Simulative Inference About Nonmonotonic Reasoners

Aaron N. Kaplan
Computer Science Department
University of Rochester
kaplan@cs.rochester.edu

Abstract

If one has attributed certain initial beliefs to an agent, it is sometimes possible to reason about further beliefs the agent must hold by observing what conclusions one's own reasoning mechanism draws when given the initial beliefs as premises. This technique is called simulative inference. In an earlier paper, we described a logic of belief in which the reasoning that generates beliefs is modeled explicitly as a computational process. We used this logic to characterize a class of computational inference mechanisms for which simulative inference is sound, under the assumption that the observer and the observed have similar mechanisms. In this paper, we present a different form of simulative inference, and show that unlike the earlier form, it is sound even for some mechanisms that perform defeasible inference.

1 Introduction

In [4], we introduced a semantics of belief that avoids the logical omniscience problem by describing the generation of a belief set from initial premises as a finite computational process. In the model, each agent has a *belief machine*, an abstraction of a computational inference mechanism, which is described by the two recursive functions *ASK* and *TELL*. Each is a function of two arguments, the first being a state of the machine, and the second a sentence of a logic. The value of $ASK(S, \varphi)$ is either *yes* or *no*, indicating whether an agent whose belief machine is in state S believes the sentence φ or not. *TELL* is the machine's state transition function: the value of $TELL(S, \varphi)$ is another state, the one to which a machine starting in state S will go when φ is asserted.

We take "α believes φ" to mean that agent α can decide with little effort that sentence φ follows from what he has learned. If a sentence follows from what α has learned, but it would take a significant amount of reflection to discover this connection, then we do not say that α believes that sentence, only that with sufficient time he could come to believe it. Therefore, the functions *ASK* and *TELL* needn't describe an agent's entire reasoning capability; they are merely intended to describe the inferences the agent makes easily and automatically.

Our motivation for introducing this computational model of belief was to explore the technique of *simulative inference*, which is reasoning about another agent's beliefs by simulating its inference processes with one's own. The idea is that if an agent is known to believe φ, and by introspective observation we see that believing φ would cause us to believe ψ as well, then we can attribute belief in ψ to the other agent. Using the vocabulary of our model of belief, this type of reasoning can be formalized by the assumption that all agents' belief machines are characterized by the same *ASK* and *TELL* functions, and the following inference rule (where S_0 is the initial state of the belief machine): if α believes $\varphi_1, \ldots, \varphi_n$, and $ASK(TELL(S_0, \varphi_1, \ldots, \varphi_n), \psi) = yes$, then α believes ψ.

Even under the assumption that our inference mechanism is identical to that of the agent we are simulating, this style of reasoning may or may not yield correct results, depending on the type of inference the mechanism performs. For example, here is a natural example of a way one might characterize the conclusions an agent makes easily and automatically: if a sentence can be proved in fewer than five steps (in a given proof system) from what the agent has learned, then that is an "easy" inference, so the agent believes the sentence; otherwise, the agent doesn't currently believe it. This characterization is simple to encode as a belief machine, by making the machine a theorem prover that prunes its search at five steps. For this characterization of easy inference, simulative inference is *not* a valid reasoning technique. If an agent has learned only φ, and ψ can be proved in fewer than five steps from φ, then the agent also believes ψ, and this fact can (correctly) be discovered by simulative inference. Once we know that the agent believes φ and ψ, we might apply simulative inference again, discover that χ can be proved in fewer than five steps from φ and ψ, and conclude that the agent also believes χ. But if χ takes more than five steps to prove from φ alone, then the agent does not in fact believe χ, so our conclusion would be wrong.

While the idea of simulative inference has long been present in the literature (e.g. Moore [7]), no previous semantics of belief could be used satisfactorily to characterize inference mechanisms for which the technique is appropriate. In [5] we gave a set of constraints relating the functions *ASK* and *TELL*, and showed that simulative inference is sound given any belief machine that satisfies these constraints. We also showed that a certain set of inference rules, which includes the rule of simulative inference, is refutation complete for a certain kind of sentence (sentences that don't use universal quantification into positively embedded belief contexts). The details of the model are reviewed briefly in Section 2.

1.1 Simulative Inference and Nonmonotonic Belief Machines

If an agent's set of beliefs can only be augmented when the belief machine is *TELL*ed a new sentence, then the belief machine reasons monotonically. If, on the other hand, some previously held beliefs might be discarded in the light of new information, then the machine reasons nonmonotonically. There are two distinct types of nonmonotonic reasoning that a belief machine might perform. First, if the belief machine discovers that the premises it has been *TELL*ed are not consistent, it might revise its beliefs in a way that involves choosing one or more of the input sentences to discard in order to avoid the inconsistency. Belief machines that can revise their beliefs in this way are permitted under the constraints given in [5], and therefore the rule of simulative inference given there can be sound for such machines.

The other way a belief machine might reason nonmonotonically is by tentatively drawing conclusions that are not strictly entailed by the *TELLed* sentences, and then withdrawing those conclusions if necessary in light of information *TELLed* at a later time. In this case it is not previously *TELLed* input sentences that are being discarded, only sentences that were inferred (defeasibly) from the inputs. Using a machine that reasons in this way, the type of simulative inference described in [5] is *not* sound. Suppose an agent believes that Tweety is a bird, and has no further information about Tweety. The agent might believe that Tweety can fly. Then suppose the agent learns that in fact Tweety is a penguin. This new information does not contradict anything the agent had learned explicitly, but it may well cause it to cease believing the inferred proposition that Tweety can fly. Given a belief machine that reasons in this way, the rule of simulative inference as we have stated it is nonmonotonic: when applied to the singleton premise set $\{B(a, bird(tweety))\}$ (meaning "a believes that Tweety is a bird"), the rule licenses the conclusion $B(a, flies(tweety))$, but when applied to

$$\{B(a, bird(tweety)), B(a, penguin(tweety))\},$$

which contains the former premise set, it does not license that same conclusion. The premises to which simulative inference is applied are the inputs to the belief machine, so if the belief machine reasons nonmonotonically from those premises, then the inference rule is nonmonotonic as well (and therefore is not sound). In this paper, we present a different form of simulative inference that is sound for a class of belief machines that includes some that perform defeasible inference.

Consider an agent who believes that all birds except penguins fly, and assumes that a given bird is not a penguin unless it has reason to believe otherwise. The fact that this agent believes Tweety is a bird is not sufficient evidence to conclude that he believes Tweety flies. However, if we also know explicitly that he doesn't believe Tweety is a penguin, then we would be justified in concluding that he believes Tweety flies. This demonstrates that simulative inference using a belief machine that reasons defeasibly may require information about the absence of certain beliefs, as well as information about the presence of beliefs.

If a belief machine is able to perform *negative introspection*, then there is a way to enter information about the absence of a belief into a simulation. Let me be a special indexical constant that each agent uses to refer to itself. A belief machine has negative introspection if whenever it answers *no* to a sentence φ, it answers *yes* to $\neg B(me, \varphi)$. For a machine with negative introspection, if $\neg B(\alpha, \varphi)$ is true, then so is $B(\alpha, \neg B(me, \varphi))$. This yields a sentence $\neg B(me, \varphi)$ that can be *TELLed* to a simulation of α's reasoning. We will show that for belief machines with negative introspection, the rule presented in this paper is complete.

2 The Computational Model

For the details of the computational model of belief, see [5]. We summarize here.

Models assign truth values to formulas of a language L, which is ordinary first-order logic plus a belief operator B. Where α is a term and φ is a sentence (a formula with no free variables), $B(\alpha, \varphi)$ is a formula whose intended meaning is that α believes φ. In [5], the formula φ was allowed to have open variables, so that quantifying-in (e.g. $\exists x B(a, P(x))$) was possible, but in this paper we only consider a simpler language that doesn't permit quantifying-in. Note that the belief φ is a sentence of L itself—the language used by the belief machine's *ASK/TELL* interface is the same as the language of the observer. Belief contexts can be nested to arbitrary depth, e.g. $B(a, B(b, B(c, P(d))))$.

A belief machine is a structure $\langle \Gamma, S_0, \mathit{TELL}, \mathit{ASK}, \rangle$, where Γ is a set of states, $S_0 \in \Gamma$ is the initial state (the state the machine is in before having been *TELL*ed anything), and *TELL* and *ASK* are the state transition function and the query function, as described above. *ASK* and *TELL* are defined only for sentences. Note that *ASK* and *TELL* are part of the semantics, not symbols of the language L.

A model is a structure like a model for ordinary FOL, but augmented with a function that maps each individual in the domain to a belief state. We will be interested in the truth values of formulas given a particular belief machine, so the machine is a parameter of the model structure rather than a constituent thereof. In other words, when we ask if $B(a, \varphi)$ logically entails $B(a, \psi)$, we are not asking if there is *any* belief machine that would conclude ψ from φ. Rather, we are asking about the belief machine we have already chosen. Where m is a belief machine, an m-model is a structure $\langle D, I, \gamma \rangle$, where D is the domain of individuals, I is an interpretation function, and γ is a function mapping each individual in D to a belief state in Γ. The notation $|\tau|^M$ means the denotation of term τ under model M.

Let $m = \langle \Gamma, S_0, \mathit{TELL}, \mathit{ASK} \rangle$ be a belief machine, and $M =< D, I, \gamma >$ an m-model. For any term α and sentence φ, the belief atom $B(\alpha, \varphi)$ is true under M if $\mathit{ASK}(\gamma(|\alpha|^M), \varphi) = yes$, i.e. if the belief machine of the agent denoted by α answers yes to the query φ.

The notation $\mathcal{B} \cdot S$ means the belief set of belief state S, i.e.

$$\mathcal{B} \cdot S = \{\varphi | \mathit{ASK}(S, \varphi) = yes\}.$$

We write the rule of simulative inference as follows, where the formulas above the line are the premises, the formula below the line is the conclusion, and the rule applies only when the condition written below it holds:

$$\frac{B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n)}{B(\alpha, \psi)}$$

if $\mathit{ASK}(\mathit{TELL}(S_0, \varphi_1, \ldots, \varphi_n), \psi) = yes$.

To the semantics introduced in [4], we now add an indexical constant me, which will be necessary for constructing an introspective belief machine. Expressions will now be evaluated with respect to a model $M = \langle D, I, \gamma \rangle$ and a reasoner $r \in D$. Denotations of terms are as before, except that $|me|^{M,r} = r$.

3 Simulative Consistency Checking

In what follows, we will say that a sequence of sentences $\varphi_1, \ldots, \varphi_n$ is *acceptable* if

$$\mathit{ASK}(\mathit{TELL}(S_0, \varphi_1, \ldots, \varphi_n), \varphi_i) = \mathit{yes}$$

for all $1 \leq i \leq n$, and if all initial subsequences of $\varphi_1, \ldots, \varphi_n$ are also acceptable (defined recursively). That is, a sequence is acceptable if, as each of its elements is *TELL*ed to a belief machine starting in the initial state, the machine continues to believe all of the elements that have been entered so far. For example, it may be the case for a particular machine that the sequence $P(c), Q(d)$ is acceptable, but the sequence $P(c), \neg P(c)$ is not. If one were to *TELL* such a machine $P(c)$ followed by $\neg P(c)$, it would detect that the two input sentences contradict each other, and therefore choose one of them not to believe.

In order to obtain the completeness result in [5], we found it necessary to introduce another inference rule that involves a different form of simulation. The rule says that if a sequence of sentences is not acceptable (a condition that can be detected by simulation), then its elements cannot be believed simultaneously. It can be expressed as follows, where $\perp$ stands for an arbitrary contradiction:

$$\frac{B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n)}{\perp}$$

if $\mathit{ASK}(\mathit{TELL}(S_0, \varphi_1, \ldots, \varphi_n), \varphi_i) = \mathit{no}$ for some $1 \leq i \leq n$.

We will call this rule *simulative consistency checking*. Theorem 1 states that it is sound for any belief machine that satisfies the following three constraints. These constraints are identical to ones used in [5] except that the condition of monotonic acceptability used in [5] has been replaced by the weaker condition of acceptability. These weakened constraints permit machines that perform defeasible inference.

C1 (closure) *For any belief state S and sentence φ, if*

$$\mathit{ASK}(S, \varphi) = \mathit{yes}$$

then

$$\mathcal{B} \cdot \mathit{TELL}(S, \varphi) = \mathcal{B} \cdot S.$$

C1 says that *TELL*ing the machine something it already believed does not change its belief set. This does not mean that the belief *state* may not change—for example, φ might be tentatively assumed as a defeasible inference in state S, but believed with full confidence in state $\mathit{TELL}(S, \varphi)$.

C2 (finite basis) *For any belief state S, there exists an acceptable sequence of sentences $\varphi_1, \ldots, \varphi_n$ such that*

$$\mathcal{B} \cdot \mathit{TELL}(S_0, \varphi_1, \ldots, \varphi_n) = \mathcal{B} \cdot S$$

Constraint C2 says that for each belief state, a state with the same belief set can be reached from the initial state by *TELL*ing the machine a finite, acceptable sequence of sentences. It requires that even if a particular state can be reached only via a non-acceptable sequence, there is another state with the same belief set that can be reached via an acceptable sequence.

C3 (commutativity) *For any belief state S and acceptable sequence of sentences* $\varphi_1, \ldots, \varphi_n$, *and for any permutation* ρ *of the integers* $1 \ldots n$, *the sequence* $\varphi_{\rho(1)}, \ldots, \varphi_{\rho(n)}$ *is also acceptable; and*

$$\mathcal{B} \cdot TELL(S, \varphi_1, \ldots, \varphi_n) = \mathcal{B} \cdot TELL(S, \varphi_{\rho(1)}, \ldots, \varphi_{\rho(n)}).$$

C3 says that if a sequence of sentences is acceptable, then it is acceptable in any order, and the belief set of the resulting state does not depend on the order. Note that this constraint does permit the belief machine to take order into account when deciding how to handle contradictory (i.e. non-acceptable) inputs.

Theorem 1 (Soundness of Simulative Consistency Checking) *Given a belief machine satisfying constraints C1–C3, if*

$$ASK(TELL(S_0, \varphi_1, \ldots, \varphi_n), \varphi_i) = no$$

for some $1 \leq i \leq n$, *then* $\{B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n)\}$ *is unsatisfiable.*

Proof: We will prove the contrapositive of the above statement. Assume that

$$\{B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n)\}$$

is satisfiable, i.e. that there exists some state S such that $ASK(S, \varphi_i) = yes$ for all $1 \leq i \leq n$. By the finite basis constraint, there is some acceptable sequence of sentences $\psi_1, \ldots, \psi_m$ such that

$$\mathcal{B} \cdot TELL(S_0, \psi_1, \ldots, \psi_m) = \mathcal{B} \cdot S,$$

which means that

$$ASK(TELL(S_0, \psi_1, \ldots, \psi_m), \varphi_i) = yes, 1 \leq i \leq n.$$

By the closure constraint, if we take a machine in state $TELL(S_0, \psi_1, \ldots, \psi_m)$ and *TELL* it each of the φ_i in turn, the resulting states will all have the same belief set as the original state, which means that the entire sequence

$$\psi_1, \ldots, \psi_m, \varphi_1, \ldots, \varphi_n$$

is acceptable. Therefore, the commutativity constraint applies; it says that the sequence

$$\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m$$

is acceptable, which by the definition of acceptability means that the initial subsequence $\varphi_1, \ldots, \varphi_n$ is acceptable. □

4 Negative Introspection

A belief machine has negative introspection if it satisfies the following constraint:

C4 (negative introspection) *For every state S and sentence φ, if*

$$ASK(S, \varphi) = no$$

then

$$ASK(S, \neg B(me, \varphi)) = yes.$$

Note that negative introspection is a form of defeasible inference: if $\varphi \notin \mathcal{B} \cdot S_0$ and $\varphi \in \mathcal{B} \cdot TELL(S_0, \varphi)$, and the machine is introspective, then $\neg B(me, \varphi)$ is believed in state S_0 but is retracted in state $TELL(S_0, \varphi)$.

For any belief machine satisfying the negative introspection constraint, the following inference rule is clearly sound:

$$\frac{\neg B(\alpha, \varphi)}{B(\alpha, \neg B(me, \varphi))}.$$

5 Completeness

The combination of the rules of simulative consistency checking and negative introspection is complete in the following sense: for any finite set Φ of belief literals (formulas of the form $B(\alpha, \varphi)$ or $\neg B(\alpha, \varphi)$), if Φ is unsatisfiable, then there is a refutation proof $\Phi \vdash \bot$. The completeness holds for any belief machine that satisfies the four constraints already presented (closure, finite basis, commutativity, and negative introspection) as well as the following, which is the converse of negative introspection:

C5 (negative faithfulness) *if*

$$ASK(S, \neg B(me, \varphi)) = yes$$

then

$$ASK(S, \varphi) = no.$$

Theorem 2 (Completeness for Belief Literals) *For any belief machine satisfying constraints C1–C5, and for any finite set of belief literals*

$$\Phi = \{B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n), \neg B(\alpha, \psi_1,), \ldots, \neg B(\alpha, \psi_m)\},$$

if Φ is unsatisfiable then $\Phi \vdash \bot$.

Proof: Assume that Φ is unsatisfiable. That means there is no belief state S such that $ASK(S, \varphi_i) = yes$ for all $1 \leq i \leq n$ and such that $ASK(S, \psi_i) = no$ for all $1 \leq i \leq m$. Since the negative faithfulness constraint holds, there must be no state S such that $ASK(S, \varphi_i) = yes$ for all $1 \leq i \leq n$ and $ASK(S, \neg B(me, \psi_i)) = yes$ for all $1 \leq i \leq m$. Therefore, the sequence $\varphi_1, \ldots, \varphi_n, \neg B(me, \psi_1), \ldots, \neg B(me, \psi_m)$ must not be acceptable.

A proof $\Phi \vdash \bot$ can be constructed as follows: from each literal $\neg B(\alpha, \psi_i)$, the rule of negative introspection licenses the conclusion $B(\alpha, \neg B(me, \psi_i))$. Then, from

$$B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n), B(\alpha, \neg B(me, \psi_1)), \ldots, B(\alpha, \neg B(me, \psi_m)),$$

the rule of simulative consistency checking licenses the conclusion $\bot$, since the sequence $\varphi_1, \ldots, \varphi_n, \neg B(me, \psi_1), \ldots, \neg B(me, \psi_m)$ is not acceptable. □

This completeness for sets of belief literals has the following corollary: for any belief machine satisfying constraints C1–C5, if the rules of simulative consistency checking and negative introspection are added to a set of inference rules that is complete for ordinary first order logic, then the resulting set of rules is complete for the entire logic (in the same sense of completeness used in Theorem 2: if a finite theory is unsatisfiable, then it has a refutation proof).

The completeness result shown in [5] for the original simulative inference rule was more general: the logic used in that paper permitted quantifying-in, and the completeness result applied to theories containing not only unquantified belief formulas such as $B(a, P(c))$, but also to theories containing existentially quantified-in belief formulas such as $\exists x B(a, P(x))$, though not universally quantified-in formulas such as $\forall x B(a, P(x))$. It remains an open question whether the same completeness condition obtains for the inference rules presented here.

6 Simulative Inference for Machines With Negative Introspection

From the rules of simulative consistency checking, negative introspection, and *reductio ad absurdum*, a new simulative inference rule can be derived:

$$\frac{B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n)}{B(\alpha, \psi)}$$

if $ASK(TELL(S_0, \varphi_1, \ldots, \varphi_n, \neg B(me, \psi)), \chi) = no$ for some $\chi \in \{\varphi_1, \ldots, \varphi_n, \neg B(me, \psi)\}$

The premises and conclusion of this rule are the same as those of the original simulative inference rule. Only the *ASK/TELL* condition under which it applies is different. Theorem 3 shows the derivation of the new inference rule.

Theorem 3 *For any term α and sentences $\varphi_1, \ldots, \varphi_n$ and ψ, if*

$$ASK(TELL(S_0, \varphi_1, \ldots, \varphi_n, \neg B(me, \psi)), \chi) = no$$

for some $\chi \in \{\varphi_1, \ldots, \varphi_n, \neg B(me, \psi)\}$, then there is a proof

$$B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n) \vdash B(\alpha, \psi)$$

using the rules of simulative consistency checking, negative introspection, and reductio ad absurdum.

Proof: The proof can be constructed as follows: begin with the premises

$$B(\alpha, \varphi_1), \ldots, B(\alpha\varphi_n),$$

and make the assumption $\neg B(\alpha, \psi)$ for the purposes of deriving a contradiction. From the assumption, the negative introspection rule licenses the conclusion $B(\alpha, \neg B(me, \psi))$. From this conclusion and the original premises, the rule of simulative consistency checking licenses the conclusion $\perp$, since the sequence $\varphi_1, \ldots, \neg B(me, \psi)$ is not acceptable. Discharging the assumption, we can conclude $B(\alpha, \psi)$ by *reductio ad absurdum.* Here is the proof in tabular form:

1.	$B(\alpha, \varphi_1), \ldots, B(\alpha, \varphi_n)$	premises
2.	Assume $\neg B(\alpha, \psi)$	
3.	$B(\alpha, \neg B(me, \psi))$	2, negative introspection
4.	$\perp$	1,3, simulative consistency checking
5.	$B(\alpha, \psi)$	*reductio ad absurdum*

□

7 Related Work

The classical possible worlds model of Hintikka [3] suffers from the "logical omniscience" problem, meaning that agents in that model must believe all of the logical consequences of their beliefs. Our computational model of belief is one of many later alternatives that model believers in a more realistic (i.e. computationally feasible) way.

The model of algorithmic knowledge of Halpern, Moses, and Vardi [2] is similar to ours in that it models an agent's belief state as the state of a computational mechanism, and its belief set as the set of sentences accepted by that mechanism in its current state. Halpern et al. do not treat the subject of simulative reasoning.

Konolige's deduction model [6] is also similar, and does include a mode of inference similar to our simulative inference. Konolige uses a very specialized model of belief computation, namely the exhaustive application of deductive inference rules. Our *ASK/TELL* model is fundamentally much more general, but our soundness and completeness results are obtained only under certain constraints on *ASK* and *TELL* that reduce this generality. The relative expressiveness of the two models is an interesting topic, which we have discussed at length in [5]. Reasoning in the deduction model is limited to deductive inference; Konolige suggests that one might extend the model to allow nonmonotonic reasoning, but to our knowledge this possibility has not been explored.

Chalupsky and Shapiro [1] describe a proof system in which simulative inference is a defeasible rule. This is a useful idea, because the assumption that the observer and the observed have identical reasoning mechanisms is likely to be wrong at times. However, as we have shown, even in cases where the approximation is correct, simulative inference may give incorrect results, depending on the kind of reasoning performed in the simulation. Our contribution has been to demonstrate conditions under which simulative inference is guaranteed to be sound.

In the work of van Arragon [8], the observer uses a default reasoning tool, and the observed is a user who reasons in the same way as the tool. Rather than using a simulative technique, the observer uses an axiomatic description of the conditions under which the user is able to make an inference. Given that the user and the observer use identical reasoning methods, it seems unlikely that reasoning with a declarative description of the user's behavior could be as efficient as simulating his behavior using the observer's own mechanism. Van Arragon's metalanguage differentiates between sentences the user believes as default assumptions and those he believes as incontrovertible facts. This contrasts with our language, which has only a single belief operator, leaving the distinction between defaults and facts to be made by the operation of the belief machine.

8 Conclusions

In an earlier paper, we introduced a model of belief in which reasoning is modeled as computation performed by a belief machine. We presented a rule of simulative inference, which allows an observer to draw conclusions about what an agent must believe by running a simulation of the agent's belief machine.

In this paper, we have presented another form of simulative inference under the same model of belief. We have shown that the new rule is sound for a broader class of belief machines, a class that includes machines that reason nonmonotonically from consistent inputs. Furthermore, we have shown that when the belief machine exhibits the properties of negative introspection and negative faithfulness, the rule is complete for the subset of the logic in which quantifying-in is disallowed.

Acknowledgments

This work was supported by NSF grant IRI-9503312 and ARPA/SSTO grant F30602-95-1-0025. Len Schubert provided invaluable suggestions and discussions.

References

[1] Hans Chalupsky and Stuart C. Shapiro. Reasoning about incomplete agents. In *Proceedings of the Fifth International Conference on User Modeling*, 1996.

[2] Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Algorithmic knowledge. In Ronald Fagin, editor, *Theoretical Aspects of Reasoning about Knowledge: Proc. Fifth Conference*, pages 255–266, San Francisco, 1994. Morgan Kaufmann.

[3] Jaakko Hintikka. *Knowledge and Belief.* Cornell University Press, Ithaca, New York, 1962.

[4] Aaron N. Kaplan and Lenhart K. Schubert. Simulative inference in a computational model of belief. In *Second International Workshop on Computational Semantics*, pages 107–121, Tilburg, The Netherlands, January 1997. Revised version to appear in Kluwer's *Studies in Linguistics and Philosophy* series.

[5] Aaron N. Kaplan and Lenhart K. Schubert. Simulative inference in a computational model of belief. Technical Report 636, University of Rochester Computer Science Department, 1997.

[6] Kurt Konolige. *A Deduction Model of Belief.* Morgan Kaufmann Publishers, Inc., Los Altos, California, 1986.

[7] Robert C. Moore. Reasoning about knowledge and action. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 1, pages 223–227, 1977.

[8] Paul van Arragon. Modeling default reasoning using defaults. *User Modeling and User-Adapted Interaction*, 1(3):259–288, 1991.