Reasoning about Knowledge by Variable Forgetting

Kaile Su and **Guanfeng Lv**

Institute of Logic and Cognition
Department of Computer Science
Zhongshan University
GuangZhou 510275, P.R. China
isskls@zsu.edu.cn

Yan Zhang

School of Computing and Information Technology
University of Western Sydney
Penrith South DC NSW 1797, Australia
yan@cit.uws.edu.au

Abstract

In this paper, we investigate knowledge reasoning within a simple framework called *knowledge structure*. We use *variable forgetting* as a basic operation for one agent to reason about its own or other agents' knowledge. In our framework, two notions namely agents' *observable variables* and the *weakest sufficient condition* play important roles in knowledge reasoning. Given a background knowledge base T and a set of observable variables O_i for each agent i, we show that the notion of agent i knowing a formula φ can be defined as a weakest sufficient condition of φ on O_i under T. Moreover, we show how to capture the notion of common knowledge by using a generalized notion of weakest sufficient condition. We also discuss possible applications of our framework in some interesting domains such as the automated analysis of the well-known muddy children puzzle and the verification of the revised Needham-Schroeder protocol.

Introduction

Epistemic logics, or logics of knowledge are usually recognized as having originated in the work of Jaakko Hintikka - a philosopher who showed how certain modal logics could be used to capture intuitions about the nature of knowledge in the early 1960s (Hintikka 1962). In the mid of 1980s, Halpern and his colleagues discovered that S5 epistemic logics could be given a natural interpretation in terms of the states of processes (commonly called agents) in a distributed system. This model now is known as the *interpreted system* model (Fagin *et al.* 1995). It was found that this model plays an important role in the theory of distributed systems and has been applied successfully in reasoning about communication protocols (Halpern & Zuck 1992). However, the work on epistemic logic has mainly focused on theoretical issues such as variants of modal logic, completeness, computational complexity, and derived notions like distributed knowledge and common knowledge.

In this paper, we explore knowledge reasoning within a more concrete model of knowledge. Our framework of reasoning about knowledge is simple and powerful enough to analyze realistic protocols such as some widely used security protocols.

To illustrate the problem investigated in this paper, let us consider the scenario that Alice sends Bob a message and Bob sends Alice an acknowledgement when receiving the message. We assume Alice and Bob commonly have the following background knowledge base T:

$$Bob\_recv\_msg \Rightarrow Alice\_send\_msg$$
$$Bob\_send\_ack \Rightarrow Bob\_recv\_msg$$
$$Alice\_recv\_ack \Rightarrow Bob\_send\_ack$$

where Bob_recv_msg and Bob_send_ack are *observable* variables to Bob, while $Alice\_send\_msg$ and $Alice\_recv\_ack$ are *observable* to Alice.

The problem we concern with is how to verify that Alice or Bob knows a statement φ. Intuitively, we should be able to prove that for a statement observable to Alice (Bob), Alice (Bob) knows the statement if and only if the statement itself holds. Moreover, Alice knows Bob_recv_msg if and only if $Alice\_recv\_ack$ holds, and Bob knows $Alice\_send\_msg$ iff Bob_recv_msg holds. Finally, it should be always false that Bob knows $Alice\_recv\_ack$.

One of the key notions introduced in our approach is agents' *observable variables*, which shares a similar spirit of the notions of *local variables* and *local propositions* in (van der Hoek & Wooldridge 2002; Engelhardt, van der Meyden, & Moses 1998; Engelhardt, van der Meyden, & Su 2003). Here we prefer to use the term "observable variable" in order to avoid any confusion from the term "local variable" used in programming, where "non-local variables" such as "global variables" may often be observable.

Our knowledge model is also closely related to the notion of *weakest sufficient condition*, which were first formalized by Lin (Lin 2001). Given a background knowledge base T and a set of observable variables O_i for each agent i, we show that the notion of agent i knowing a formula φ can be defined as the weakest sufficient condition of φ on O_i under T, which can be computed via the operation of *variable forgetting* (Lin & Reiter 1994). Moreover, we generalize the notion of weakest sufficient condition and capture the notion of common knowledge.

The notion of *variable forgetting* or *eliminations of middle terms* (Boole 1854) has various applications in knowledge representation and reasoning. For example, Weber (Weber 1986) applied it for updating propositional knowledge bases. More recently, Lang and Marquis (Lang & Marquis 2002) used it for merging a set of knowledge bases when simply taking their union may result in inconsistency.

Now we briefly discuss the role of variable forgetting in our knowledge model. Let us examine the scenario described above again. Consider the question: how can Alice figure out Bob's knowledge when she receives the acknowledgement from Bob? Note that Alice's knowledge is the conjunction of the background knowledge base T and her observations $Alice\_recv\_ack$ etc. Moreover, all Alice knows about Bob's knowledge is the conjunction of the background knowledge base T and all she knows about Bob's observations. Thus, Alice gets Bob's knowledge by computing all she knows about Bob's observations. In our setting, Alice gets her knowledge on Bob's observations simply by forgetting Bob's non-observable variables in her own knowledge.

To show the significance of our framework, we investigate some of its interesting applications to the automated analysis of the well-known muddy children puzzle and the verification of the revised Needham-Schroeder protocol (Lowe 1996).

The organization of this paper is as follows. In the next section, we briefly introduce the concept of forgetting and the notion of weakest sufficient and strongest necessary conditions. In section 3, we define our framework of reasoning about knowledge via variable forgetting. In section 4, we generalize the notion of weakest sufficient condition and strongest necessary condition to capture common knowledge in the reasoning within our framework. In section 5, we consider a case study by applying our framework to deal with the well known muddy children puzzle. In section 6, we further apply our framework of knowledge reasoning to security protocols verification. Finally, in secion 7 we conclude the paper with some remarks.

Preliminaries

Forgetting

Given a set of propositional variables P, we sometimes do not distinguish a subset of P from its characteristic function, i.e. a truth assignment for P. We say a formula φ *over* P if each propositional variable occurring in φ is in P. For convenience, we define $\mathbf{true}$ as an abbreviation for a fixed valid propositional formula, say $p \vee \neg p$, where p is primitive proposition in P. We abbreviate $\neg\mathbf{true}$ by $\mathbf{false}$.

We also use $\models$ to denote the usual satisfaction relation between a truth assignment and a formula. Moreover, for a set of formulas Γ and a formula φ, we use $\Gamma \models \varphi$ to denote that for every assignment σ, if $\sigma \models \alpha$ for all $\alpha \in \Gamma$, then $\sigma \models \varphi$.

Given a propositional formula φ, and a propositional variable p, we denote by $\varphi(\frac{p}{\mathbf{true}})$ the result of replacing every p in φ by $\mathbf{true}$. We define $\varphi(\frac{p}{\mathbf{false}})$ similarly.

The notion of *Variable forgetting* (Lin & Reiter 1994), or eliminations of middle terms (Boole 1854), can be defined as follows:

Definition 1 Let φ be a formula over P, and $V \subseteq P$. The *forgetting of* V *in* φ , denoted as $\exists V\varphi$, is a quantified formula over P, defined inductively as follows:

1. $\exists\emptyset\varphi = \varphi$;
2. $\exists\{p\}\varphi = \varphi\left(\frac{p}{\mathbf{true}}\right) \vee \varphi\left(\frac{p}{\mathbf{false}}\right)$;
3. $\exists(V \cup \{p\})\varphi = \exists V(\exists\{p\}\varphi)$.

For convenience, we use $\forall V\varphi$ to denote $\neg\exists V(\neg\varphi)$.

Clearly, $\exists V\varphi$ is a logical consequence of φ that is independent of V; moreover, it is the strongest consequence of φ. Many characterizations of variable forgetting, together with complexity results, are reported in (Lang & Marquis 1998).

Weakest Sufficient Conditions

The formal definitions of *weakest sufficient conditions* and *strongest necessary conditions* were first formalized via the notion of variable forgetting by (Lin 2001), which in turn play an essential role in our approach.

Definition 2 Let V be a set of propositional variables and $V' \subseteq V$. Given a set of formulas Γ over V as a background knowledge base and a formula α over V.

- A formula φ over V' is called a *sufficient condition of* α *on* V' under Γ if $\Gamma \models \varphi \Rightarrow \alpha$. It is called a *weakest sufficient condition of* α *on* V' under Γ if it is a sufficient condition of α on V' under Γ, and for any other sufficient condition φ' of α on V' under Γ, we have $\Gamma \models \varphi' \Rightarrow \varphi$.
- A formula φ over V' is called a necessary condition of α on V' under Γ if $\Gamma \models \alpha \Rightarrow \varphi$. It is called a *strongest necessary condition of* α *on* V' under Γ if it is a necessary condition of α on V' under Γ, and for any other necessary condition φ' of α on V' under Γ, we have $\Gamma \models \varphi \Rightarrow \varphi'$.

The notions given above are closely related to theory of abduction. Given an observation, there may be more than one abduction conclusion that we can draw. It should be useful to find the weakest of such conclusions, i.e. the weakest sufficient condition of the observation (Lin 2001). The notions of strongest necessary and weakest sufficient conditions of a proposition also have many potential applications in other areas such as reasoning about actions. The following proposition, which is due to Lin (Lin 2001), shows how to compute the two conditions.

Proposition 3 *Given a background knowledge base θ on V, and a formula ϕ on V. Let $V' \subseteq V$. Suppose that SNC and WSC are a strongest necessary condition and a weakest sufficient condition of ϕ on V' under θ respectively. Then*

- *WSC is equivalent to $\forall(V - V')(\theta \Rightarrow \phi)$; and*
- *SNC is equivalent to $\exists(V - V')(\theta \wedge \phi)$.*

The following gives a generalized notion of weakest sufficient conditions and strongest necessary conditions.

Definition 4 Given a set of formulas Γ over V as a background knowledge base. Let α be a formula over V, and $\mathcal{V}$ a collection of subsets of V.

- A formula φ is called $\mathcal{V}$*-definable* under Γ (or simply called $\mathcal{V}$*-definable* if there is no confusion in the context), if for each $P \in \mathcal{V}$, there is a formula ψ_P on P such that $\Gamma \models \varphi \Leftrightarrow \psi_P$.

- A formula φ is called a *$\mathcal{V}$-sufficient condition of* α under Γ if it is $\mathcal{V}$-definable and $\Gamma \models \varphi \Rightarrow \alpha$. It is called a *weakest $\mathcal{V}$-sufficient condition of* α under Γ if it is a $\mathcal{V}$-sufficient condition of α under Γ, and for any other $\mathcal{V}$-sufficient condition φ' of α under Γ, we have $\Gamma \models \varphi' \Rightarrow \varphi$.
- The notions of *$\mathcal{V}$-necessary conditions of* α and *strongest $\mathcal{V}$-necessary conditions of* α under Γ can be defined in the same way.

Given a set of formulas Γ over V as a background knowledge base and $P \subseteq V$, a formula is a weakest $\{P\}$-sufficient condition of α under Γ iff it is equivalent to a weakest sufficient condition of α on P.

Let Γ be a set of formulas, V a set of propositional variables, and $\mathcal{V}$ a set of subsets of V. For convenience, we use $\mathcal{E}_{\mathcal{V}}$ to denote a relation between two assignments s, s' on V satisfying Γ such that $(s, s') \in \mathcal{E}_{\mathcal{V}}$ iff there exists a $P \in \mathcal{V}$ with $s \cap P = s' \cap P$. We use $\mathcal{E}_{\mathcal{V}}^*$ to denote the transitive closure of $\mathcal{E}_{\mathcal{V}}$. The following proposition gives the existence of weakest $\mathcal{V}$-sufficient and strongest $\mathcal{V}$-necessary conditions.

Proposition 5 *Given a finite set V of propositional variables, a set Γ of formulas over V as a background knowledge base, a formula α over V, and a set $\mathcal{V}$ of subsets of V. Denote by S_{WSC} the set of assignments s over V such that $s \models \Gamma$, and for all assignments s' satisfying Γ with $(s, s') \in \mathcal{E}_{\mathcal{V}}^*$, $s' \models \alpha$. Also denote by S_{SNC} the set of assignments s over V such that s satisfies Γ, and there exists an s' such that $s' \models \Gamma$, $s' \models \alpha$ and $(s', s) \in \mathcal{E}_{\mathcal{V}}^*$. Then,*

- *if a formula satisfies exactly those assignments in S_{WSC}, then the formula is a weakest $\mathcal{V}$-sufficient condition of α under Γ; and*
- *if a formula satisfies exactly those assignments in S_{SNC}, then the formula is a strongest $\mathcal{V}$-necessary condition of α under Γ.*

Proof: We prove only the first point because the second can be done in a similar way. Let ϕ_1 be a boolean formula over V such that, for all assignment s, $s \models \phi_1$ iff $s \in S_{WSC}$. Then, for every assignment $s \in S_{WSC}$, we have $s \models \alpha$ because $(s, s) \in \mathcal{E}_{\mathcal{V}}^*$. Thus, $\phi_1 \models \alpha$.

To prove ϕ_1 is $\mathcal{V}$-definable, we show that, for each $P \in \mathcal{V}$, $\phi_1 \models \forall(V - P)\phi_1$, which implies that ϕ_1 is equivalent to the formula $\forall(V - P)\phi_1$ over P. To prove $\phi_1 \models \forall(V - P)\phi_1$, in a semantical way, it suffices to show that, for every assignment $s \in S_{WSC}$ and $s' \models \Gamma$, if $s \cap P = s' \cap P$, then $s' \in S_{WSC}$. Let s and s' be given as above and suppose $s \cap P = s' \cap P$. Then, $(s, s') \in \mathcal{E}_{\mathcal{V}}$. Given an assignment t satisfying Γ, if $(s', t) \in \mathcal{E}_{\mathcal{V}}^*$, then $(s, t) \in \mathcal{E}_{\mathcal{V}}^*$ by $(s, s') \in \mathcal{E}_{\mathcal{V}}$. Thus, $s' \in S_{WSC}$. This proves that ϕ_1 is $\mathcal{V}$-definable.

Now we show that ϕ_1 is a weakest $\mathcal{V}$-sufficient condition under Γ. Suppose ϕ is a $\mathcal{V}$-definable and sufficient condition of α under Γ, we want to prove that $\Gamma \models \phi \Rightarrow \phi_1$. The semantical argument of such a proof is as follows. Let s be an assignment satisfying Γ and ϕ, we must show that $s \in S_{WSC}$, i.e., for every assignment s' satisfying Γ such that $(s, s') \in \mathcal{E}_{\mathcal{V}}^*$, $s' \models \alpha$. Because $\Gamma \models \phi \Rightarrow \alpha$, it suffices to show that $s' \models \phi$. By the condition $(s, s') \in \mathcal{E}_{\mathcal{V}}^*$, there is a finite sequence of assignments $s_0, \cdots, s_k$ satisfying Γ with $s_0 = s$ and $s_k = s'$, and for every $j < k$, $(s_j, s_{j+1}) \in \mathcal{E}_{\mathcal{V}}$. By the $\mathcal{V}$-definability of ϕ, we know that for every $j < k$, $s_j \models \phi$ implies $s_{j+1} \models \phi$. Thus, we have $s' \models \phi$ by induction. ∎

The above proposition can be thought of as a semantical characterization of weakest $\mathcal{V}$-sufficient and strongest $\mathcal{V}$-necessary conditions.

Knowledge and Weakest Sufficient Conditions

In our framework, a *knowledge structure* is a simple model of reasoning about knowledge. The advantage of this model is, as will be shown later, that agents' knowledge can be computed via the operation of variable forgetting.

Knowledge Structure

Definition 6 A *knowledge structure* $\mathcal{F}$ with n-agents is a $(n + 2)$-tuple $(V, \Gamma, O_1, \cdots, O_n)$ where (1) V is a set of propositional variables; (2) Γ is a set of boolean formulas over V; and (3) for each agent i, $O_i \subseteq V$.

The variables in O_i are called agent i's *observable variables*. An assignment that satisfies Γ is called a *state* of knowledge structure $\mathcal{F}$. Given a state s of $\mathcal{F}$, we define *agent i's local state* at state s as $s \cap O_i$.

A pair $(\mathcal{F}, s)$ of knowledge structure $\mathcal{F}$ and a state s of $\mathcal{F}$ is called a *scenario*.

In our framework, the language of epistemic logic, denoted by $\mathcal{L}_n^C$, is a propositional language augmented with modal operator K_i for each agent i, and modal operator C_Δ for each set of agents Δ. For a formula α, $K_i\alpha$ means that agent i knows α, and $C_\Delta\alpha$ indicates that it is common knowledge among agents in Δ that α holds. Based on scenarios, we define the semantics of language $\mathcal{L}_n^C$ as follows.

- For each primitive proposition p, $(\mathcal{F}, s) \models p$ iff $s \models p$.
- For any formulas α and β, $(\mathcal{F}, s) \models \alpha \wedge \beta$ iff $(\mathcal{F}, s) \models \alpha$ and $(\mathcal{F}, s) \models \beta$; and $(\mathcal{F}, s) \models \neg\alpha$ iff not $(\mathcal{F}, s) \models \alpha$.
- $(\mathcal{F}, s) \models K_i\alpha$ iff for all s' of $\mathcal{F}$ such that $s' \cap O_i = s \cap O_i$, $(\mathcal{F}, s') \models \alpha$.
- $(\mathcal{F}, s) \models C_\Delta\alpha$ iff $(\mathcal{F}, s') \models \alpha$ for all s' of $\mathcal{F}$ such that $(s, s') \in \mathcal{E}_{\mathcal{V}_\Delta}^*$, where $\mathcal{E}_{\mathcal{V}_\Delta}^*$ is defined as in the previous section.

Let $\mathcal{F} = (V, \Gamma, O_1, \cdots, O_n)$ be a knowledge structure. For convenience, by $\mathcal{F} \models \alpha$, we mean that for every state s of $\mathcal{F}$, $(\mathcal{F}, s) \models \alpha$. We say that a formula is an *i-local* formula if it is on O_i. Clearly, agent i knows an i-local formula φ in $\mathcal{F}$ iff $\Gamma \models \varphi$.

Lemma 7 *Let V be a finite set of variables, $\mathcal{F} = (V, \Gamma, O_1, \cdots, O_n)$ be a knowledge structure, and s be a state of $\mathcal{F}$. Also suppose that $\Delta \subseteq \{1, \cdots, n\}$, and $\mathcal{V}_\Delta = \{O_i \mid i \in \Delta\}$. Then*

1. *for any boolean formula ψ over V, $(\mathcal{F}, s) \models \psi$ iff $s \models \psi$;*
2. *for any formula $\gamma \in \Gamma$, $(\mathcal{F}, s) \models \gamma$;*
3. *for any i-local formula β, $(\mathcal{F}, s) \models K_i\beta \Leftrightarrow \beta$;*
4. *For any $\mathcal{V}_\Delta$-definable formula β, $(\mathcal{F}, s) \models C_\Delta\beta \Leftrightarrow \beta$;*
5. *for any formulas α_1 and α_2, $(\mathcal{F}, s) \models K_i(\alpha_1 \Rightarrow \alpha_2) \Rightarrow (K_i\alpha_1 \Rightarrow K_i\alpha_2)$;*

6. *For any formulas* α_1 *and* α_2, $(\mathcal{F}, s) \models C_\Delta(\alpha_1 \Rightarrow \alpha_2) \Rightarrow (C_\Delta\alpha_1 \Rightarrow C_\Delta\alpha_2)$.
7. *For any formula* α *and* $i \in \Delta$, $(\mathcal{F}, s) \models C_\Delta\alpha \Rightarrow K_i C_\Delta\alpha$.

Proof: Here we only prove point 4, the proofs of the rest of points will immediately follow from the definition of the satisfaction relationship $\models$ between a scenario and a formula. Suppose that formula β is $\mathcal{V}_\Delta$-definable, we need to show $(\mathcal{F}, s) \models C_\Delta\beta \Leftrightarrow \beta$. It suffices to prove that $(\mathcal{F}, s) \models \beta \Rightarrow C_\Delta\beta$. Assume $(\mathcal{F}, s) \models \beta$. To prove that $(\mathcal{F}, s) \models C_\Delta\beta$, we need to show that for every assignment s' such that $(s, s') \in \mathcal{E}^*_{\mathcal{V}_\Delta}$, $(\mathcal{F}, s') \models \beta$. From the definition of $\mathcal{E}^*_{\mathcal{V}_\Delta}$, it suffices to show that for every finite sequence of assignments $s_0, \cdots, s_k$ with $s_0 = s$ and $(s_j, s_{j+1}) \in \mathcal{E}_{\mathcal{V}_\Delta}$ $(0 \leq j < k)$, we have that for every $j \leq k$, $(\mathcal{F}, s_j) \models \beta$. We show this by induction on j. When $j = 0$, the result is clearly true. Assume $(\mathcal{F}, s_j) \models \beta$. Now we prove $(\mathcal{F}, s_{j+1}) \models \beta$. Because $(s_j, s_{j+1}) \in \mathcal{E}_{\mathcal{V}_\Delta}$, there is an $i \in \Delta$ such that $O_i \cap s_j = O_i \cap s_{j+1}$. On the other hand, β is equivalent to a i-local formula. Thus, $s_j \models \beta$ iff $s_{j+1} \models \beta$. Hence, $(\mathcal{F}, s_{j+1}) \models \beta$ as desired. ▮

Remark 8 It is worth mentioning that we can actually associate a knowledge structure $\mathcal{F} = (V, \Gamma, O_1, \cdots, O_n)$ with a Kripke structure $M(\mathcal{F}) = (W, \pi, \mathcal{K}_1, \cdots, \mathcal{K}_n)$, where

1. W is the set of all states of $\mathcal{F}$;
2. for each $w \in W$, the assignment $\pi(w)$ is the same as w; and
3. for each agent i and assignments $w, w' \in W$, we have that $w\mathcal{K}_i w'$ iff $w \cap O_i = w' \cap O_i$.

It is easy to show that for any formula α, $(\mathcal{F}, s) \models \alpha$ iff the situation $(M(\mathcal{F}), s)$ satisfies α. In this sense, a knowledge structure can be viewed as a specific Kripke structure.

Knowledge as Weakest Sufficient Conditions

The following theorem establishes a bridge between the notion of knowledge and the notion of weakest sufficient and strongest necessary conditions.

Theorem 9 *Let* V *be a finite set of variables,* $\mathcal{F} = (V, \Gamma, O_1, \cdots, O_n)$ *a knowledge structure,* α *a formula over* V*, and for an agent* i*,* WSC_i^α *and* SNC_i^α *a weakest sufficient condition and a strongest necessary condition of* α *on* O_i *under* Γ *respectively. Then, for each state* s *of* $\mathcal{F}$*,*

$$(\mathcal{F}, s) \models K_i\alpha \Leftrightarrow WSC_i^\alpha$$

and

$$(\mathcal{F}, s) \models \neg K_i \neg\alpha \Leftrightarrow SNC_i^\alpha.$$

Proof: We only show $(\mathcal{F}, s) \models K_i\alpha \Leftrightarrow WSC_i^\alpha$, while the other part can be proved in a similar way. Because WSC_i^α is a sufficient condition of α under Γ, we have $\Gamma \models WSC_i^\alpha \Rightarrow \alpha$. Let θ be the conjunction of all formulas in Γ, then we have $\models \theta \Rightarrow (WSC_i^\alpha \Rightarrow \alpha)$, which leads to $(\mathcal{F}, s) \models K_i WSC_i^\alpha \Rightarrow K_i\alpha$ (by Lemma 7.) Because WSC_i^α is i-local, by Lemma 7 again, we have $(\mathcal{F}, s) \models WSC_i^\alpha \Rightarrow K_i WSC_i^\alpha$. Hence, $(\mathcal{F}, s) \models WSC_i^\alpha \Rightarrow K_i\alpha$.

To show the other direction $(\mathcal{F}, s) \models K_i\alpha \Rightarrow WSC_i^\alpha$, we consider the formula $\forall(V - O_i)(\theta \Rightarrow \alpha)$, where θ is the same as above. By Proposition 3, we have $\Gamma \models \forall(V - O_i)(\theta \Rightarrow \alpha) \Rightarrow WSC_i^\alpha$. On the other hand, we know that $(\mathcal{F}, s) \models K_i\alpha \Rightarrow \forall(V - O_i)(\theta \Rightarrow \alpha)$ by the definition of $K_i\alpha$. This proves $(\mathcal{F}, s) \models K_i\alpha \Rightarrow WSC_i^\alpha$. ▮

The following corollary presents a symbolic way to compute an agent's knowledge.

Corollary 10 *Let* V *be a finite set of variables,* $\mathcal{F} = (V, \{\theta\}, O_1, \cdots, O_n)$ *a knowledge structure with* n *agents, and* α *a formula over* V*. Then, for every state* s *of* $\mathcal{F}$*,*

$$(\mathcal{F}, s) \models K_i\alpha \Leftrightarrow \forall(V - O_i)(\theta \Rightarrow \alpha).$$

Proof: Immediately by Theorem 9. ▮

Example 11: Now we consider the communication scenario between Alice and Bob addressed in section 1 once again. To show how our system can deal with the knowledge reasoning issue in this scenario, we define a knowledge structure $\mathcal{F}$ as follows:

$$\mathcal{F} = (V, \{\theta\}, O_A, O_B),$$

where

- $O_A = \{Alice\_send\_msg, Alice\_recv\_ack\}$,
- $O_B = \{Bob\_recv\_msg, Bob\_send\_ack\}$,
- $V = O_A \cup O_B$, and
- θ is the conjunction of the following three formulas:
  $Bob\_recv\_msg \Rightarrow Alice\_send\_msg$,
  $Bob\_send\_ack \Rightarrow Bob\_recv\_msg$,
  $Alice\_recv\_ack \Rightarrow Bob\_send\_ack$,

Now given a state of $\mathcal{F}$

$$s = \left\{ \begin{array}{l} Alice\_send\_msg, \\ Alice\_recv\_ack, \\ Bob\_recv\_msg, \\ Bob\_send\_ack \end{array} \right\},$$

we would like to know whether Alice knows that Bob received the message. Consider the formula

$$\forall \left\{ \begin{array}{l} Bob\_recv\_msg, \\ Bob\_send\_ack \end{array} \right\} (\theta \Rightarrow Bob\_recv\_msg).$$

From Definition 1, the above formula is simplified as $\neg Alice\_send\_msg \vee Alice\_recv\_ack$, which, obviously, is satisfied in the scenario $(\mathcal{F}, s)$, i.e.

$$(\mathcal{F}, s) \models \neg Alice\_send\_msg \vee Alice\_recv\_ack.$$

Then from Corollary 10, we have

$$(\mathcal{F}, s) \models K_A Bob\_recv\_msg.$$

Similarly, we can show that

$$(\mathcal{F}, s) \models K_A Alice\_send\_msg$$

and

$$(\mathcal{F}, s) \models K_A Alice\_recv\_ack,$$

which indicate that Alice knows that she sent the message and she knows that she received acknowledgement from Bob. ▮

Given a set of states S of a knowledge structure $\mathcal{F}$ and a formula α, by $(\mathcal{F}, S) \models \alpha$, we mean that for each $s \in S$, $(\mathcal{F}, s) \models \alpha$. The following proposition presents an alternative way to compute an agent's knowledge.

Proposition 12 *Let V be a finite set of variables, $\mathcal{F} = (V, \Gamma, O_1, \cdots, O_n)$ a knowledge structure with n agents, and α and ψ two formulas over V. Suppose that SNC_i^ψ is a strongest necessary condition of ψ on O_i under Γ, S_ψ denotes the set of those states s of $\mathcal{F}$ such that $(\mathcal{F}, s) \models \psi$, and $S_{SNC_i^\psi}$ denotes the set of those states s such that $(\mathcal{F}, s) \models SNC_i^\psi$. Then, for each agent i, we have that*

$$(\mathcal{F}, S_\psi) \models K_i\alpha \text{ iff } (\mathcal{F}, S_{SNC_i^\psi}) \models \alpha.$$

Proof: Let S_1 be the set of all states s satisfying $(\mathcal{F}, s) \models \exists(V - O_i)(\theta \wedge \psi)$. Because $\Gamma \models SNC_i^\psi \Leftrightarrow \exists(V - O_i)(\theta \wedge \psi)$, we have $S_1 = S_{SNC_i^\psi}$. Also it is easy to see that for state s of $\mathcal{F}$, $s \in S_1$ iff there is a state s' of $\mathcal{F}$ such that $s' \models \psi$ and $s \cap O_i = s' \cap O_i$. Therefore we have $(\mathcal{F}, S_\psi) \models K_i\alpha$ iff $S_1 \subseteq \{s \mid (\mathcal{F}, s) \models \alpha\}$. This leads to $(\mathcal{F}, S_\psi) \models K_i\alpha$ iff $(\mathcal{F}, S_1) \models \alpha$ iff $(\mathcal{F}, S_{SNC_i^\psi}) \models \alpha$. ∎

The intuitive meaning behind Proposition 12 is that if all we know about the current state is ψ, then all we know about agent i's knowledge (or agent i's observations) is the strongest necessary condition of ψ on O_i. A useful method of knowledge computation can be extracted from this proposition when the nested depth of knowledge operators is no more than 2.

Proposition 13 *Let V be a finite set of variables, $\mathcal{F} = (V, \{\theta\}, O_1, \cdots, O_n)$ a knowledge structure with n agents, α and ψ two formulas over V, and S_ψ denote the set of states s of $\mathcal{F}$ such that $(\mathcal{F}, s) \models \psi$. Then, for each agent i and each agent j, we have*

1. *$(\mathcal{F}, S_\psi) \models K_i\alpha$ holds iff*
$$\models (\theta \wedge \exists(V - O_i)(\theta \wedge \psi)) \Rightarrow \alpha;$$
2. *$(\mathcal{F}, S_\psi) \models K_jK_i\alpha$ holds iff*
$$\models (\theta \wedge \exists(V - O_i)(\theta \wedge \exists(V - O_j)(\theta \wedge \psi))) \Rightarrow \alpha.$$

Proof: The first part of the theorem follows immediately from Proposition 12. To show the second part, let $S_{\exists(V-O_j)(\theta\wedge\psi)}$ be the set of all states s satisfying $(\mathcal{F}, s) \models \exists(V - O_j)(\theta \wedge \psi)$. By Proposition 12, we have that

$$(\mathcal{F}, S_\psi) \models K_jK_i\alpha \text{ iff } (\mathcal{F}, S_{\exists(V-O_j)(\theta\wedge\psi)}) \models K_i\alpha.$$

By the first part of this theorem, we have that $(\mathcal{F}, S_{\exists(V-O_j)(\theta\wedge\psi)}) \models K_i\alpha$ iff

$$\models \theta \wedge \exists(V - O_i)(\theta \wedge \exists(V - O_j)(\theta \wedge \psi)) \Rightarrow \alpha.$$

∎

As will be illustrated in our analysis of security protocols (i.e. Section 6), the part 2 of Proposition 13 is useful for verifying protocol specifications with nested knowledge operators. Given a background knowledge base θ, when we face the task of testing whether $K_jK_i\alpha$ holds in those states satisfying ψ, by part 2 of Proposition 13, we can first get $\phi_1 = \exists(V - O_j)(\theta \wedge \psi)$, which is a strongest necessary condition of ψ on O_j. This is all we know about what agent j observes from ψ. Then we compute $\phi_2 = \exists(V - O_i)(\theta \wedge \phi_1)$, i.e. the strongest necessary condition of ϕ_1 on O_i which is, from the viewpoint of agent j, about what agent i observes. In this way, the task of checking $K_jK_i\alpha$ is reduced to a task of checking $\theta \wedge \phi_2 \Rightarrow \alpha$.

Corollary 14 *Let V be a finite set of propositional variables and $\mathcal{F} = (V, \{\theta\}, O_1, \cdots, O_n)$ a knowledge structure with n agents, α and ψ two formulas over V. Suppose that S_ψ denotes the set of all states s of $\mathcal{F}$ such that $(\mathcal{F}, s) \models \psi$, and SNC_i^ψ and WSC_i^α are a strongest necessary condition of ψ on O_i and a weakest necessary condition of α on O_i under $\{\theta\}$ respectively. Then*

1. *$(\mathcal{F}, S_\psi) \models K_i\alpha$ iff $\models (\theta \wedge \psi) \Rightarrow WSC_i^\alpha$; and*
2. *$(\mathcal{F}, S_\psi) \models K_i\alpha$ iff $\models (\theta \wedge SNC_i^\psi) \Rightarrow \alpha$.*

Proof: The firs part of the corollary follows from Theorem 9 and Lemma 7, while the second part follows immediately by Proposition 12. ∎

In our analysis of security protocols, we observe that very often, it seems more efficient to check an agent's knowledge via the second part of Corollary 14 rather than via the first part. But this may not be always true for some other applications (e.g. see the example of the muddy children puzzle in the next section).

Common Knowledge

Common knowledge plays an important role in reasoning about knowledge (Fagin *et al.* 1995). In this section, we generalize the concept of weakest sufficient and strongest conditions so that they can be used to compute common knowledge.

Generalized Weakest Sufficient Condition

We first investigate the computation of the weakest $\mathcal{V}$-sufficient and strongest $\mathcal{V}$-necessary conditions by using the notions of a least and a greatest fixed points of an operator, which is introduced as follows.

Let ξ be an operator from the set of boolean formulas over $\mathbf{x}$ to the set of boolean formulas over $\mathbf{x}$. We say a ψ is a *fixed point* of ξ, if $\models \xi(\psi) \Leftrightarrow \psi$. We say a ψ_0 is a *greatest fixed point* of ξ, if ψ_0 is a fixed point of ξ and for every fixed point ψ of ξ, we have $\models \psi \Rightarrow \psi_0$. Clearly, any two greatest fixed points are logically equivalent to each other. Thus, we denote a greatest fixed point of ξ by $\mathbf{gfp}Z\xi(Z)$. Similarly, We say a ψ_0 is a *least fixed point* of ξ, if ψ_0 is a fixed point of ξ and for every fixed point ψ of ξ, we have $\models \psi_0 \Rightarrow \psi$. We denote a least fixed point of ξ by $\mathbf{lfp}Z\xi(Z)$. We say ξ is *monotonic*, if for every two formulas ψ_1 and ψ_2 such that $\models \psi_1 \Rightarrow \psi_2$, we have $\models \xi(\psi_1) \Rightarrow \xi(\psi_2)$. For a finite set $\mathbf{x}$ of boolean formulas if ξ is monotonic, then there exist a least fixed point and a greatest fixed point (Tarski 1955).

Theorem 15 *Let V be a finite set of variables, $\mathcal{F} = (V, \{\theta\}, O_1, \cdots, O_n)$ a knowledge structure, α a formula* *over V, $\Delta \subseteq \{1, \cdots, n\}$, $\mathcal{V}_\Delta = \{O_i \mid i \in \Delta\}$. Assume that Λ_1 and Λ_2 be two operators such that*

$$\Lambda_1(Z) = \bigwedge_{i \in \Delta} \forall(\mathbf{x} - O_i)(\theta \Rightarrow Z)$$

and

$$\Lambda_2(Z) = \bigwedge_{i \in \Delta} \exists(\mathbf{x} - O_i)(\theta \wedge Z).$$

Then,

- *a weakest $\mathcal{V}_\Delta$-sufficient condition of α under $\{\theta\}$ is equivalent to* $\mathbf{gfp}\ Z(\alpha \wedge \Lambda_1(Z))$*; and*
- *a strongest $\mathcal{V}_\Delta$-necessary condition of α under $\{\theta\}$ is equivalent to* $\mathbf{lfp}\ Z(\alpha \vee \Lambda_2(Z))$.

Proof: We only prove the first point of this theorem, the proof of the other point is similar. Let WSC^α_Δ be a weakest $\mathcal{V}_\Delta$-sufficient condition of α under $\{\theta\}$. Note that the operator $(\alpha \wedge \Lambda_1(Z))$ is monotonic and thus there exists a greatest fixed point of it. Let $\psi_1 = \mathbf{gfp}\ Z(\alpha \wedge \Lambda_1(Z))$. We must show that $\theta \models WSC^\alpha_\Delta \Leftrightarrow \psi_1$.

We first show that $\theta \models WSC^\alpha_\Delta \Rightarrow \psi_1$. For this purpose, we only need to prove

1. $\theta \models WSC^\alpha_\Delta \Rightarrow (\alpha \wedge \Lambda_1(\mathbf{true}))$; and
2. for all formulas φ on V, if $\theta \models WSC^\alpha_\Delta \Rightarrow \varphi$, then $\theta \models WSC^\alpha_\Delta \Rightarrow (\alpha \wedge \Lambda_1(\varphi))$.

The first point is trivially true because $\Lambda_1(\mathbf{true})$ is equivalent to $\mathbf{true}$ and WSC^α_Δ is a sufficient condition of α under $\{\theta\}$. To show the second point, suppose $\theta \models WSC^\alpha_\Delta \Rightarrow \varphi$. For $i \in \Delta$, Let α_i be the formula on O_i such that $\theta \models WSC^\alpha_\Delta \Leftrightarrow \alpha_i$. Then, $\theta \models \alpha_i \Rightarrow \varphi$. It follows that $\models \alpha_i \Rightarrow (\theta \Rightarrow \varphi)$ and hence $\models \alpha_i \Rightarrow \forall(V - O_i)(\theta \Rightarrow \varphi)$ because α_i does not depend on the variables in $(V - O_i)$. So, we have that, for all $i \in \Delta$, $\theta \models WSC^\alpha_\Delta \Rightarrow \forall(V - O_i)(\theta \Rightarrow \varphi)$. The conclusion of the second point follows immediately.

We now show that $\theta \models \psi_1 \Rightarrow WSC^\alpha_\Delta$, or $\theta \models (\theta \Rightarrow \psi_1) \Rightarrow WSC^\alpha_\Delta$. It suffices to show that $\theta \Rightarrow \psi_1$ is $\mathcal{V}_\Delta$-sufficient condition of α under $\{\theta\}$, that is,

1. $\theta \Rightarrow \psi_1$ is $\mathcal{V}_\Delta$ definable; and
2. $\theta \models (\theta \Rightarrow \psi_1) \Rightarrow \alpha$.

By the fact that ψ_1 is a fixed point of the operator $(\alpha \wedge \Lambda_1(Z))$, we have that

$$\models \psi_1 \Rightarrow (\alpha \wedge \bigwedge_{i \in \Delta} \forall(\mathbf{x} - O_i)(\theta \Rightarrow \psi_1)).$$

It follows that $\models \psi_1 \Rightarrow \alpha$, and hence $\theta \models (\theta \Rightarrow \psi_1) \Rightarrow \alpha$. To show the other point, for $i \in \Delta$, we need to prove that $\theta \Rightarrow \psi_1$ is equivalent to a formula over O_i. By the above, we have that $\psi_1 \Rightarrow \forall(V - O_i)(\theta \Rightarrow \psi_1)$. It follows that $\theta \models (\theta \Rightarrow \psi_1) \Rightarrow \forall(V - O_i)(\theta \Rightarrow \psi_1)$, and hence

$$\theta \models (\theta \Rightarrow \psi_1) \Leftrightarrow \forall(V - O_i)(\theta \Rightarrow \psi_1)$$

because $\models \forall(V - O_i)(\theta \Rightarrow \psi_1) \Rightarrow (\theta \Rightarrow \psi_1)$ holds trivially. Thus $(\theta \Rightarrow \psi_1)$ is equivalent under θ to $\forall(V - O_i)(\theta \Rightarrow \psi_1)$, which is over O_i. This completes the first point of the conclusion of the theorem. ∎

Common Knowledge as Weakest $\mathcal{V}$-sufficient Conditions

Given a set Δ of agents and a family $\mathcal{V}_\Delta$ of observable variable sets of these agents, we investigate the relationship between common knowledge and the weakest $\mathcal{V}_\Delta$-sufficient and strongest $\mathcal{V}_\Delta$-necessary conditions.

Theorem 16 *Let V be a finite set of variables, $\mathcal{F} = (V, \Gamma, O_1, \cdots, O_n)$ a knowledge structure, $\Delta \subseteq \{1, \cdots, n\}$, $\mathcal{V}_\Delta = \{O_i \mid i \in \Delta\}$, α a formula over V, and WSC^α_Δ and SNC^α_Δ a weakest $\mathcal{V}_\Delta$-sufficient condition and a strongest $\mathcal{V}_\Delta$-necessary condition of α under Γ respectively. Then, for every state s of $\mathcal{F}$,*

$$(\mathcal{F}, s) \models C_\Delta \alpha \Leftrightarrow WSC^\alpha_\Delta$$

and

$$(\mathcal{F}, s) \models \neg C_\Delta \neg\alpha \Leftrightarrow SNC^\alpha_\Delta.$$

Proof: We only show $(\mathcal{F}, s) \models C_\Delta \alpha \Leftrightarrow WSC^\alpha_\Delta$; the other part can be done in a similar way. Because WSC^α_Δ is a sufficient condition of α, we have that $\Gamma \models WSC^\alpha_\Delta \Rightarrow \alpha$. Let θ be the conjunction of all formulas in Γ, we have that $\models \theta \Rightarrow (WSC^\alpha_\Delta \Rightarrow \alpha)$, which leads to $(\mathcal{F}, s) \models C_\Delta WSC^\alpha_\Delta \Rightarrow C_\Delta \alpha$ (by point 6 of Lemma 7). Because WSC^α_Δ is $\mathcal{V}_\Delta$-definable, we have, by point 4 of Lemma 7, $(\mathcal{F}, s) \models WSC^\alpha_\Delta \Rightarrow C_\Delta WSC^\alpha_\Delta$. Hence, $(\mathcal{F}, s) \models WSC^\alpha_\Delta \Rightarrow C_\Delta \alpha$.

To show the other direction $(\mathcal{F}, s) \models C_\Delta \alpha \Rightarrow WSC^\alpha_\Delta$, we consider the formula ψ_1 in the proof of Theorem 15, i.e., the greatest fixed point of the operator

$$\xi(Z) = \alpha \wedge \bigwedge_{i \in \Delta} \forall(V - O_i)(\theta \Rightarrow Z).$$

Because we already have $(\mathcal{F}, s) \models \psi_1 \Rightarrow WSC^\alpha_\Delta$ by Theorem 15, it suffices to show $(\mathcal{F}, s) \models C_\Delta \alpha \Rightarrow \psi_1$. Because the greatest fixed point ψ_1 of the operator ξ can be obtained by a finite iteration of the operator with the starting point $\xi(\mathbf{true})$, we only need to prove that

1. $\mathcal{F} \models C_\Delta \alpha \Rightarrow \xi(\mathbf{true})$; and
2. for arbitrary boolean formula φ over V, if $\mathcal{F} \models C_\Delta \alpha \Rightarrow \varphi$, then $\mathcal{F} \models C_\Delta \alpha \Rightarrow \xi(\varphi)$.

The first point is trivially true because $\xi(\mathbf{true})$ is equivalent to α. To prove the second, suppose $\mathcal{F} \models C_\Delta \alpha \Rightarrow \varphi$. Then, for each $i \in \Delta$, $\mathcal{F} \models K_i(C_\Delta \alpha \Rightarrow \varphi)$. Thus, we have that $\mathcal{F} \models C_\Delta \alpha \Rightarrow K_i \varphi$ by points 5 and 7 of Lemma 7. Hence, $\mathcal{F} \models C_\Delta \alpha \Rightarrow \forall(V - O_i)(\theta \Rightarrow \varphi)$ (by Corollary 10). It follows that $\mathcal{F} \models C_\Delta \alpha \Rightarrow \bigwedge_{i \in \Delta} \forall(V - O_i)(\theta \Rightarrow \varphi)$ and hence $\mathcal{F} \models C_\Delta \alpha \Rightarrow \xi(\varphi)$. We thus get $\mathcal{F} \models C_\Delta \alpha \Rightarrow \psi_1$. This completes the proof. ∎

Proposition 17 *Given V, $\mathcal{F}$, Δ, $\mathcal{V}_\Delta$, α as defined in Theorem 16. Let ψ be a formula over V. Assume that a strongest $\mathcal{V}_\Delta$-necessary condition of ψ is SNC^ψ_Δ. Denote by S_ψ the set of those states s of $\mathcal{F}$ such that $(\mathcal{F}, s) \models \psi$, and by $S_{SNC^\psi_\Delta}$ the set of those states s such that $(\mathcal{F}, s) \models SNC^\psi_\Delta$. Then, for each agent i, we have*

$$(\mathcal{F}, S_\psi) \models C_\Delta \alpha \textit{ iff } (\mathcal{F}, S_{SNC^\psi_\Delta}) \models \alpha.$$

Proof: Let S_1 be the set of all states s such that there is a state s' with $s' \models \psi$ and $(s', s) \in \mathcal{V}_\Delta$. We have that $(\mathcal{F}, S_\psi) \models C_\Delta \alpha$ iff for every $s \in S_1$, $(\mathcal{F}, s) \models \alpha$. This leads to $(\mathcal{F}, S_\psi) \models C_\Delta \alpha$ iff $(\mathcal{F}, S_1) \models \alpha$. On the other hand, by Proposition 5, we have that $S_1 = S_{SNC^\psi_\Delta}$. Then the conclusion of the proposition follows immediately. ∎

Note that, in Proposition 17, if α is a formula, we have that $(\mathcal{F}, S_\psi) \models C_\Delta \alpha$ iff $\Gamma \models SNC^\psi_\Delta \Rightarrow \alpha$. Moreover, by Thoerem 16, we have $(\mathcal{F}, S_\psi) \models C_\Delta \alpha$ iff $\Gamma \models \psi \Rightarrow WSC^\alpha_\Delta$, where WSC^α_Δ is a weakest $\mathcal{V}_\Delta$-sufficient of α.

A Case Study: the Muddy Children Puzzle

In this section, we demonstrate how our framework can be applied to practical problems by using the example of the muddy children puzzle.

Muddy Children Puzzle

The muddy children puzzle is a well-known variant of the wise men puzzle. The story goes as follows (Fagin *et al.* 1995): Imagine n children playing together. Some of the children, say k of them, get mud on their foreheads. Each can see the mud on others but not on his/her own forehead. Along comes the father, who says, "at least one of you has mud on your forehead." The father then asks the following question, over and over: "Does any of you know whether you have mud on your own forehead?"

Assuming that all children are perceptive, intelligent, truthful, and they answer simultaneously, what we want to show is that the first $(k-1)$ times the father asks the question, they will say "No" but the k^{th} time the children with muddy foreheads will all answer "Yes."

Modelling the Muddy Children Puzzle

To model the muddy children puzzle, let m_i be a propositional variable, which means that child i is muddy $(i < n)$. Denote by V the set $\{m_i \mid i < n\}$. Suppose the assignment $s_0 = \{m_i \mid i < k\}$ represents the actual state: child 0, $\cdots$, child $k-1$ have mud on their foreheads; and the other children have not.

Each step $j \leq k$ is associated with a knowledge structure

$$\mathcal{F}_j = (V, \Gamma_j, O_0, \cdots, O_{n-1})$$

where $O_i = V - \{m_i\}$ for each $i < n$, and Γ_j is defined as follows:

- At step 1: $\Gamma_1 = \{\bigvee_{i<n} m_i\}$.
- At step $j+1$: Let ψ^j_i, for each $i < n$, be the formula $\forall m_i(\Gamma_j \Rightarrow m_i)$. We have that $\Gamma_{j+1} = \Gamma_j \cup \left\{\varphi^j_i \mid i < n\right\}$ where

$$\varphi^j_i = \begin{cases} \neg\psi^j_i, & \text{if } (\mathcal{F}_j, s_0) \models \neg K_i m_i \\ \psi^j_i, & \text{if } (\mathcal{F}_j, s_0) \models K_i m_i. \end{cases}$$

Note that, we cannot just add $K_i(m_i)$ or $\neg K_i(m_i)$ to Γ_j because Γ_{j+1} should be a formula over V and $K_i(m_i)$ or $\neg K_i(m_i)$ depends on Γ_j. On the other hand, by Proposition 12, ψ^j_i indeed indicates that agent i knows m_i (i.e. $K_i m_i$) with respect to $\mathcal{F}_j$ because $V - O_i = \{m_i\}$. We have that, for $0 < j < k$ and $i < n$, $(\mathcal{F}_j, s_0) \models \neg K_i m_i$, and for $i < k$, $(\mathcal{F}_k, s_0) \models K_i m_i$ as desired.

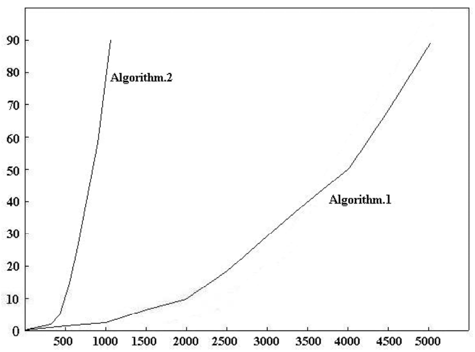

Figure 1: Performances of the two algorithms for the muddy children puzzle

Experimental Results

Our framework of knowledge structure has been implemented by using the BDD library (CUDD) developed by Fabio Somenzi at Colorado University. To check agents' knowledge, we implemented two different algorithms in terms of Part 1 and 2 of Corollary 14 in Section 3, respectively. Algorithm 1, which is based on part 1 of Corollary 14, seems much more efficient than Algorithm 2, which is based on part 2 of Corollary 14, for this particular example. The reason is as follows. It is clear that the main task of both algorithms is to check whether $(\mathcal{F}_j, s_0) \models K_i(m_i)$. However, Algorithm 1's method is to compute $s_0 \models \forall m_i(\mathcal{F}_j \Rightarrow m_i)$, while Algorithm 2 is to compute $\models \exists m_i(\mathcal{F}_j \wedge s_0) \Rightarrow m_i$. Now the main reason why Algorithm 1 is much more efficient for this particular problem is clear: $\forall m_i(\mathcal{F}_j \Rightarrow m_i)$ is simply equivalent to $\mathcal{F}_j(\frac{m_i}{false})$. Assuming half of the children are muddy, Fig. 1 gives the performances for a Pentium IV PC at 2.4GHz, with 512RAM. In the figure, the x-axis is for the number of children, and the y-axis for the CPU run time in seconds.

The muddy children puzzle as a famous benchmark problem of reasoning about knowledge can be resolved by both proof-theoretic and semantical approaches, for example, (Baltag, Moss, & Solecki 1998; Gerbrandy 1999; Lomuscio 1999). Proof-theoretic approaches depend on efficient provers for multi-modal logics; and semantical ones may suffer from the state-explosion problem. Our approach is essentially a semantical one, but we give a syntactical and compact way to represent Kripke structures by using knowledge structures, and hence may avoid the state-explosion problem to some extent.

Application to Verification of Security Protocols

In this section, we apply our knowledge model to security protocols verification. Security protocols that set up credits of the parties and deal with the distribution of cryptographic keys are essential in communication over vulnerable networks. Authentication plays a key role in security protocols. Subtle bugs that lead to attack are often found when the protocols are used for many years. This presents a challenge of how to prove the correctness of a security protocol. Formal methods are introduced to establish and prove whether a secure protocol satisfies a certain authentication specification.

Background on Authentication Protocols

Authentication protocols aim to coordinate the activity of different parties (usually referred to as *principals*) over a network. They generally consist of a *sequence* of message exchanges whose format is fixed in advance and must conform to. Usually, a principal can take part into a protocol run in different ways, as the *initiator*or the *responder*; we often call the principal has different *roles*. Very often a principal can take part into several protocols runs simultaneously with different roles.

The design of authentication protocols must have the conscious in mind that the message may be intercepted and someone with malicious intention can impersonate an honest principal. One of the key issues in authentication is to ensure the *confidentiality*, that is, to prevent private information from being disclosed to unauthorized entities. Another issue is to avoid intruder impersonating other principals. In general, a principal should ensure that the message he receives was created *recently* and sent by the principal who claims to have sent it.

Cryptography is a fundamental element in authentication. A message transmitted over a channel without any cryptographic converting is called *plaintext*. The intention of cryptography is to transform a given message to some form that is unrecognizable by anyone except the intended receiver. The procedure is called *encryption* and the corresponding parameter is known as *encryption key*. The encoded message is referred to as *ciphertext*. The reverse procedure is called *decryption* and uses the corresponding *decryption key*. The *symmetric-key cryptography*, which is also called *secret-key cryptography*, uses the same key for both encryption and decryption. The *asymmetric-key cryptography*, which is also called *public-key cryptography*, uses different keys for encryption and decryption. The one for the encryption is *public key* that is generally available for anyone. Corresponding to the public key is the *private key*, which is for the decryption and only owned by one principal.

The Dolev-Yao Intruder Model

The standard adversary model for the analysis of security protocols was introduced by Dolev and Yao in 1983 and is commonly known as *Dolev-Yao model* (Dolev & Yao 1983). According to this model, a set of conservative assumptions is made as follows:

1. Messages are considered as indivisible abstract values instead of sequences of bits.
2. All the messages from one principal to any other principals must pass through the adversary and the adversary acts as a general router in the communication.
3. The adversary can read, alter and redirect any messages.
4. The adversary can only decrypt a message if he has the right keys, can only compose new messages from keys and messages that he already possesses.
5. The adversary cannot perform any statistical or other cryptanalytic attacks.

Although this model has drawback of finding implementation dependent attacks, it simplifies the protocol analysis. It has been proved to be the the most powerful modelling of the adversary (Cervesato 2001) because it can simulate any other possible attackers.

The Revised Needham-Schroeder Protocol

As Lowe (Lowe 1996) pointed out that the Needham-Schroeder protocol has the problem of lacking the identity of the responder and can be fixed by a small modification. However, it is not clear if the revised version is correct. Our approach provides a method to automatically prove the correctness of security protocols instead of just finding bugs as usual analysis tools do for security protocols.

In the cryptography literature, the revised Needham-Schroeder protocol is described as follows:

1. $A \rightarrow B$: $\{Na, A\}_{Kb}$
2. $B \rightarrow A$: $\{B, Na, Nb\}_{Ka}$
3. $A \rightarrow B$: $\{Nb\}_{Kb}$

where $A \rightarrow B : M$ is a notation for "A sends B the message M " or "B receives the message M from A". The notation $\{M\}_K$ means the encryption of M with the key K. Also, A, B denote the principal identifiers, Ka, Kb indicate, respectively, A's and B's public keys. Moreover, Na and Nb are the *nonces* which are newly generated unguessable values by A and B, respectively, to guarantee the freshness of messages.

Two informal goals or specifications of the protocol are "A knows that B knows A said Na and Na is fresh," and "B knows that A knows B said Nb and Nb is fresh ."

To analyze the protocol, we introduce A and B *local histories* for the protocol: If A plays the role of the initiator in the protocol, and assumes that B be the responsor, then A's local history is that

1. A said $\{Na, A\}_{Kb^A}$
2. A sees $\{B^A, Na, Nb^A\}_{Ka}$
3. A said $\{Nb^A\}_{Kb^A}$

where "A said M" means that A sent the message M, or other message containing M; "A sees M" indicates that A receives M or got M by some received messages; B^A is the responsor of the protocol from A's local view; Kb^A and Nb^A are, from A's local view, the responsor's public key and nonce, respectively.

If B plays the role of responsor in the protocol, and assumes A be the initiator, then A's local history is that

1. B sees $\{Na^B, A^B\}_{Kb}$
2. B said $\{B, Na^B, Nb\}_{Ka}$

3. B sees $\{Nb\}_{Kb}$

where A^B is the initiator of the protocol from A's local observations; Ka^B and Na^B are, from B's local view, the initiator's public key and nonce, respectively.

The main point of our analysis is that if an agent is involved in the protocol, then the agent's real observations should be compatible with the so-called *local history*. For example, if A is the initiator of the protocol, A sees $\{B, Na^B, Nb\}_{Ka}$, then according to A's local history for the protocol we have that A assumes that B is the responsor of the protocol, the responsor's nonce is Nb, and from the responsor's view, the initiator's nonce is Na (see the 4th formula of the background knowledge Γ bellow).

Let us see how our framework of reasoning about knowledge can be applied to this protocol.

The variable set V consists of the following atoms:

- $fresh(Na)$: Nonce Na is fresh.
- $fresh(Nb)$: Nonce Nb is fresh.
- $role(Init, A)$: A plays the role of the initiator of the protocol.
- $role(Resp, B)$: B plays the role of the responder of the protocol.
- $Resp^A = B$: A assumes that the responder of the protocol is B.
- $Init^B = A$: B assumes that the initiator of the protocol is A.
- $Na^B = Na$: B assumes that the partner's nonce in the execution of the protocol is Na.
- $Nb^A = Nb$: A assumes that the partner's nonce in the execution of the protocol is Nb.
- $said(B, Na)$: B said Na by sending a message containing Na.
- $said(A, Nb)$: A said Nb.
- $sees(B, \{Na, A\}_{Kb})$: B sees $\{Na, A\}_{Kb}$ (possibly by decrypting the messages received.)
- $sees(A, \{B, Na^B, Nb\}_{Ka})$: A sees $\{B, Na^B, Nb\}_{Ka}$.

The background knowledge Γ consists of the following formulas:

1. $\begin{pmatrix} sees(B, \{Na, A\}_{Kb})\wedge \\ said(B, Na)\wedge \\ fresh(Na) \end{pmatrix} \Rightarrow role(Resp, B)$

2. $\begin{pmatrix} sees(A, \{B, Na^B, Nb\}_{Ka})\wedge \\ said(A, Nb)\wedge \\ fresh(Nb) \end{pmatrix} \Rightarrow role(Init, A)$

3. $\begin{pmatrix} role(Resp, B)\wedge \\ sees(B, \{Na, A\}_{Kb})\wedge \\ said(B, Na)\wedge \\ fresh(Na) \end{pmatrix} \Rightarrow \begin{pmatrix} Init^B = A\wedge \\ Na^B = Na \end{pmatrix}$

4. $\begin{pmatrix} role(Init, A)\wedge \\ sees(A, \{B, Na^B, Nb\}_{Ka})\wedge \\ said(A, Nb)\wedge \\ fresh(Nb) \end{pmatrix} \Rightarrow \begin{pmatrix} Resp^A = B\wedge \\ Na^B = Na\wedge \\ Nb^A = Nb \end{pmatrix}$

5. $\begin{pmatrix} role(Init, A)\wedge \\ Resp^A = B \end{pmatrix} \Rightarrow \begin{pmatrix} sees(B, \{Na, A\}_{Kb})\wedge \\ said(B, Na) \end{pmatrix}$

6. $\begin{pmatrix} role(Resp, B)\wedge \\ Init^B = A \end{pmatrix} \Rightarrow \begin{pmatrix} sees(A, \{B, Na^B, Nb\}_{Ka})\wedge \\ said(A, Nb) \end{pmatrix}$

7. $\begin{matrix} (role(Init, A) \Rightarrow fresh(Na))\wedge \\ (role(Resp, B) \Rightarrow fresh(Nb)) \end{matrix}$

Notice that the first two formulas are required for the rationality of the agents A and B. The other formulas in Γ can be obtained automatically by some fixed set of meta rules. We obtain the third and fourth formulas by the comparing their local history for the protocols to the conditions appearing in the formulas. To get the fifth formula informally, consider A's local history under the conditions $role(Init, A)$ and $Resp^A = B$, which should be that

1. A said $\{Na, A\}_{Kb}$
2. A sees $\{B, Na, Nb^A\}_{Ka}$
3. A said $\{Nb^A\}_{Kb}$.

According to A's local history, A sees the nonce Na generated by A itself. Because Na is only said in the message $\{Na, A\}_{Kb}$, thus B, who has the inverse key of Kb, must see this message and said Na. Similarly, we can see that sixth formula holds. The last formula follows immediately by the definition of the protocol.

The set O_A of observable variables to A is

$$\{fresh(Na), role(Init, A), Resp^A = B\}.$$

The set O_B of observable variables to B is

$$\{fresh(Nb), role(Resp, B), Init^B = A\}.$$

Now consider the knowledge structure

$$\mathcal{F} = (V, \Gamma, O_A, O_B).$$

Let $Spec_A$ be the formal specification:

$$\begin{pmatrix} fresh(Na)\wedge \\ role(Resp, A)\wedge \\ Resp^A = B \end{pmatrix} \Rightarrow K_A K_B \begin{pmatrix} said(A, Na)\wedge \\ fresh(Na) \end{pmatrix}$$

and $Spec_B$ be the formal specification:

$$\begin{pmatrix} fresh(Nb)\wedge \\ role(Resp, B)\wedge \\ Init^B = A \end{pmatrix} \Rightarrow K_B K_A \begin{pmatrix} said(B, Nb)\wedge \\ fresh(Nb) \end{pmatrix}.$$

It is easy to show that, for all states s of $\mathcal{F}$,

$$(\mathcal{F}, s) \models Spec_A \wedge Spec_B$$

as desired.

We should mention that, in the original Needham-Schroeder protocol (R.M.Needham & M.D.Schroeder 1978), the second message is $B \rightarrow A$: $\{Na, Nb\}_{Ka}$ instead of $B \rightarrow A$: $\{B, Na, Nb\}_{Ka}$. Therefore, the fourth formula in Γ would change to

$$\begin{pmatrix} role(Init, A)\wedge \\ sees(A, \{Na^B, Nb\}_{Ka})\wedge \\ said(A, Nb)\wedge \\ fresh(Nb) \end{pmatrix} \Rightarrow \begin{pmatrix} Na^B = Na\wedge \\ Nb^A = Nb \end{pmatrix}$$

Thus, $Resp^A = B$ do not necessarily hold under the condition $role(Init, A) \wedge sees(A, \{Na^B, Nb\}_{Ka}) \wedge said(A, Nb) \wedge fresh(Nb)$. This is why the specifications $Spec_A$ and $Spec_B$ do not hold for the original Needham-Schroeder protocol.

Discussion

BAN logic (Burrows, Abadi, & Needham 1990) is one of the most successful logical tools to reason about security protocols. However, the semantics of BAN is always arguable, and it is not clear under what assumption the rules of BAN logic is sound and complete. This motivated the research of seeking more adequate frameworks (models). Providing a model-theoretic semantics for BAN logic has been a central idea in the development of BAN-like logics such as AT (Abadi & Tuttle 1991) and SVO (Syversion & van Oorschot 1996). The major advantage of our approach is that we can prove the correctness of a protocol systematically, rather than finding a flaw (bug) of a protocol. Also, our method operates on the actual definition of the protocols, not on some kind of their abstract specifications.

Conclusion

In this paper, we have investigated knowledge reasoning within a simple framework called knowledge structure. Variable forgetting is used as a basic operation for one agent to reason about its own or other agents' knowledge. Given a background knowledge base T, and a set of observable variables O_i for each agent i, we have showed that the notion of agent i knowing a formula φ can be defined as the weakest sufficient condition of φ on O_i under T. Moreover, we generalize the notion of weakest sufficient conditions to capture the notion of common knowledge in framework. To illustrate the applications of our knowledge structures, we have discussed the automated analysis of the well-known muddy children puzzle and the verification of the corrected Needham-Schroeder protocol.

Our work presented in this paper can be further extended in several directions. First, we will investigate whether our knowledge structures can be extended and used as a basis for knowledge based programming (Fagin *et al.* 1995). Secondly, in our current framework of knowledge structures, we have not considered the issue of *only knowing* which has been extensively studied in other knowledge reasoning models, e.g. (Halpern & Lakemeyer 1996; van der Hock, Jaspars, & Thijsse 2003; Levesque 1990). It will be an interesting topic of how our knowledge model handles only knowing in reasoning about knowledge.

Finally, recent research has shown that *knowledge update* has many important applications in reasoning about actions and plans and dynamic modelling of multi-agent systems (Zhang 2003). Baral and Zhang have proposed a general model for performing knowledge update based on the standard single agent S5 modal logic (Baral & Zhang 2001). We believe that their work can be extended to many agents modal logics by using the knowledge structure defined in this paper and therefore to develop a more general system for knowledge update. Along this direction, an interesting question is what is the underlying relationship between *knowledge forgetting* - a specific type of knowledge update, and variable forgetting as addressed in this paper.

Acknowledgement

The authors thank Ron van der Meyden and Fangzhen Lin for their valuable comments on an earlier version of this paper. This work was supported by the National Science Foundation of China under grants 60073056 and 60273062. Also, the third author was supported in part by the Australian Research Council Linkage-Project grant LP034787.

References

Abadi, M., and Tuttle, M. 1991. A semantics for a logic of authentication. In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, 201–216.

Baltag, A.; Moss, L.; and Solecki, S. 1998. The logic of public announcement, common knowledge, and private suspicions. In Gilboa, I., ed., *Proc. TARK-98*, 125–132. San Francisco: Morgan Kaufmann.

Baral, C., and Zhang, Y. 2001. On the semantics of knowledge update. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 97–102.

Boole, G. 1854. *An Investigation of the Laws of Thought*.

Burrows, M.; Abadi, M.; and Needham, R. M. 1990. A logic of authentication. *ACM Transactions on Computer Systems* 8(1).

Cervesato, I. 2001. The Dolev-Yao intruder is the most powerful attacker. In *Proc. 16th Annual Int. Symp on Logic in Computer Science*.

Dolev, D., and Yao, A. 1983. On the security of public-key protocols. *Communications of the ACM* 29(8):198–208.

Engelhardt, K.; van der Meyden, R.; and Moses, Y. 1998. Knowledge and the logic of local propositions. In *Theoretical Aspects of Rationality and Knowledge, Proc. of TARK 1998*, 29–41. Morgan Kaufmann.

Engelhardt, K.; van der Meyden, R.; and Su, K. 2003. Modal logics with a hierarchy of local propositional quantifiers. In *Advance in Modal Logic*, volume 4, 9–30. Kings College Publications.

Fagin, R.; Halpern, J.; Moses, Y.; and Vardi, M. 1995. *Reasoning about knowledge*. Cambridge, MA: MIT Press.

Gerbrandy, J. 1999. *Bisimulation on Plant Kripke*. Ph.D thesis, Institute for Logic, Language and Computation, University of Amsterdam.

Halpern, J., and Lakemeyer, G. 1996. Multi-agent only knowing. In *TARK 1996*, 251–265.

Halpern, J., and Zuck, L. 1992. A little knowledge goes a long way: Simple knowledge based derivations and correctness proofs for a family of protocols. *Journal of the ACM* 39(3):449–478.

Hintikka, J. 1962. *Knowledge and Belief.* Ithaca, NY: Cornell University Press.

Lang, J., and Marquis, P. 1998. Complexity results for independence and definability. In *Proc. the 6th International Conference on Knowledge Representation and Reasoning*, 356–367.

Lang, J., and Marquis, P. 2002. Resolving inconsistencies by variable forgetting. In *Proc. of KR'2002*, 239–250.

Levesque, H. 1990. All I know: a study in autoepistemic logic. *Artificial Intelligence* 42:263–309.

Lin, F., and Reiter, R. 1994. Forget it! In Greiner, R., and Subramanian, D., eds., *Working Notes of AAAI Fall Symposium on Relevance*, 154–159.

Lin, F. 2001. On the strongest necessary and weakest sufficient conditions. *Artificial Intelligence* 128:143–159.

Lomuscio, A. 1999. *Knowledge Sharing among Ideal Agents*. Ph.D thesis, School of Computer Science, University of Birmingham.

Lowe, G. 1996. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In Margaria, and Steffen., eds., *Tools and Algorithms for the Construction and Analysis of Systems*, Vol 1055 of Lecture Notes in Computer Science, 147–166. Springer Verlag.

R.M.Needham, and M.D.Schroeder. 1978. Using encryption for authentication in large networks of computers. *Communication of the ACM* 21(12):993–999.

Syversion, P. F., and van Oorschot, P. 1996. An unified cryptographic protocol logic. Technical Report NRL Publication 5540-227, Naval Research Lab.

Tarski, A. 1955. A lattice-theoretical fixpoint theorem ans its applications. *Pacific J. Math.* 5:285–309.

van der Hock, W.; Jaspars, J.; and Thijsse, E. 2003. Theories of knowledge and ignorance. In S. Rahman, J. Symons, D. G., and van Bendegem, J., eds., *Logic, Epistemology and the Unity of Science*. Kluwer.

van der Hoek, W., and Wooldridge, M. 2002. Model checking knowledge and time. In *Proc. 19th Workshop on SPIN (Model Checking Software)*, 95–111.

Weber, A. 1986. Updating propositional formulas. In *Proc. First Conference on Expert Database Systems*, 487–500.

Zhang, Y. 2003. Minimal change and maximal coherence for epistemic logic program updates. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, 112–117.