begin{aligned}\mathbf{K}(a_1, \mathbf{K}(a_2, clipped(t_1, f, t_2) \Leftrightarrow \exists\, e, t\ .\ happens(e, t) \wedge \\ t_1 < t < t_2 \wedge termindates(e, f, t)))\end{aligned} \tag{1.8}$$

4. From (8), P_4, and $[DR_{15}]$ we conclude that a_1 knows that a_2 knows that f has not been clipped between t_1 and t_2:

$$\mathbf{K}(a_1, \mathbf{K}(a_2, \neg \mathit{clipped}(t_1, f, t_2))) \tag{1.9}$$

5. From $[A_2]$, $[DR_{11}]$, and $[DR_1]$ we get:

$$\begin{aligned}\mathbf{K}(a_1, \mathbf{K}(a_2, [&\mathit{happens}(e, t_1) \wedge \mathit{initiates}(e, f, t_1) \wedge \\ &t_1 < t_2 \wedge \neg \mathit{clipped}(t_1, f, t_2)] \Rightarrow \mathit{holds}(f, t_2)))\end{aligned} \tag{1.10}$$

6. From (10), premise P_2, (6), (7), (9), and $[DR_{17}]$ we get:

$$\mathbf{K}(a_1, \mathbf{K}(a_2, \mathit{holds}(f, t_2))) \tag{1.11}$$

– *Method 3*: The last method, M_3, shows that when (1) it is common knowledge that t_1 is prior to t_2; (2) an agent a_1 knows that an agent a_2 knows that a fluent f holds at t_1; and (3) a_1 believes that nothing happened between t_1 and t_2 that would cause a_2 to believe that f no longer holds; then a_1 believes that a_2 believes that f holds at t_2:

1. The starting premises are:

$$P_1: \quad \mathbf{C}(t_1 < t_2);$$

$$P_2: \quad \mathbf{K}(a_1, \mathbf{K}(a_2, \mathit{holds}(f, t_1)));$$

$$P_3: \quad \begin{aligned}\mathbf{B}(a_1, \neg\exists\, e, t\ .\ &\mathbf{B}(a_2, \mathit{happens}(e, t)) \wedge \mathbf{B}(a_2, t_1 < t < t_2) \wedge \\ &\mathbf{B}(a_2, \mathit{terminates}(e, f, t))).\end{aligned}$$

2. From premise P_2, $[DR_5]$, and $[DR_{12}]$, we get:

$$\mathbf{B}(a_1, \mathbf{B}(a_2, \mathit{holds}(f, t_1))) \tag{1.12}$$

3. From $[A_6]$, $[DR_3]$, and universal specialization we get:

$$\begin{aligned}[\mathbf{B}(a_1, \mathbf{B}(a_2, \mathit{holds}(f, t_1))) \wedge \mathbf{B}(a_1, \neg\exists\, e, t\ .\ &\mathbf{B}(a_2, \mathit{happens}(e, t)) \wedge \\ &\mathbf{B}(a_2, t_1 < t < t_2) \wedge \\ &\mathbf{B}(a_2, \mathit{terminates}(e, f, t)))] \Rightarrow \mathbf{B}(a_1, \neg\mathbf{B}(a_2, \mathit{clipped}(t_1, f, t_2)))\end{aligned} \tag{1.13}$$

4. By P_3, (13), (12), conjunction introduction, and modus ponens, we get:

$$\mathbf{B}(a_1, \neg\mathbf{B}(a_2, \mathit{clipped}(t_1, f, t_2))) \tag{1.14}$$

5. From $[A_5]$, $[DR_{11}]$, and $[DR_2]$ we get:

$$\begin{aligned}\mathbf{K}(a_1, [&\mathbf{B}(a_2, \mathit{holds}(f, t_1)) \wedge \mathbf{B}(a_2, t_1 < t_2) \wedge \\ &\neg\mathbf{B}(a_2, \mathit{clipped}(t_1, f, t_2))] \Rightarrow \mathbf{B}(a_2, f))\end{aligned} \tag{1.15}$$

6. From (15) and $[DR_5]$ we get:

$$\mathbf{B}(a_1, [\mathbf{B}(a_2, holds(f, t_1)) \wedge \mathbf{B}(a_2, t_1 < t_2) \wedge \neg\mathbf{B}(a_2, clipped(t_1, f, t_2))] \Rightarrow \mathbf{B}(a_2, holds(f, t_2)) \tag{1.16}$$

7. From P_1, $[DR_1]$, $[DR_5]$, and $[DR_{12}]$ we get:

$$\mathbf{B}(a_1, \mathbf{B}(a_2, t_1 < t_2)) \tag{1.17}$$

8. From (16), (12), (17), (14), and $[DR_{19}]$ we get:

$$\mathbf{B}(a_1, \mathbf{B}(a_2, holds(f, t_2))) \tag{1.18}$$

The correct conclusion for the false-belief task, produced by our implementation in a fraction of a second, is now obtained in the following manner:

1. Method M_1 fires, invoked with Alice, Bob, the action type

$$places(cookie, cabinet),$$

   and time point *beginning*.

2. Axiom $[D_1]$ is repeatedly instantiated (via $[DR_{11}]$) with *Bob*, *cookie*, and *cabinet*.

3. Method M_2 fires, invoked with *Alice*, *Bob*, the action that *Bob* has placed the cookie in the cabinet, the fluent that the cookie is located in the cabinet, and the two time points *beginning* and *departure*.

4. Method M_3 fires, invoked with Alice, Bob, the fluent that the cookie is located in the cabinet, and the two time points *departure* and *return*.

5 Embedding the Calculus in First-Order Logic

Despite their significantly greater expressivity and modeling power, systems combining modal operators and quantification are often met with resistance from AI researchers on the grounds that reasoning in them is hopelessly inefficient. As we mentioned in the introduction, one of our chief aims is to provide an expressive formal framework for the rigorous modeling and analysis of interesting cognitive phenomena involving the propositional attitudes, so for our purposes the lack of automation is not a vitiating factor. Nevertheless, the problem is obviously of crucial importance, particularly insofar as such a framework is to be used *by* artificial agents in order to interpret, predict, or influence the behavior of other agents. We have already suggested one avenue for achieving a certain degree of automation, namely, the introduction of specialized reasoning methods. When a sufficiently large number of such methods are available concurrently, as demons running on a highly parallel architecture, it is plausible that reasoning of the kind that is required for the false-belief task can be carried out efficiently. In this section we sketch out an alternative approach to automating reasoning in our framework.

It is worth emphasizing from the outset that *theoretically* efficient reasoning does not seem possible even for propositional logic (on the assumption that P $\neq$ NP). In practice, however, there are reasoning systems that perform fairly well even in the case of first-order logic. In particular, resolution-based automated theorem provers (ATPs) such as Vampire[20] and Spass[22] have made significant progress over the last decade, and it is not unreasonable to hope that ATP technology will continue to improve. In fact, first-order logic has long been an attractive option for AI researchers working on propositional-attitude systems, because the language of first-order logic is "the *lingua franca* of knowledge representation," and because "there is good theorem-proving technology for this language" [14, p. 2]. Accordingly, several researchers have tried to formalize intensional logics *in* first-order logic by taking a so-called "syntactic" approach, whereby an intensional propositional operator (such as knowledge or belief) becomes a first-order *predicate* symbol that takes as an argument a term denoting a proposition. However, naive syntactic formalizations of first-order intensional logics in classical first-order logic using devices for quotation and unquotation are prone to inconsistency, as was shown by R. Montague[13], R. Thomason[19], and others. While J. des Rivières and H. J. Levesque[9] have shown how to avoid such inconsistency by essentially restricting the range over which sentence variables range in the various intensional axiom schemas (a result that was later somewhat extended[14]), the exercise, when carried out in unsorted first-order logic, is still delicate and faces certain difficulties. These difficulties can be avoided by deploying a sort discipline to impose a sharp separation between object language and metalanguage, thus precluding the type of self-reference that is familiar from the liar paradox and which ultimately leads to the aforementioned inconsistency results. (This kind of separation between object- and meta-language, of course, was pioneered by Tarski precisely in order to avert the type of self-reference responsible for the truth paradoxes.)

In the remainder of this section we will pursue this kind of syntactic route. Due to the explicit separation between object- and meta-language, our effort will have to be a bit more involved than the usual syntactic approaches—we will encode the entire proof system for our original intensional system. Nevertheless, we will stay entirely within classical first-order logic, which means that the various powerful ATPs mentioned above will be available for reasoning in the encoded system. This suggests a two-tiered use of our framework. For purposes of modeling and analysis, we can use the original implementation, which is properly intensional in that the various modal operators are directly applied to sentences. For purposes of automated reasoning, we can resort to the encoded system in multi-sorted first-order logic and help ourselves to available ATPs. Translating a specific problem description to the encoded system can be performed automatically. The encoding given below is carried out in Athena[2], an interactive theorem-proving system for multi-sorted first-order-logic that comes integrated with both Vampire and Spass. A brief presentation of Athena's syntax and semantics can be found elsewhere [4, ch. 2].

To facilitate the task, we will encode a sequent-calculus presentation of our system, rather than a natural-deduction version of it, since derivation in sequent calculi is easier to formalize. The transcription of the inference rules of Section 2 in sequent form is straightforward. Sample sequent rules are shown in Fig. 2. The substitution operation $P[x \mapsto t]$ is partial, defined only when variable capture does not occur, in

which case $P[x \mapsto t]$ denotes the proposition obtained from P by replacing every free occurrence of variable x by the term t. We start by introducing appropriate syntactic domains for variables, terms, and propositions. Athena provides the built-in domain `Ide` (for "identifiers") as a generic variable category, precisely in order to facilitate the representation of formal systems. Typical identifiers are strings prefixed by `'`, e.g., `'x` and `'foo`:

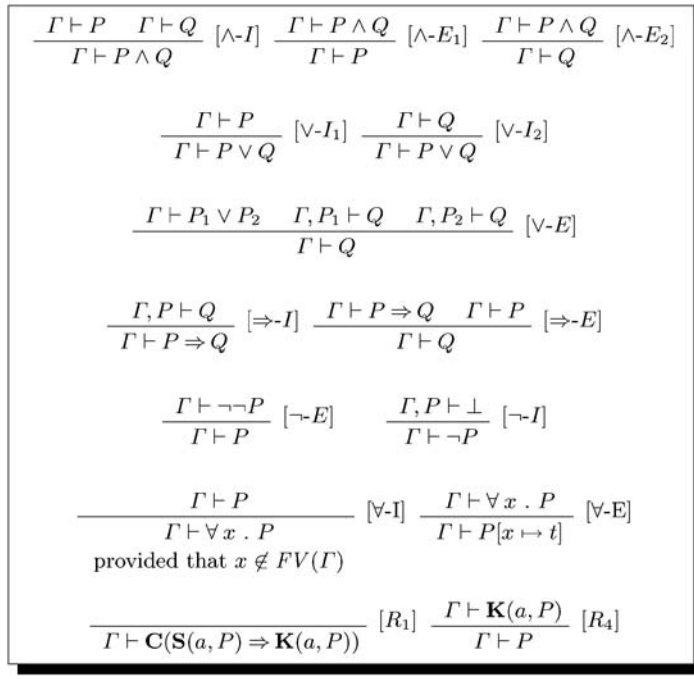

Figure 2. Sample inference rules for a sequent-based version of the system of Section 2

```
>(exists ?x (= ?x 'foo))

Proposition: (exists ?x:Ide
               (= ?x 'foo))
```

Thus, assuming a domain `Symbol` for function symbols such as *action* and *clipped*, we can readily encode terms as follows:

```
(datatype Term
  (Var Ide)
  (App Symbol (List-Of Term)))
```

This simply says that a term is either a variable (an application of the constructor `Var` to an identifier), or else a function application of a function symbol to a list of terms. Thus, the term *happens(e,t)*, where e and t are variables, is represented by the following:

```
(App happens (Cons (Var 'e) (Cons (Var 't) Nil))),
```

where `Cons` and `Nil` are the two constructors of the polymorphic datatype constructor `List-Of`. A substitution operation is declared as follows:

```
(declare sub (-> (Ide Term Term) Term))
```

The intended meaning is that `(sub ?x ?t ?s)` is the term obtained from `?s` by replacing every occurrence of `?x` by `?t`. It is convenient to have a version that can operate on an entire list of terms:

```
(declare subLst (->(Ide Term(List-Of Term))(List-Of Term)))
```

The two functions are defined by the following four universally quantified identities:

```
(define sub-axiom-1
  (forall ?x ?t
    (= (sub ?x ?t (Var ?x)) ?t)))

(define sub-axiom-2
  (forall ?x ?t ?f ?terms
    (= (sub ?x ?t (App ?f ?terms))
       (App ?f (subLst ?x ?t ?terms)))))

(define subLst-axiom-1
  (forall ?x ?t
    (= (subLst ?x ?t Nil) Nil)))

(define subLst-axiom-2
  (forall ?x ?t ?s ?rest
    (= (subLst ?x ?t (Cons ?s ?rest))
       (Cons (sub ?x ?t ?s)
             (subLst ?x ?t ?rest)))))

(assert sub-axiom-1 sub-axiom-2 subLst-axiom-1 subLst-axiom-2)
```

Sorts are introduced as elements of a datatype:

```
(datatype Sort
  ObjectSort
  AgentSort
  ...
  FluentSort)
```

We introduce a `sort` function from `Symbol` to sort signatures:

```
(declare symbol-sort(->(Symbol)(Pair-Of(List-Of Sort)Sort)))
```

and a function `term-has-sort` and predicate `prop-well-sorted` for performing sort (type) checking on terms and propositions, respectively. We omit the details here, which are not complicated.[5] Given a domain of `Agent`s, propositions are defined as follows:

```
(domain Agent)

(datatype Prop
  (atom Term)
  (neg Prop)
  (conj Prop Prop)
  (disj Prop Prop)
  (cond Prop Prop)
  (for-every Ide Prop)
```

[5] Note, however, that sort checking here needs to be done with respect to a sort context, i.e., a mapping from symbols to sorts, since variables are not explicitly annotated with their sorts.

```
  (for-some Ide Prop)
  (common Prop)
  (sees Agent Prop)
  (believes Agent Prop)
  (knows Agent Prop))
```

This directly mirrors the structure of the proposition grammar shown in Fig.1 (except that there are no explicit sort annotations for quantified variables). The formalization of matters involving free variables is standard; we present a small sample:

```
(declare occurs (-> (Ide Term) Boolean))

(declare occursLst (-> (Ide (List-Of Term)) Boolean))

(define occurs-axiom-1
  (forall ?x ?y
    (iff (occurs ?x (Var ?y))
         (= ?x ?y))))

(define occurs-axiom-2
  (forall ?x ?f ?terms
    (iff (occurs ?x (App ?f ?terms))
         (occursLst ?x ?terms))))

(define occursLst-axiom-1
  (forall ?x
    (not (occursLst ?x Nil))))

(define occursLst-axiom-2
  (forall ?x ?t ?rest
    (iff (occursLst ?x (Cons ?t ?rest))
         (or (occurs ?x ?t)
             (occursLst ?x ?rest)))))

(declare occursFree (->(Ide Prop)Boolean))

(declare occursFreeLst (->(Ide(List-Of Prop))Boolean))

(define occursFree-atom-axiom
  (forall ?x ?t
    (iff (occursFree ?x (atom ?t))
         (occurs ?x ?t))))

(define occursFree-conj-axiom
  (forall ?x ?p ?q
    (iff (occursFree ?x (conj ?p ?q))
         (or (occursFree ?x ?p)
             (occursFree ?x ?q)))))

(define occursFree-all-axiom
  (forall ?x ?y ?p
    (iff (occursFree ?x (for-every ?y ?p))
         (and (occursFree ?x ?p)
              (not (= ?x ?y))))))
```

A sequent is formalized as pair of a list of propositions (the context) and a single proposition (the conclusion):

```
(declare sequent (-> ((List-Of Prop) Prop) Boolean))
```

Finally, the inference rules of the system (figure 2) are captured by straightforward axioms. We demonstrate with conjunction introduction, universal generalization, and the truth requirement on knowledge:

```
(define conj-intro
  (forall ?Gamma ?p ?q
    (if (and (sequent ?Gamma ?p)
             (sequent ?Gamma ?q))
        (sequent ?Gamma (conj ?p ?q)))))

(define ugen
  (forall ?Gamma ?x ?p
    (if (and (sequent ?Gamma ?p)
             (not (occursFreeLst ?x ?Gamma)))
        (sequent ?Gamma (for-every ?x ?p)))))

(define R4-sequent
  (forall ?Gamma ?a ?p
    (if (sequent ?Gamma (knows ?a ?p))
        (sequent ?Gamma ?p))))
```

Now let p be any proposition in the original intensional system, and let $\widehat{p}$ be the Athena term of sort `Prop` denoting its translation into the first-order theory T we have specified here. (The translation is trivial, requiring no more than linear time in the size of p.) Then, to derive a proposition p from a set of propositions $\{p_1, \ldots, p_n\}$ in the original system, it suffices to derive the sequent

$$[\widehat{p_1}, \ldots, \widehat{p_n}] \vdash \widehat{p} \tag{1.19}$$

from the theory T (where T contains all the definitions we have given above, and axioms such as `conj-intro` and `ugen`, describing the behaviors of the inference rules of the intensional system). Since (19) is a first-order formula and the theory T is also a set of first-order formulas, the latter problem is amenable to classical first-order techniques and ATP systems. Moreover, special-purpose proof methods can be programmed in Athena to further aid the automation of reasoning in this system.

6 Related Work and Conclusions

We have presented a formal system for representing and reasoning about certain important kinds of mental states, and used it to provide a formal analysis of false-belief tasks. Such tasks have been extensively discussed, particularly in the debate between theory-theory and simulation[7], but there are few rigorous models to be found. The only computational treatments of which we are aware are by Bello, Bignoli, and Cassimatis[5]; and by Watt[21]. Neither is based on a formal inference system. Goodman et al.[12] present a rational analysis of false belief reasoning based on causal Bayesian models.

Technically, our system is a multi-sorted multi-modal first-order logic. There is a growing recognition of the importance of quantification in epistemic contexts. Propositional multi-modal logics are just not sufficiently expressive. For instance, they cannot capture the difference between de dicto and de re knowledge. The versatility of first-order logic is necessary, alongside constructs such as common knowledge.

Our approach has been thoroughly proof-theoretic; we have not given a model-theoretic semantics for our logic. Coming up with an appropriate formal semantics for

propositional attitudes is exceedingly difficult, and should not hold back experimentation with and implementation of various proof systems. The usual possible-world semantics[11] are mathematically elegant and well-understood, and they can be a useful tool in certain situations (e.g., in security protocol analysis). But they are notoriously implausible from a cognitive viewpoint. (Indeed, knowledge, belief, desire, intention, provability, etc., all receive the exact same formal analysis in possible-world semantics.) In an apt assessment of the situation, Anderson[1] wrote that epistemic logic "has been a pretty bleak affair." Fagin et al.[11] describe various attempts to deal with some of the problems arising in a possible-worlds setting, none of which has been widely accepted as satisfactory.

At any rate, even in the standard Kripke framework, the question of how to combine quantification with epistemic constructs (particularly with common knowledge) is a difficult open problem: there have been no complete recursive axiomatizations, and indeed such logics are not even recursively enumerable[24]. Some decidable fragments have been investigated, such as the space of monodic formulas[18], but such restrictions limit expressivity, which in our view is a more important consideration. Indeed, we see no reason to insist on a computationally tractable—or even decidable—formalism, or on a complete logic, at the expense of expressivity. First-order logic is undecidable, but it is routinely used for the analysis and verification of a wide variety of extensional systems, by deploying interactive theorem-proving systems. Higher-order logic is both undecidable and incomplete, but it too is used widely for similar purposes. Things need not be different when it comes to the representation, analysis, and verification of rational agents. Our concern here has been to design and implement a fairly expressive logic that can be readily used for such purposes; and to gain experience with constructing machine-checkable proofs in that logic, and particularly with writing powerful proof tactics in it.

LORA[25] is a multi-sorted language that extends first-order branching-time temporal logic with modal constructs for beliefs, desires, and intentions (drawing on the seminal work of Cohen and Levesque[6], and particularly on the BDI paradigm that followed it[16]), as well as a dynamic logic for representing and reasoning about actions. It does not have any constructs for perception or for common knowledge, and does not allow for the representation of events that are not actions. Its semantics for the propositional attitudes are standard Kripke semantics, with the possible worlds being themselves branching time structures. We are not aware of any implementations of LORA.

CASL (Cognitive Agents Specification Language)[17] is another system which combines an action theory, defined in terms of the situation calculus, with modal operators for belief, desire, and intention. Like LORA, CASL does not have any constructs for perception or for group knowledge (shared, distributed, or common). Also like LORA, the semantics of all intensional operators in CASL are given in terms of standard possible worlds. They are, in fact, explicitly defined in the higher-order logic PVS[15] by quantifying over states. Insofar as both LORA and CASL base their treatment of intensional operators on Kripke structures, they inherit all the conceptual difficulties associated with them. An advantage of CASL from our viewpoint is that it is implemented and allows for mechanized proofs, given in PVS. However, PVS is not readily programmable, and the use of sequents complicates the formulation of

tactics. The natural deduction style of our framework is more conducive to that task.

References

[1] Anderson CA. The paradox of the knower. Journal of Philosophy, 1983, 80(6): 338–355.

[2] Arkoudas K. Athena. http://www.pac.csail.mit.edu/athena

[3] Arkoudas K, Bringsjord S. Metareasoning for multi-agent epistemic logics. Fifth International Conference on Computational Logic In Multi-Agent Systems. Lecture Notes in Artificial Intelligence. Springer: New York, 2005, 3487: 111-125.

[4] Arvizo T. A virtual machine for a type-ω denotational proof language [Masters Thesis]. MIT, 2002.

[5] Bello P, Bignoli P, Cassimatis N. Attention and association explain the emergence of reasoning about false beliefs in young children. Proc. of the 8th International Conference on Cognitive Modeling, 2007.

[6] Cohen PR, Levesque HJ. Intention is choice with commitment. Artificial Intelligence, 1990, 42: 213–261.

[7] Davies M, Stone T. Folk psychology: the theory of mind debate. Blackwell Publishers, 1995.

[8] Dennett D. Conditions of Personhood. Brainstorms: philosophical essays on mind and psychology. Bradford Books, Montgomery, VT, 1978: 267–285.

[9] des Rivières J, Levesque HJ. The consistency of syntactical treatments of knowledge. Proc. of the 1st Conference on Theoretical Aspects of Reasoning about Knowledge, Morgan Kaufmann, 1986: 115–130.

[10] Engelmore R, Morgan T. Blackboard Systems. Addison-Wesley, 1988.

[11] Fagin R, Halpern J, Moses Y, et al. Reasoning about Knowledge. MIT Press, Cambridge, Massachusetts, 1995.

[12] Goodman ND, Bonawitz EB, Baker CL, Mansinghka VK, Gopnik A, Wellman H, Schulz L, Tenenbaum JB. Intuitive theories of mind: A rational approach to false belief. Proc. of the Twenty-Eight Annual Conference of the Cognitive Science Society, 2006.

[13] Montague R. Syntactical treatment of modality, with corollaries on reflection principles and finite axiomatizability. Acta Philosophica Fennica, 1963, 16: 153–167.

[14] Morreau M, Kraus S. Syntactical treatments of propositional attitudes. Artificial Intelligence, 1998, 106: 161–177.

[15] Owre S, Shankar N, Rushby JM. The PVS specification language (draft). Research Report, Computer Science Laboratory, SRI International, Menlo Park, California, February 1993.

[16] Rao AS, Georgeff MP. Modeling rational agents within a BDI-architecture. Proc. of Knowledge Representation and Reasoning (KR&R-91), 1999: 473–484.

[17] Shapiro S, Lespérance Y, Levesque HJ. The cognitive agents specification language and verification environment for multiagent systems. The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, : 19–26.

[18] Sturm H, Wolter F, Zakharyaschev M. Common Knowledge and Quantification. Economic Theory, 2002, 19: 157–186.

[19] Thomason R. A Note on Syntactical Treatments of Modality. Synthese, 1980, 44: 391–395.

[20] Voronkov A. The anatomy of Vampire: implementing bottom-up procedures with code trees. Journal of Automated Reasoning, 1995, 15(2).

[21] Watt SNK. Seeing things as people [Ph.D. Thesis]. Knowledge Media Institute, Open University, UK, 1997.

[22] Weidenbach C. Combining superposition, sorts, and splitting. In: Robinson A, Voronkov A, eds. Handbook of Automated Reasoning, North-Holland, 2001, 2.

[23] Wimmer H, Perner J. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. Cognition, 1983, 13: 103–128.

[24] Wolter F. First Order Common Knowledge Logics. Studia Logica, 2000, 65: 249–271.

[25] Wooldridge M. Reasoning About Rational Agents. MIT Press, 2000.