A Modelling Language for Capturing and Analysing Complex Epistemic Situations

September 13, 2012

Abstract

Epistemic logic uses Kripke semantics to model knowledge. However, one of the problems in using Kripke semantics to analyse systems and situations is that it is difficult to confirm whether a Kripke model that putatively models a particular system does indeed model it correctly. This is because epistemic Kripke models are difficult for human to construct from a scenario. Moreover the subtleties involved in a more complex epistemic scenario are hard to capture formally in a Kripke model. In this paper we propose a modelling language which makes it easier to capture and analyse complex epistemic situations. We will take a specific epistemic scenario as an example. We will then use this language to represent and reason about the example. Using the semantics of our language, we will look at the properties derived from the Kripke model we have generated. We will then use our language to formally reason about $S5_n$ models in general. Finally we will describe an application for model checking using this language.

1 Introduction and Background

The information a person possesses, and the reasoning he does based on that information, constitutes his knowledge and belief. Knowledge is an important part of the human decision making process. An agent is an intelligent unit that can sense the environment it is located in and act upon it to achieve its objectives [5]. The information an agent possesses is its knowledge, which plays an important role in the decision making process.

A logic of knowledge, rooted in philosophy, was first presented by Jaakko Hintikka in 1962 [3]. This logic was later adapted by Computer Scientists and is widely used in Artificial intelligence (AI).

An artificial agent (or a group of agents) needs to have a mechanism which can capture knowledge and reason about it, to be able to make decisions. For instance, a group of agents may be working together to reach a common goal. To achieve this they are required to communicate with each other. A faulty communication medium can result in a message sent from Agent A being lost before reaching Agent B. Agent A needs to knows when the message has not reached B and also to know that B knows this [6]. This is an example where agents need the ability to capture and to reason about knowledge.

Knowledge representation and reasoning is the area of AI which studies symbolic representation of knowledge and uses automated reasoning for manipulating pieces of knowledge. Logic is used as a tool to reason about knowledge. The semantics of the language of logic are well defined, making it a suitable tool for this purpose.

1.1 Possible Worlds Model

The "possible worlds model", also known as the "Kripke model", is commonly used to represent the knowledge agents possess. In this model, an agent's knowledge is characterised in terms of possible worlds or states. A Kripke model is a directed labelled graph where nodes represent the reachable states, and edges illustrate their accessibility relations. A labelling function is used to associate with each node a set of properties holding in that state. Truth values are represented by primitive or complex propositions at each state. A complex proposition consists of primitive propositions and logical operators. The idea is that, apart from the current state, other states are also considered. If an agent knows something in one state then in all the accessible states from there he considers it to be true. This means that he cannot distinguish the actual state from other accessible states. This is formally represented as follows:

Definition 1. *Let A_g be a set of n agents and Φ a set of propositional constants. The Kripke model* M *is a tuple $\langle S, \pi, R_1, R_2, \cdots, R_n \rangle$ where:*

i. S is a non-empty set of states,

ii. $\pi : S \to 2^{\Phi}$ is a truth assignment to the propositional constants per state,

iii. $R_i \subseteq S \times S$ is the accessibility relation for $i \in A_g$.

(M, s) represents a state $s \in S$ in the Kripke model M. The meaning of $R_i(s,t)$ is that from (M, s), agent i, given his knowledge in s, considers (M, t) possible.
A classic example of how the possible worlds model is used to capture and represent knowledge is as follows:

Example 1. *Alice and Bob both know that it is currently raining in Liverpool. Alice also knows that it is also raining in London while Bob does not know this.*

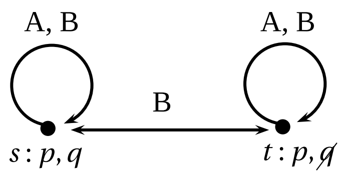

Figure 1: A Kripke model illustrating Alice's and Bob's knowledge

Figure 1 is a Kripke model representing the above scenario. There are two states in this model labelled s and t. p is a primitive proposition and is true when *"It is raining in Liverpool"*. q is a second primitive proposition and is true when *"It is raining in London"*. Directed arrows are labelled **A** for "Alice" and **B** for "Bob".
Suppose that the actual state is s. In this state, Alice finds only s accessible (self-loop) in which it rains both in Liverpool and London (p, q). This means that Alice knows that *"It is raining in Liverpool and London"*. In the same state, Bob finds t accessible as well. In both s and t, *"It is raining in Liverpool"*, (p), but while in s, *"It is raining in London"*, (q), in t this does not hold $(\not q)$. This means that Bob does not know whether it is raining in London or not. It is said that "Bob is *ignorant* about raining in London".

As can be seen from this example, Alice considers possible only the states that correspond to her knowledge. These worlds which correspond to an agent's knowledge are called *epistemic alternatives.*

1.2 Epistemic Logic

Epistemic logic is formulated using modal logic [1] and is used to reason about knowledge. Modal logic is concerned with *necessary* and *possible* truths [7]. A necessary truth is something that is true because it is not possible for it to be otherwise. Most mathematical facts, such as "the square root of 4 is 2", are like this. By contrast a possible truth is one that could have been otherwise. For example "Spain is the winner of World Cup 2010" is a possible truth as another team could equally have won.

1.2.1 Syntax

The following defines the syntax of epistemic logic.

Definition 2. *Let* $A_g = \{1, \ldots, n\}$ *be a set of agents and* $\Phi = \{p, q, \ldots\}$ *a set of Boolean variables. The language of* $S5_n$ *with respect to these sets is defined via the following grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \psi \mid K_i\varphi$$

Where "$\neg$" indicates negation, "$\vee$" indicates disjunction, $p \in \Phi$ and $i \in A_g$. K_i is the individual knowledge operator. φ is a formula and $K_i\varphi$ is read as "*agent i knows* φ". The abbreviation $M_i\varphi$ is normally used for $\neg K_i \neg\varphi$ and is read as "*agent i considers* φ *possible*". We also take $\top$ as an abbreviation for a propositional tautology such as $p \vee \neg p$.

1.2.2 Semantics

We now explain what it means for a formula φ to be true in a state s of a model M, written as $\mathsf{M}, s \models \varphi$ and read as *"φ is true in* M, s*".*

Definition 3. *Let* M *be a Kripke model ,* $s \in \mathsf{M}$ *a state in this model,* φ *and* ψ *epistemic formulae and* i *an agent. The semantics of epistemic logic are defined as follows [4]:*

- $\mathsf{M}, s \models \top$
- $\mathsf{M}, s \models p \Leftrightarrow p \in \pi(s),$ *where* $p \in \Phi$
- $\mathsf{M}, s \models \neg\varphi \Leftrightarrow \mathsf{M}, s \not\models \varphi$
- $\mathsf{M}, s \models \varphi \vee \psi \Leftrightarrow \mathsf{M}, s \models \varphi$ *or* $\mathsf{M}, s \models \psi$
- $\mathsf{M}, s \models K_i\varphi \Leftrightarrow \forall s' : (R_i(s, s') \Rightarrow \mathsf{M}, s' \models \varphi)$
- $\mathsf{M}, s \models M_i\varphi \Leftrightarrow \exists s' : (R_i(s, s') \& \mathsf{M}, s' \models \varphi)$

Name	Property	Epistemic Formula	First-order Formula
T	Reflexive	$K_i\varphi \to \varphi$	$\forall s \in S : R_i(s,s)$
D	Serial	$K_i\varphi \to M_i\varphi$	$\forall s \in S, \exists s' \in S : R_i(s,s')$
B	Symmetric	$\varphi \to K_i M_i \varphi$	$\forall s,s' \in S : R_i(s,s') \to R_i(s',s)$
4	Transitive	$K_i\varphi \to K_i K_i \varphi$	$\forall s,s',s'' \in S : R_i(s,s') \wedge R_i(s',s'') \to R_i(s,s')$
5	Euclidean	$\neg K_i\varphi \to K_i \neg K_i \varphi$	$\forall s,s',s'' \in S : R_i(s,s') \wedge R_i(s,s'') \to R_i(s',s'')$

Table 1: Correspondence theory

A formula φ is *satisfiable* in the model M if it is true in at least one of its states. A formula φ is *valid* in the model M if it is true in all the states in M. This is written as $\mathsf{M} \vDash \varphi$ and read as *"φ is true in M"*. If φ is *valid* in all models it will be indicated as $\vDash \varphi$.
In a state of a Kripke model, $K_i\varphi$ holds when φ is true in all the accessible states whereas $M_i\varphi$ holds when there is at least one accessible state where φ is true.
In Figure 1, s is the only **A** accessible world from s in which both p and q holds. t is a **B** accessible world from s where p holds while q does not. In both **B** accessible worlds from s, which are s and t, **A** knows $(p \wedge q)$ or $(p \wedge \neg q)$. We show this as follows:

$$\mathsf{M}, s \vDash K_A(p \wedge q) \wedge \neg K_B(p \wedge q) \wedge K_B(K_A(p \wedge q) \vee K_A(p \wedge \neg q))$$

We also make use of the following definitions.

Definition 4. *Let i be an agent and φ an epistemic formula. Then*

- $KW_i\varphi \equiv (K_i\varphi) \vee (K_i\neg\varphi)$
- $IGN_i\varphi \equiv (\neg K_i\varphi) \wedge (\neg K_i\neg\varphi)$

Thus, for example, $KW_i p$ expresses the fact that agent i knows that p is true or knows that p is false. $IGN_j p$ intuitively means that j is ignorant about the value of p. It can be seen that $KW_i\varphi \equiv \neg IGN_i\varphi$.

1.2.3 Properties of Knowledge

In epistemic logic, accessibility relations are *equivalence relations*. This means that they are binary relations which are *reflexive* and *euclidean* (or *reflexive*, *symmetric* and *transitive*). Table 1 expresses this in more detail [8].
The following axiom system, known as **K**, derives all the valid sentences in all Kripke models.

Axiom 1. *All propositional tautologies*

Axiom 2. ***(K)**: $\vDash (K_i\varphi \wedge K_i(\varphi \to \psi)) \to K_i\psi$*

Rule 1. ***(Modus Ponens)**: $\vdash \varphi,\ \vdash \varphi \to \psi \Rightarrow\ \vdash \psi$*

Rule 2. ***(Necessitation)**: $\vdash \varphi \Rightarrow\ \vdash K_i\varphi$*

Adding the above axioms and rules to the following axioms results in a system called S_5, also known as $KT5$. S_5 is the standard epistemic logic system.

Axiom 3. ***(T)***: $K_i\varphi \rightarrow \varphi$

Axiom 4. ***(4)***: $K_i\varphi \rightarrow K_iK_i\varphi$

Axiom 5. ***(5)***: $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$

Axiom 3 which derives from reflexivity means that "known facts are true" and is called the *truth* or *knowledge* axiom. This axiom distinguishes knowledge from belief. An agent can have false belief but he cannot know something that is false. Every state in an S_5 model is reflexive which means that the actual state is always one of the states that an agent considers possible.
Axioms 4 and 5 mean that an agent can do introspection regarding his knowledge. Axiom 4 derives from transitivity and means that "an agent knows that he knows something". This is called the *positive introspection* axiom. Finally Axiom 5 which is derived from the relation being Euclidean, means that "an agent knows that he does not know something" and is called the *negative introspection* axiom.

2 Difficulties of using a Kripke model

Kripke models provide a framework which is used for modelling knowledge. An agent's knowledge or ignorance is based on whether he can distinguish the actual world from other worlds. To reason about an epistemic scenario it is first required to construct a corresponding Kripke model. A single epistemic scenario can be represented by many Kripke models.
There are certain difficulties in using epistemic logic based on Kripke model, one of which is the problem in checking whether a Kripke model that putatively models an epistemic scenario does indeed model it. As we will see in an example later, it is not easy to capture formally an epistemic scenario within a Kripke model, especially considering the subtleties frequently involved in such scenarios.
There is also the difficulty of verifying an epistemic formula in a model; i.e. checking whether the model satisfies the formula. We use the Muddy Children puzzle to elaborate these problems.
The Muddy Children puzzle is an example commonly used in the literature to reason about the knowledge of a group of agents [2]. The idea is that by repeatedly announcing a certain statement that the agents already know, and without giving out any new information, the state of knowledge in a group of agents will change. They will eventually reach common knowledge. We will use a modified version of the Muddy Children puzzle as follows:

Example 2. *Imagine that* n *children are gathered in a circle after playing in the garden. They have been told by their mother not to get muddy otherwise there will be consequences. Some of them, say* k *of them, happen to have mud on their forehead. Each can see the mud on other's foreheads but not on his own. All this is common knowledge among the children.*

Let us assume that there are three children, or agents as they will be called from now on. The first two, Agent 1 and Agent 2, have muddy foreheads while Agent 3 is clean. We can describe this situation by a tuple of 0's and 1's of the form $(x_1, \ldots, x_n)$ where $x_i = 1$ if child i is muddy and $x_i = 0$ otherwise. Therefore the above example is represented as $(1, 1, 0)$. We are now required to find the worlds each agent considers possible. For example for agent 3, $(1, 1, 1)$ is a possible world since his only doubt is about his own forehead. Thus the only other possible world for him is one where he

has a muddy forehead. Of course in this world Agent 1 and 2 still have muddy foreheads. To go further, in every world Agent 3 finds possible, Agents 1 and 2 have muddy foreheads. Similarly Agent 1 will consider $(0,1,0)$ and Agent 2 will consider $(1,0,0)$ possible. The Kripke model for a general situation where there are n agents will have 2^n states.
The next stage is to find out the properties which need to be included at each state. Since we are reasoning about muddy children, we use $\Phi = \{m_1, m_2, m_3\}$, where m_i states that *"child i has a muddy forehead"*. Therefore we define π so that $(\mathsf{M}, (x_1, x_2, x_3)) \models m_i$ iff $x_i = 1$. A world is considered possible by child i when he agrees on all except possibly the i^{th} element. To define the R_i relations, we take $(s, s') \in R_i$ such that s and s' agree on all except the i^{th} element.

It can be seen that generating a Kripke model for a simple scenario such as the Muddy Children puzzle, requires consideration of many elements, making it a complicated process.
Let us return to the Muddy Children puzzle. Agent 1 can see both Agents 2's forehead and 3's forehead which means that he knows whether they are muddy or not. At the same time he does not know about his own forehead. A more natural way of expressing this would be with a statement such as

`1 sees` m_2`;`

where 1 refers to Agent 1 and m_2 to Agent 2's forehead. The same can be used in the form of

`1 doesn't see` m_1`;`

to represent that Agent 1 is ignorant about his own forehead.

3 Epistemic Modelling Language

We now introduce the Epistemic Modelling Language (EML). The idea of this language is to use a collection of "`sees`" and "`doesn't see`" statements to place constraints on the Kripke model of an epistemic scenario. Suppose a system with n agents, $A_g = \{a_1, \ldots, a_n\}$ and m propositional variables $\Phi = \{p_1, \ldots, p_m\}$. The visibility of these variables is restricted, so that a_1, let us say, sees variable p_1. Take any equivalence class of possible worlds through agent a_1's accessibility relation, then the value of p_1 will be the same in each world in this equivalence class. This defines the constraints on any Kripke model representing this system.

3.1 Syntax

EML is defined by the following grammar.

```
<scenario> ::=
   scenario [
     <vignette>,
     ...,
     <vignette>
    ]

<vignette> ::=
```

```
   <atomicVignette>
|  <namedVignette>


<atomicVignette> ::=
   <AGENT> sees <VARIABLE>;
|  <AGENT> doesn't see <VARIABLE>;
|  <AGENT> sees <vignette>;
|  <AGENT> doesn't see <vignette>;

<namedVignette> ::=
   vignette <IDENTIFIER> [
      <vignette>,
      ... ,
      <vignette>
   ]
```

<AGENT> is the name of an agent and <VARIABLE> is a propositional variable. An <atomicVignette> is a collection of "sees" or "doesn't see" statements. A <vignette> is a collection of <atomicVignette>s or <namedVignette>s. An <IDENTIFIER> is a name given to a <namedVignette> for identification purposes. Finally a <scenario> is a collection of <vignette>s.
An example scenario is as follows.

EML code 1.

```
scenario [
  A sees p;
  B sees A sees p;
  B doesn't see p;
]
```

The constraints this scenario defines on a model ensure that A will know what the value of variable p is, B will know that A will know what the value of p is, while B itself remains ignorant about the value of p.

3.2 Semantics

The semantics of EML with respect to $S5_n$ are defined as follows, where $[\![\ldots]\!]$ defines a function mapping vignettes to $S5_n$ formulae:

- $[\![i \text{ sees } p]\!] = KW_i p$
- $[\![i \text{ doesn't see } p]\!] = IGN_i p$
- $[\![i \text{ sees } V]\!] = K_i [\![V]\!]$
- $[\![i \text{ doesn't see } V]\!] = \neg K_i [\![V]\!]$

Using these symantics the EML code 1 will translate to the following epistemic formulae.

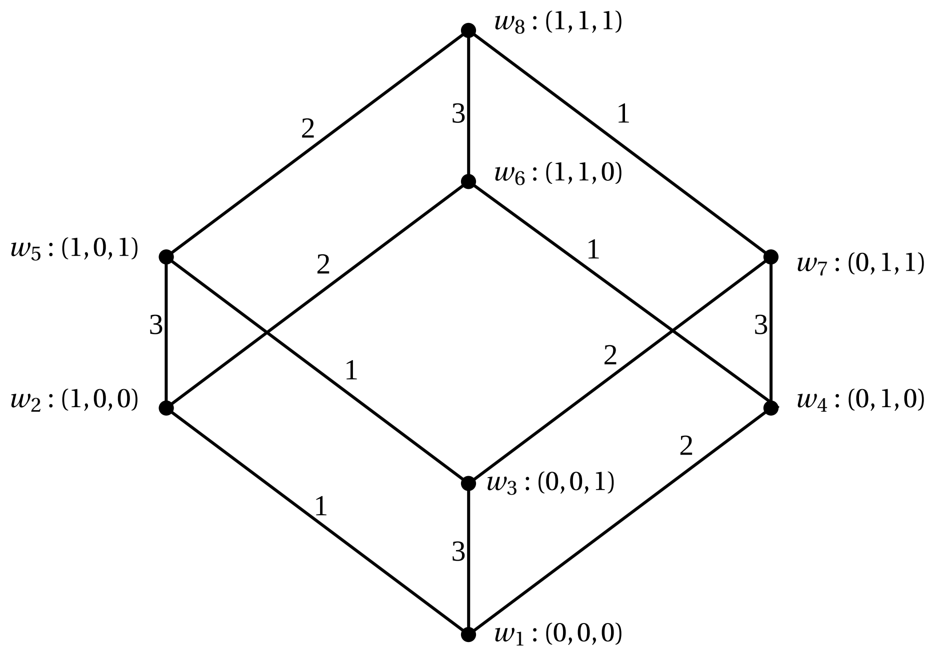

Figure 2: Muddy Children puzzle Kripke model

- $K_A p \vee K_A \neg p$
- $K_B(K_A p \vee K_A \neg p)$
- $\neg K_B p \wedge \neg K_B \neg p$

4 EML Applied

Now we will use EML to represent and reason about the Muddy Children puzzle, explained in Example 2. The Kripke model for the Muddy Children puzzle with three agents is illustrated in Figure 2. This is an S_5 model which means that the accessibility relations are *reflexive*, *symmetric* and *transitive*. To simplify the diagram, self-loops and arrowheads have been omitted. Each edge is labelled by numbers corresponding to Agent 1, 2 or 3. In this Kripke model the actual state is w_6 where Agent 1 and 2 are muddy and Agent 3 is clean. This is shown as $(1,1,0)$.

4.1 EML and the Muddy Children puzzle

Agent 1 can see that Agent 2 has a muddy forehead and Agent 3 is clean. He also knows that both Agent 2 and 3 can see his forehead while he himself does not know if it is muddy or clean. This is how EML represents this for Agent 1.

EML code 2.

```
vignette V1 [
 1 doesn't see m1;
 1 sees m2;
 1 sees m3;
```

]

The first line of the vignette V_1, `1 doesn't see` m_1, refers to Agent 1's ignorance about his forehead. Lines two and three say that Agent 1 either knows that agents 2 and 3 have muddy foreheads or knows that they do not. We use m_1 when Agent 1 is muddy and $\neg m_1$ when he is clean. Using the semantics of EML, this vignette will be translated to the following formulae.

1. $\neg K_1 m_1 \wedge \neg K_1 \neg m_1$
2. $K_1 m_2 \vee K_1 \neg m_2$
3. $K_1 m_3 \vee K_1 \neg m_3$

We now show that the properties derivable from our EML semantics in fact correspond to those that can be derived from the Kripke model. The actual world in the Kripke model of Figure 2 is w_6. Let us see what properties we can derive in this world for Agent 1. From w_6 Agent 1 accessible worlds are w_6 and w_4. While in w_6, m_1 is true, it is false in w_4. This means that Agent 1 is ignorant about m_1, hence

$$\mathsf{M}, w_6 \vDash \neg K_1 m_1 \wedge \neg K_1 \neg m_1$$

From every state of this model where m_1 holds, there is an Agent 1 accessible world where $\neg m_1$ holds and vice versa. Therefore Agent 1's ignorance about m_1 not only holds in w_6 but also in every other state which makes it *valid* in M and thus

$$\mathsf{M} \vDash \neg K_1 m_1 \wedge \neg K_1 \neg m_1$$

This shows that our EML semantics gives some properties that are *valid* in Figure 2. In the same way for Agents 2 and 3 we have

$$\mathsf{M} \vDash \neg K_2 m_2 \wedge \neg K_2 \neg m_2$$

$$\mathsf{M} \vDash \neg K_3 m_3 \wedge \neg K_3 \neg m_3$$

Hence, regardless of whether an agent is muddy or not, he does not know his own condition, which is the essence of the Muddy Children puzzle.
On the other hand in both states, w_6 and w_4, Agent 1 considers m_2 and $\neg m_3$ possible. Therefore we have:

$$\mathsf{M}, w_6 \vDash K_1 m_2$$

$$\mathsf{M}, w_6 \vDash K_1 \neg m_3$$

It can be seen that while for Agent 1 we have $\mathsf{M} \vDash K_1 m_2 \vee K_1 \neg m_2$, in w_6 where m_2 hold we have $\mathsf{M}, w_6 \vDash K_1 m_2$. Lemma 1 looks at this property in every $S5_n$ model.

Lemma 1. *For every $S5_n$ model M and for an arbitrary formula φ we have*

$$\mathsf{M} \vDash ((K\varphi \vee K\neg\varphi) \wedge \varphi) \rightarrow K\varphi$$

Proof. Take M and $s \in \mathsf{M}$ arbitrary. Suppose

$$\mathsf{M}, s \vDash (K\varphi \vee K\neg\varphi) \wedge \varphi \tag{4.1}$$

Let us assume that $\mathsf{M}, s \models \neg K\varphi$ or, equivalently, $\mathsf{M}, s \models M\neg\varphi$. Then

$$\exists t : (R(s,t) \& \mathsf{M}, t \models \neg\varphi) \tag{4.2}$$

Since R is reflexive, we have for s that $\mathsf{M}, s \models M\varphi$ also from (4.2) we have $\mathsf{M}, s \models M\neg\varphi$. Therefore $\mathsf{M}, s \models M\varphi \wedge M\neg\varphi$ or, equivalently, $\mathsf{M}, s \models \neg K\varphi \wedge \neg K\neg\varphi$. This is in contradiction with (4.1). Hence our assumption was wrong and we conclude $\mathsf{M}, s \models K\varphi$. □

EML code 2 looks at the first level of agent's knowledge. This consists of whether an agent knows the value of a proposition or is ignorant about it. The second level of agent's knowledge looks at what an agent knows about other agents' knowledge. To cover this we need to expand the EML code 2 as follows:

EML code 3.

```
vignette V1,
vignette V2 [
 2 sees m1;
 2 doesn't see m2;
 2 sees m3;
]
vignette V3 [
 3 sees m1;
 3 sees m2;
 3 doesn't see m3;
]
scenario [
 1 sees V2;
 1 sees V3;
 2 sees V1;
 2 sees V3;
 3 sees V1;
 3 sees V2;
]
```

Vignettes V_1 is repeated from EML code 2. Vignettes V_2 and V_3 capture what agents 2 and 3 know, following the same pattern as Agent 1, discussed above. The scenario represents the second level of knowledge which results in the following formulae for Agent 1.

1. $K_1(K_2m_1 \vee K_2\neg m_1)$
2. $K_1(\neg K_2m_2 \wedge \neg K_2\neg m_2)$
3. $K_1(K_2m_3 \vee K_2\neg m_3)$
4. $K_1(K_3m_1 \vee K_3\neg m_1)$
5. $K_1(K_3m_2 \vee K_3\neg m_2)$
6. $K_1(\neg K_3m_3 \wedge \neg K_3\neg m_3)$

These formulae hold in every state in M which makes them *valid*.
Let us look again at the properties derived from these formulae in w_6. For the second formula in the above list we have

$$\mathsf{M}, w_6 \models K_1(\neg K_2m_2 \wedge \neg K_2\neg m_2) \to (K_1\neg K_2m_2 \wedge K_1\neg K_2\neg m_2)$$

This can be checked in the Kripke model of Figure 2. In w_6, where m_2 is true, there is an Agent 2 accessible world, w_2, where m_2 is false therefore we have $\neg K_2 m_2$. w_6 itself is also an Agent 2 accessible world and so we also have $\neg K_2 \neg m_2$. This means that in w_6 we have $\neg K_2 m_2 \wedge \neg K_2 \neg m_2$. Similarly in w_4, m_2 is true while in w_1 which is Agent 2 accessible from w_4, m_2 is false and therefore we have $\neg K_2 m_2 \wedge \neg K_2 \neg m_2$. In both w_6 and w_4, Agent 1 accessible worlds from w_6, we have $\neg K_2 m_2 \wedge \neg K_2 \neg m_2$. This means for Agent 1 in w_6 we have $K_1(\neg K_2 m_2 \wedge \neg K_2 \neg m_2)$.
Lemma 2 generalises this for every $S5_n$ model.

Lemma 2. *For every $S5_n$ model M and for an arbitrary formulae φ and ψ we have*

$$\mathsf{M} \vDash K(\varphi \wedge \psi) \to (K\varphi \wedge K\psi)$$

Proof. Take M and $s \in \mathsf{M}$ arbitrary and suppose $\mathsf{M}, s \vDash K(\varphi \wedge \psi)$. Then

$$\forall t : (R(s,t) \Rightarrow \mathsf{M}, t \vDash \varphi \wedge \psi) \tag{4.3}$$

From (4.3) for an arbitrary t such that $R(s,t)$ we have

$$\mathsf{M}, t \vDash \varphi \text{ and } \mathsf{M}, t \vDash \psi$$

This means

$$\mathsf{M}, s \vDash K\varphi \text{ and } \mathsf{M}, s \vDash K\psi$$

Therefore

$$\mathsf{M}, s \vDash K\varphi \wedge K\psi$$

□

For the third formula of this list, $K_1(K_2 m_3 \vee K_3 \neg m_3)$, in w_6, we can conclude

$$\mathsf{M}, w_6 \vDash (K_1(K_2 m_3 \vee K_2 \neg m_3)) \to K_1 K_2 \neg m_3$$

This is because in all the Agent 2 accessible worlds from w_6, which are w_6 and w_2, $\neg m_3$ holds. This is the same in w_4 where in both Agent 2 accessible worlds from there, w_4 and w_1, $\neg m_3$ holds again. This means that in all Agent 1 accessible worlds from w_6 we have $K_2 \neg m_3$ hence in w_6 we have $K_1 K_2 \neg m_3$. The following lemma looks at this property for every $S5_n$ model.

Lemma 3. *For every $S5_n$ model M and for an arbitrary formula φ and arbitrary agents i and j we have*

$$\mathsf{M} \vDash (K_i(K_j\varphi \vee K_j \neg\varphi) \wedge (K_i\varphi \vee K_i \neg\varphi) \wedge \varphi \to K_i K_j \varphi$$

Proof. Take M and $s \in \mathsf{M}$ arbitrary and suppose

$$\mathsf{M}, s \vDash (K_i(K_j\varphi \vee K_j \neg\varphi) \wedge (K_i\varphi \vee K_i \neg\varphi) \wedge \varphi)$$

From $\mathsf{M}, s \vDash K_i(K_j\varphi \wedge K_j \neg\varphi)$ we infer that

$$\forall t : (R_i(s,t) \Rightarrow \mathsf{M}, t \vDash K_j\varphi \wedge K_j \neg\varphi) \tag{4.4}$$

From $\mathsf{M}, s \vDash (K_i\varphi \vee K_i \neg\varphi) \wedge \varphi)$ and using Lemma 1 we infer that

$$\forall t : (R_i(s,t) \Rightarrow \mathsf{M}, t \vDash \varphi) \tag{4.5}$$

Let us assume that $\mathsf{M}, s \vDash \neg K_i K_j \varphi$ or, equivalently $\mathsf{M}, s \vDash M_i M_j \neg\varphi$. Then

$$\exists t', \exists u' : (R_i(s,t') \;\&\; R_j(t',u') \text{ and } \mathsf{M}, u' \vDash \neg\varphi) \tag{4.6}$$

From (4.5) and the fact that R_j is reflexive we have for this t' that $\mathsf{M}, t' \vDash M_j \varphi$ and from (4.6) we have $\mathsf{M}, t' \vDash M_j \neg\varphi$. Therefore $\mathsf{M}, t' \vDash M_j \varphi \wedge M_j \neg\varphi$ or $\mathsf{M}, t' \vDash \neg K_j \varphi \wedge \neg K_j \neg\varphi$. This is in contradiction with (4.4) hence our assumption was wrong and we conclude $\mathsf{M}, s \vDash K_i K_j \varphi$ □

4.2 Extended Muddy Children puzzle

The previous section looked at the first level of agents' knowledge. This is what each agent knows about his own forehead and about others' foreheads. It reasoned about formulae such as $K_1 m_2 \vee K_1 \neg m_2$, where Agent 1 knows m_2 or knows $\neg m_2$. It also looked at the second level of agents' knowledge. This is what each agent knows about what other agents know. The second level of agents' knowledge is captured in formulae such as $K_1(K_2 m_1 \vee K_2 \neg m_1)$ where Agent 1 knows that Agent 2 knows m_1 or knows $\neg m_1$.

This section expands agents' knowledge to the third level. It will look at formulae such as $K_1 K_2 (K_3 m_1 \vee K_3 \neg m_1)$, where Agent 1 knows that Agent 2 knows that Agent 3 knows m_1 or knows $\neg m_1$. To capture these situations the following EML code is used.

EML code 4.

```
vignette V1,
vignette V2,
vignette V3,
vignette V10 [
 1 sees V2;
 1 sees V3;
]
vignette V20 [
 2 sees V1;
 2 sees V3;
]
vignette V30 [
 3 sees V1;
 3 sees V2;
]
scenario [
 1 sees V20;
 1 sese V30;
 2 sees V10;
 2 sees V30;
 3 sees V10;
 3 sees V20;
]
```

Looking at the EML code 4 for Agent 1, it can be seen that vignette V_1 is repeated from EML code 2. This is because we are still covering the first and the second level of agents' knowledge. Also vignette V_{10} expresses the same formulae which has been covered in the scenario of EML code 3. However in the scenario of EML code 4, for Agent 1, we have these formulae:

1. $K_1K_2(\neg K_1 m_1 \wedge \neg K_1 \neg m_1)$
2. $K_1K_2(K_1 m_2 \vee K_1 \neg m_2)$
3. $K_1K_2(K_1 m_3 \vee K_1 \neg m_3)$
4. $K_1K_2(K_3 m_1 \vee K_3 \neg m_1)$
5. $K_1K_2(K_3 m_2 \vee K_3 \neg m_2)$
6. $K_1K_2(\neg K_3 m_3 \wedge \neg K_3 \neg m_3)$
7. $K_1K_3(\neg K_1 m_1 \wedge \neg K_1 \neg m_1)$
8. $K_1K_3(K_1 m_2 \vee K_1 \neg m_2)$
9. $K_1K_3(K_1 m_3 \vee K_1 \neg m_3)$
10. $K_1K_3(K_2 m_1 \vee K_2 \neg m_1)$
11. $K_1K_3(\neg K_2 m_2 \wedge \neg K_2 \neg m_2)$
12. $K_1K_3(K_2 m_3 \vee K_2 \neg m_3)$

In w_6, the actual state, certain properties can be derived for Agent 1 from the above formulae. Let us look at number 6 of the above list and what can be derived from this in w_6.

$$\mathsf{M}, w_6 \models (K_1K_2(\neg K_3 m_3 \wedge \neg K_3 \neg m_3)) \rightarrow (K_1K_2 \neg K_3 m_3 \wedge K_1K_2 \neg K_3 \neg m_3)$$

Looking at the Kripke model in Figure 2 it can be seen that in w_6 there are two Agent 1 accessible worlds which are w_6 itself and w_4. From w_6, Agent 2 accessible worlds are w_6 and w_2 and from w_4 they are w_4 and w_1. In w_6, where m_3 is false, w_8 is an Agent 3 accessible world where m_3 is true. Therefore in w_6 we have $\neg K_3 m_3 \wedge \neg K_3 \neg m_3$. Similarly in w_2, where m_3 is false, w_5 is an Agent 3 accessible world where w_3 is true. For w_4, m_3 is false and w_7 is an Agent 3 accessible world where m_3 is true. Finally m_3 is false in w_1 but true in w_3, Agent 3 accessible from w_1. In all the worlds mentioned for Agent 3 we have $\neg K_3 m_3 \wedge \neg K_3 \neg m_3$ which means that for Agent 1 in w_6 we have $K_1K_2 \neg K_3 m_3 \wedge K_1K_2 \neg K_3 \neg m_3$.
This also applies to formulae number 1, 7 and 11 of the list. The following Lemma looks at this for every $S5_n$ model.

Lemma 4. *For every $S5_n$ model* M *and for arbitrary formulae φ and ψ and arbitrary agents i and j we have*

$$\mathsf{M} \models K_iK_j(\varphi \wedge \psi) \rightarrow (K_iK_j\varphi \wedge K_iK_j\psi)$$

Proof. Take M and $s \in \mathsf{M}$ arbitrary. Suppose $\mathsf{M}, s \models K_iK_j(\varphi \wedge \psi)$. Then

$$\forall t\ (R_i(s,t) \Rightarrow \mathsf{M}, t \models K_j(\varphi \wedge \psi)) \tag{4.7}$$

Using Lemma 2, we then get

$$\forall t (R_i(s,t) \Rightarrow \mathsf{M}, t \models (K_j\varphi \wedge K_j\psi)) \tag{4.8}$$

But using the truth-defintion of K_i, we see that (4.8) is equivalent to

$$\mathsf{M}, s \models K_i(K_j\varphi \wedge K_j\psi) \tag{4.9}$$

Applying Lemma 2 now to (4.9) gives the desired conclusion, i.e. $\mathsf{M}, s \models K_iK_j\varphi \wedge K_iK_j\psi$. □

Another property that can be derived for Agent 1 in w_6 is

$$\mathsf{M}, w_6 \models (K_1 K_2 (K_1 m_3 \lor K_1 \neg m_3)) \to K_1 K_2 K_1 \neg m_3$$

This can be checked in the Kripke model of Figure 2. In w_6, there are two Agent 1 accessible worlds which are w_6 itself and w_4. From w_6, Agent 2 accessible worlds are w_6 and w_2 and from w_4 they are w_4 and w_1. In w_6 for both Agent 1 accessible worlds which are w_6 and w_4, $\neg m_3$ holds. From w_2, w_1 is an Agent 1 accessible world where $\neg m_3$ holds. It is the same in w_4, where w_6 is Agent 1 accessible and w_1 where w_2 is Agent 1 accessible and in all, $\neg m_3$ holds. This means that in w_6 we have $K_1 K_2 K_1 \neg m_3$.
This leads us to the following Lemma.

Lemma 5. *For every $S5_n$ model M and for arbitrary agents i and j and arbitrary formula φ we have*

$$\mathsf{M} \models (K_i K_j (K_i \varphi \lor K_i \neg\varphi) \land K_i (K_j \varphi \lor K_j \neg\varphi) \land \varphi) \to K_i K_j K_i \varphi$$

Proof. Take M and $s \in \mathsf{M}$ arbitrary and suppose

$$\mathsf{M}, s \models K_i K_j (K_i \varphi \lor K_i \neg\varphi) \land K_i (K_j \varphi \lor K_j \neg\varphi) \land \varphi$$

From $\mathsf{M}, s \models K_i K_j (K_i \varphi \lor K_i \neg\varphi)$, using the truth definition, we infer that

$$\forall t, \forall u : (R_i(s,t) \;\&\; R_j(t,u) \Rightarrow \mathsf{M}, u \models K_i \varphi \lor K_i \neg\varphi) \tag{4.10}$$

Also, from $\mathsf{M}, s \models K_i K_j (K_i \varphi \lor K_i \neg\varphi)$, we derive (using the truth axiom twice) that $\mathsf{M}, s \models K_i \varphi \lor K_i \neg\varphi$. We can combine this with $\mathsf{M}, s \models K_i (K_j \varphi \lor K_j \neg\varphi) \land \varphi$, together with Lemma 3 to conclude $\mathsf{M}, s \models K_i K_j \varphi$. The latter means

$$\forall t, \forall u : (R_i(s,t) \;\&\; R_j(t,u) \Rightarrow \mathsf{M}, u \models \varphi) \tag{4.11}$$

Let us assume that $\mathsf{M}, s \models \neg K_i K_j K_i \varphi$ or, equivalently, $\mathsf{M}, s \models M_i M_j M_i \neg\varphi$. Then

$$\exists t' \exists u' \exists v' : (R_i(s,t') \;\&\; R_j(t',u') \;\&\; R_i(u',v') \text{ and } \mathsf{M}, v' \models \neg\varphi) \tag{4.12}$$

From (4.11) and the fact that R_i is reflexive, we have for this u' that $\mathsf{M}, u' \models M_i \varphi$, and from (4.12) we have $\mathsf{M}, u' \models M_i \neg\varphi$. Therefore $\mathsf{M}, u' \models M_i \varphi \land M_i \neg\varphi$ or $\mathsf{M}, u' \models \neg K_i \varphi \land \neg K_i \neg\varphi$. This is in contradiction with (4.10). Hence our assumption was wrong and we conclude $M, s \models K_i K_j K_i \varphi$ □

We have seen how a set of constraints on a Kripke model, expressed by EML, can be used to represent an epistemic scenario. We have also used EML to reason about $S5_n$ systems and have demonstrated that EML is expressive enough for this purpose.

5 An application for EML

It has been shown above that EML defines a set of constraints on a Kripke model. An application for EML would be to create a tool which takes as input a Kripke model and a set of constraints in EML and checks whether the model satisfies these constraints. The Kripke model should be defined by a set of propositional constants, Φ, a set of states, W, a truth assignment function assigning propositional constants to each state, π and $R_1, \ldots, R_n$ accessibility relations for agents 1

to n. The application will then check whether the Kripke model satisfies the constraints set by the EML code.
For example a Kripke model such as the following:

- $\Phi = \{p\}$
- $W = \{s, t\}$
- $\pi(s)(p) = \text{true}$
- $\pi(t)(p) = \text{false}$
- $R_A = \{(s, s), (t, t)\}$
- $R_B = \{(s, s), (t, t), (s, t), (t, s)\}$

with the EML code 1, will result in the application confirming that the code satisfies the Kripke model. This is because the epistemic formulae that EML code 1 represents is valid in the Kripke model above.

6 Conclusion

We have established the difficulties of using a Kripke model to analyse an epistemic situation. As a solution to this we introduced a high level modelling language, EML, that uses a collection of "sees" and "doesn't see" statements to define a collection of constraints on an epistemic Kripke model. The Muddy Children puzzle was an example of an epistemic scenario. We used EML to represent and reason about this scenario. We then extended our EML code to represent deeper levels of agents' knowledge in this particular example. We demonstrated that the properties derived from our EML code correspond to those derived from the Kripke model. We have proved that these properties hold in every $S5_n$ model thus demonstrating that EML is expressive enough to represent and reason about epistemic scenarios.

References

[1] Chellas, B. *Modal Logic: An Introduction.* Cambridge University Press: Cambridge, England, 1993.

[2] Fagin, R.,Halpern, J. Y., Moses, Y., and Vardi, M. Y. *Reasoning About Knowledge.* The MIT press: Cambridge, MA, 1995.

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

[4] Meyer, J.-J. C. and van der Hoek, W. *Epistemic Logic for AI and Computer Science.* Cambridge University Press: Cambridge, England, 1995.

[5] Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, 1995.

[6] Shoham, Y. and Leyton-Brown, K. *Multiagent Systems: algorithmic, game-theoretic, and logical foundations.* Cambridge University Press: Cambridge, England, 2009.

[7] Wooldridge, M. *The Logical Modelling of Computational Multi-Agent Systems*. PhD thesis, Department of Computation, UMIST, Manchester UK, 1992.

[8] Wooldridge, M. *An Introduction to multiagent systems*. John Wiley and Sons Ltd, 2nd Edition, 2009.