A Semantic Characterization of an Algorithm for Estimating Others' Beliefs from Observation

Hideki Isozaki and Hirofumi Katsuno
NTT Basic Research Laboratories
3-1 Morinosato-Wakamiya, Atsugi, Kanagawa, Japan 243-01
{isozaki,katsuno}@theory.brl.ntt.jp

Abstract

Human beings often estimate others' beliefs and intentions when they interact with others. Estimation of others' beliefs will be useful also in controlling the behavior and utterances of artificial agents, especially when lines of communication are unstable or slow. But, devising such estimation algorithms and background theories for the algorithms is difficult, because of many factors affecting one's belief. We have proposed an algorithm that estimates others' beliefs from observation in the changing world. Experimental results show that this algorithm returns natural answers to various queries. However, the algorithm is only heuristic, and how the algorithm deals with beliefs and their changes is not entirely clear. We propose certain semantics based on a nonstandard structure for modal logic. By using these semantics, we shed light on a logical meaning of the belief estimation that the algorithm deals with. We also discuss how the semantics and the algorithm can be generalized.

Introduction

Human beings often estimate others' beliefs and intentions when they interact with others. Estimation of others' beliefs will be useful also in controlling the behavior and utterances of artificial agents such as robots, especially when lines of communication are unstable or slow.

Suppose Alice, Bob, and Charlie are researchers in a research laboratory. The laboratory has a computer room whose light can be turned off with a switch on its outside wall. One cannot see whether the light is on or off when the door is closed. In the initial world, Alice and Bob were working in the computer room, while Charlie was working in his office. Later, Alice and Bob left the computer room. Alice closed and locked the door, and turned off the switch. Since Bob saw Alice turn off the switch, he believes that the light is off without checking this by opening the door. Then Alice believes that Bob believes that the light is off because Alice believes that Bob observed her action. On the other hand, Alice does not believe that Charlie believes that the light is off, because she believes he can observe neither the light nor her actions. Hence, she might want to inform Charlie that the light is off and that the door is locked, but she will not inform Bob of these facts.

The above example illustrates how one can estimate others' beliefs based on one's observations, without the use of utterances. We have proposed an algorithm that estimates others' beliefs from observation in the changing world (Isozaki 1995). Experimental results show that this algorithm returns natural answers to various queries. For this estimation, it uses domain knowledge represented by a database about the initial world, postconditions of events, incompatibility relations of propositions, and observability conditions of events and propositions.

We use a symbol a_0 to indicate the agent that executes our algorithm. The function $\mathsf{belief}(p, k, m)$ defined in the algorithm (Fig. 1) returns yes, no, or unknown depending on whether a_0 should believe that a_1 believes that a_2 believes that ... that a_k believes that p is true when a_0 has observed a sequence of events $e^1, \ldots, e^m$ in this order. e^j indicates an event that a_0 observed on the transition from time point $j-1$ to j. If a_0 was not able to observe any event at that time, we assume $e^j = \mathsf{nop}$.

However, the algorithm is only heuristic, and how the algorithm deals with beliefs and their changes is not entirely clear. We propose certain semantics based on a nonstandard structure for modal logic. By using these semantics, we shed light on a logical meaning of the belief estimation that the algorithm deals with. We also discuss how the semantics and the algorithm can be generalized.

In modal logic, belief is usually represented by modal operators characterized by the axiom system $\mathrm{KD45}_n$ or $\mathrm{K45}_n$ (Fagin *et al.* 1995). The modal formula $B_a\phi$ means that agent a believes ϕ. Some researchers are working on extensions of the standard semantics of the logic to mitigates the problem of logical omniscience (i.e., the constraint that each agent has to believe all logical consequences of its belief) and to represent changes in belief. Fagin et al. proposes an *alternate nonstandard structure* to represent incomplete and incoherent beliefs (Fagin *et al.* 1995). As another approarch, Meyden proposes *labelled trees* equivalent to

$K45_n$ situations to represent changes in belief (van der Meyden 1994b).

We employ *labelled trees for alternate nonstandard structures* as a semantic structure for logic of belief. If a_0 has an initial belief modelled by a labelled tree $\mathcal{T}(0)$ and observes a sequence of events $e^1, \ldots, e^m$, then the tree becomes $\mathcal{T}(1)$ after e^1, then $\mathcal{T}(2)$, and finally $\mathcal{T}(m)$. In the next section, we describe the algorithm and define satisfaction of a formula $\models$ under $\mathcal{T}$s. Next, we show that the algorithm corresponds to the trees in the following way.

- $\mathsf{belief}(p, k, m) = \mathsf{yes}$ iff $\mathcal{T}(m) \models B_{a_0} B_{a_1} \cdots B_{a_k} p$.
- $\mathsf{belief}(p, k, m) = \mathsf{no}$ iff $\mathcal{T}(m) \models B_{a_0} B_{a_1} \cdots B_{a_k} \neg p$.
- $\mathsf{belief}(p, k, m) = \mathsf{unknown}$ iff $\mathcal{T}(m) \not\models B_{a_0} B_{a_1} \cdots B_{a_k} p$ and $\mathcal{T}(m) \not\models B_{a_0} B_{a_1} \cdots B_{a_k} \neg p$.

Finally, we discuss the results and related works.

Methodology

Here we will show a propositional version of the algorithm and semantics of beliefs. Let Φ be the finite set of primitive propositions and E the finite set of events. The set of agents is $A = \{1, \ldots, n\}$. We simplify the task of belief estimation by the following assumptions.

Disregard of messages The interaction between messages (or speech acts) and participants' beliefs is very complicated and controversial (Perrault 1990; Cohen & Levesque 1990; Appelt & Konolige 1988; van der Meyden 1994b). To circumvent the problems associated with such interaction, the algorithm ignores the influence of messages on beliefs, and estimates others' beliefs only from observation of the external world. We have found a way to extend the algorithm for *honest utterances* (Isozaki 1996), but for simplicity in the present work, we will not discuss it here.

Persistence of states While we can find many propositions that tend to persist. we usually do not consider the possibility that they may change. Therefore, we assume that an agent believes that an observed proposition persists until the agent observes another proposition or an event that negates the proposition. It is well known that simple persistence does not always hold true when one guesses about events (ex. stolen car problem (Kautz 1986)). In the present work, we separate such abductive reasoning from belief estimation, and focus on estimation of current beliefs about the current world.

Disregard of preconditions We assume that the algorithm is being applied to an observed action sequence. This implies that every action must have satisfied its preconditions immediately before it took place. This in turn obviates the need to check the preconditions of others' actions. Although it is possible to make an algorithm that checks preconditions, it will not work for complex worlds, because the reasoner's belief alone is often insufficient for checking preconditions. Moreover, preconditions are usually less significant than postconditions in estimating agents' beliefs about a current situation. It is more important to make a belief estimation algorithm simple and fast, even if it fails in some cases.

Simplification of the world changes It is assumed that every event has zero duration and no two events occur simultaneously. We assume a discrete time structure, and specify a time point with a natural number; 0 means the initial state, 1 means the next time point, and so on. We allow only one event e^t on the transition from time point $t-1$ to t. No occurrence of an event is represented by $e^t = \mathsf{nop}$.

Simplification of observations Although in reality one is not always aware of all observable things, we assume here that every agent is in fact always aware of all observable things.

Moreover, we assume that agents have the following **common knowledge**.

Postconditions The postcondition $\mathsf{post}(e)$ of an event e is a set of primitive propositions that become true by the fact of the event's occurrence. We assume that all postconditions are common knowledge. When an agent was not able to observe any event at a certain time, define this as the agent observing the event nop where $\mathsf{post}(\mathsf{nop}) = \{\}$.

Observability conditions An observability condition $\mathsf{ob}[a, x]$ is a propositional formula that determines whether agent a can observe $x (\in \Phi \cup E)$. Agent a can observe p if both p and $\mathsf{ob}[a, p]$ are true. Note that $\mathsf{ob}[a, p]$ does not address the observability of $\neg p$.[1] Agent a can observe an occurrence of event e if $\mathsf{ob}[a, e]$ is true immediately before the occurrence. We assume that all observability conditions are common knowledge and are given in DNF (Disjunctive Normal Form).

Integrity constraints When an agent receives new information, it changes its belief to make its belief consistent with the new information. A constraint that must be satisfied after every belief change is called an *integrity constraint*. We cannot expect that all agents have the ability to maintain arbitrary integrity constraints.

However, there is a restricted class of integrity constraints that most agents can maintain. For example, if one is in California, one is not in Oregon at the same time. This illustrates that a certain pair of primitive propositions cannot hold at the same time. From this point, we will consider only this restricted class of integrity constraints: *binary nogoods*. Accordingly

[1] Suppose a security guard is watching a department store through a TV camera. If the guard percieves a lady on the TV screen, she must necessarily be in the store. However, the guard cannot say definitely that she is not in the store when she is not on the screen.

each $p \in \Phi$ has a fixed set of primitive propositions that cannot hold when p holds. This set $N(\{p\})$ is called p's *incompatible proposition set*, and we define $N(S) \stackrel{\text{def}}{=} \cup_{p \in S} N(\{p\})$.

Belief estimation algorithm

Figure 1 shows Algorithm 1, a propositional version of our algorithm (Isozaki 1995). Algorithm 1 checks the following three factors.

Observation If one is observing a proposition now, one believes the proposition now.

Effects If one has just observed an event, then one believes in all of its expected effects, even if one has not yet observed them.

Memory If no new information is available, one's previous belief remains valid.

The upper half of the definition of the function belief computes one's own belief. The lower half computes one's belief about someone else's belief. Each part checks observation, effects, and memory, in that order. Functions obs_one and obs_all check observability.

One's own belief If the query is about a_0's initial belief, Algorithm 1 determines its value by the input $\mathcal{I}^+$ and $\mathcal{I}^-$ where $\mathcal{I}^+$ (or $\mathcal{I}^-$) is the set of primitive propositions that a_0 believes true (or false) in the initial state. They should satisfy the following conditions:

- $\mathcal{I}^+ \cap \mathcal{I}^- = \{\}$: a_0 does not believe both p and $\neg p$ in the initial state.
- $N(\mathcal{I}^+) \subseteq \mathcal{I}^-$: if a_0 believes p, he/she should believe that any proposition in $N(\{p\})$ is false.
- $\mathcal{I}^+$ should contain all of the primitive propositions that a_0 is able to observe in the initial state.

If the query is about a_0's belief at time $m(\geq 1)$, Algorithm 1 checks a_0's *observation at time* m (the input $Ob(m)$) and the expected *effects* of e^m. If the above two factors do not give yes/no, Algorithm 1 checks a_0's *memory*, i.e., a_0's belief at time $m-1$.

Someone else's belief If the given query is about someone else's (a_k's) initial belief, Algorithm 1 checks a_k's *observation* in the initial state. If the query is about a_k's belief at time $m(\geq 1)$, it checks a_k's *observation*, e^m's *effects*, and a_k's *memory*.

The function obs_one(x, k, m) determines if a_k can observe x according to a_0's belief about a_1's belief about $\ldots a_{k-1}$'s belief. Since ob$[a, x]$ is in DNF, the function tries to find a true disjunct in ob$[a, x]$. T_l in the algorithm is a disjunct and L_u is a literal in the disjunct. $|L_u|$ is a primitive proposition in L_u.

If a_h does not notice the occurrence of e^m, a_h will not take e^m into account when it estimates others' beliefs. To compensate this, the function obs_all$(e^m, k, m-1)$ checks the observability conditions of all agents in the list $a_1, \ldots, a_k$.

INPUT:

p	a query proposition
$\mathcal{I}^+, \mathcal{I}^-$	the initial belief
$a_1, \ldots, a_k$	an agent list ($a_h \neq a_{h-1}$)
$e^1, \ldots, e^m$	a sequence of observed events
$Ob(1), \ldots, Ob(m)$	a sequence of the sets of observed propositions
$N(\{p\})$	p's incompatible proposition set ($p \in \Phi$)
post(e)	e's postconditions ($e \in E$)
ob$[a, x]$	observability condition ($a \in A$, $x \in \Phi \cup E$)

OUTPUT: belief(p, k, m)

```
function belief(p, k, m) {
  if (k = 0) { % ONE'S OWN BELIEF
    if (m = 0) { % INITIAL BELIEF
      if (p ∈ I+) {return (yes);}
      if (p ∈ I-) {return (no);}
      return (unknown);}
    % OBSERVATION
    if (p ∈ Ob(m)) {return (yes);}
    if (∃q ∈ N({p}) ∩ Ob(m)) {return no;}
    % EFFECTS
    if (p ∈ post(e^m)) {return (yes);}
    if (∃q ∈ N({p}) ∩ post(e^m)) {return (no);}
    % MEMORY
    return belief(p, k, m - 1);
  }
  % ESTIMATION OF OTHERS' BELIEF
  % OBSERVATION
  if (belief(p, k - 1, m) = yes and
      obs_one(p, k, m) = yes) {return (yes);}
  if (∃q ∈ N({p}) s.t. belief(q, k - 1, m) = yes
      and obs_one(q, k, m) = yes) {return (no);}
  % EFFECTS
  if (m = 0) {return (unknown);}
  if (obs_all(e^m, k, m - 1) = yes) {
    if (p ∈ post(e^m)) {return (yes);}
    if (∃q ∈ N({p}) ∩ post(e^m)) {return (no);}
  }
  % MEMORY
  return belief(p, k, m - 1);
}
function obs_one(x, k, m){
  if (∃T_l ∈ ob[a_k, x](= T_1 ∨ ··· ∨ T_t) s.t.
      ∀L_u ∈ T_l(= L_1 ∧ ··· ∧ L_s) :
      (L_u = |L_u| and belief(|L_u|, k - 1, m) = yes) or
      (L_u = ¬|L_u| and belief(|L_u|, k - 1, m) = no)) {
      return (yes);} else {return (no);}
}
function obs_all(e^m, k, m - 1){
  if (∀a_i(1 ≤ i ≤ k) : obs_one(e^m, i, m - 1) = yes) {
      return (yes);} else {return (no);}
}
```

Figure 1: Algorithm 1 — A propositional version of the belief estimation algorithm (Isozaki 1995)

Semantics of beliefs

We use a propositional modal language $\mathcal{L}_n$ generated over the set of primitive propositions Φ. $\mathcal{L}_n$ contains a modal operator B_a for each agent a to represent a's belief.

Kripke structures A Kripke structure is a tuple $M = \langle W, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$ where W is a set of worlds, π is an interpretation that gives a truth assignment $\pi(w) : \Phi \to \{\mathsf{true}, \mathsf{false}\}$ for each $w \in W$, and $\mathcal{B}_i$ is a binary relation on W. A K45$_n$ structure is a Kripke structure where each $\mathcal{B}_i$ is transitive and Euclidean (Chellas 1980). A K45$_n$ situation is a pair of (M, w) where M is a K45$_n$ structure and w is a world of M. Satisfaction of a formula in a situation can be defined as usual (Fagin *et al.* 1995).

If a is smart enough, we can expect $B_a\phi \wedge B_a(\phi \Rightarrow \psi) \Rightarrow B_a\psi$ (distribution axiom), $B_a\phi \Rightarrow B_aB_a\phi$ (positive introspection), and $\neg B_a\phi \Rightarrow B_a\neg B_a\phi$ (negative introspection). K45$_n$ structures have these properties.

Alternate nonstandard structures However, Kripke structures are not appropriate for representing the beliefs of logically incomplete agents. Fagin et al. proposed an *alternate nonstandard structure* ((Fagin *et al.* 1995), p. 350), which is a tuple $\langle W, \pi, \mathcal{B}_1{}^+, \ldots, \mathcal{B}_n{}^+, \mathcal{B}_1{}^-, \ldots, \mathcal{B}_n{}^- \rangle$ where W is a set of worlds, π is an interpretation that gives a *nonstandard truth assignment* $\pi(w)$ for each $w \in W$, and each $\mathcal{B}_i{}^+$ and $\mathcal{B}_i{}^-$ is a binary relation in W. A nonstandard truth assignment τ gives each literal a truth value: $\tau : \Phi \cup \neg\Phi \to \{\mathsf{true}, \mathsf{false}\}$ where $\neg\Phi \stackrel{\text{def}}{=} \{\neg p \mid p \in \Phi\}$. Satisfaction of formulas in a nonstandard situation is defined as follows:

- $(M, w) \models p$ $(p \in \Phi)$ iff $\pi(w)(p) = \mathsf{true}$,
- $(M, w) \models \phi \wedge \psi$ iff $(M, w) \models \phi$ and $(M, w) \models \psi$,
- $(M, w) \models B_i\phi$ iff $(M, x) \models \phi$ for any x that satisfies $(w, x) \in \mathcal{B}_i{}^+$,
- $(M, w) \models \neg p$ $(p \in \Phi)$ iff $\pi(w)(\neg p) = \mathsf{true}$,
- $(M, w) \models \neg(\phi \wedge \psi)$ iff $(M, w) \models \neg\phi$ or $(M, w) \models \neg\psi$,
- $(M, w) \models \neg\neg\phi$ iff $(M, w) \models \phi$,
- $(M, w) \models \neg B_i\phi$ iff $(M, x) \not\models \phi$ for some x that satisfies $(w, x) \in \mathcal{B}_i{}^-$.

Disjunction $\phi \vee \psi$ is given by $\neg(\neg\phi \wedge \neg\psi)$ as usual. Since implication $\phi \Rightarrow \psi$ given by $\neg\phi \vee \psi$ does not capture the intuition of if-then, we introduce *strong implication* $\phi \hookrightarrow \psi$ (Fagin *et al.* 1995).

- $(M, w) \models \phi \hookrightarrow \psi$ iff $(M, w) \not\models \phi$ or $(M, w) \models \psi$.

Labelled trees Since a K45$_n$ situation describes one's beliefs, belief change can be considered as an operation that transforms one K45$_n$ situation into another K45$_n$ situation. But handling transformations of situations directly is difficult to cope with this. Meyden has introduced a *labelled tree*, which can be constructed by *unravelling* a K45$_n$ structure (van der Meyden 1994b). Figure 2 shows a labelled tree. Its root

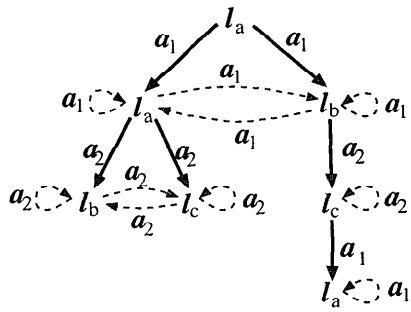

Figure 2: A labelled tree (van der Meyden 1994b)

vertex corresponds to the external world. Two vertices accessible from the root show worlds that agent a_1 in this world regards as possible. l_a, l_b, and l_c are vertex labels, and a_1 and a_2 are edge labels. Meyden shows the equivalence of labelled trees and K45$_n$ situations and represents nested belief change as rewriting a labelled tree.

Let L be the whole set of vertex labels. Each vertex label $l \in L$ has a truth assignment $\pi(l)$. Each vertex v in a labelled tree is uniquely represented by an alternate sequence of vertex labels and edge labels $l_0a_0l_1a_1 \cdots l_{k-1}a_{k-1}l_k$ where $k \geq 0$, $l_j \in L$, and $a_j \neq a_{j+1}$ for all $j \geq 0$. Such a sequence is called an (L, n)-sequence. The last label l_k of v is denoted $\lambda(v)$. v's parent $\mathsf{par}(v)$ is a vertex specified by $l_0a_0l_1a_1 \cdots l_{k-1}$. Thus, a labelled tree $\mathcal{T}$ is a set of (L, n)-sequences that is *closed under prefixes*[2] and that contains a unique (L, n)-sequence l_0, which is $\mathcal{T}$'s root ($\mathsf{root}(\mathcal{T})$).

We can construct from a labelled tree $\mathcal{T}$ a K45$_n$ structure $M(\mathcal{T}) = \langle \mathcal{T}, \pi_M, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$ where $\pi_M(w) = \pi(\lambda(w))$ and each $\mathcal{B}_i (\subseteq \mathcal{T} \times \mathcal{T})$ is given by:

$$\mathcal{B}_i \stackrel{\text{def}}{=} \{(v, w) \mid w = v\, i\, \lambda(w) \text{ or } (v = \mathsf{par}(v)\, i\, \lambda(v) \text{ and } w = \mathsf{par}(v)\, i\, \lambda(w))\}.$$

The former half, $w = v\, i\, \lambda(w)$, means that vertices are accessible from their parents (see the solid arrows in Fig. 2), and the latter half means that child vertices of a vertex are accessible to each other (see the broken arrows). Satisfaction of formulas under $\mathcal{T}$ can be defined by identifying $\mathcal{T}$ with a K45$_n$ situation $(M(\mathcal{T}), \mathsf{root}(\mathcal{T}))$.

Incomplete standard structure Instead of a K45$_n$ situation, we regard a labelled tree as a situation in a restricted alternate nonstandard structure called an *incomplete standard structure* (ISS). An ISS is a tuple $M = \langle W, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$ where W is a set of worlds, π is an interpretation that gives an *incomplete truth assignment* $\pi(w)$ for each $w \in W$, and each $\mathcal{B}_i$ is a

[2] Whenever $l_0a_0l_1a_1 \cdots l_{k-1}a_{k-1}l_k \in \mathcal{T}$, we have $l_0a_0l_1a_1 \cdots l_{h-1}a_{h-1}l_h \in \mathcal{T}$ for all $h \leq k$.

binary relation in W. An incomplete truth assignment is a nonstandard truth assignment that does not assign true to both p and $\neg p$ at the same time. By assuming $\mathcal{B}_i = \mathcal{B}_i^+ = \mathcal{B}_i^-$, we can regard an ISS as an alternate nonstandard structure. Semantics under an ISS follows from semantics under an alternate nonstandard structure. We can easily show that $\phi (\in \mathcal{L}_n)$ and $\neg\phi$ do not become true at the same time under ISSs, but they may become false at the same time. Occurrence of the latter case implies that the truth value of ϕ is unknown.

An *incomplete labelled tree* is defined from a labelled tree by replacing each label's truth assignment with an incomplete one. From an incomplete labelled tree $\mathcal{T}$, we can construct an ISS $M(\mathcal{T})$ in the same way as we constructed a K45$_n$ structure from a labelled tree. Hence, we can identify $\mathcal{T}$ with a situation $(M(\mathcal{T}), \mathsf{root}(\mathcal{T}))$. The semantics used with incomplete labelled trees have the following properties.

Property 1 *The following formulas are true under any incomplete labelled tree:* $B_a\phi \hookrightarrow B_aB_a\phi$, $\neg B_a\phi \hookrightarrow B_a\neg B_a\phi$, *and* $B_a\phi \wedge B_a(\phi \hookrightarrow \psi) \hookrightarrow B_a\psi$.

Representation of integrity constraints We can represent a finite set of binary nogoods with a formula $\wedge_i \neg(p_i \wedge q_i)$. According to the constraints for ISSs, the formula is true when $\neg p_i$ or $\neg q_i$ is true for all i. However, Algorithm 1 sometimes returns unknown for both p_i and q_i. In such a case, the above formula is not true under the correspondence given by Theorem 2. Hence, we represent it by the formula $IC \stackrel{\text{def}}{=} \wedge_i((p_i \hookrightarrow \neg q_i) \wedge (q_i \hookrightarrow \neg p_i))$.

In the next section, we will show that IC is a kind of *mutual belief* (Perrault 1990). We use a modal operator MB_G to represent mutual belief among group G of agents:

> $(M, w) \models MB_G\phi$ iff $G \subseteq A$ and $(M, x) \models \phi$ for all x that is accessible from w by applying $\cup_{a\in G}\mathcal{B}_a$ once or more. See the definition of common knowledge (Fagin *et al.* 1995).

Belief change First, we define an operation that transforms an incomplete truth assignment to another incomplete truth assignment. We denote an incomplete truth assignment τ by $\langle T, F\rangle$ where $T = \{p \in \Phi \mid \tau(p) = \mathsf{true}\}$ and $F = \{p \in \Phi \mid \tau(\neg p) = \mathsf{true}\}$. Then we get $T \cap F = \{\}$ and $T \cup F \subseteq \Phi$. When one accepts a set of primitive propositions S as true, one will add S to T and delete S from F. Since $T \cup S$ might violate integrity constraints, $N(S)$ should be removed from T. And all elements of $N(S)$ should be false. Hence, we define the successor of $\langle T, F\rangle$ as follows.

Definition 1 (Atomic change of a vertex label) *If a vertex has a label* $\langle T, F\rangle$, *its successor vertex has a label* $\langle T \circ S, F \bullet S\rangle$ *where* $T \circ S \stackrel{\text{def}}{=} (T - N(S)) \cup S$ *and* $F \bullet S \stackrel{\text{def}}{=} (F - S) \cup N(S)$.

Minimum coherent trees A *minimum coherent tree* (MCT) is an infinitely deep incomplete labelled tree of which each vertex has just one outgoing edge for each agent. To characterize Algorithm 1, it is sufficient to consider only MCTs whose vertex label set L is the whole set of incomplete truth assignments on Φ. We assume that every $\mathcal{T}(j)$ mentioned in the introduction is an MCT. Then we can define $\mathcal{T}(j)$ by Definition 1.

Since an agent list $a_0, \cdots, a_h$ specifies a unique vertex $l_0a_0l_1a_1\cdots l_ha_hl_{h+1}$ in $\mathcal{T}(j)$, l_{h+1} is denoted $v(h+1, j) (= \langle T(h+1, j), F(h+1, j)\rangle \in L)$. Moreover, we use $v(h+1, j) \models \phi$ or $\langle T(h+1, j), F(h+1, j)\rangle \models \phi$ to indicate that a propositional formula ϕ is true for a labelled tree that contains only one vertex whose label is $v(h+1, j)$.

Construction of trees Each tree $\mathcal{T}(j)$ requires a root vertex. Since a_0 cannot know the actual truth assignment in the external world, we do not define the label $v(0, j)$ of $\mathcal{T}(j)$'s root. Instead, we assume that $Ob(j)$ representing what a_0 observed at time j is given, and that the following conditions hold because postconditions and integrity constraints are assumed to be correct.

> $Ob(j) = \{p \in \Phi \mid v(0, j) \models p \wedge \mathsf{ob}[a_0, p]\}$ for all $j \geq 1$,
> $\mathsf{post}(e^j) \subseteq T(0, j)$, $N(\mathsf{post}(e^j)) \subseteq F(0, j)$, and $v(0, j) \models IC$ for all $j \geq 0$.

The label $v(h, j)$ of a non-root vertex is given by rewriting $v(h, j-1)$ by e^j's expected postcondition $Eff(h, j)$ and a set of newly observed propositions $New(h, j)$. We can estimate what a_h observed at time j by using a_0's belief about a_1's belief about ... about a_{h-1}'s belief. If one does not observe an event, one will not consider its effects in estimating another's belief. Hence, we get the following mutually recursive definition:

Definition 2 (Belief change) *Each non-root vertex in* $\mathcal{T}(j)$ *has a label* $v(h, j) (= \langle T(h, j), F(h, j)\rangle)$ *given by the following definition:*

- $T(1, 0) \stackrel{\text{def}}{=} \mathcal{I}^+$, $\quad F(1, 0) \stackrel{\text{def}}{=} \mathcal{I}^-$,
- $T(h, 0) \stackrel{\text{def}}{=} New(h, 0)$ *for* $h \geq 2$,
- $F(h, 0) \stackrel{\text{def}}{=} N(New(h, 0))$ *for* $h \geq 2$,
- $T(h, j) \stackrel{\text{def}}{=} (T(h, j-1) \circ Eff(h, j)) \circ New(h, j)$ *for* $h \geq 1$ *and* $j \geq 1$,
- $F(h, j) \stackrel{\text{def}}{=} (F(h, j-1) \bullet Eff(h, j)) \bullet New(h, j)$ *for* $h \geq 1$ *and* $j \geq 1$.

where $Eff(h, j)$ *is* e^j*'s expected postcondition and* $New(h, j)$ *is a set of newly observed propositions defined by:*

- $New(1, j) \stackrel{\text{def}}{=} Ob(j)$ *for* $j \geq 1$,
- $New(h, j) \stackrel{\text{def}}{=} \{p \in \Phi \mid v(h-1, j) \models p \wedge \mathsf{ob}[a_{h-1}, p]\}$ *for* $h \geq 2$ *and* $j \geq 0$.

- $Eff(1,j) \stackrel{\text{def}}{=} \mathsf{post}(e^j)$ *if* $j \geq 1$,
- $Eff(h,j) \stackrel{\text{def}}{=} Eff(h-1,j)$ *if* $v(h-1,j-1) \models \mathsf{ob}[a_{h-1}, e^j]$ *and* $h \geq 2$ *and* $j \geq 1$,
- $Eff(h,j) \stackrel{\text{def}}{=} \{\}$ *otherwise.*

The above definition has the following property.

Property 2 (Lower bound) *Every incompatible proposition of a true proposition is false:* $N(T(k, m)) \subseteq F(k,m)$.

The next lemma shows that the above method for changing beliefs does not violate integrity constraints if the original incomplete truth assignment satisfies the constraints.

Lemma 1 (Invariance of integrity constraints) *If* $T \cup F \subseteq \Phi$ *and* $T \cap F = \{\}$, *the following properties hold.*

- $\langle T, F\rangle \models IC$ *iff* $N(T) \subseteq F$.
- *If* $\langle T, F\rangle \models IC$ *and* $S \cap N(S) = \{\}$ *hold,* $(T \circ S) \cap (F \bullet S) = \{\}$ *and* $\langle T \circ S, F \bullet S\rangle \models IC$ *hold.*

Then we can show that IC remain as mutual belief by Lemma 1, Property 2, and Definition 2.

Theorem 1 (Mutual belief) *Integrity constraints remain as mutual belief. That is,* $\mathcal{T}(m) \models MB_A IC$ *for all* $m \geq 0$.

Results

Now we can compare output of Algorithm 1 and a sequence of labelled trees. The next theorem shows the correspondence of Algorithm 1 and the semantics described above.

Theorem 2 (Correspondence) *If* $a_{h-1} \neq a_h$ *for any* h $(1 \leq h \leq k)$[3], *Algorithm 1 satisfies the following properties.*

- $\mathsf{belief}(p,k,m) = \mathsf{yes}$ *iff* $\mathcal{T}(m) \models B_{a_0} B_{a_1} \cdots B_{a_k} p$.
- $\mathsf{belief}(p,k,m) = \mathsf{no}$ *iff* $\mathcal{T}(m) \models B_{a_0} B_{a_1} \cdots B_{a_k} \neg p$.
- $\mathsf{belief}(p,k,m) = \mathsf{unknown}$ *iff* $\mathcal{T}(m) \not\models B_{a_0} B_{a_1} \cdots B_{a_k} p$ *and* $\mathcal{T}(m) \not\models B_{a_0} B_{a_1} \cdots B_{a_k} \neg p$.

Since $\mathcal{T}(m) \models B_{a_0} B_{a_1} \cdots B_{a_k} \phi$ is equivalent to $v(k+1,m) \models \phi$ for any propositional formula ϕ, the above theorem can be proved by the following induction hypothesis. We will give the proof in the full paper.

Assumption 1 (Induction hypothesis) *We assume that the following relations hold if both* $h < k$ *and* $j \leq m$ *hold or both* $h \leq k$ *and* $j < m$ *hold.*

- $\mathsf{belief}(p,h,j) = \mathsf{yes}$ *iff* $p \in T(h+1,j)$.
- $\mathsf{belief}(p,h,j) = \mathsf{no}$ *iff* $p \in F(h+1,j)$.
- $\mathsf{belief}(p,h,j) = \mathsf{unknown}$ *iff* $p \notin T(h+1,j)$ *and* $p \notin F(h+1,j)$.

[3] If $a_{h-1} = a_h$, we can use Property 1.

In order to prove the main theorem from this induction hypothesis, we have to show that obs_all and obs_one correspond to observability conditions.

Lemma 2 (Observability condition) *The following properties hold under Assumption 1.*

1. $\mathsf{obs\_one}(x,k,m) = \mathsf{yes}$ *iff* $v(k,m) \models \mathsf{ob}[a_k, x]$.
2. *If* $\mathsf{post}(e^m) = \{\}$ *holds,* $Eff(k+1,m) = \{\}$ *holds for any* $k \geq 0$.
3. *If* $\mathsf{post}(e^m) \neq \{\}$ *holds,* $\mathsf{obs\_all}(e^m, k, m-1) = \mathsf{yes}$ *is equivalent to* $Eff(k+1,m) = \mathsf{post}(e^m)$.

Discussion

Complexity of algorithm To determine a_k's belief about p, Algorithm 1 checks a_{k-1}'s belief about $\mathsf{ob}[a_k, p]$ that depends on a_{k-1}'s belief about a_k's location. If $\mathsf{ob}[a_k, p]$ turns out to be false, Algorithm 1 picks up an element q from $N(\{p\})$ and checks a_{k-1}'s belief about $\mathsf{ob}[a_k, q]$ that also depends on a_{k-1}'s belief about a_k's location. Thus, Algorithm 1 computes the same function value several times to process a given query.

Our experiments show that a variant, Algorithm 2, of Algorithm 1 that records and reuses function values is several times faster than Algorithm 1. Aside from the extra computation time for the reuse, Algorithm 2 requires much less computation than a progressive algorithm, Algorithm 3, that can be easily specified from Definition 2. We can estimate the time complexity of Algorithm 3. Let C be the time complexity of computing $v(h,j)$ from $v(h-1,j-1)$, $v(h-1,j)$, and $v(h, j-1)$. Then Algorithm 3 computes $v(1,j), \ldots, v(k, j)$ from $v(1,j-1), \ldots, v(k,j-1)$ in $O(k\,C)$ and $v(k, m)$ from $v(1,0)$ in $O(m\,k\,C)$.

Improvement of expressiveness Algorithm 1 is not very general, but we can improve it in various ways. Since the original algorithm (Isozaki 1995) accepts domain knowledge that contains variables, we would like to extend the above result to first-order cases. We might be able to find another algorithm based on magic sets (Ullman 1988) for Datalog-like domain knowledge. Our latest algorithm (Isozaki 1996) can take simple *honest utterances* as well as *negation by introspection* into account. Introduction of probability and utility will improve the verisimilitude of the estimation. Introduction of an abductive reasoning mechanism will allow estimation of unobserved events based on various evidences.

However, we should consider the time complexity of extended algorithms. Estimating others' beliefs by means of an overly complex algorithm might take more time than just asking them directly. Hence, it is important to refine the algorithm by applying it to real-world problems.

We have considered only binary nogoods such as $\neg(p \wedge q)$, and translated it as $(p \hookrightarrow \neg q) \wedge (q \hookrightarrow \neg p)$. The constraint behaves just like Boolean Constraint Propagation or BCP (Forbus & de Kleer 1993). If we employ

the idea of BCP, we can translate any CNF (Conjunctive Normal Form) formula into a formula with strong implications. For example, we can translate $\neg p \vee q \vee \neg r$ as $(p \wedge \neg q \hookrightarrow \neg r) \wedge (\neg q \wedge r \hookrightarrow \neg p) \wedge (r \wedge p \hookrightarrow q)$. We expect such representation will not increase the size of labelled trees much even if a general CNF formula is given as an integrity constraint.

Related work Estimating the change in another's belief is more difficult than changing one's own belief and has not been studied well. Katsuno and Mendelzon proposed the semantic differences between *revisions* and *updates* (Katsuno & Mendelzon 1992). *Revision* incorporates newly obtained information about a static world into beliefs, while *update* conforms beliefs to the most recent facts when the world changes. Definition 2 is a combination of revision and update. The first operation for $Eff(h, j)$ corresponds to update. The second operation for $New(h, j)$ corresponds to revision.

Fagin et al. studied knowledge update of observing agents (a.k.a. communicating scientists) (Fagin *et al.* 1995). Meyden applied labelled trees to analyze belief revision (van der Meyden 1994b). However, they ignored changes of the world. Meyden (van der Meyden 1994a) analyzed knowledge update in the changing world, but did not analyze belief change. Chou and Winslett proposed an interesting belief revision algorithm is proposed for first-order logic with equality (Chou & Winslett 1994). However, they ignored nested beliefs.

Perrault applied default logic to speech act theory to represent how speech influences one's belief (Perrault 1990). Appelt and Konolige employed hierarchic autoepistemic logic to control belief change (Appelt & Konolige 1988). However, these theories tell us nothing about how one's beliefs about others' beliefs change when the external world changes.

Brafman and Tennenholtz proposed an interesting model of belief ascription. It employs a decision-theoretic utility function to ascribe beliefs to various objects (Brafman & Tennenholtz 1994). It might be possible to unify our algorithm with their model.

Concluding remarks

We have shown that our belief estimation algorithm correctly computes others' beliefs according to non-standard valuation on a sequence of labelled trees. We also discussed restrictions and generalizations found in this approach. We hope the above results contribute to the development of intelligent agents. We are applying the algorithm to natural language processing (Isozaki 1996).

We would like to thank Ken'ichiro Ishii for supporting this research and Akihiro Umemura for discussion. We also thank anonymous referees for their useful comments.

References

Appelt, D., and Konolige, K. 1988. A practical nonmonotonic theory for reasoning about speech acts. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, 170–178.

Brafman, R. I., and Tennenholtz, M. 1994. Belief ascription and mental-level modelling. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, 87–98. Morgan Kaufmann.

Chellas, B. F. 1980. *Modal logic, an introduction.* Cambridge University Press.

Chou, T. S.-C., and Winslett, M. 1994. A model-based belief revision system. *Journal of Automated Reasoning* 12:157–208.

Cohen, P. R., and Levesque, H. J. 1990. Rational interaction as the basis for communication. In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communication*, 221–255. MIT Press.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning About Knowledge.* MIT Press.

Forbus, K. D., and de Kleer, J. 1993. *Building Problem Solvers.* MIT Press.

Isozaki, H. 1995. Reasoning about belief based on common knowledge of observability of actions. In *Proceedings of the First International Conference on Multi-Agent Systems*, 193–200. MIT Press.

Isozaki, H. 1996. An application of a belief estimation algorithm to selection of Japanese sentence final particles (in Japanese). In technical group notes on natural language understanding and models of communication, The Institute of Electronics, Information and Communication Engineers.

Katsuno, H., and Mendelzon, A. O. 1992. On the difference between updating a knowledge base and revising it. In *Belief Revision.* Cambridge University Press. 183–203.

Kautz, H. 1986. The logic of persistence. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 401–405. Morgan Kaufmann.

Perrault, C. R. 1990. An application of default logic to speech act theory. In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communication.* MIT Press. chapter 9, 161–185.

Ullman, J. D. 1988. *Principles of database and knowledge-base systems II.* Computer Science Press.

van der Meyden, R. 1994a. Common knowledge and update in finite environments. I (extended abstract). In *Proceedings of the Fifth Conference on Theoretical Aspects of Reasoning About Knowledge*, 225–242. Morgan Kaufmann.

van der Meyden, R. 1994b. Mutual belief revision. In *Proceedings of the Fifth Conference on Principles of Knowledge Representation and Reasoning*, 595–606. Morgan Kaufmann.