Bulletin of the Section of Logic
Volume 54/4 (2025), pp. 607–642
https://doi.org/10.18778/0138-0680.2025.19

Yuki Nishimura

AGENT-KNOWLEDGE LOGIC FOR ALTERNATIVE EPISTEMIC LOGIC

Abstract

Epistemic logic is known as a logic that captures the knowledge and beliefs of agents and has undergone various developments. In this paper, we propose a new logic called agent-knowledge logic by taking the product of individual knowledge structures and the set of relationships among agents. This logic is based on the Facebook logic and the Logic of Hide and Seek Game. We show two main results; one is that this logic can embed the standard epistemic logic, and the other is that there is a proof system of tableau calculus that works in finite time. We also discuss various sentences and inferences that this logic can express.

Keywords: agent-knowledge logic, modal logic, epistemic logic, hybrid logic, tableau calculus.

Presented by: Michał Zawidzki
Received: December 25, 2024, **Received in revised form**: December 13, 2025,
Accepted: December 29, 2025, **Published online**: March 13, 2026

1. Introduction

Investigations into knowledge and beliefs form the part of philosophy, which is now called epistemology. This area has been the subject of various studies from a logical standpoint. One of these was conducted by applying modal logic, which is nowadays called epistemic logic. The operator K_i, which is the key element of this logic, constitutes formulas of the form $K_i\varphi$, which express that "agent i knows that φ." On this basis, it is possible to represent various concepts related to knowledge and belief in formal language. Hintikka did pioneering work on epistemic logic in 1962 [9], and there is a wide range of research today; see Fagin et al. [6] and van Benthem [18].

A more recent logic for human knowledge is Facebook logic, developed by Seligman et al. in 2011 [16]. This logic was invented to describe personal knowledge plus the friendships of agents in a two-dimensional hybrid logic. For instance, consider this sentence: "An agent is Andy's friend, and Andy knows he has a pollen allergy. Then, one of the agent's friends knows that they have a pollen allergy." This inference can be written using the language of Facebook logic as follows:

$$\langle \texttt{Friend} \rangle i \wedge @_i [\texttt{Know}] p \rightarrow \langle \texttt{Friend} \rangle [\texttt{Know}] p,$$

where $p =$ "they have pollen allergy." and $i =$ "This is Andy." The at sign @ in the logical formula is the operator of hybrid logic, where $@_i p$ can be read as "p holds at point i." Facebook logic uses nominals, a tool of hybrid logic, to make reference to individual agents. For a thorough introduction to hybrid logics, we refer the reader to Blackburn & ten Cate [2], Indrzejczak [10], and Braüner [4]. Sano [14] provides further details on two-dimensional hybrid logic.

In fact, Facebook logic treats propositional variables differently from epistemic logic. The truth of a propositional variable p depends not only on the epistemic alternative but also on the agent under consideration. Therefore, the propositions represented by the propositional variables here are personal properties, such as, "I have a pollen allergy."

The new logic proposed in this paper — we will call it *agent-knowledge logic* — is a modification of the aforementioned Facebook logic. One feature of this logic is that the fragment of it is compatible with epistemic logic. This property allows us to use agent-knowledge logic as an alternative to epistemic logic. Indeed, this paper shows how to embed epistemic logic into our new logic. Furthermore, agent-knowledge logic is able to formalize a variety of sentences that cannot be represented by traditional epistemic logic, such as "one of an agent's friends knows p."

In this paper, we also introduce a proof system for the logic by constructing a suitable tableau calculus. The tableau calculus is not only a proof system but also a system for discovering a counterexample model in which the formula is not valid. In particular, by constructing a tableau calculus with the termination property — in short, that the proof ends in finite time — we can show that the logic is decidable.

This logic has two parents: one is Facebook logic, and the other is a logic which seems to have nothing to do with epistemic logic, namely the Logic of Hide and Seek Game (LHS, in short) created by Li et al. in 2021 [11, 12]. This logic was originally invented to illustrate the hide and seek game (also known as cops and robbers). In LHS, propositional variables are split into two sets, which are related to hider and seeker, respectively. We borrow this idea to express the *agent-free* propositions ("the sun rises in the east," for example).

We proceed as follows: Section 2 reviews the well-known epistemic logic and explains the parents of agent-knowledge logic, Facebook logic, and LHS, briefly. In Section 3, we introduce our new logic, that is, agent-knowledge logic. Section 4 shows how we embed epistemic logic into our new logic. In Section 5, we construct a tableau calculus with the termination property and completeness. Finally, in Section 6, we write about some future prospects.

2. Preliminary

2.1. Epistemic Logic

This section is mostly based on the work of Fagin et al. [6, Chapter 2].

In epistemic logic, we have another set $\mathbf{A}$ of agents besides a usual set $\mathbf{Prop}$ of propositional variables. The elements of $\mathbf{A}$ occur in a new operator K_i. The intuitive meaning of $K_i\varphi$ is that "agent i knows φ."

Definition 2.1. We have two disjoint sets, $\mathbf{Prop}$ and $\mathbf{A}$. A formula φ of the *epistemic logic* $\mathcal{L}_{\mathrm{EL}}$ is defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi$$

where $p \in \mathbf{Prop}$ and $i \in \mathbf{A}$.

We only use $\neg$ and $\wedge$ as primitives since other Boolean operators, such as $\vee$ and $\rightarrow$, can be defined as compounds of the first two operators.

Definition 2.2. A *Kripke model for epistemic logic* (we call it *EL model*) $\mathcal{M}_{\mathrm{EL}}$ is a tuple $(W, (R_i)_{i\in\mathbf{A}}, V)$ where:

- W is a non-empty set.
- For each $i \in \mathbf{A}$, R_i is a binary relation on W.
- $V : \mathbf{Prop} \to \mathcal{P}(W)$.

Definition 2.3. Given an EL model $\mathcal{M}_{\mathrm{EL}}$, its point w, and a formula $\varphi \in \mathcal{L}_{\mathrm{EL}}$, the *satisfaction relation* $\mathcal{M}_{\mathrm{EL}}, w \models \varphi$ is defined inductively as follows:

$$\begin{aligned}
\mathcal{M}_{\mathrm{EL}}, w \models p &\iff w \in V(p), \text{ where } p \in \mathbf{Prop}\\
\mathcal{M}_{\mathrm{EL}}, w \models \neg\varphi &\iff \text{Not } \mathcal{M}_{\mathrm{EL}}, w \models \varphi\ (\mathcal{M}_{\mathrm{EL}}, w \not\models \varphi)\\
\mathcal{M}_{\mathrm{EL}}, w \models \varphi \wedge \psi &\iff \mathcal{M}_{\mathrm{EL}}, w \models \varphi \text{ and } \mathcal{M}_{\mathrm{EL}}, w \models \psi\\
\mathcal{M}_{\mathrm{EL}}, w \models K_i\varphi &\iff \text{For all } v \in W, wR_iv \text{ implies } \mathcal{M}_{\mathrm{EL}}, v \models \varphi.
\end{aligned}$$

As for epistemic logic, we define the validity of a formula. Later we discuss embedding epistemic logic into our new logic, so the formal definition is needed.

Definition 2.4. A formula φ is *valid* with respect to the class of EL models (denoted by $\models_{\mathrm{EL}} \varphi$) if $\mathcal{M}_{\mathrm{EL}}, w \models \varphi$ for every model $\mathcal{M}_{\mathrm{EL}}$ and its every world w.

2.2. Facebook Logic

Facebook logic, first invented by Seligman et al. [16], has two characteristics compared to classical modal logic.

First, we have two modal operators, K and F. These modal operators correspond to knowledge and friendship, respectively. Correspondingly, a possible world is decomposed into two components: one representing an agent and the other representing an epistemic alternative of an individual.

Another addition is the introduction of special propositional variables called *nominals*. A nominal n is a proposition corresponding to only one agent, that is, it is a proposition for the *name* of the agent. In addition, we introduce the satisfaction operator @ used in hybrid logic. The intuitive meaning of $@_n p$ is that "p holds for agent n."

Let us introduce a formal definition. We have two disjoint infinite sets, **Prop** of propositional variables and **Nom** of nominals. A formula φ of the *Facebook logic* is defined as follows:

$$\varphi ::= p \mid n \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi \mid F\varphi \mid @_n\varphi,$$

where $p \in$ **Prop** and $n \in$ **Nom**. If needed, we can define the dual $\langle K\rangle$ and $\langle F\rangle$ of each modal operator as $\langle K\rangle\varphi := \neg K\neg\varphi$ and $\langle F\rangle\varphi := \neg F\neg\varphi$.

The semantics of Facebook logic is based on *epistemic social network models*. An epistemic social network model is a tuple $(W, A, (\sim_a)_{a\in A}, (\asymp_w)_{w\in W}, V)$, where:

- W is a set of epistemic alternatives.
- A is a set of agents.
- For each $a \in A$, $\sim_a$ is an equivalence relation on W.

- For each $w \in W$, $\asymp_w$ is an irreflexive and symmetric relation of friendship on A.
- V is a valuation function, which assigns a propositional variable p to a subset of $W \times A$ and a nominal n to a set $W \times \{a\}$ for some $a \in A$.

The reason for a relation $\asymp_w$ over A being irreflexive and symmetric can be understood when we assume it as a friendship; no one is a friend to oneself, and if a person is your friend, then you are a friend of them.

Then, the truth of formulas in Facebook logic is defined inductively. The Boolean cases are omitted since they are the same as those in classical modal logic. Also, the element $a \in A$ such that $V(n) = W \times \{a\}$ holds is abbreviated as n^V.

$$\begin{array}{rcl}
\mathcal{M}, w, a \models p & \iff & (w, a) \in V(p) \text{ where } p \in \mathbf{Prop}, \\
\mathcal{M}, w, a \models n & \iff & n^V = a, \text{ where } n \in \mathbf{Nom} \\
\mathcal{M}, w, a \models K\varphi & \iff & \mathcal{M}, v, a \models \varphi \text{ for every } v \sim_a w, \\
\mathcal{M}, w, a \models F\varphi & \iff & \mathcal{M}, w, b \models \varphi \text{ for every } b \asymp_w a, \\
\mathcal{M}, w, a \models @_n\varphi & \iff & \mathcal{M}, w, n^V \models \varphi.
\end{array}$$

As mentioned in the Introduction, the truth of a propositional variable depends on both an epistemic alternative and an agent.

Example 2.5. The following formulas of Facebook logic can be translated into natural language as follows:

- Kp: An agent knows that they are p.
- KFp: An agent knows that all of their friends are p.
- FKp: Each of an agent's friends knows that they are p.
- $\langle F \rangle n$: An agent has a friend n.
- $@_n Kp$: An agent n knows that they are p.

For readers who would like to study it deeper, Seligman et al. [16] and its sequel, Seligman et al. [17], should be of help.

2.3. Logic of Hide and Seek Game

The logic of hide and seek game (LHS), as the name implies, is a logic for describing a hide and seek game. There are two players, a hider and a seeker, and a set of propositional variables $\mathbf{Prop}_H$ and $\mathbf{Prop}_S$ for each player to describe their state. Moreover, there is a special propositional variable I. This is a proposition to describe that the hider and seeker are in the same place, i.e., expressing "I find you!"

The main difference from Facebook logic is that we use the same structure (W, R, V) as in usual modal logic, which is appropriate considering that the hide and seek game is played by two players on the same board.

Here is a definition of a formula of LHS φ, where $p_H \in \mathbf{Prop}_H$ and $p_S \in \mathbf{Prop}_S$:

$$\varphi ::= p_H \mid p_S \mid I \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Diamond_H \varphi \mid \Diamond_S \varphi.$$

The truth value of LHS formulas is defined inductively as follows (note that both x and y are elements of W):

$$\begin{array}{rcl}
\mathcal{M}, x, y \models p_H & \iff & x \in V(p_H) \text{ where } p_H \in \mathbf{Prop}_H \\
\mathcal{M}, x, y \models p_S & \iff & y \in V(p_S) \text{ where } p_S \in \mathbf{Prop}_S \\
\mathcal{M}, x, y \models I & \iff & x = y \\
\mathcal{M}, x, y \models \Diamond_H \varphi & \iff & \text{there is some } x' \text{ such that } xRx' \text{ and } \mathcal{M}, x', y \models \varphi \\
\mathcal{M}, x, y \models \Diamond_S \varphi & \iff & \text{there is some } y' \text{ such that } yRy' \text{ and } \mathcal{M}, x, y' \models \varphi.
\end{array}$$

Using this language, we can describe the hide and seek game. For example, $\Box_H \Diamond_S I$ means that no matter how the hider moves, the seeker has a one-step move to catch the hider. This expression shows the existence of a winning strategy for the seeker.

In addition to the already mentioned Li et al. [11], Li et al. [12] may also help readers who want to know more about LHS.

3. Agent-Knowledge Logic

Here, we introduce a new logic, called *agent-knowledge logic*. As you read in Section 1, this logic is a mixture of Facebook logic and LHS. We have two dimensions, which correspond to agents and their knowledge, respectively. This structure and the intention behind it are very similar to that of Facebook logic. On the other hand, the idea that we use both $\mathbf{Prop}_A$ and $\mathbf{Prop}_K$ is unique for LHS.

3.1. Agent-Knowledge Model

To construct the vocabulary, we require four sets of variables in total: two for propositional variables and two for nominals, each associated with agents and knowledge, respectively. Among these, $p_K \in \mathbf{Prop}_K$ can be viewed as a proposition that does not depend on an agent, such as "the Earth goes around the Sun." Furthermore, the two types of nominals, $a \in \mathbf{Nom}_A$ and $k \in \mathbf{Nom}_K$, can be interpreted as an agent name and a label for an epistemic alternative, respectively. Interpreting the elements of $\mathbf{Prop}_A$ is somewhat more difficult by comparison, but $p_A \in \mathbf{Prop}_A$ could be understood as a proposition representing some property of an agent, such as "an agent has a pollen allergy."

Definition 3.1. We have four disjoint sets $\mathbf{Prop}_A$, $\mathbf{Prop}_K$, $\mathbf{Nom}_A$, and $\mathbf{Nom}_K$. A formula φ of the *agent-knowledge logic* $\mathcal{L}_{\mathrm{AK}}$ is defined as follows:

$$\varphi ::= p_A \mid p_K \mid a \mid k \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_A\varphi \mid \Box_K\varphi \mid @_a\varphi \mid @_k\varphi,$$

where $p_A \in \mathbf{Prop}_A$, $p_K \in \mathbf{Prop}_K$, $a \in \mathbf{Nom}_A$, and $k \in \mathbf{Nom}_K$.

As we mentioned in Section 2.2, nominals in $\mathbf{Nom}_A$ and $\mathbf{Nom}_K$ point to a specific agent and a specific epistemic alternative, respectively. As well as $\vee$ and $\rightarrow$, if we need, we can define $\Diamond_A$ and $\Diamond_K$ in the usual way.

Definition 3.2. An *agent-knowledge model (AK model)* $\mathcal{M}_{\mathrm{AK}}$ is a tuple $(W_A \times W_K, (R_y)_{y \in W_K}, (S_x)_{x \in W_A}, V_A, V_K)$ where:

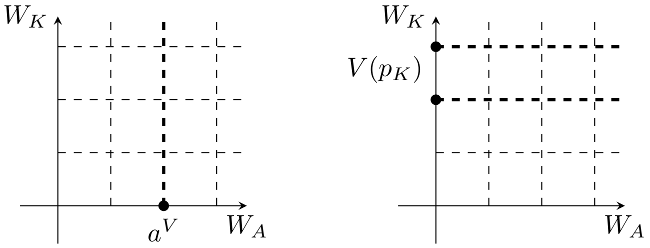

Figure 1: An agent-knowledge model.

- W_A, W_K are disjoint non-empty sets.
- For each $y \in W_K$, R_y is a binary relation on W_A.
- For each $x \in W_A$, S_x is a binary relation on W_K.
- $V_A : \mathbf{Prop}_A \cup \mathbf{Nom}_A \to \mathcal{P}(W_A)$, where if $a \in \mathbf{Nom}_A$, then $V_A(a) = \{x\}$ for some $x \in W_A$.
- $V_K : \mathbf{Prop}_K \cup \mathbf{Nom}_K \to \mathcal{P}(W_K)$, where if $k \in \mathbf{Nom}_K$, then $V_K(k) = \{y\}$ for some $y \in W_K$.

Note that the image of a nominal $\mathbf{Nom}_A$ by V_A is a singleton (the same fact holds for $\mathbf{Nom}_K$ and V_K). Owing to this definition, a nominal behaves as a *name* for each possible world.

We can illustrate an agent-knowledge model as if we write Cartesian coordinates in Figure 1. In this circumstance, a nominal is true in the worlds on the corresponding horizontal or vertical line. Likely, a propositional variable holds in the worlds on some set of parallel lines.

We write V to express $V_A \cup V_K$. For instance, $V(p_A) = V_A(p_A)$. Moreover, we abbreviate $x \in W_A$ such that $V_A(a) = \{x\}$ by a^V. We do the same for k^V.

Definition 3.3. Given a model $\mathcal{M}_{\mathrm{AK}}$, its points $(x, y) \in W_A \times W_K$, and a formula $\varphi \in \mathcal{L}_{\mathrm{AK}}$, the *satisfaction relation* $\mathcal{M}_{\mathrm{AK}}, (x, y) \models \varphi$ is defined inductively as follows:

$$\begin{aligned}
\mathcal{M}_{\mathrm{AK}}, (x, y) \models p_A &\iff x \in V(p_A), \text{ where } p_A \in \mathbf{Prop}_A \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models p_K &\iff y \in V(p_K), \text{ where } p_K \in \mathbf{Prop}_K \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models a &\iff x = a^V, \text{ where } a \in \mathbf{Nom}_A \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models k &\iff y = k^V, \text{ where } k \in \mathbf{Nom}_K \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models \neg\varphi &\iff \text{Not } \mathcal{M}_{\mathrm{AK}}, (x, y) \models \varphi \ (\mathcal{M}_{\mathrm{AK}}, (x, y) \not\models \varphi) \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models \varphi \wedge \psi &\iff \mathcal{M}_{\mathrm{AK}}, (x, y) \models \varphi \text{ and } \mathcal{M}_{\mathrm{AK}}, (x, y) \models \psi \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models \Box_A\varphi &\iff \text{For all } x' \in W_A, xR_yx' \text{ implies} \\
&\qquad \mathcal{M}_{\mathrm{AK}}, (x', y) \models \varphi \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models \Box_K\varphi &\iff \text{For all } y' \in W_K, yS_xy' \text{ implies} \\
&\qquad \mathcal{M}_{\mathrm{AK}}, (x, y') \models \varphi \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_a\varphi &\iff \mathcal{M}_{\mathrm{AK}}, (a^V, y) \models \varphi \\
\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_k\varphi &\iff \mathcal{M}_{\mathrm{AK}}, (x, k^V) \models \varphi.
\end{aligned}$$

The truth of each propositional variable is determined by either $x \in W_A$ or $y \in W_K$. Especially whether p_K is true or false is independent of the element of W_A, so p_K can be assumed as an agent-free proposition.

The usage of the satisfaction operator @ should also be mentioned. It refers to a specific agent or epistemic alternative while ignoring the current one. For example, the meaning of $@_a\varphi$ is "for an agent whose name is a, φ holds." The current element of W_A is no longer necessary information to determine the truth of that formula.

Definition 3.4. A formula φ is *valid* with respect to the class of $\mathcal{M}_{\mathrm{AK}}$ (denoted by $\models_{\mathrm{AK}} \varphi$) if $\mathcal{M}_{\mathrm{AK}}, (x, y) \models \varphi$ for every model $\mathcal{M}_{\mathrm{AK}}$ and its every pair (x, y).

Before finishing this section, it is necessary to mention the globality of the satisfaction operator for readers familiar with hybrid logic. In ordinary (one-dimensional, you might say) hybrid logic, a formula of the form $@_i\varphi$

has globality, that is, if $@_i\varphi$ is true in any possible world, then it is true in every possible world.

On the other hand, this is not the case with agent-knowledge logic. Suppose $\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_a\varphi$ ($a \in \mathbf{Nom}_A$). Then, we have $\mathcal{M}_{\mathrm{AK}}, (a^V, y) \models \varphi$. However, it can be the case that there is some $z \in W_K$ such that $\mathcal{M}_{\mathrm{AK}}, (a^V, z) \not\models \varphi$. In this case, $@_a\varphi$ is not true in (x, z). Likewise, $@_k\varphi$ ($k \in \mathbf{Nom}_K$) is not a global expression.

However, this does not mean that the agent-knowledge logic has completely lost its global expression. If we use two satisfaction operators together and create a formula $@_a@_k\varphi$ ($a \in \mathbf{Nom}_A, k \in \mathbf{Nom}_K$), then it has globality. In fact, $\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_a@_k\varphi$ is equivalent to $\mathcal{M}_{\mathrm{AK}}, (a^V, k^V) \models \varphi$, which shows that the truth of $@_a@_k\varphi$ does not depend at all on the current state.

3.2. Examples

As we do in Facebook logic, we can compound friendship and knowledge in agent-knowledge logic. We read $\Box_K\varphi$ as "An agent knows φ," and $\Box_A\varphi$ as "All of an agent's friends are φ." For example, we can write some sentences as follows:

- $\Box_A\Box_K p_K$: All of an agent's friends know p_K.
- $\Diamond_A\Box_K p_K$: Some of an agent's friends know p_K.
- $\Box_K\Diamond_A\Box_K p_K$: An agent knows that some of their friends know p_K.

Moreover, we can designate an individual by calling their name owing to nominals. Consider this sentence:

> An agent is Andy's friend, and if Andy knows that the Earth goes around the Sun, then one of the agent's friends knows the heliocentric theory.

This inference can be symbolized in the agent-knowledge logic as follows:

$$\Diamond_A a \wedge @_a\Box_K p_K \to \Diamond_A\Box_K p_K,$$

where p_K shows "the Earth goes around the Sun" and a shows "This is Andy."

The difference between agent-knowledge logic and Facebook logic becomes more pronounced when we assume that the binary relations over epistemic alternatives are equivalence relations. For example, in Facebook logic, the formula $@_n Kp \to p$ is not valid even if $\asymp_w$ is an equivalence relation. Define $\mathcal{M} = (W, A, (\sim_a)_{a \in A}, (\asymp_w)_{w \in W}, V)$ as follows:

$$\begin{aligned} W &= \{w, v\} \\ A &= \{a, b\} \\ \sim_a &= \sim_b = W \times W \\ \asymp_w &= \asymp_v = A \times A \\ V(p) &= \{(w, b), (v, b)\} \\ V(n) &= W \times \{b\}. \end{aligned}$$

Then, $\mathcal{M}, (w, a) \models @_n Kp$ holds but we have $\mathcal{M}, (w, a) \not\models p$. However, in agent-knowledge logic, the situation changes.

Proposition 3.5. The formula $@_a \Box_K p_K \to p_K$ is valid with respect to the class of $\mathcal{M}_{\mathrm{AK}}$ where all of S_x are equivalence relations.

Proof: Suppose that $\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_a \Box_K p_K$. Then, we have $\mathcal{M}_{\mathrm{AK}}, (a^V, y) \models \Box_K p_K$. By the reflexivity of S_y, especially we have $\mathcal{M}_{\mathrm{AK}}, (a^V, y) \models p_K$. Since the truth value of p_K is determined only by an element of W_K, we have $\mathcal{M}_{\mathrm{AK}}, (x, y) \models p_K$. □

This fact may be better understood if we interpret those formulas in natural language. Even though Andy knows he has a pollen allergy, it does not mean so do all agents. However, if he knows that the Earth goes around the Sun, then it is true; the Earth really goes around the Sun.

In addition to the relationships between epistemic alternatives, we can also impose restrictions on the relationships between agents as needed. For example, in Facebook logic, the relationship between agents should be irreflexive and symmetric. Also, we have another way to capture relationships between agents, for example, to read xR_yx' as "in the situation y, the agent x can see the post of x'" on X[1]. Then, we can read $\Box_A \Box_K p_K$ as "all the people know p_K, *as far as I know*."

4. Embedding Epistemic Logic into Agent-Knowledge Logic

One of the aims of our new logic is to make it an alternative to Facebook logic. In fact, any sentence we can express in basic epistemic logic can be rewritten in this agent-knowledge logic. In this section, we show that we can embed epistemic logic into agent-knowledge logic.

To begin with, let us define how to translate a formula of epistemic logic.

Definition 4.1. We define a *translation* $T : \mathcal{L}_{\text{EL}} \to \mathcal{L}_{\text{AK}}$ as follows:

$$\begin{aligned} T : \mathbf{Prop} \ni p \mapsto p_K \in \mathbf{Prop}_K &\text{ is a bijection} \\ T : \mathbf{A} \ni i \mapsto a \in \mathbf{Nom}_A &\text{ is a bijection} \\ T(\neg\varphi) &= \neg T(\varphi) \\ T(\varphi \wedge \psi) &= T(\varphi) \wedge T(\psi) \\ T(K_i\varphi) &= @_{T(i)} \Box_K T(\varphi). \end{aligned}$$

Example 4.2. Here is one example of translation:

$$T(K_i(p \wedge K_j \neg q)) = @_{a_i} \Box_K (p_K \wedge @_{a_j} \Box_K \neg q_K).$$

We write a_i to abbreviate $T(i)$ $(i \in \mathbf{A})$.

In fact, the idea of rewriting $K_i\varphi$ as $@_{T(i)}\Box_K T(\varphi)$ was presented in Sano's review in 2011 [15]. This was written in Japanese to introduce the paper by Seligman et al. [16]. Unfortunately, this translation does not work for Facebook logic, but it does work when the target logic is agent-knowledge logic.

We aim to prove the following theorem.

[1] Most of the readers are familiar with the name once it had: Twitter.

Theorem 4.3. *For all $\varphi \in \mathcal{L}_{EL}$,*

$$\models_{EL} \varphi \iff \models_{AK} T(\varphi).$$

First, we construct an AK model from an EL model.

Definition 4.4. Given an EL model $\mathcal{M}_{\text{EL}} = (W, (R_i)_{i\in\mathbf{A}}, V)$, the induced AK model $\mathcal{M}^{\alpha}_{\text{AK}}$ is defined as follows:

$$\mathcal{M}^{\alpha}_{\text{AK}} = (\mathbf{A} \times W, \emptyset, (R_i)_{i\in\mathbf{A}}, V^{\alpha}),$$

where:

- For any $p_A \in \mathbf{Prop}_A$, $V^{\alpha}(p_A) = \emptyset$.
- For any $p_K \in \mathbf{Prop}_K$, $V^{\alpha}(p_K) = V(T^{-1}(p_K))$.
- For any $a \in \mathbf{Nom}_A$, $V^{\alpha}(a) = \{T^{-1}(a)\}$.
- Take one $y_0 \in W$, and for any $k \in \mathbf{Nom}_K$, $V^{\alpha}(k) = \{y_0\}$.

Note that we do not care about the definitions of $(R_y)_{y\in W_K}, V^{\alpha}(p_A)$, and $V^{\alpha}(k)$. It is because the formula translated by T requires only $\mathbf{Prop}_K$, $\mathbf{Nom}_A$, Boolean operators, $\Box_K$, and $@_a$.

Lemma 4.5. *For any $\varphi \in \mathcal{L}_{EL}$ and for any $i \in \mathbf{A}$, we have:*

$$\mathcal{M}_{EL}, w \models \varphi \iff \mathcal{M}^{\alpha}_{AK}, (i, w) \models T(\varphi).$$

Proof: By induction on the complexity of φ.

$(\varphi = p)$ For all $i \in \mathbf{A}$,

$$\begin{aligned} \mathcal{M}_{\text{EL}}, w \models p &\iff w \in V(p) \\ &\iff w \in V^{\alpha}(T(p)) \\ &\iff \mathcal{M}^{\alpha}_{\text{AK}}, (i, w) \models T(p). \end{aligned}$$

$(\varphi = \neg\psi, \psi \wedge \chi)$ Straightforward.

$(\varphi = K_j\psi)$ First, we prove the left-to-right direction.

Suppose that $\mathcal{M}_{\mathrm{EL}}, w \models K_j\psi$. Then, for all v such that wR_jv, we have $\mathcal{M}_{\mathrm{EL}}, v \models \psi$. We divide the proof into two cases depending on whether such a world $v \in W$ exists.

(i) If $v \in W$ reachable by R_j from w exists, take arbitrary one. Then, we have $\mathcal{M}_{\mathrm{EL}}, v \models \varphi$. By the induction hypothesis, especially $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (j, v) \models T(\varphi)$. Since we took v arbitrarily, it follows that $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (j, w) \models \Box_K T(\varphi)$. By the definition of V^{α}, we finally get that $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (i, w) \models @_{T(j)}\Box_K T(\varphi)$ for all $i \in A$.

(ii) If there is no $v \in W$ such that wR_jv, we straightforwardly get that $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (j, w) \models \Box_K T(\varphi)$. In the same way as in the former case, we have $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (i, w) \models @_{T(j)}\Box_K T(\varphi)$ for all $i \in \mathbf{A}$.

In both cases, we can reach the result $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (i, w) \models @_{T(j)}\Box_K T(\varphi)$ for all $i \in \mathbf{A}$. Therefore, we have $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (i, w) \models T(K_j\varphi)$.

Next, we prove the other direction. Take one $i \in A$ and suppose that $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (i, w) \models T(K_j\psi)$. It means that for all v such that wR_jv, $\mathcal{M}^{\alpha}_{\mathrm{AK}}, (j, v) \models T(\psi)$ holds. Take one v such that wR_jv (if we cannot, then $\mathcal{M}_{\mathrm{EL}}, w \models K_j\psi$ is straightforward). By the induction hypothesis, we have $\mathcal{M}_{\mathrm{EL}}, v \models \psi$. Since we took v arbitrarily, it follows that $\mathcal{M}_{\mathrm{EL}}, w \models K_j\psi$. □

Definition 4.6. Given an AK model $\mathcal{M}_{\mathrm{AK}} = (W_A \times W_K, (R_y)_{y \in W_K}, (S_x)_{x \in W_A}, V)$, the induced EL model $\mathcal{M}^{\beta}_{\mathrm{EL}}$ is defined as follows:
$\mathcal{M}^{\beta}_{\mathrm{EL}} = (W_K, (S^{\beta}_i)_{i \in \mathbf{A}}, V^{\beta})$, where:

- $\mathbf{A}$ is the set used in Definition 2.1.
- $yS^{\beta}_i z$ in $\mathcal{M}^{\beta}_{\mathrm{EL}}$ iff $yS_{T(i)^V} z$ in $\mathcal{M}_{\mathrm{AK}}$.
- $V^{\beta}(p) = V(T(p))$.

Let us consider a function $\beta : W_A \to \mathbf{A}$ such that $\beta(T(i)^V) = i$ for all $i \in \mathbf{A}$. It expresses the correspondence between an agent in W_A and an agent in $\mathbf{A}$. The illustration of this condition in Figure 2 may help your understanding.

$$\begin{tikzcd}
\mathbf{A} \ni i \arrow[r, mapsto, "T"] & a \in \mathbf{Nom}_1 \arrow[d, mapsto, "(\cdot)^V"] \\
\mathbf{A} \ni i \arrow[u, mapsto, dashed, "\mathrm{id}_{\mathbf{A}}"] & x \in W_A \arrow[l, mapsto, "\beta"']
\end{tikzcd}$$

Figure 2: The condition β satisfies ($\mathrm{id}_{\mathbf{A}}$ is the identity on $\mathbf{A}$).

LEMMA 4.7. *For any $\varphi \in \mathcal{L}_{EL}$ and for any $x \in W_A$,*

$$\mathcal{M}_{AK}, (x, y) \models T(\varphi) \iff \mathcal{M}^{\beta}_{EL}, y \models \varphi.$$

PROOF: By induction on the complexity of φ.

($\varphi = p$) For all $x \in W^A$,

$$\begin{aligned}
\mathcal{M}_{\mathrm{AK}}, (x, y) \models T(p) &\iff y \in V(T(p)) \\
&\iff y \in V^{\beta}(p) \\
&\iff \mathcal{M}^{\beta}_{\mathrm{EL}}, y \models p.
\end{aligned}$$

($\varphi = \neg\psi, \psi \land \chi$) Straightforward.

($\varphi = K_j\psi$) First, we prove the left-to-right direction.

Suppose that $\mathcal{M}_{\mathrm{AK}}, (x, y) \models T(K_j\psi)$. That is, we assume that $\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_{T(j)}\Box_K T(\psi)$. Then, for all z such that $yS_{T(j)^V}z$, we have $\mathcal{M}_{\mathrm{AK}}, (T(j)^V, z) \models T(\psi)$. Bearing in mind the definition of S^{β}_i, it suffices to pick up one $z \in W_K$ such that $yS^{\beta}_j z$ (if we cannot, it is straightforward that $\mathcal{M}^{\beta}_{\mathrm{EL}}, y \models K_j\psi$ holds). By the assumption, we have $\mathcal{M}_{\mathrm{AK}}, (T(j)^V, z) \models T(\psi)$. By the induction hypothesis, $\mathcal{M}^{\beta}_{\mathrm{EL}}, z \models \psi$. Since we picked up z arbitrarily, we have $\mathcal{M}^{\beta}_{\mathrm{EL}}, y \models K_j\psi$.

Next, we prove the other direction. Suppose that $\mathcal{M}^{\beta}_{\mathrm{EL}}, y \models K_j\psi$. It means that for all $z \in W_K$ such that $yS^{\beta}_{j}z$, $\mathcal{M}^{\beta}_{\mathrm{EL}}, z \models \psi$ holds. Now, pick $z \in W_A$ such that $yS_{T(j)^V}z$ arbitrarily (if we cannot, we have $\mathcal{M}_{\mathrm{AK}}, (x, y) \models T(K_j\psi)$ for all $x \in W_A$), and we have $yS^{\beta}_{j}z$. Then, we have $\mathcal{M}_{\mathrm{AK}}, z \models \psi$. By the induction hypothesis, $\mathcal{M}_{\mathrm{AK}}, (T(j)^V, z) \models T(\psi)$. Since we pick up z arbitrarily, it follows that $\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_{T(j)}\Box_K T(\psi)$ for any $x \in W_A$, which means $\mathcal{M}_{\mathrm{AK}}, (x, y) \models T(K_j\psi)$. □

Now, we are ready to prove the main theorem, Theorem 4.3. Here is the proof.

Proof: We prove it by showing the contraposition. To prove the left-to-right direction, suppose that we have some φ such that $\not\models_{\mathrm{AK}} T(\varphi)$. Then, there is a model $\mathcal{M}_{\mathrm{AK}}$ and its pair of points (x, y) such that $\mathcal{M}_{\mathrm{AK}}, (x, y) \models \neg T(\varphi)$, which means that $\mathcal{M}_{\mathrm{AK}}, (x, y) \models T(\neg\varphi)$. Then, by Lemma 4.7, we have $\mathcal{M}^{\beta}_{\mathrm{EL}}, y \models \neg\varphi$, which leads us to the conclusion that $\not\models_{\mathrm{EL}} \varphi$. The case of the other direction can be done by using Lemma 4.5. □

We usually treat binary relations of EL models as equivalence relations. Moreover, once we want to deal with beliefs by means of a modal operator, we impose yet another condition on accessibility relations. The following corollary shows how embedding can reflect these restrictions.

Proposition 4.8. We have the following properties:

(i) For every $i \in \mathbf{A}$, if R_i in $\mathcal{M}_{\mathrm{EL}}$ is reflexive (or serial, symmetric, transitive, Euclidean), then so is R_i in $\mathcal{M}^{\alpha}_{\mathrm{AK}}$.

(ii) For every $x \in W_A$, if S_x in $\mathcal{M}_{\mathrm{AK}}$ is reflexive (or serial, symmetric, transitive, Euclidean), then so is S^{β}_{i} in $\mathcal{M}^{\beta}_{\mathrm{EL}}$.

Proof: The former is obvious, and the latter is straightforward from the definition of S^{β}_{i}. □

5. Proof System

In this section, we introduce a tableau calculus as a proof system.

In constructing a tableau calculus for agent-knowledge logic, we have made significant use of that for hybrid logic. The primary reference is the work of Bolander and Blackburn [3]. We also refer to Nishimura [13], who studies tableau calculi for some two-dimensional hybrid logics.

For simplicity, this section deals only with the negation normal form (NNF, in short) of formulas. Moreover, if we write $\neg\varphi$, we assume it as its NNF. For the satisfaction operators, a formula $\neg @_a \varphi$ is equivalent to $@_a \neg\varphi$. That is, for any model and its possible world (x, y), a formula φ, and a nominal $a \in \mathbf{Nom}_A$, we have

$$\mathcal{M}_{\mathrm{AK}}, (x, y) \models @_a \neg\varphi \iff \mathcal{M}_{\mathrm{AK}}, (x, y) \models \neg @_a \varphi.$$

The same equivalence holds for the case of $k \in \mathbf{Nom}_K$. Transformations to the NNF involving Boolean and modal operators can be done in the usual way.

5.1. Tableau Calculus

Here we provide a tableau calculus of agent-knowledge logic, denoted by $\mathbf{T}_{\mathrm{AK}}$.

Definition 5.1. A *tableau* is a well-founded tree constructed in the following way:

- Start with a formula of the form $@_a @_k \varphi$ (called the *root formula*), where φ is a formula of agent-knowledge logic and $a \in \mathbf{Nom}_A, k \in \mathbf{Nom}_K$ does not occur in φ.
- For each branch, extend it by applying rules (see Definition 5.3) to all nodes as often as possible. However, we can no longer add any formula in a branch if at least one of the following conditions is satisfied:
  - (i) Every new formula generated by applying any rule already exists in the branch.
  - (i) The branch is closed (see Definition 5.2).

Here, a *branch* means a maximal path of a tableau. If a formula φ occurs in a branch Θ, we write $\varphi \in \Theta$.

Definition 5.2. A branch of a tableau Θ is *closed* if one of the following conditions holds:

(ii) There are $a \in \mathbf{Nom}_A$, $k, l \in \mathbf{Nom}_K$, and $s \in \mathbf{Prop}_A \cup \mathbf{Nom}_A$ such that $@_a @_k s, @_a @_l \neg s \in \Theta$.

(ii) There are $a, b \in \mathbf{Nom}_A$, $k \in \mathbf{Nom}_K$, and $t \in \mathbf{Prop}_K \cup \mathbf{Nom}_K$ such that $@_a @_k t, @_b @_k \neg t \in \Theta$.

We say that Θ is *open* if it is not closed. A tableau is called *closed* if all branches in the tableau are closed.

Definition 5.3. We provide the rules of $\mathbf{T}_{\mathrm{AK}}$ in Figure 3.

In these rules, the formulas above the line show the formulas that have already occurred in the branch, and the formulas below the line show the formulas that will be added to the branch. The vertical line in the $[\vee]$ means that the branch splits to the left and right.

Definition 5.4 (provability). Given a formula φ, we say that φ is *provable* in $\mathbf{T}_{\mathrm{AK}}$ if there is a closed tableau whose root formula is $@_a @_k \neg\varphi$, where $a \in \mathbf{Nom}_A$ and $k \in \mathbf{Nom}_K$ do not occur in φ.

5.2. Termination

A tableau calculus has *the termination property* if, for any tableau constructed in the system, all branches have a finite length. We first prove that the tableau calculus $\mathbf{T}_{\mathrm{AK}}$ introduced above has the termination property. This proof is based on the termination proof in [3]; however, it is more complicated because we are now dealing with two-dimensional circumstances.

Definition 5.5. Let $@_a @_k \varphi, @_b @_l \psi$ $(a, b \in \mathbf{Nom}_A,\ k, l \in \mathbf{Nom}_K)$ be formulas. We say that $@_a @_k \varphi$ is a *prefixed subformula* of $@_b @_l \psi$ if φ is a subformula of ψ.

$$\frac{@_a@_k\neg b}{@_b@_k b}\ [\neg_A]^{*1} \qquad \frac{@_a@_k\neg l}{@_a@_l l}\ [\neg_K]^{*1}$$

$$\frac{@_a@_k(\varphi\wedge\psi)}{\begin{array}{c}@_a@_k\varphi\\ @_a@_k\psi\end{array}}\ [\wedge] \qquad \frac{@_a@_k(\varphi\vee\psi)}{@_a@_k\varphi \mid @_a@_k\psi}\ [\vee]$$

$$\frac{@_a@_k\Diamond_A\varphi}{\begin{array}{c}@_a@_k\Diamond_A b\\ @_b@_k\varphi\end{array}}\ [\Diamond_A]^{*1,*2,*3} \qquad \frac{@_a@_k\Diamond_K\varphi}{\begin{array}{c}@_a@_k\Diamond_K l\\ @_a@_l\varphi\end{array}}\ [\Diamond_K]^{*1,*2,*4}$$

$$\frac{\begin{array}{c}@_a@_k\Box_A\varphi\\ @_a@_k\Diamond_A b\end{array}}{@_b@_k\varphi}\ [\Box_A]^{*5} \qquad \frac{\begin{array}{c}@_a@_k\Box_K\varphi\\ @_a@_k\Diamond_K l\end{array}}{@_a@_l\varphi}\ [\Box_K]^{*5}$$

$$\frac{@_a@_k@_b\varphi}{@_b@_k\varphi}\ [@_A] \qquad \frac{@_a@_k@_l\varphi}{@_a@_l\varphi}\ [@_K] \qquad \frac{\begin{array}{c}@_a@_k\varphi\\ @_a@_k b\end{array}}{@_b@_k\varphi}\ [Id_A]^{*2} \qquad \frac{\begin{array}{c}@_a@_k\varphi\\ @_a@_k l\end{array}}{@_a@_l\varphi}\ [Id_K]^{*2}$$

*1: This rule can be applied only one time per formula.
*2: The formula above the line is not an accessibility formula. Here, an *accessibility formula* is the formula of the form $@_a@_k\Diamond_A b$ ($@_a@_k\Diamond_K l$) generated by $[\Diamond_A]$ ($[\Diamond_K]$), where b (l) is a new nominal.
*3: $b \in \mathbf{Nom}_A$ does not occur in the branch.
*4: $l \in \mathbf{Nom}_K$ does not occur in the branch.
*5: The second formula above the line is an accessibility formula.

Figure 3: The rules of $\mathbf{T}_{\mathrm{AK}}$

Lemma 5.6. *For any formula $@_a @_k \varphi$ occurring in a branch Θ of a tableau, at least one of the following conditions holds:*

- *$@_a @_k \varphi$ is a prefixed subformula of the root formula.*
- *$@_a @_k \varphi$ is an accessibility formula.*

Proof: By induction on the length of Θ. □

Definition 5.7. Let Θ be a branch. For every pair (a, k) $(a \in \mathbf{Nom}_A, k \in \mathbf{Nom}_K)$ of nominals occurring in Θ, the set $T^{\Theta}((a, k))$ is defined as follows:

$$T^{\Theta}((a,k)) = \{\varphi \mid @_a @_k \varphi \in \Theta \text{ is a prefixed subformula of the root formula}\}.$$

Since the number of prefixed subformulas is finite for any formula $@_a @_k \varphi$, the set $T^{\Theta}((a, k))$ is finite for any pair (a, k) of nominals.

Definition 5.8. Let Θ be a branch of a tableau, and let $a, b \in \mathbf{Nom}_A$ and $k, l \in \mathbf{Nom}_K$ be nominals occurring in Θ. A pair (b, l) of nominals is *generated* by (a, k) in Θ (denoted by $(a, k) \prec_{\Theta} (b, l)$) if one of the following conditions holds:

(i) $k = l$ and b is introduced by applying $[\Diamond_A]$ to $@_a @_k \Diamond_A \varphi$.

(ii) $a = b$ and l is introduced by applying $[\Diamond_K]$ to $@_a @_k \Diamond_K \varphi$.

Observe that the following equivalences hold: $(a, k) \prec_{\Theta} (b, l)$ if and only if one of the following conditions holds:

(i) $k = l$ and there is an accessibility formula $@_a @_k \Diamond_A b \in \Theta$.

(ii) $a = b$ and there is an accessibility formula $@_a @_k \Diamond_K l \in \Theta$.

Thus far, the discussion has proceeded in nearly the same manner as in previous studies such as [3]. However, the situation changes from the following lemma.

Lemma 5.9. *Given a branch* Θ *of a tableau, we define a structure* $G^\Theta = (N^\Theta, \prec_\Theta)$ *where:*

- $N^\Theta = \{(a, k) \mid$ *there is a* φ *such that* $@_a @_k \varphi \in \Theta\}$,
- $\prec_\Theta$ *is the relation defined in Definition 5.8.*

Then G^Θ *is a disjoint union of well-founded and finitely branching trees.*

Proof: Properties that G is well-founded and finitely branching are proved in a similar way to the proof of [3, Lemma 4.2]. Then the rest of the proof is to show that G is a disjoint union of trees.

Suppose there are pairs (a, k) and (a', k') of nominals such that both $(a, k) \prec_\Theta (b, l)$ and $(a', k') \prec_\Theta (b, l)$. This proof is split into two cases.

(a) Suppose that $k = k' = l$ and there are accessibility formulas $@_a @_k \Diamond_A b$, $@'_a @_k \Diamond_A b \in \Theta$. However, it never occurs since we have to introduce a new nominal whenever we apply $[\Diamond_A]$ or $[\Diamond_K]$ to a branch.

(b) Suppose that $k = l$, $a' = b$ and there are accessibility formulas $@_a @_k \Diamond_A b, @_{a'} @_{k'} \Diamond_K l \in \Theta$. If $@_a @_k \Diamond_A b$ appears first, then the occurrence $@_{a'} @_{k'} \Diamond_K l$ in Θ contradicts the restriction of $[\Diamond_K]$ since $l = k$ already exists in Θ before adding $@_{a'} @_{k'} \Diamond_K l$. □

Note that, unlike [3, Lemma 6.4], the lemma shown above does not claim that there are only finitely many trees. In the tableau for one-dimensional hybrid logic, there can be only finitely many nominals that serve as roots in G^Θ; they are restricted to those occurring in the root formula. Consequently, if N^Θ is infinite, we immediately obtain an infinite sequence of $\prec_\Theta$ by König's Lemma. In the present case, however, if at least one member of the pair (a, k) is a nominal from the root formula, it has the potential to become a root of G^Θ.

While this issue must be addressed, we first define a function m_Θ for a pair of nominals (a, k) that returns the maximum length of formulas labeled by it, and then show that this value decreases along the transitions of $\prec_\Theta$.

Definition 5.10. Let Θ be a branch of a tableau. For any pair (a,k) of nominals occurring in Θ. We define a function $m_\Theta : \mathbf{Nom}_A \times \mathbf{Nom}_K \to \mathbb{N}$ as follows:

$$m_\Theta((a,k)) = \max\{|\varphi| \mid @_a @_k \varphi \in \Theta\}.$$

Then, we want to show that $(a,k) \prec_\Theta (b,l)$ implies $m_\Theta((a,k)) > m_\Theta((b,l))$. However, directly proving it does not work well. Thus, we show an extended result as a lemma.

Lemma 5.11. *Let Θ be a branch of a tableau.*

(i) If $(a,k) \prec_\Theta (b,k)$, then for all $l \in \mathbf{Nom}_K$, we have $m_\Theta((a,k)) > m_\Theta((b,l))$.

(ii) If $(a,k) \prec_\Theta (a,l)$, then for all $b \in \mathbf{Nom}_A$, we have $m_\Theta((a,k)) > m_\Theta((b,l))$.

Proof: We only show (i). The other part of the lemma can be shown similarly.

Assume $(a,k) \prec_\Theta (b,k)$. Take a nominal $l \in \mathbf{Nom}_K$ and a formula φ such that $m_\Theta((b,l)) \geq m_\Theta((b,l'))$ for all $l' \in \mathbf{Nom}_K$ and $|\varphi| = m_\Theta((b,l))$. We show that $m_\Theta((a,k)) > |\varphi|$ or contradiction by dividing it into cases depending on which rule $@_b @_l \varphi$ was introduced by.

(a) Suppose that $@_b @_l \varphi$ is introduced by $[\neg_A]$. Straightforwardly, we have $@_b @_l b \in \Theta$. This is obviously not an accessibility formula, so by Lemma 5.6, $@_b @_l b$ is a prefixed subformula of the root formula of Θ. Then, b is in the root formula, which contradicts that b is a generated nominal.

(b) Suppose that $@_b @_l \varphi$ is introduced by $[\neg_K]$. Then, we have $\varphi = l$ and there is another nominal $l \in \mathbf{Nom}_K$ such that $@_b @_{l'} \neg l \in \Theta$. However, it contradicts the maximality of l and φ.

(c) If $@_b @_l \varphi$ is introduced by $[\wedge]$, then there is another formula ψ such that $@_b @_l (\varphi \wedge \psi) \in \Theta$. However, it contradicts the maximality of φ. We can prove this in a similar way in the case of $[\vee]$.

(d) Suppose that $@_b @_l \varphi$ is introduced by $[\Diamond_A]$. This means that there is another nominal $b' \in \mathbf{Nom}_A$ (it may be different from a) such that $@_b @_l \varphi$ is introduced from $@_{b'} @_l \Diamond_A \varphi$. Then we have $(b', l) \prec_\Theta (b, l)$, which means b is introduced by some formula $@_{b'} @_l \Diamond_A \varphi$. However, since $(a, k) \prec_\Theta (b, k)$ and $G = (N^\Theta, \prec_\Theta)$ defined in Lemma 5.9 is a disjoint union of trees, we have $(b', l) = (a, k)$. Therefore, it follows that

$$m_\Theta((a, k)) = m_\Theta((b', l)) \geq |\Diamond_A \varphi| > |\varphi|.$$

We can prove this in a similar way in the case of $[\Box_A]$.

(e) Suppose that $@_b @_l \varphi$ is introduced by $[\Diamond_K]$. Then we have that there is another nominal $l' \in \mathbf{Nom}_K$ such that $@_b @_{l'} \Diamond_K \varphi \in \Theta$. However, it contradicts the maximality of l and φ. We can prove this in a similar way in the case of $[\Box_K]$.

(f) If $@_b @_l \varphi$ is introduced by $[@_A]$, then there is another nominal $b' \in \mathbf{Nom}_A$ such that $@_{b'} @_l @_b \varphi \in \Theta$. By Lemma 5.6, $@_b \varphi$ is a prefixed subformula of the root formula of Θ. Then, b is in the root formula, which contradicts that b is a generated nominal.

(g) If $@_b @_l \varphi$ is introduced by $[@_K]$, then there is another nominal $l' \in \mathbf{Nom}_K$ such that $@_b @_{l'} @_l \varphi \in \Theta$. However, it contradicts the maximality of l and φ.

(h) Suppose that $@_b @_l \varphi$ is introduced by $[Id_A]$. Then, we have another nominal $b' \in \mathbf{Nom}_A$ such that $@_{b'} @_l b \in \Theta$. Thus, by Lemma 5.6, b is in the root formula, which contradicts that b is a generated nominal.

(i) Suppose that $@_b @_l \varphi$ is introduced by $[Id_K]$. Then, we have another nominal $l' \in \mathbf{Nom}_K$ such that $@_b @_{l'} \varphi \in \Theta$. In this case, take l' instead and check which rule derives $@_b @_{l'} \varphi$. Note that this case cannot be applied infinitely. In this case, we also have $@_b @_{l'} l \in \Theta$, so l is in the root formula. However, the root formula contains only a finite number of nominals. □

Now, we address the remaining issue. As previously mentioned, G^Θ can be an infinite disjoint union of trees. Therefore, we consider constructing a single large tree by "grafting" the root of each tree onto a branch of another tree.

Definition 5.12. We define a relation $\lhd_\Theta$ over $\mathbf{Nom}_A \times \mathbf{Nom}_K$. For each root (b, l) of G^Θ, insert $(a, k) \lhd_\Theta (b, l)$, where (a, k) is the earliest occurrence of a label in Θ among those satisfying the following conditions:

(i) $@_b@_l\psi$ is derived from $@_a@_k\varphi$ by either $[\neg_M], [@_M]$, or $[Id_M]$ $(M \in \{A, K\})$ and

(ii) $m_\Theta((a, k)) \geq m_\Theta((b, l))$.

The following lemma guarantees that the definition of $\lhd_\Theta$ works well.

Lemma 5.13. *Let (b, l) be a root of G^Θ. If the root formula of Θ does not have the form $@_b@_l\varphi$, i.e., (b, l) is not a label of the root formula of Θ, then there is at least one pair (b', l') satisfying the conditions (i) and (ii) in Definition 5.12.*

Proof: For any root (b, l) of G^Θ, take a formula ψ such that $|\psi| = m_\Theta((b, l))$. If $@_b@_l\psi$ is introduced by $[\wedge]$ or $[\vee]$, then there is a formula $@_b@_l(\psi \wedge \chi)$ or $@_b@_l(\psi \vee \chi)$ in Θ, which contradicts the maximality of ψ. Moreover, if $@_b@_l\psi$ is derived by $[\Diamond_A]$, there exists a nominal $b' \in \mathbf{Nom}_A$ such that $@_{b'}@_l\Diamond_A\psi, @_{b'}@_l\Diamond_A b \in \Theta$. Then, we have $(b', l) \prec_\Theta (b, l)$, which contradicts that (b, l) is a root of G^Θ (the same applies to the cases for $[\Diamond_K]$, $[\Box_A]$, and $[\Box_K]$).

Therefore, $@_b@_l\varphi$ must be derived by either $[\neg_M]$, $[@_M]$, or $[Id_M]$ $(M \in \{A, K\})$. If $@_b@_l\psi$ is derived by $[\neg_A]$, we have $\psi = b$ and there is another nominal $b' \in \mathbf{Nom}_A$ such that $@_{b'}@_l\neg b \in \Theta$. Then, we have $m_\Theta((b', l)) \geq |\neg b| \geq |b| = |\psi|$. If $@_b@_l\psi$ is derived by $[@_A]$, there is another nominal $b' \in \mathbf{Nom}_A$ such that $@_{b'}@_l@_b\psi \in \Theta$. Then, we have $m_\Theta((b', l)) \geq |@_b\psi| \geq |\psi|$. In the case for $[Id_A]$, there is $b' \in \mathbf{Nom}_A$ such that $@_{b'}@_l\psi \in \Theta$. Then, we have $m_\Theta((b', l)) \geq |\psi|$. We leave the remainder of the proof of the cases for $[\neg_K]$, $[@_K]$, and $[Id_K]$. □

Recall that all the formulas in a tableau, except for accessibility formulas, are quasi-subformulas in the root formula. Observing the rules $[\neg_M]$, $[@_M]$, and $[Id_M]$ $(M \in \{A, K\})$, we have the following restriction:

(R) If $(a, k) \lhd_\Theta (a, l)$ holds, l occurs in a root formula. Moreover, if $(a, k) \lhd_\Theta (b, k)$ holds, b occurs in a root formula.

Since the number of nominals occurring in the root formula is finite, we cannot have the following infinite sequence

$$(a_0, k_0) \lhd_\Theta (a_1, k_1) \lhd_\Theta (a_2, k_2) \lhd_\Theta \cdots ,$$

where $(a_i, k_i) \neq (a_j, k_j)$ $(i \neq j)$ for all $i, j \in \mathbb{N}$.

Owing to $\lhd_\Theta$, we can construct a single tree by grafting trees in G^Θ.

Lemma 5.14. *Given a branch Θ of a tableau, we define a structure $\tilde{G}^\Theta = (N^\Theta, (\prec_\Theta \cup \lhd_\Theta))$. Then $\tilde{G}^\Theta$ is a finitely branching tree.*

Proof: When a new pair of nominals emerges in N^Θ, we apply either $[\Diamond_M]$, $[\neg_M]$, $[@_M]$, or $[Id_M]$ $(M \in \{A, K\})$ to Θ. Thus, any node of $\tilde{G}^\Theta$, except for the pair of nominals that serves as a label for the root formula, has the antecedent by either $\prec_\Theta$ or $\lhd_\Theta$. Since all formulas in the tableau are derived from the root formula, any node of $\tilde{G}^\Theta$ is connected to the label of the root formula.

We have checked that G^Θ is finitely-branching. Moreover, from the fact (R) and the fact that a root formula of Θ has only a finite number of nominals, $\lhd_\Theta$ is also finitely-branching.

It is not the case that both $(a, k) \prec_\Theta (b, l)$ and $(a', k') \lhd_\Theta (b, l)$ hold—otherwise, (b, l) would be and yet not be a root of G^Θ. Also, by the definition of $\lhd_\Theta$, any antecedent of $(b, l) \in N^\Theta$ by $\lhd_\Theta$ is unique. Together with the fact that G^Θ is a disjoint union of trees, we show that $\tilde{G}^\Theta$ is a tree. □

We do not show that $\tilde{G}$ is well-founded. In fact, the following infinite ascending branch may occur:

$$(a_0, k_0) \rhd_\Theta (a_1, k_1) \rhd_\Theta (a_2, k_2) \rhd_\Theta \ldots .$$

However, considering (R) and the finiteness of nominals occurring in the root formula, only a finite pair of nominals occurs in that sequence. Then, it may occur only as a finite loop. Therefore, this infinite ascending branch does not cause any problem in applying König's lemma. Now, we are ready to prove the main theorem.

Theorem 5.15. *The tableau calculus* $\mathbf{T}_{AK}$ *has the termination property.*

Proof: By *reductio ad absurdum.*

Suppose that there is a branch Θ of a tableau that is infinite. If only a finite number of nominals occur in Θ, then a finite number of formulas occur in Θ by Lemma 5.6, which contradicts the infinity of Θ. Thus, Θ contains infinitely many nominals, so N^{Θ} is infinite. Since $\tilde{G}^{\Theta}$ is a finitely branching tree, we find an infinite sequence

$$(a_0, k_0) \lhd_{\Theta}^{n_0} (a_1, k_1) \prec_{\Theta}^{+} (a_2, k_2) \lhd_{\Theta}^{n_1} (a_3, k_3) \prec_{\Theta}^{+} \cdots ,$$

by König's Lemma (note that we cannot make an infinite ascending chain by $\lhd_{\Theta}$). Applying Lemma 5.11 and Definition 5.12, we have an infinite descent

$$m_{\Theta}((a_0, k_0)) \geq m_{\Theta}((a_1, k_1)) > m_{\Theta}((a_2, k_2)) \geq m_{\Theta}((a_3, k_3)) > \cdots ,$$

which contradicts the definition of m_{Θ}. □

5.3. Completeness

The soundness of $\mathbf{T}_{\mathrm{AK}}$ can be proved in a similar way to the method introduced in [13]. Then, we proceed to prove the completeness of $\mathbf{T}_{\mathrm{AK}}$.

In preparation, we define some terminology. We say a branch Θ *saturated* if every new formula generated by applying some rules already exists in Θ. Moreover, $s \in \mathbf{Nom}_A \cup \mathbf{Nom}_K$ is a *right nominal* in a branch Θ if there are some $a \in \mathbf{Nom}_A$ and $k \in \mathbf{Nom}_K$ such that $@_a @_k s \in \Theta$. From Lemma 5.6, it is straightforward that all the right nominals in Θ occur in the root formula of Θ.

Definition 5.16. Given a branch Θ of a tableau, we define binary relations $\sim_{\Theta}^{A} \subset \mathbf{Nom}_A \times \mathbf{Nom}_A$ and $\sim_{\Theta}^{K} \subset \mathbf{Nom}_K \times \mathbf{Nom}_K$ on a set of right nominals in Θ as follows.

- $a \sim_{\Theta}^{A} b$ if there is a nominal $k \in \mathbf{Nom}_K$ such that $@_a @_k b \in \Theta$.
- $k \sim_{\Theta}^{K} l$ if there is a nominal $a \in \mathbf{Nom}_A$ such that $@_a @_k l \in \Theta$.

We can show that if Θ is saturated, then both $\sim^A_\Theta$ and $\sim^K_\Theta$ are equivalence relations (see [13, Lemma 5.7]). Then, we define equivalence classes $[a]^A_\Theta$ and $[k]^K_\Theta$ of a right nominal by the relation $\sim^A_\Theta$ and $\sim^K_\Theta$, respectively. Moreover, if we arrange the elements in $\mathbf{Nom}_A$ and $\mathbf{Nom}_K$ in the order of occurrence in the branch, respectively, we can fix the minimal element $\min(A)$ for any subset A in $\mathbf{Nom}_A$ or $\mathbf{Nom}_K$. They enable us to take a representative of nominals.

Definition 5.17. Let Θ be a saturated branch and $s \in \mathbf{Nom}_A \cup \mathbf{Nom}_K$ a nominal occurring in Θ. The *urfather* of s on Θ (denoted by $u_\Theta(s)$) is defined as follows:

$$u_\Theta(s) = \begin{cases} \min([b]^A_\Theta) & \text{if } s \in \mathbf{Nom}_A \text{ and } @_s@_k b \in \Theta \\ \min([l]^K_\Theta) & \text{if } s \in \mathbf{Nom}_K \text{ and } @_a@_s l \in \Theta \\ s & \text{otherwise.} \end{cases}$$

Given an $s \in \mathbf{Nom}_A \cup \mathbf{Nom}_K$, $u_\Theta(s)$ is well-defined (see [13, Proposition 5.10]). Moreover, the following lemma holds like [13, Lemma 5.11].

Lemma 5.18. *Let Θ be a saturated branch. Then, the following properties hold:*

1. *If $a \in \mathbf{Nom}_A$ is a right nominal in Θ, then there is some $k \in \mathbf{Nom}_K$ such that $@_{u_\Theta(a)}@_k a \in \Theta$. Likely, if $k \in \mathbf{Nom}_K$ is a right nominal in Θ, then there is some $a \in \mathbf{Nom}_A$ such that $@_a@_{u_\Theta(k)}k \in \Theta$.*

2. *If $@_a@_k b \in \Theta$ ($b \in \mathbf{Nom}_A$), then $u_\Theta(a) = u_\Theta(b)$. Likely, if $@_a@_k l \in \Theta$ ($l \in \mathbf{Nom}_K$), then $u_\Theta(k) = u_\Theta(l)$.*

3. *For every prefixed subformula $@_a@_k\varphi \in \Theta$ of the root formula of Θ, we have $@_{u_\Theta(a)}@_{u_\Theta(k)}\varphi \in \Theta$.*

Definition 5.19. Given an open saturated branch Θ, a model $\mathcal{M}^\Theta_{\mathrm{AK}} = (W^\Theta_A \times W^\Theta_K, (R^\Theta_y)_{y \in W^\Theta_K}, (S^\Theta_x)_{x \in W^\Theta_A}, V^\Theta)$ generated from Θ is defined as follows:

$$\begin{aligned}
W_A^\Theta &= \{u_\Theta(a) \mid a \in \mathbf{Nom}_A \text{ occurs in } \Theta\}\\
W_K^\Theta &= \{u_\Theta(k) \mid k \in \mathbf{Nom}_K \text{ occurs in } \Theta\}\\
R_{u_\Theta(k)}^\Theta &= \{(u_\Theta(a), u_\Theta(b)) \mid \text{accessibility formula } @_a @_k \Diamond_A b \in \Theta\}\\
S_{u_\Theta(a)}^\Theta &= \{(u_\Theta(k), u_\Theta(l)) \mid \text{accessibility formula } @_a @_k \Diamond_K l \in \Theta\}\\
V^\Theta(p_A) &= \{u_\Theta(a) \mid \text{there is } k \in \mathbf{Nom}_K \text{ such that } @_a @_k p_A \in \Theta\},\\
&\quad \text{where } p_A \in \mathbf{Prop}_A\\
V^\Theta(p_K) &= \{u_\Theta(k) \mid \text{there is } a \in \mathbf{Nom}_A \text{ such that } @_a @_k p_K \in \Theta\},\\
&\quad \text{where } p_K \in \mathbf{Prop}_K\\
V^\Theta(a) &= \{u_\Theta(a)\}, \text{ where } a \in \mathbf{Nom}_A\\
V^\Theta(k) &= \{u_\Theta(k)\}, \text{ where } k \in \mathbf{Nom}_K.
\end{aligned}$$

LEMMA 5.20. *Let Θ be an open saturated branch and let $@_a @_k \varphi$ be a prefixed subformula of the root formula of Θ. Then, we have:*

$$\textit{if } @_a @_k \varphi \in \Theta, \textit{ then } \mathcal{M}_{AK}^\Theta, (u_\Theta(a), u_\Theta(k)) \models \varphi.$$

PROOF: By induction on the complexity of φ.

[$\varphi = p_A$] Suppose that $@_a @_k p_A \in \Theta$. Then, by definition, we have $u_\Theta(a) \in V^\Theta(p_A)$. Therefore, $\mathcal{M}_{\mathrm{AK}}^\Theta, (u_\Theta(a), u_\Theta(k)) \models p_A$ holds. We can do the same in the case $\varphi = p_K$.

[$\varphi = \neg p_A$] Suppose that $@_a @_k \neg p_A \in \Theta$. Since Θ is open, $@_a @_l p_A \notin \Theta$ is valid for all $l \in \mathbf{Nom}_K$. This means that $u_\Theta(a) \notin V^\Theta(p_A)$. Therefore, we have $\mathcal{M}_{\mathrm{AK}}^\Theta, (u_\Theta(a), u_\Theta(k)) \models \neg p_A$. We can do the same in the case $\varphi = \neg p_K$.

[$\varphi = b$] Suppose that $@_a @_k b \in \Theta$ ($b \in \mathbf{Nom}_A$). By Lemma 5.18, we have $u_\Theta(a) = u_\Theta(b)$. Thus, we have $\mathcal{M}_{\mathrm{AK}}^\Theta, (u_\Theta(a), u_\Theta(k)) \models b$ by the definition of $V^\Theta(b)$. We can do the same in the case $\varphi = l$ ($l \in \mathbf{Nom}_K$).

[$\varphi = \neg b$] Suppose that $@_a @_k \neg b \in \Theta$ ($b \in \mathbf{Nom}_A$). Since Θ is saturated, we also have $@_b @_k b \in \Theta$. Then by Lemma 5.18, it follows that $@_{u_\Theta(a)} @_{u_\Theta(k)} \neg b, @_{u_\Theta(b)} @_{u_\Theta(k)} b \in \Theta$. However, since Θ is open, $u_\Theta(a) \neq u_\Theta(b)$, which means $\mathcal{M}^\Theta_{\mathrm{AK}}, (u_\Theta(a), u_\Theta(k)) \models \neg b$. We can do the same in the case $\varphi = \neg l$ ($l \in \mathbf{Nom}_K$).

[$\varphi = \psi \wedge \chi, \psi \vee \chi$] Straightforward.

[$\varphi = \Diamond_A \psi$] If $@_a @_k \Diamond_A \psi \in \Theta$, then there is a nominal $b \in \mathbf{Nom}_A$ such that $@_a @_k \Diamond_A b, @_b @_k \psi \in \Theta$. It follows that $u_\Theta(a) R^\Theta_{u_\Theta(k)} u_\Theta(b)$ from the former formula. Moreover, by the latter formula and the induction hypothesis, $\mathcal{M}^\Theta_{\mathrm{AK}}, (u_\Theta(b), u_\Theta(k)) \models \psi$ holds. Therefore, we have $\mathcal{M}^\Theta_{\mathrm{AK}}, (u_\Theta(a), u_\Theta(k)) \models \Diamond_A \psi$. We can do the same in the case $\varphi = \Diamond_K \psi$.

[$\varphi = \Box_A \psi$] Suppose that $@_a @_k \Box_A \psi \in \Theta$. Now, assume that there is a nominal $b \in \mathbf{Nom}_A$ such that there is an accessibility formula $@_a @_k \Diamond_A b \in \Theta$ (If not, then there is no state reachable from $u_\Theta(a)$. Thus, $\mathcal{M}^\Theta_{\mathrm{AK}}, (u_\Theta(a), u_\Theta(k)) \models \Box_A \psi$ is straightforward). Then, from the definition of R^Θ, we have $u_\Theta(a) R^\Theta_{u_\Theta(k)} u_\Theta(b)$. Moreover, since Θ is saturated, we obtain $@_b @_k \psi \in \Theta$. By the induction hypothesis, $\mathcal{M}^\Theta_{\mathrm{AK}}, (u_\Theta(b), u_\Theta(k)) \models \psi$ holds. Since we pick up an arbitrary b, we have $\mathcal{M}^\Theta_{\mathrm{AK}}, (u_\Theta(a), u_\Theta(k)) \models \Box_A \psi$. We can do the same in the case $\varphi = \Box_K \psi$.

[$\varphi = @_b \psi, @_k \psi$] Straightforward. □

This lemma is called *model existence lemma*. Note that by combining it with the termination property of $\mathbf{T}_{\mathrm{AK}}$, we can show the finite model property of agent-knowledge logic as well as the completeness.

Theorem 5.21. *The tableau calculus $\mathbf{T}_{AK}$ is complete for the class of all AK models.*

Proof: We show the contraposition.

Suppose that φ is not provable in $\mathbf{T}_{\mathrm{AK}}$. Then, we can find an open and saturated branch Θ with the root formula $@_a @_k \neg \varphi$, where $a \in \mathbf{Nom}_A$ and $k \in \mathbf{Nom}_K$ do not occur in φ. Then, by Lemma 5.20, we have

$\mathcal{M}^{\Theta}_{\mathrm{AK}}, (u_{\Theta}(a), u_{\Theta}(k)) \models \neg\varphi$. It means that there is an AK model and its possible world that falsifies φ. □

The termination property and completeness of the tableau calculus tell us about the decidability of logic. If φ is provable, then it is provable in finite time. By contrast, if φ is unprovable, we can make a finite counterexample model. From them, the following corollary holds.

Corollary 5.22. The agent-knowledge logic is decidable.

6. Future Work and Perspective

6.1. Adding More Operators

One of the proposed future work to make agent-knowledge logic more fruitful is to add new operators.

For example, let us imitate some operators of epistemic logic. Given a group $G \subseteq \mathbf{A}$ of agents, the *everybody knows operator* E_G is defined as follows:

$$\mathcal{M}_{\mathrm{EL}}, w \models E_G\varphi \iff \mathcal{M}, w \models K_i\varphi \text{ for all } i \in G.$$

Intuitively, this formula says that everyone in the group G knows φ. Recalling that a formula $K_i\varphi$ is translated into $@_{T(i)}\Box_K T(\varphi)$, we can define the everybody knows operator in agent-knowledge logic as follows, where G is a subset of $\mathbf{Nom}_A$:

$$\mathcal{M}_{\mathrm{AK}}, (x, y) \models E_G\varphi \iff \mathcal{M}, (x, y) \models @_a\Box_K\varphi \text{ for all } a \in G.$$

This definition works well even if two nominals $a, b \in \mathbf{Nom}_A$ point to the same agent, but only one of a or b is in G. If they point to the same agent, then the two formulas $@_a\varphi$ and $@_b\varphi$ are equivalent for any world in any model.

Based on these definitions, we may mimic other operators used in epistemic logic, such as the operator for common knowledge C_G and the operator for distributed knowledge D_G. Can we analyze problems that have been considered in epistemic logic, such as the muddy children puzzle, using agent-knowledge logic? Research in this direction may be able to reflect various results in epistemic logic in agent-knowledge logic as well.

Additionally, there is another direction to research agent-knowledge logic, to introduce a universal operator used in hybrid logic, which may enable us to symbolize more expressions in natural language. The definition of the universal operators A_A and E_A are as follows:

$$\begin{aligned}\mathcal{M}_{\mathrm{AK}}, (x, y) \models \mathsf{A}_A\varphi &\iff \mathcal{M}_{\mathrm{AK}}, (z, y) \models \varphi \text{ for all } z \in W_A,\\ \mathcal{M}_{\mathrm{AK}}, (x, y) \models \mathsf{E}_A\varphi &\iff \text{there is some } z \in W_A\\ &\qquad \text{such that } \mathcal{M}_{\mathrm{AK}}, (z, y) \models \varphi.\end{aligned}$$

Owing to these operators, we can write some expressions as follows:

- $\mathsf{E}_A\Box_K p_K$: Someone knows p_K.
- $\Box_K\mathsf{A}_A\Box_K p_K$: An agent knows that all the people know p_K.
- $\mathsf{E}_A\Box_A\Box_K p_K$: There is a person all of whose friends know p_K.

6.2. Seeking More Usage

Research in agent-knowledge logic is not merely about adding new operators. In contrast, some of the languages proposed in this paper lack effective application. For example, agent names in $\mathbf{Nom}_A$ and agent-independent propositions in $\mathbf{Prop}_K$ play essential roles to embed epistemic logic into agent-knowledge logic. However, this paper does not go so far as to present practical applications for the sets $\mathbf{Prop}_A$ and $\mathbf{Nom}_K$, although it provides interpretations for the elements of them.

It might be an interesting direction to assign interpretations to the two sets other than agents and their epistemic alternative. For example, as often seen in modal logics applied to computer science, elements of W_K can be interpreted as states of a computer. In that case, W_A can be seen as a set of named computers. Under this interpretation, agent-knowledge logic might be able to describe a system where computers mutually monitor each other's behavior.

6.3. Hilbert-Style Axiomatization

In this paper, we have given the tableau calculus for agent-knowledge logic as a proof system. We can give another proof system, for example, Hilbert-style axiomatization.

Fortunately, there is already abundant prior research in related fields, such as the aforementioned Sano's work [14] on two-dimensional hybrid logic (see also [1], which focused on Facebook logic). For LHS, recent research by Chen and Li [5] gives the axiomatization.

6.4. Complexity

In this paper, we have shown the decidability of the agent-knowledge logic using tableau calculus. But what about its computational complexity? As is already known, the satisfiability problem for epistemic logic is PSPACE-complete [8]. If we want to use agent-knowledge logic as an alternative to epistemic logic, we expect it to be PSPACE-complete.

The analysis of computational complexity for a fusion in modal logic may provide a clue to solving this problem. An explanation for a fusion is in [7, p. 111]:

> Let L_1 and L_2 be two multimodal logics formulated in languages $\mathcal{L}_1$ and $\mathcal{L}_2$, both containing the language $\mathcal{L}$ of classical propositional logic, but having disjoint sets of modal operators. Denote by $\mathcal{L}_1 \otimes \mathcal{L}_2$ the union of $\mathcal{L}_1$ and $\mathcal{L}_2$. Then the *fusion* $L_1 \otimes L_2$ of L_1 and L_2 is the smallest multimodal logic L in the language $\mathcal{L}_1 \otimes \mathcal{L}_2$ containing $L_1 \cup L_2$.

From the results of Halpern and Moses [8], we can obtain that the satisfiability problem for $\mathbf{K} \otimes \mathbf{K}$ is PSPACE-complete. Since agent-knowledge logic is based on $\mathbf{K} \otimes \mathbf{K}$, we may be able to answer the question with reference to this proof.

Acknowledgements. I would like to thank Prof. Ryo Kashima and Leonardo Pacheco for their invaluable advice in writing this paper. I am also thankful for invaluable comments from Prof. Valentin Goranko, Michał Zawidzki, and Paweł Płaczek, who met in NCL'24.

References

[1] P. Balbiani, S. F. González, *Indexed Frames and Hybrid Logics*, [in:] N. Olivetti, R. Verbrugge, S. Negri, G. Sandu (eds.), **Advances in Modal Logic 13**, College Publications, Chichester (2020), pp. 56–72.

[2] P. Blackburn, B. ten Cate, *Pure Extensions, Proof Rules, and Hybrid Axiomatics*, **Studia Logica**, vol. 84(2) (2006), pp. 277–322, DOI: https://doi.org/https://doi.org/10.1007/s11225-006-9009-6.

[3] T. Bolander, P. Blackburn, *Termination for Hybrid Tableaus*, **Journal of Logic and Computation**, vol. 17(3) (2007), pp. 517–554, DOI: https://doi.org/https://doi.org/10.1093/logcom/exm014.

[4] T. Braüner, **Hybrid Logic and its Proof-Theory**, vol. 37, Springer Science & Business Media (2011), DOI: https://doi.org/https://doi.org/10.1007/978-94-007-0002-4.

[5] Q. Chen, D. Li, *Logic of the Hide and Seek Game: Characterization, Axiomatization, Decidability*, [in:] N. Gierasimczuk, F. R. Velázquez-Quesada (eds.), **Dynamic Logic. New Trends and Applications**, Springer Nature Switzerland (2024), pp. 20–34, DOI: https://doi.org/https://doi.org/10.1007/978-3-031-51777-8_2.

[6] R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi, **Reasoning about knowledge**, MIT press (1995), DOI: https://doi.org/https://doi.org/10.7551/mitpress/5803.001.0001.

[7] D. M. Gabbay, **Many-Dimensional Modal Logics: Theory and Applications**, Elsevier North Holland (2003).

[8] J. Y. Halpern, Y. Moses, *A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief*, **Artificial Intelligence**, vol. 54(3) (1992), pp. 319–379, DOI: https://doi.org/https://doi.org/10.1016/0004-3702(92)90049-4.

[9] J. Hintikka, **Knowledge and Belief: An Introduction to the Logic of the Two Notions**, 2nd ed., Cornell University Press (1962).

[10] A. Indrzejczak, *Modal Hybrid Logic*, **Logic and Logical Philosophy**, vol. 16(2-3) (2007), pp. 147–257, DOI: https://doi.org/https://doi.org/10.12775/LLP.2007.006.

[11] D. Li, S. Ghosh, F. Liu, Y. Tu, *On The Subtle Nature of a Simple Logic of The Hide and Seek Game*, [in:] **Logic, Language, Information, and Computation: 27th International Workshop, WoLLIC 2021, Virtual Event, October 5–8, 2021, Proceedings 27**, Springer (2021), pp. 201–218, DOI: https://doi.org/https://doi.org/10.1007/978-3-030-88853-4_13.

[12] D. Li, S. Ghosh, F. Liu, Y. Tu, *A Simple Logic of The Hide and Seek Game*, **Studia Logica**, vol. 111(5) (2023), pp. 821–853, DOI: https://doi.org/https://doi.org/10.1007/s11225-023-10039-4.

[13] Y. Nishimura, *Completeness of Tableau Calculi for Two-dimensional Hybrid Logics*, **Journal of Logic and Computation**, vol. 35(3) (2025), p. exae018, DOI: https://doi.org/https://doi.org/10.1093/logcom/exae018.

[14] K. Sano, *Axiomatizing Hybrid Products: How Can We Reason Many-Dimensionally in Hybrid Logic?*, **Journal of Applied Logic**, vol. 8(4) (2010), pp. 459–474, DOI: https://doi.org/https://doi.org/10.1016/j.jal.2010.08.006.

[15] K. Sano, *Paper Review: Logic in The Community*, **Journals of The Japanese Society for Artificial Intelligence**, vol. 26(6) (2011), pp. 703–707, DOI: https://doi.org/https://doi.org/10.11517/jjsai.26.6_703, (written in Japanese).

[16] J. Seligman, F. Liu, P. Girard, *Logic in the Community*, [in:] **Proceedings of the 4th Indian Conference on Logic and Its Applications**, Springer Berlin Heidelberg (2011), pp. 178–188, DOI: https://doi.org/https://doi.org/10.1007/978-3-642-18026-2_15.

[17] J. Seligman, F. Liu, P. Girard, *Facebook and the Epistemic Logic of Friendship*, [in:] **Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge** (2013), pp. 229–238.

[18] J. van Benthem, *Epistemic Logic and Epistemology: The State of their Affairs*, **Philosophical Studies**, vol. 128 (2006), pp. 249–76, DOI: https://doi.org/https://doi.org/10.1007/s11098-005-4052-0.

Yuki Nishimura
Institute of Science Tokyo
School of Computing
152-8550, Ookayama 2-12-1
Meguro-ku, Tokyo, Japan
e-mail: nishimura.y.as@m.titech.ac.jp

Funding information: The research of the author was supported by JST SPRING, Grant Number JPMJSP2106 and JPMJSP2180.

Conflict of interests: None.

Ethical considerations: The Author assures of no violations of publication ethics and takes full responsibility for the content of the publication.

Declaration regarding the use of GAI tools: Not used.