Melvin Fitting

How True It Is = Who Says It's True

Abstract. This is a largely expository paper in which the following simple idea is pursued. Take the truth value of a formula to be the set of agents that accept the formula as true. This means we work with an arbitrary (finite) Boolean algebra as the truth value space. When this is properly formalized, complete modal tableau systems exist, and there are natural versions of bisimulations that behave well from an algebraic point of view. There remain significant problems concerning the proper formalization, in this context, of natural language statements, particularly those involving negative knowledge and common knowledge. A case study is presented which brings these problems to the fore. None of the basic material presented here is new to this paper—all has appeared in several papers over many years, by the present author and by others. Much of the development in the literature is more general than here—we have confined things to the Boolean case for simplicity and clarity. Most proofs are omitted, but several of the examples are new. The main virtue of the present paper is its coherent presentation of a systematic point of view—identify the truth value of a formula with the set of those who say the formula is true.

Keywords: modal logic, many-valued logic, logic of knowledge, boolean algebra, bisimulation, tableau.

1. Introduction

A wide range of many valued logics are in the literature, but one family is notably missing: those whose truth value space is a Boolean algebra other than {*false*, *true*}. There is a good reason for this, embodied in the following classic theorem due to Stone. *If $\mathcal{B}$ is a Boolean algebra and b is a member other than the top element, there is a homomorphism from $\mathcal{B}$ to the two-element Boolean algebra {false, true}, mapping b to false.* (Proposition 2.19 in Volume 1 of [19] is a particularly simple version of this.) An easy consequence is that if we have a formula, constructed using $\wedge$, $\vee$, $\neg$, say, where the semantics for these connectives is given by the analogous operations in some Boolean algebra $\mathcal{B}$, and the formula is not true in some $\mathcal{B}$ valued model, it is falsified in some standard two valued model. This is the basis for the completeness proof for classical propositional logic using Lindenbaum algebras.

Special Issue: Truth Values. Part I
Edited by **Yaroslav Shramko and Heinrich Wansing**; *Received* August 21, 2008

DOI: 10.1007/s11225-009-9178-1

It is also the supporting idea behind Boolean valued forcing in set theory. It would seem that allowing arbitrary Boolean algebras as truth-value spaces does not provide anything new. It is the job of this paper to show that this is a narrow and misleading point of view.

The present paper is really a partial summation of work by the author and others over several years. It makes a case for the utility of finite Boolean algebras, which we can think of as representing situations involving multiple agents. Areas in need of further research will be described. In particular a case study, organized around the well-known 'muddy children' puzzle, will reveal many of the open problems of natural language interpretation in the Boolean valued context. It will be seen that the title of this paper is not just metaphor, but is a correct summary of the content of the paper.

2. Propositional Logic

Suppose we have a finite set $\mathcal{A} = \{a_1, \ldots, a_n\}$ of agents. The power set $\mathcal{P}(\mathcal{A})$ of this set is a finite Boolean algebra, using intersection, union, and complementation as the operations and subset as the ordering. (Every finite Boolean algebra is isomorphic to such a set algebra, so there is no loss of generality if we always work with sets of agents.) We can take $\mathcal{P}(\mathcal{A})$ as a space of truth values. There is a natural intuition behind this. For a given propositional letter P, some agents may take P to be true, some to be false. It is reasonable to think of the set of agents who accept P as *being* the truth value of P, at least under some circumstances. Here are some simple examples to show how this can facilitate the natural expression of things.

Suppose we have three agents, say 1, 2, and 3, and we want to express that some of them are tall and some are not. Conventionally we might introduce three propositional letters, say T_1, T_2, and T_3 with the intention that T_i expresses that i is tall. Then if we want to say that agents 1 and 2 are tall but 3 is not, we can do this by saying $T_1 \wedge T_2 \wedge \neg T_3$ has the truth value *true* in the standard two-element Boolean algebra. Instead, in our setting we could have a *single* propositional letter, T, which we think of as representing "is tall". Further we give it, and other formulas, truth values in the space $\mathcal{P}(\{1, 2, 3\})$. Then we assert that 1 and 2 are tall but 3 is not by saying the truth value of T is $\{1, 2\}$. This is a first pass at the theme of the paper, and will be elaborated as we continue.

The muddy children puzzle is widely familiar—we describe it briefly for those who have not encountered it. Several children sit in a circle. All are perfect reasoners. Each can see the forehead of the other children, but not their own. A parent puts mud on some of the foreheads of the children (at least one), so that each child does not know the status of their own forehead. The parent announces, "at least one of you has a muddy forehead." Then the parent asks, "does anyone know that their forehead is muddy?" Presumably nobody answers. This is asked again, and again. Eventually those with muddy foreheads answer yes. If k children have muddy foreheads, answers come at round k of questioning. The problem is to account for, and formalize, the reasoning behind the answers of the children.

We examine the muddy children problem in more detail in Section 7. For the moment, let us assume there are three children, and each has a muddy forehead. A standard formalization begins as follows. Say the three children are a, b, and c. Let A stand for "a has a muddy forehead." Let B stand for "b has a muddy forehead." And similarly let C stand for "c has a muddy forehead." Then the initial situation is formalized by $A \wedge B \wedge C$.

As an alternative in the style of the present paper, suppose we use the Boolean algebra $\mathcal{P}(\{a, b, c\})$ as our space of truth values. Let M be a propositional letter whose intended meaning is "has a muddy forehead." Then the initial setup—recall we assume each child has a muddy forehead—amounts to giving M the truth value $\{a, b, c\}$.

What is common to these and other examples is that we have a finite Boolean algebra, $\mathcal{P}(\mathcal{A})$, of sets of agents, and truth values for propositional letters come from this algebra. In the usual way we form more complex formulas from propositional letters, with connectives $\wedge$, $\vee$, $\neg$, $\supset$, say. A Boolean valued propositional semantics is then straightforward. Suppose v is a mapping from propositional letters to $\mathcal{P}(\mathcal{A})$ (call this a $\mathcal{P}(\mathcal{A})$ *valuation*). We extend v as follows.

$$
\begin{aligned}
v(X \wedge Y) &= v(X) \cap v(Y) \\
v(X \vee Y) &= v(X) \cup v(Y) \\
v(\neg X) &= \overline{v(X)} \\
v(X \supset Y) &= v(X) \Rightarrow v(Y) \\
&= \overline{v(X)} \cup v(Y)
\end{aligned}
\tag{1}
$$

So far we have a mild alternative to the two valued approach—not much has been added. Indeed, the following is easily shown: for any non-empty set of agents $\mathcal{A}$, a formula X is a tautology if and only if $v(X) = \mathcal{A}$ for every $\mathcal{P}(\mathcal{A})$ valuation (where $\mathcal{A}$ contains at least one agent).

3. Modal Logic

Suppose we enlarge the language by adding a necessity symbol, $\Box$. (A possibility symbol, $\Diamond$, can be defined in the usual dual way, if needed.) We can extend the usual notion of Kripke model to a Boolean valued one—formulas take on values in $\mathcal{P}(\mathcal{A})$ at possible worlds. *And we can also allow the accessibility relation to be Boolean valued.* Here is the construction.

DEFINITION 3.1. Let $\mathcal{A}$ be a finite set of agents. A $\mathcal{P}(\mathcal{A})$ valued modal *frame* is a structure $\mathcal{F} = \langle \mathcal{G}, \mathcal{R} \rangle$ meeting the following conditions.

1. $\mathcal{G}$ is a non-empty set (of *states*, or *possible worlds*);
2. $\mathcal{R}$ is a *Boolean valued accessibility relation*, mapping pairs of states to truth values,
$$\mathcal{R} : \mathcal{G} \times \mathcal{G} \longrightarrow \mathcal{P}(\mathcal{A})$$

A $\mathcal{P}(\mathcal{A})$ valued modal *model* is a structure $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$ where $\langle \mathcal{G}, \mathcal{R} \rangle$ is a $\mathcal{P}(\mathcal{A})$ valued frame and also:

3. v maps atoms, at states, to truth values,
$$v : \mathcal{G} \times \text{propositional letters} \longrightarrow \mathcal{P}(\mathcal{A})$$

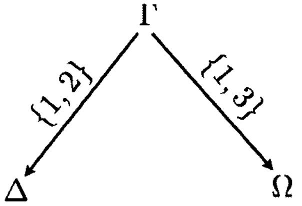
Given a $\mathcal{P}(\mathcal{A})$ valued model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$, the behavior of v is extended to all formulas as follows. For each $\Gamma \in \mathcal{G}$:

$$\begin{aligned} v(\Gamma, X \wedge Y) &= v(\Gamma, X) \cap v(\Gamma, Y) \\ v(\Gamma, X \vee Y) &= v(\Gamma, X) \cup v(\Gamma, Y) \\ v(\Gamma, \neg X) &= \overline{v(\Gamma, X)} \\ v(\Gamma, X \supset Y) &= v(\Gamma, X) \Rightarrow v(\Gamma, Y) \\ &= \overline{v(\Gamma, X)} \cup v(\Gamma, Y) \end{aligned} \tag{2}$$

$$\begin{aligned} v(\Gamma, \Box X) &= \bigcap_{\Delta \in \mathcal{G}} [\mathcal{R}(\Gamma, \Delta) \Rightarrow v(\Delta, X)] \\ &= \bigcap_{\Delta \in \mathcal{G}} [\overline{\mathcal{R}(\Gamma, \Delta)} \cup v(\Delta, X)] \end{aligned} \tag{3}$$

We will not make use of frames until Section 8. The conditions in (2) are the counterparts of (1), relativized to states. If there is only one agent, a and *true* is identified with $\{a\}$, and *false* with $\emptyset$, then (3) becomes the familiar way of evaluating $\Box X$ at worlds of Kripke models.

Here is a simple example, displayed in Figure 1. Suppose we have three agents, $\mathcal{A} = \{1, 2, 3\}$. Let $\mathcal{G} = \{\Gamma, \Delta, \Omega\}$, and let accessibility be as shown. Thus, for instance, $\mathcal{R}(\Gamma, \Delta) = \{1, 2\}$, which we could read as "Δ is accessible from Γ for agents 1 and 2." Arrows not explicitly shown have truth values of $\emptyset$. Also assume A is a propositional letter—behavior of v on A is also displayed in Figure 1.



$$v(\Gamma, A) = \emptyset$$
$$v(\Delta, A) = \{1, 3\}$$
$$v(\Omega, A) = \{2, 3\}$$

Figure 1. Three Agent Example

Then, with respect to the $\mathcal{P}(\mathcal{A})$ valued model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$ we have the following calculation.

$$\begin{aligned} v(\Gamma, \Box A) &= [\overline{\mathcal{R}(\Gamma, \Gamma)} \cup v(\Gamma, A)] \cap [\overline{\mathcal{R}(\Gamma, \Delta)} \cup v(\Delta, A)] \cap [\overline{\mathcal{R}(\Gamma, \Omega)} \cup v(\Omega, A)] \\ &= [\{1, 2, 3\} \cup \emptyset] \cap [\{3\} \cup \{1, 3\}] \cap [\{2\} \cup \{2, 3\}] \\ &= \{1, 2, 3\} \cap \{1, 3\} \cap \{2, 3\} \\ &= \{3\} \end{aligned}$$

Here is another example of a calculation, this time for *any* set of agents, and *any* model. In the single agent case it amounts to the usual way that $\Diamond X$ is evaluated in Kripke models.

$$\begin{aligned} v(\Gamma, \Diamond X) &= v(\Gamma, \neg\Box\neg X) \\ &= \overline{v(\Gamma, \Box\neg X)} \\ &= \overline{\bigcap_{\Delta \in \mathcal{G}} [\overline{\mathcal{R}(\Gamma, \Delta)} \cup v(\Delta, \neg X)]} \\ &= \bigcup_{\Delta \in \mathcal{G}} [\mathcal{R}(\Gamma, \Delta) \cap \overline{v(\Delta, \neg X)}] \\ &= \bigcup_{\Delta \in \mathcal{G}} [\mathcal{R}(\Gamma, \Delta) \cap v(\Delta, X)] \end{aligned}$$

The models introduced in this section originate in [4, 5], in a more general version using *Heyting algebras*, which include Boolean algebras as a special case. This will be discussed further in Section 9.

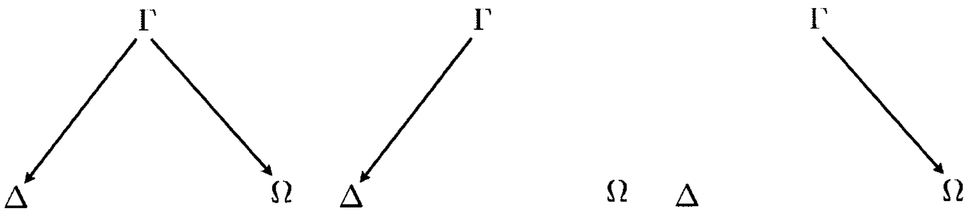
4. Agent Slices

Given a $\mathcal{P}(\mathcal{A})$ valued model, we can ask how things in the model "look" to each of the agents in $\mathcal{A}$, individually. We call this an agent slice.

DEFINITION 4.1. Let $c \in \mathcal{A}$ be an agent.

1. For a $\mathcal{P}(\mathcal{A})$ valued modal frame $\mathcal{F} = \langle \mathcal{G}, \mathcal{R} \rangle$, by the *c slice* of $\mathcal{F}$ we mean the conventional frame $\mathcal{F}_c = \langle \mathcal{G}, \mathcal{R}_c \rangle$ defined as follows. The set of possible worlds, $\mathcal{G}$, is the same as in $\mathcal{F}$. For worlds Γ and Δ, we set $\Gamma \mathcal{R}_c \Delta$ to be true if and only if $c \in \mathcal{R}(\Gamma, \Delta)$.
2. For a $\mathcal{P}(\mathcal{A})$ valued modal model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$, by the *c slice* of $\mathcal{M}$ we mean the conventional Kripke model $\mathcal{M}_c = \langle \mathcal{G}, \mathcal{R}_c, v_c \rangle$ where the frame $\langle \mathcal{G}, \mathcal{R}_c \rangle$ is the c slice of the $\mathcal{P}(\mathcal{A})$ valued frame $\langle \mathcal{G}, \mathcal{R} \rangle$, and for a propositional letter P, we set $v_c(P)$ to be true if $c \in v(\Gamma, P)$, and otherwise we set it to be false.

In the usual way, (conventional) truth at a world in a slice model is defined. We symbolize it by $\mathcal{M}_c, \Gamma \Vdash X$; formula X is true at world Γ of (standard) Kripke model $\mathcal{M}_c$.



$v_1(\Gamma, A) = \textit{false}$ $v_2(\Gamma, A) = \textit{false}$ $v_3(\Gamma, A) = \textit{false}$

$v_1(\Delta, A) = \textit{true}$ $v_2(\Delta, A) = \textit{false}$ $v_3(\Delta, A) = \textit{true}$

$v_1(\Omega, A) = \textit{false}$ $v_2(\Omega, A) = \textit{true}$ $v_3(\Omega, A) = \textit{true}$

Figure 2. Three Agent Slices

We continue with the example begun in Figure 1. Since there are three agents there are three slices, given schematically in Figure 2. Note that in the three agent slices we have $\mathcal{M}_1, \Gamma \nVdash \Box A$ and $\mathcal{M}_2, \Gamma \nVdash \Box A$, but $\mathcal{M}_3, \Gamma \Vdash \Box A$. This accords with our earlier calculation that in the $\mathcal{P}(\mathcal{A})$ valued model from Figure 1, $v(\Gamma, \Box A) = \{3\}$. In fact, this behavior is quite generally the case. A Boolean valued modal model summarizes the behavior of all its slices.

THEOREM 4.2. *Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$ be a $\mathcal{P}(\mathcal{A})$ valued modal model. For each formula X, and for each $\Gamma \in \mathcal{G}$,*

$$v(\Gamma, X) = \{c \in \mathcal{A} \mid \mathcal{M}_c, \Gamma \Vdash X\}.$$

We omit the proof of this theorem—it is a straightforward induction on formula degree. But what it says, quite literally, is that in one of our Boolean valued models, at each state, the truth value of a formula is the set of agents who accept the formula as true.

5. Extending Expressivity

Using the machinery so far, we can name the empty set of agents, using $P \wedge \neg P$, and the entire set of agents, using $P \vee \neg P$, but nothing in between. We now add straightforward machinery to do more. From now on assume that for each set of agents there is some *propositional constant* that 'names' it. These propositional constants play a central role in the tableau system presented in Section 6. We use the obvious notation. For agents a, b, c, say, take $\{a, b, c\}$ as a propositional letter of the formal language, and similarly for $\emptyset$, $\mathcal{A}$ (the set of all agents), and so on. Of course syntactically these have structure, so to be proper perhaps what we should do is assign some conventional propositional letter to each agent set, and keep track of which is assigned to what, but a certain amount of syntactic sugar makes things more readable here.

Semantically, how to treat these new propositional constants is obvious. In the $\mathcal{P}(\mathcal{A})$ valued model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$, for any $\Gamma \in \mathcal{G}$, we take $v(\Gamma, \{a, b, c\})$ to be the member $\{a, b, c\}$ of $\mathcal{P}(\mathcal{A})$. Likewise $v(\Gamma, \emptyset) = \emptyset$, $v(\Gamma, \mathcal{A}) = \mathcal{A}$, and so on. Of course we are using expressions in two quite different ways, but context will keep things sorted out. The essential idea is that we have a variety of propositional constants, one for each set of agents, and we use the set itself (or more formally, a string of symbols that designates it in standard notation) as a name for the set.

It is almost a triviality that in any $\mathcal{P}(\mathcal{A})$ valued model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$, for each $\Gamma \in \mathcal{G}$, $v(A \supset B) = \mathcal{A}$ if and only if $v(A) \subseteq v(B)$. This means we can use propositional constants and implication to 'bound' truth values. For any propositional constant t, asserting $t \supset X$ in effect says the truth value of X must be at least t. Likewise asserting $X \supset t$ says the truth value of X is at most t.

DEFINITION 5.1. Let t be a propositional constant—a member of $\mathcal{P}(\mathcal{A})$. We call a formula of the form, $t \supset X$ or $X \supset t$ a *bounding formula.*

6. Tableaus

Let $\mathcal{A}$ be a non-empty, finite set of agents, fixed for this section. We present a proof system corresponding to the $\mathcal{P}(\mathcal{A})$ valued modal semantics of Section 3. It is tableau based, and is drawn from [7], which in turn derives from [5]. The basic building block is the *bounding formula*, Definition 5.1. All formulas appearing in tableaus will be of this form. We also make use of *signs*, T and F, in the familiar Smullyan style, [8, 21]. If X is a formula then $T\,X$ and $F\,X$ are *signed formulas.* (Notice that signs cannot be iterated.) The signed formula $T\,X$ should be thought of intuitively as asserting the truth of X (at some particular world of some particular model), and $F\,X$ as asserting falsehood of X. Then, combining signs with bounding formulas, $T\,(t \supset X)$ can be thought of intuitively as asserting that the truth value of X is at least t and $F\,(t \supset X)$ as asserting that the truth value of X is not at least t. Similarly for $T\,(X \supset t)$ and $F\,(X \supset t)$. This machinery provides considerable expressive power.

A *tableau* is a tree with nodes labeled with signed bounding formulas, constructed according to rules which are given below. A tableau is *for* a particular signed bounding formula if that item appears at the root. A branch of a tableau is called *closed* if it contains a syntactic contradiction, and the specific sets of contradictions are also specified below. A tableau itself is *closed* if each branch is closed. In the usual way we can think of a tableau as the disjunction of its branches, and a branch as the conjunction of what appears on it. Then a closed tableau represents an impossible situation.

To prove a formula X in this system there are two equivalent approaches. First, one could construct a closed tableau for the signed bounding formula $F\,(\mathcal{A} \supset X)$. Intuitively such a closed tableau tells us that it is impossible for the truth value of X not to be above $\mathcal{A}$, which tells us that X must have the value $\mathcal{A}$ at every world of every $\mathcal{P}(\mathcal{A})$ valued model. Equivalently we could construct a family of closed tableaus, one for $F\,\{a\} \supset X$ for each agent $a \in \mathcal{A}$.

Now we turn to the details, which are spread out over several subsections. In Section 6.1 we give rules for the propositional connectives only, and we do this in a schematic way, for an arbitrary set $\mathcal{A}$ of agents. In Section 6.2 we give a concrete example, using a specific set of agents. Then in Section 6.3 we give the modal rules, again for an arbitrary set of agents, followed by the concrete example extended to the modal case in Section 6.4. A lengthy example of a tableau is given in Section 7.

6.1. Propositional Rules—General Version

Recall that $\mathcal{A}$ is a finite set of agents. We give the closure rules for tableaus, and the rules for the propositional connectives, assuming $\mathcal{P}(\mathcal{A})$ is the intended space of truth values.

Branch Closure Conditions In this, x and y are arbitrary propositional constants, and X is an arbitrary formula. A tableau branch is *closed* if it contains:

$T\, x \supset y$ where $x \not\subseteq y$	
$F\, x \supset y$ where $x \subseteq y$ and $x \neq \emptyset, y \neq \mathcal{A}$	
$F\, \emptyset \supset X$	
$F\, X \supset \mathcal{A}$	
$T\, y \supset X$ $F\, x \supset X$	where $x \subseteq y$ and $x \neq \emptyset$

Remark: some restrictions on the closure conditions are simply because these cases are subsumed under other closure rules.

Notice that each of the five closure conditions above represents an intuitive impossibility. For instance, suppose we have both $T\, y \supset X$ and $F\, x \supset X$, where $x \subseteq y$. By our intuitive reading of the first item, at some world Γ of some model we have $y \subseteq v(\Gamma, X)$. Since we have $x \subseteq y$ we would also have $x \subseteq v(\Gamma, X)$, and this contradicts the intuitive reading of the second item, $F\, x \supset X$. This informal understanding of the conditions becomes the basis of a proper soundness proof, which we begin to formalize now.

DEFINITION 6.1.

1. We say a formula X is *true* at world Γ of $\mathcal{P}(\mathcal{A})$ valued model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$ if $v(\Gamma, X) = \mathcal{A}$. Otherwise X is *not true* at Γ.

2. A formula X is *valid* if it is true at every world of every $\mathcal{P}(\mathcal{A})$ valued model.

3. A signed bounding formula $T\, B$ is *true* at a world if B is true there. $F\, B$ is true at a world if B is not true there.

4. A set S of signed bounding formulas is *satisfiable* if there is some $\mathcal{P}(\mathcal{A})$ valued model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$ and some $\Gamma \in \mathcal{G}$ such that every member of S is true at Γ.

5. A tableau branch is *satisfiable* if the set of signed formulas on it is satisfiable.

6. A tableau is *satisfiable* if some branch is satisfiable.

Note that for a bounding formula $t \supset X$, part 1 of the definition above tells us that it is true at Γ in model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$ exactly when $t \subseteq v(\Gamma, X)$. Similarly for bounding formulas $X \supset t$. Also part 2 easily yields that X is valid just in case $\mathcal{A} \supset X$ is valid, just in case $\{a\} \supset X$ is valid for every $a \in \mathcal{A}$.

Given this definition, and the informal discussion following the Branch Closure Conditions, it is simple to verify the following.

LEMMA 6.2. *A closed tableau is not satisfiable.*

Next we give the *Branch Extension* rules. These say that if one has a tableau, one can extend branches to produce a new tableau. We allow more than binary branching. We begin with what we call *Reversal Rules.* These allow us to replace bounding formulas with an F sign by others with a T sign, with the bound reversed, and also the other way around. For instance, if we know that the truth value of X is not above some particular member of $\mathcal{P}(\mathcal{A})$, we can conclude it must be below certain other members, depending on circumstances. The practical effect of these rules is that they allow us to cut the number of rules for connectives and modal operators in half.

The Reversal Rules, and all subsequent rules, should be read as follows. If the signed formula above the line appears on a tableau branch, the branch can be extended. If there is only one signed formula below the line, that is added to the branch end. If n signed formulas are below the line, the last node of the branch becomes a branch point with n immediate successors, and each signed formula below the line is used as a label on one of the new nodes.

Reversal Rules In these rules, X is restricted to be any formula other than a propositional constant. Also, x, t_1, $\ldots$, t_n, u_1, $\ldots$, u_k are propositional constants.

$F \geq$

$$\frac{F\, x \supset X}{T\, X \supset t_1 \mid \ldots \mid T\, X \supset t_n}$$

Where $x \neq \emptyset$, and $t_1,\ldots,t_n$ are all the maximal members of $\mathcal{P}(\mathcal{A})$ such that, for each i, $x \not\subseteq t_i$.

$T \geq$

$$\frac{T\, x \supset X}{F\, X \supset t_i}$$

Where $x \neq \emptyset$, and t_i is any maximal member of $\mathcal{P}(\mathcal{A})$ such that $x \not\subseteq t_i$.

$F \leq$

$$\frac{F\, X \supset x}{T\, u_1 \supset X \mid \ldots \mid T\, u_k \supset X}$$

Where $x \neq \mathcal{A}$, and $u_1,\ldots,u_k$ are all minimal members of $\mathcal{P}(\mathcal{A})$ such that, for each i, $u_i \not\subseteq x$.

$T \leq$

$$\frac{T\, X \supset x}{F\, u_i \supset X}$$

Where $x \neq \mathcal{A}$, and u_i is any minimal member of $\mathcal{P}(\mathcal{A})$ with $u_i \not\subseteq x$.

In the $F \geq$ rule above, n-way branching is indicated. Similarly there is k-way branching in the $F \leq$ rule. The conditions that $x \neq \emptyset$ and $x \neq \mathcal{A}$ would not give unsound rules if dropped, but they would add cases that cannot help in proofs.

To understand the intuition behind these rules, consider rule $F \geq$ as an example. Suppose we have $F\, x \supset X$ on a tableau branch. Intuitively this tells us that at some world Γ in some model we have $x \not\subseteq v(\Gamma, X)$. We can extend $v(\Gamma, X)$ to a maximal member of $\mathcal{P}(\mathcal{A})$, call it t, such that $x \not\subseteq t$. Since t extends $v(\Gamma, X)$, intuitively we should have $T\, X \supset t$. Similar considerations apply to the other three rules as well. Formally, this amounts to saying that if we have a satisfiable tableau, and one of the Reversal Rules is applied, the result is another satisfiable tableau. This is the key point behind each of the rules below as well.

Next we have the rather obvious rules for $\wedge$ and $\vee$. The exceptions concerning $\emptyset$ and $\mathcal{A}$ are simply to avoid adding useless formulas.

Conjunction Rules x is any propositional constant other than $\emptyset$, and A and B are any formulas.

$T\wedge$

$$\frac{T\, x \supset (A \wedge B)}{\begin{array}{c} T\, x \supset A \\ T\, x \supset B \end{array}}$$

$F\wedge$

$$\frac{F\, x \supset (A \wedge B)}{F\, x \supset A \mid F\, x \supset B}$$

Disjunction Rules x is any propositional constant other than $\mathcal{A}$, and A and B are any formulas.

$T\vee$

$$\frac{T\,(A \vee B) \supset x}{\begin{array}{c} T\,A \supset x \\ T\,B \supset x \end{array}}$$

$F\vee$

$$\frac{F\,(A \vee B) \supset x}{F\,A \supset x \mid F\,B \supset x}$$

There are no rules for negation since we can take $\neg X$ as an abbreviation for $X \supset \emptyset$. Here are the implication rules.

Implication Rules Once more x is any propositional constant other than $\emptyset$, and A and B are any formulas.

$F \supset$

$$\frac{F\,x \supset (A \supset B)}{\begin{array}{c|c|c} T\,t_1 \supset A & \dots & T\,t_n \supset A \\ F\,t_1 \supset B & & F\,t_n \supset B \end{array}}$$

Where $t_1,\dots,t_n$ are all the non-empty members of $\mathcal{P}(\mathcal{A})$ such that $t_i \subseteq x$.

$T \supset$

$$\frac{T\,x \supset (A \supset B)}{F\,t_i \supset A \mid T\,t_i \supset B}$$

Where t_i is any non-empty member of $\mathcal{P}(\mathcal{A})$ such that $t_i \subseteq x$.

A case-by-case analysis allows us to verify the following key result.

LEMMA 6.3. *Any Branch Extension Rule applied to a satisfiable tableau produces another satisfiable tableau.*

Now soundness is simple to establish.

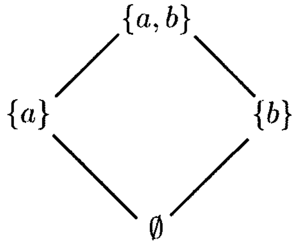
THEOREM 6.4. *If there is a closed tableau for $F\,B$ where B is a bounding formula, then B is valid. It follows that, for an arbitrary formula X, X is valid provided there is a closed tableau for $F\,\mathcal{A} \supset X$, or equally well, provided there are closed tableaus for $F\,\{a\} \supset X$ for each $a \in \mathcal{A}$.*

PROOF. Suppose B is not valid. Then there is some world of some model at which B is not true, and hence at which $F\,B$ is true. Then the set $\{F\,B\}$ is satisfiable. A tableau construction begins with the tableau consisting of only a root node, labeled with $F\,B$, and this tableau will be satisfiable. By Lemma 6.3, applying Branch Extension Rules to a tableau beginning with $F\,B$ can only produce satisfiable tableaus. But no satisfiable tableau can be closed, by Lemma 6.2.

The other parts of the theorem are straightforward. ■

6.2. Propositional Rules—A Specific Case

The general rule schemes are, perhaps, a bit much. We take a look at what they become in the simplest non-classical example. Let us say there are two agents, a and b, so that our Boolean algebra, $\mathcal{P}(\mathcal{A})$, looks like this.



In this setting we have the following, instantiating the conditions of the previous section.

Branch Closure Conditions A branch is closed if it contains the following.

$T\,\{a\} \supset \emptyset$	$T\,\{a\} \supset \{b\}$	$T\,\{a,b\} \supset \{a\}$
$T\,\{b\} \supset \emptyset$	$T\,\{b\} \supset \{a\}$	$T\,\{a,b\} \supset \{b\}$
$T\,\{a,b\} \supset \emptyset$		
$F\,\{a\} \supset \{a\}$	$F\,\{b\} \supset \{b\}$	
$F\,\emptyset \supset X$		
$F\,X \supset \mathcal{A}$		
$T\,\{a\} \supset X$	and	$F\,\{a\} \supset X$
$T\,\{a,b\} \supset X$	and	$F\,\{a\} \supset X$
$T\,\{b\} \supset X$	and	$F\,\{b\} \supset X$
$T\,\{a,b\} \supset X$	and	$F\,\{b\} \supset X$
$T\,\{a,b\} \supset X$	and	$F\,\{a,b\} \supset X$

The Reversal Rules instantiate as follows.

Reversal Rules

$F \geq$ $$\frac{F\,\{a\} \supset X}{T\,X \supset \{b\}} \qquad \frac{F\,\{b\} \supset X}{T\,X \supset \{a\}} \qquad \frac{F\,\{a,b\} \supset X}{T\,X \supset \{a\} \mid T\,X \supset \{b\}}$$

$T \geq$ $$\frac{T\,\{a\} \supset X}{F\,X \supset \{b\}} \qquad \frac{T\,\{b\} \supset X}{F\,X \supset \{a\}} \qquad \frac{T\,\{a,b\} \supset X}{F\,X \supset \{a\}} \qquad \frac{T\,\{a,b\} \supset X}{F\,X \supset \{b\}}$$

$F \leq$ $$\frac{F\,X \supset \emptyset}{T\,\{a\} \supset X \mid T\,\{b\} \supset X} \qquad \frac{F\,X \supset \{a\}}{T\,\{b\} \supset X} \qquad \frac{F\,X \supset \{b\}}{T\,\{a\} \supset X}$$

$$T \leq \qquad \frac{T\,X \supset \emptyset}{F\,\{a\} \supset X} \quad \frac{T\,X \supset \emptyset}{F\,\{b\} \supset X} \quad \frac{T\,X \supset \{a\}}{F\,\{b\} \supset X} \quad \frac{T\,X \supset \{b\}}{F\,\{a\} \supset X}$$

Conjunction and Disjunction Rules These are exactly as before.

And finally, the implication rules specify as follows.

Implication Rules

$F \supset$

$$\frac{F\,\{a\} \supset (A \supset B)}{\begin{array}{c} T\,\{a\} \supset A \\ F\,\{a\} \supset B \end{array}} \qquad \frac{F\,\{b\} \supset (A \supset B)}{\begin{array}{c} T\,\{b\} \supset A \\ F\,\{b\} \supset B \end{array}}$$

$$\frac{F\,\{a,b\} \supset (A \supset B)}{\begin{array}{c|c|c} T\,\{a\} \supset A & T\,\{b\} \supset A & T\,\{a,b\} \supset A \\ F\,\{a\} \supset B & F\,\{b\} \supset B & F\,\{a,b\} \supset B \end{array}}$$

$T \supset$

$$\frac{T\,\{a\} \supset (A \supset B)}{F\,\{a\} \supset A \mid T\,\{a\} \supset B} \qquad \frac{T\,\{b\} \supset (A \supset B)}{F\,\{b\} \supset A \mid T\,\{b\} \supset B}$$

$$\frac{T\,\{a,b\} \supset (A \supset B)}{F\,\{a\} \supset A \mid T\,\{a\} \supset B} \qquad \frac{T\,\{a,b\} \supset (A \supset B)}{F\,\{b\} \supset A \mid T\,\{b\} \supset B}$$

$$\frac{T\,\{a,b\} \supset (A \supset B)}{F\,\{a,b\} \supset A \mid T\,\{a,b\} \supset B}$$

Tableau examples are postponed until Section 7.

6.3. Modal Rules—General Version

There are several different kinds of modal tableaus. Here we use Boolean valued versions of *destructive* tableaus, and we begin with a brief sketch of the usual rules. After this we present the extension to many agents. We only treat the multi-agent analog of the modal logic K, but systems do exist for several other standard modal logics, [14, 16]. A fuller discussion of tableaus for standard modal logics can be found in [11], among other places.

For the moment, assume we are working with a standard tableau system for classical propositional logic, using signed formulas. Rules for the propositional connectives are well-known, and we skip over them. For the modal operator $\Box$ we begin with a definition, and then the rule is easy to state. For a set S of signed formulas, let $S^\sharp = \{T\,X \mid T\,\Box X \in S\}$ (this assumes $\Box$

is primitive and $\Diamond$ is defined). Now, here is the destructive modal rule for K, followed by an explanation.

$$\frac{\begin{array}{c}\mathcal{S}\\ F\,\Box X\end{array}}{\begin{array}{c}\mathcal{S}^{\sharp}\\ F\,X\end{array}}$$

This is a *destructive* rule because it does not lengthen a tableau branch—it replaces it, generally loosing information in the process. Read it as follows. If we have a tableau branch on which the set of signed formulas consists of $\mathcal{S}$ together with $F\,\Box X$, that entire branch may be *replaced* with a new branch on which the formulas are $\mathcal{S}^{\sharp}$ together with $F\,X$. Note that in displaying a tableau we use a double horizontal line to indicate a branch replacement. If several branches have nodes in common and one of those branches is being replaced, the other branches are unaffected. With pencil and paper, it is easiest to cross out all formulas on a branch being replaced, make fresh copies of shared formulas at the ends of unaffected branches, and append the replacement branch at the end of the branch with crossed out formulas. If a tableau is represented as a list of branches, with branches represented as lists of formulas, as is common in computer implementations of tableau theorem provers, the issue of shared formulas does not come up.

The idea behind this rule is actually quite simple. In a Kripke K model, suppose all members of a set $\mathcal{S}$ of signed formulas are true at some possible world Γ. (Recall, $T\,Z$ is true at the world if Z is, and $F\,Z$ is true if Z is not.) It is easy to see that if Δ is any possible world accessible from Γ, then all members of $\mathcal{S}^{\sharp}$ must be true at Δ. It follows that if $\mathcal{S}, F\,\Box X$ is satisfiable (its members are all true at some world of some model), then $\mathcal{S}^{\sharp}, F\,X$ is also satisfiable. This gives the soundness of the destructive tableau rule. Completeness proofs for the system can be found in many places.

We now move to the multi-agent setting. To adapt this rule we recall that the accessibility relation itself is now multi-valued. This means we need a multi-valued sharp operation on sets of signed bounding formulas. We continue to use $\mathcal{A}$ for a (finite) set of agents, and $\mathcal{P}(\mathcal{A})$ for the Boolean algebra of truth values, and we have propositional constants in the language for each member of $\mathcal{P}(\mathcal{A})$, as in Section 5.

DEFINITION 6.5. Let $\mathcal{S}$ be a set of signed bounding formulas, and x be a propositional constant.

$$\mathcal{S}^{\#}(x) = \{T\,(t \cap x) \supset X \mid T\,t \supset \Box X \in \mathcal{S} \text{ and } t \cap x \neq \emptyset\}$$

The motivation for the definition is very similar to that of the conventional version. Suppose we have a $\mathcal{P}(\mathcal{A})$ valued model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, v \rangle$. Let $\Gamma, \Delta \in \mathcal{G}$ and suppose $\mathcal{R}(\Gamma, \Delta) = x$. Then, if all members of $\mathcal{S}$ are true at Γ, it follows that all members of $\mathcal{S}^{\sharp}(x)$ will be true at Δ. We leave the checking of this to you. An easy consequence is the soundness of the rule given below.

Now, here is the only modal rule for the multi-agent system. As usual, assume the set of agents is $\mathcal{A}$.

Modal Rule $F\Box$ In the following, $t_1, \ldots, t_n$ are propositional constants for all members of $\mathcal{P}(\mathcal{A})$ for which $t \cap t_i \neq \emptyset$.

$$\begin{array}{c}
\mathcal{S} \\
F\, t \supset \Box X \\
\hline\hline
\begin{array}{c|c|c}
\mathcal{S}^{\#}(t_1) & \cdots & \mathcal{S}^{\#}(t_n) \\
F\,(t \cap t_1) \supset X & & F\,(t \cap t_n) \supset X
\end{array}
\end{array}$$

This is, again, a *destructive* rule, indicated by the double horizontal line. Branch contents are replaced, just as in the conventional case. Soundness of the rule was sketched informally above, and this can be converted into a formal proof of system soundness, along the lines of Theorem 6.4, without much difficulty. Completeness was established, in a more general setting, in [7], based on the completeness of a related sequent calculus in [5].

6.4. Modal Rules—A Specific Case Again

We return to the specific two agent example of Section 6.2, extending it to encompass the modal rule. Our set of agents is $\mathcal{A} = \{a, b\}$. We begin with the sharp operation. Since $t \cap \emptyset = \emptyset$, there is no branch below the double line in the Modal Rule above corresponding to $t_i = \emptyset$ so $\mathcal{S}^{\sharp}(\emptyset)$ needs no definition. This leaves us with three cases.

$$\begin{aligned}
\mathcal{S}^{\sharp}(\{a\}) &= \{T\,\{a\} \supset X \mid T\,\{a\} \supset \Box X \in \mathcal{S}\} \cup \\
&\quad\ \{T\,\{a\} \supset X \mid T\,\{a, b\} \supset \Box X \in \mathcal{S}\} \\
\mathcal{S}^{\sharp}(\{b\}) &= \{T\,\{b\} \supset X \mid T\,\{b\} \supset \Box X \in \mathcal{S}\} \cup \\
&\quad\ \{T\,\{b\} \supset X \mid T\,\{a, b\} \supset \Box X \in \mathcal{S}\} \\
\mathcal{S}^{\sharp}(\{a, b\}) &= \{T\,\{a\} \supset X \mid T\,\{a\} \supset \Box X \in \mathcal{S}\} \cup \\
&\quad\ \{T\,\{b\} \supset X \mid T\,\{b\} \supset \Box X \in \mathcal{S}\} \cup \\
&\quad\ \{T\,\{a, b\} \supset X \mid T\,\{a, b\} \supset \Box X \in \mathcal{S}\}
\end{aligned}$$

This gives us three separate modal rules. The $\emptyset$ case contributes no rule, or more precisely the case can be seen as a version of one of our closure rules.

Modal Rules

$F\Box - \{a\}$

$$\frac{\begin{array}{c} \mathcal{S} \\ F\{a\} \supset \Box X \end{array}}{\begin{array}{c|c} \mathcal{S}^{\#}(\{a\}) & \mathcal{S}^{\#}(\{a,b\}) \\ F\{a\} \supset X & F\{a\} \supset X \end{array}}$$

$F\Box - \{b\}$

$$\frac{\begin{array}{c} \mathcal{S} \\ F\{b\} \supset \Box X \end{array}}{\begin{array}{c|c} \mathcal{S}^{\#}(\{b\}) & \mathcal{S}^{\#}(\{a,b\}) \\ F\{b\} \supset X & F\{b\} \supset X \end{array}}$$

$F\Box - \{a,b\}$

$$\frac{\begin{array}{c} \mathcal{S} \\ F\{a,b\} \supset \Box X \end{array}}{\begin{array}{c|c|c} \mathcal{S}^{\#}(\{a\}) & \mathcal{S}^{\#}(\{b\}) & \mathcal{S}^{\#}(\{a,b\}) \\ F\{a\} \supset X & F\{b\} \supset X & F\{a,b\} \supset X \end{array}}$$

And now it is time to turn to a specific example.

7. A Tableau Example—With Problems

We gave general tableau rules in Sections 6.1 and 6.3, and concrete two agent versions in Sections 6.2 and 6.4. Soundness was sketched in this paper; completeness is proved in [5, 7]. It is time to consider tableau examples. We do so, but we work in a context with open problems and unresolved difficulties. These have largely to do with the translation of statements from natural language into the language family presented here. On the one hand, a proper intuition about such translations has yet to be developed. On the other hand, it may be that the formal language is not sufficiently expressive, and needs supplementation. A few small examples will be presented to illustrate these points, then a case study will be developed to help make the range of difficulties clear. It is hoped that all this will induce people to think about the issues involved, and perhaps contribute to the subject.

The origins of the present paper are in a talk given at the Workshop on Truth Values, held in Dresden in May of 2008, organized by Heinrich

Wansing and Yaroslav Shramko. In part of that talk I presented an example applying multi-agent tableaus to the muddy children puzzle. I later came to realize there was a problem with my formulation. In trying to give a better formulation for this paper, I further came to realize that my understanding of the behavior of common knowledge was not sufficient. It is well-known that common knowledge plays a central role in the muddy children puzzle. Common knowledge is not a simple concept to capture generally, but fortunately in the puzzle common knowledge is used positively, that is, only pre-existing common knowledge is used. Achieving common knowledge, which is more difficult, never arises. Using common knowledge positively is not hard to manage with tableaus, provided we use an underlying two-valued logic. The problem here is how to use positive common knowledge in a logic with sets of agents as truth values. This is something that needs serious work. The hope is that others will be motivated to think about the issues involved.

The muddy children puzzle was discussed in Section 2, with three children as a representative case. Even three is too many for present clarity, so we consider only two children now. Let us say we have two children, a and b, and so our underlying truth value space is $\mathcal{P}(\{a, b\})$. Also let us assume that both children have muddy foreheads.

Since the puzzle involves the interplay of the knowledge of two children, for this section we read the modal operator, $\Box$, epistemically. A truth value for $\Box X$, in $\mathcal{P}(\{a, b\})$, is the set of children who know X. Then the sentence $\{a\} \supset \Box X$ asserts that a knows X (and possibly b does also). The sentence $\Box X \supset \{b\}$ asserts that at most b knows X, and so a does not know X.

At the start the parent announces, "at least one of you has a muddy forehead." Since everybody hears this, and everybody witnesses that everybody hears it, and so on, this is common knowledge. The problem is how to represent this, and it is here that my presentation in the workshop began to go wrong. As we did in Section 2, we use the propositional letter M with the intended meaning "has a muddy forehead." Then representing that a has a muddy forehead simply amounts to asserting $\{a\} \supset M$. Likewise asserting $\{b\} \supset M$ corresponds to b having a muddy forehead. In the workshop talk I went on to represent that at least one child has a muddy forehead by $\{a\} \supset M \vee \{b\} \supset M$, but this cannot be a correct representation. Suppose, for the moment, that nobody has a muddy forehead, so the truth value of M is $\emptyset$. It follows that the truth value of $\{a\} \supset M$ would be $\{a\} \supset \emptyset = \overline{\{a\}} \cup \emptyset = \{b\}$. Likewise the truth value of $\{b\} \supset M$ would be $\{a\}$, and so the truth value of $\{a\} \supset M \vee \{b\} \supset M$ would be $\{a, b\}$. The sentence intended to express that somebody has a muddy forehead turns out to have maximum truth under the circumstance where nobody has a muddy forehead! This is a good example of the difficulties our intuition has where the present language is concerned.

Fortunately there is a natural way of representing that somebody has a muddy forehead, using tableau machinery. It is easy to see that the bounding formula $M \supset \emptyset$ asserts that nobody has a muddy forehead. That is, its truth value is maximum, $\{a, b\}$, exactly when the truth value of M is empty. So we need to say that $M \supset \emptyset$ is not the case, and we can do this using a signed formula, $F\,M \supset \emptyset$. The problem remains, to capture that this is to be common knowledge, true and unchanging throughout the problem.

In modal logic *based on classical two valued logic*, one talks about consequences of premises as well as about simple validity. It turns out that, unlike in the classical non-modal setting, modal premises can be *local* or *global*, and these are quite different. Semantically, X is a consequence of a local premise if X is true at those possible worlds where the premise is true. But X is a consequence of a global premise if X is true at all possible worlds of models meeting the condition that the global premise is also true at all possible worlds. In our context, we can think of the parent's announcement that somebody has a muddy forehead as a global premise. In classical modal tableaus, local premises can be added to a branch only at the beginning of a tableau construction, but global premises can be added at any point. The situation with multi-agent modal logic has not been properly investigated, but it seems likely that what works classically will carry over. Consequently we assume the following, as our way of representing the initial announcement:

$$F\,M \supset \emptyset \text{ can be added at any point of a tableau construction.} \tag{4}$$

At the start of the puzzle, everybody understands that nobody can see the status of their own forehead. That the forehead of a is muddy is expressed by $\{a\} \supset M$. If a knows this fact, we would have that the truth value of $\Box(\{a\} \supset M)$ is at least $\{a\}$. Then if it is not the case, that is, if $\Box(\{a\} \supset M)$ is not bounded below by $\{a\}$, it must be bounded above by $\{b\}$. So, we express that a does not know its own forehead is muddy by the bounding formula $\Box(\{a\} \supset M) \supset \{b\}$. To express that b does not know its forehead is muddy we need a similar formula. We introduce an abbreviation, N for 'nobody knows'.

$$N = [\Box(\{a\} \supset M) \supset \{b\}] \wedge [\Box(\{b\} \supset M) \supset \{a\}] \tag{5}$$

Even though (5) seems like a reasonable representation of the fact that nobody knows the status of their own forehead, difficulties can arise with this sort of formulation. An anonymous referee helpfully pointed out the following anomaly. Suppose, temporarily, that a does not have a muddy forehead, and b does know this. That a does not have a muddy forehead amounts to saying $\{a\} \supset M$ is false. How should we represent this falsehood? One natural way, following the ideas above, is to represent it by saying that $M \supset \{b\}$ is true—at most b has a muddy forehead. Then the assertion that b knows a does not have a muddy forehead becomes $\{b\} \supset \Box(M \supset \{b\})$. As it happens, this is simply valid! (A tableau proof is a simple matter.) So this cannot be the correct representation, although it appeared in earlier versions of this paper. Alternately one might introduce a negation operator in the usual way, $\neg A$ stands for $A \supset \emptyset$. And then we might represent the intended situation by $\{b\} \supset \Box\neg(\{a\} \supset M)$. This is not a validity, and so conveys information. But it is unsatisfactory that our intuitions are not strongly in favor of this formula. There is a third, unexplored, alternative. The negation just introduced has an intuitionistic flavor—indeed the most general version of the logic presented here uses Heyting algebras and not Boolean algebras. In formulation (4) we made use of F, which is part of the tableau machinery, but not part of the formal language itself. This is a kind of classical negation that, perhaps, could be brought in, at least for implications. After all, what we need to express is that $\{a\} \supset M$ is false, $\{a\} \not\supset M$. Such a connective, perhaps, could be introduced. As I remarked earlier, this section presents difficulties—it does not make them all go away. The intention is to encourage further work.

Despite the problem just discussed, we will use N from (5) to represent that nobody knows the status of their forehead. Unlike in the referee's example above, information inside the $\Box$ operator is positive, not negative. Formula (5) does not reduce to a simple validity—it conveys information. At the start N is not only true, but common knowledge. This cannot be represented as a global assumption since by the end of the puzzle everybody does know that their forehead is muddy. The truth of N changes. Since we cannot use the global assumption mechanism, we introduce a common knowledge operator, $\mathcal{C}$. We make the following assumptions about it. First, if something is common knowledge for anybody, it is common knowledge for everybody. Second, if something is common knowledge, it is true, and the fact that it is common knowledge is known to everybody. Formally, we make the following assumptions, formulated as tableau rules.

$$\frac{T\,\{a\} \supset \mathcal{C}X}{T\,\{a,b\} \supset \mathcal{C}X} \qquad \frac{T\,\{b\} \supset \mathcal{C}X}{T\,\{a,b\} \supset \mathcal{C}X} \tag{6}$$

$$\frac{T\,\{a,b\} \supset CX}{T\,\{a,b\} \supset X} \qquad \frac{T\,\{a,b\} \supset CX}{T\,\{a,b\} \supset \Box CX} \tag{7}$$

This is not intended to be a complete formalization of C in a multi-agent logic, though it may be enough for formulas in which C only occurs positively. These are simply intended to get the muddy children problem off the ground. A proper investigation of C in this context is needed.

Next we have conditions that are harder to capture. Everybody sees everybody else's forehead, and this is always the situation. So, if the forehead of a is muddy, b knows it, and if the forehead of a is not muddy, b knows that. This is the case at the start, and remains the case throughout. In some sense these are global assumptions. The difficulty comes in how to formalize them. In particular, if a does not have a muddy forehead, b should know that. As we saw above, to say that a does not have a muddy forehead is to assert $M \supset \{b\}$. To assert that b knows a does not have a muddy forehead apparently would be to assert $\{b\} \supset \Box(M \supset \{b\})$, but this is exactly the problematic formula considered earlier—it conveys no information because it is a validity. It is the wrong formulation.

In part the problem arises because $\supset$ has an easily understood behavior only in isolation. $M \supset \{b\}$ is true, that is, it has truth value $\{a, b\}$ just in case the truth value of M is less than or equal to $\{b\}$. Understanding $M \supset \{b\}$ in the context of a more complicated formula, such as $\{b\} \supset \Box(M \supset \{b\})$, is something for which our intuitions are less well-developed. But there is promise to the idea that some sort of global assumption is involved in formalizing that each child can see the other's forehead. We propose building this into the tableau rules, much as we did with (4). But things are more complicated now, because we have conditionals, not absolutes. Child b knows the forehead of a is not muddy *provided* it is, in fact, not muddy. What we will do is build this into the sharp operation. Here is the idea. If we are at a possible world where a does not have a muddy forehead, then b should know this, and so in any move to a world that is an alternative for b, it should still be the case that a does not have a muddy forehead. More formally, if we have $T\,M \supset \{b\}$ at a world, we should continue to have it at any b-alternative. More formally yet, if $T\,M \supset \{b\} \in S$, we want $T\,M \supset \{b\}$ to be in $S^\sharp(\{b\})$ and in $S^\sharp(\{a, b\})$. And similarly for a, of course. Here are the revised versions. In these X is an arbitrary formula, while M is a specific propositional letter informally representing muddiness.

$$\begin{aligned}\mathcal{S}^\sharp(\{a\}) = & \{T\{a\} \supset X \mid T\{a\} \supset \Box X \in \mathcal{S}\} \cup \\ & \{T\{a\} \supset X \mid T\{a,b\} \supset \Box X \in \mathcal{S}\} \cup \\ & \{T\{b\} \supset M \mid T\{b\} \supset M \in \mathcal{S}\} \cup \\ & \{T M \supset \{a\} \mid T M \supset \{a\} \in \mathcal{S}\}\end{aligned}$$

$$\begin{aligned}\mathcal{S}^\sharp(\{b\}) = & \{T\{b\} \supset X \mid T\{b\} \supset \Box X \in \mathcal{S}\} \cup \\ & \{T\{b\} \supset X \mid T\{a,b\} \supset \Box X \in \mathcal{S}\} \cup \\ & \{T\{a\} \supset M \mid T\{a\} \supset M \in \mathcal{S}\} \cup \\ & \{T M \supset \{b\} \mid T M \supset \{b\} \in \mathcal{S}\} \qquad (8)\end{aligned}$$

$$\begin{aligned}\mathcal{S}^\sharp(\{a,b\}) = & \{T\{a\} \supset X \mid T\{a\} \supset \Box X \in \mathcal{S}\} \cup \\ & \{T\{b\} \supset X \mid T\{b\} \supset \Box X \in \mathcal{S}\} \cup \\ & \{T\{a,b\} \supset X \mid T\{a,b\} \supset \Box X \in \mathcal{S}\} \cup \\ & \{T\{b\} \supset M \mid T\{b\} \supset M \in \mathcal{S}\} \cup \\ & \{T M \supset \{a\} \mid T M \supset \{a\} \in \mathcal{S}\} \cup \\ & \{T\{a\} \supset M \mid T\{a\} \supset M \in \mathcal{S}\} \cup \\ & \{T M \supset \{b\} \mid T M \supset \{b\} \in \mathcal{S}\}\end{aligned}$$

Now we represent the puzzle itself. Initially the parent announces that somebody has a muddy forehead, that is, S from (4) becomes a global assumption. The question is then asked whether anyone knows the status of their forehead, and nobody does, represented by N from (5). Everybody sees that nobody knows, so the current situation is common knowledge, that is, $\mathcal{C}N$. It is asked again if anybody knows, and this time everybody does, that is, $\Box(\{a,b\} \supset M)$ is the case. So $\mathcal{C}N$ is sufficient to yield $\Box(\{a,b\} \supset M)$ provided (4) is assumed. Thus we need to prove $\mathcal{C}N \supset \Box(\{a,b\} \supset M)$. To do this we may try to give a tableau proof of $\{a,b\} \supset (\mathcal{C}N \supset \Box(\{a,b\} \supset M))$. Or equivalently we could give tableau proofs of both $\{a\} \supset (\mathcal{C}N \supset \Box(\{a,b\} \supset M))$ and $\{b\} \supset (\mathcal{C}N \supset \Box(\{a,b\} \supset M))$. To keep the clutter down (somewhat) we just work with $\{a\} \supset (\mathcal{C}N \supset \Box(\{a,b\} \supset M))$.

The tableau starts as follows, using an implication rule $F \supset$ and common knowledge rules (6) followed by (7). Then rule $F\Box - \{a\}$ applies, replacing the branch with two new ones, one with $\mathcal{S}^\sharp(\{a\})$, and the other with $\mathcal{S}^\sharp(\{a,b\})$. For this step it doesn't matter if we use the modified version of the sharp operation in (8) or the original one.

$$\begin{array}{c|c}
\multicolumn{2}{c}{F\,\{a\} \supset (CN \supset \Box(\{a,b\} \supset M))} \\
\multicolumn{2}{c}{T\,\{a\} \supset CN} \\
\multicolumn{2}{c}{F\,\{a\} \supset \Box(\{a,b\} \supset M)} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset CN} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset \Box CN} \\
\hline\hline
T\,\{a\} \supset CN & T\,\{a,b\} \supset CN \\
F\,\{a\} \supset (\{a,b\} \supset M) & F\,\{a\} \supset (\{a,b\} \supset M)
\end{array}$$

On the new branch displayed on the left above we can apply rule (6) adding $T\,\{a,b\} \supset CN$ which makes the right branch a subset of the left. So we continue with the right branch only, since its closure will imply closure of the left branch.

$$\begin{array}{c|c}
\multicolumn{2}{c}{T\,\{a,b\} \supset CN} \\
\multicolumn{2}{c}{F\,\{a\} \supset (\{a,b\} \supset M)} \\
\multicolumn{2}{c}{T\,\{a\} \supset \{a,b\}} \\
\multicolumn{2}{c}{F\,\{a\} \supset M} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset N} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset [\Box(\{a\} \supset M) \supset \{b\}]} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset [\Box(\{b\} \supset M) \supset \{a\}]} \\
\hline
F\,\{b\} \supset \Box(\{b\} \supset M) & T\,\{b\} \supset \{a\}
\end{array} \tag{9}$$

We begin by applying $F \supset$ to the second line. We then apply one of (7), then a $T\wedge$ rule, recalling the definition of N, (5). There are three $T \supset$ rules that can be applied to the last formula before the branching, above. We have used one of the rules, and leave it to the reader to try the other two. The right branch is closed because of $T\,\{b\} \supset \{a\}$. We continue with the left branch only.

$$\begin{array}{c|c}
\multicolumn{2}{c}{T\,\{a,b\} \supset CN} \\
\multicolumn{2}{c}{F\,\{a\} \supset (\{a,b\} \supset M)} \\
\multicolumn{2}{c}{T\,\{a\} \supset \{a,b\}} \\
\multicolumn{2}{c}{F\,\{a\} \supset M} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset N} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset [\Box(\{a\} \supset M) \supset \{b\}]} \\
\multicolumn{2}{c}{T\,\{a,b\} \supset [\Box(\{b\} \supset M) \supset \{a\}]} \\
\multicolumn{2}{c}{F\,\{b\} \supset \Box(\{b\} \supset M)} \\
\multicolumn{2}{c}{T\,M \supset \{b\}} \\
\hline\hline
T\,M \supset \{b\} & T\,M \supset \{b\} \\
F\,\{b\} \supset (\{b\} \supset M) & F\,\{b\} \supset (\{b\} \supset M)
\end{array} \tag{10}$$

Above, $T\,M \supset \{b\}$ comes from $F\,\{a\} \supset M$ using a Reversal Rule, $F \geq$. Now rule $F\Box - \{b\}$ applies, and the branch is replaced by two new ones. One of the new branches involves $\mathcal{S}^\sharp(\{b\})$ and the other $\mathcal{S}^\sharp(\{a, b\})$, using the revised definition of sharp, (8). The two new branches happen to be identical in this case. We continue with the common version.

$$\begin{array}{c}
T\,M \supset \{b\} \\
F\,\{b\} \supset (\{b\} \supset M) \\
T\,\{b\} \supset \{b\} \\
F\,\{b\} \supset M \\
F\,\{a\} \supset M \\
F\,M \supset \emptyset \\
\hline
T\,\{a\} \supset M \;\big|\; T\,\{b\} \supset M
\end{array}$$

There is an application of $F \supset$ and then an application of Reversal Rule $T \leq$. These are followed by an application of (4), representing the parent's initial announcement. Reversal Rule $F \leq$ causes branching, and each branch is closed.

The tableau construction above should be considered to be exploratory. Common knowledge is always difficult to treat formally. We are not interested in how to achieve common knowledge, but in how to use it given its existence. As we observed above, with conventional logics of knowledge this is rather straightforward, but with the kind of multi-agent systems considered here, things are not well understood. The approach based on modifying the sharp operation was shown to work, but it was not really justified. A properly formulated soundness and completeness theorem is needed. This is an open problem.

8. Bisimulations

Bisimulations are basic and important tools in modal logic, and discussions of their properties and uses can be found extensively in the literature—[2] will serve as a representative. We do not go into the applications of bisimulations here, but we do consider how they may be generalized to the Boolean valued setting. We only discuss *frame* bisimulations, though the ideas extend naturally to bisimulations of *models* as well. A fuller presentation of our approach can be found in [9].

We begin with a brief summary of the classic bisimulation notion. Suppose we have two Kripke frames, in the standard sense. Say one is $\mathcal{F}_1 = \langle \mathcal{G}_1, \mathcal{R}_1 \rangle$ and the other is $\mathcal{F}_2 = \langle \mathcal{G}_2, \mathcal{R}_2 \rangle$. Let $\mathcal{B}$ be a relation between $\mathcal{G}_1$

and $\mathcal{G}_2$, that is, a subset of $\mathcal{G}_1 \times \mathcal{G}_2$. $\mathcal{B}$ is a *frame bisimulation* provided the following two conditions are met. First, suppose $\Gamma_1, \Delta_1 \in \mathcal{G}_1$ with $\Gamma_1 \mathcal{R}_1 \Delta_1$, and also suppose $\Gamma_1 \mathcal{B} \Gamma_2$, where $\Gamma_2 \in \mathcal{G}_2$. Then there is some $\Delta_2 \in \mathcal{G}_2$ such that $\Gamma_2 \mathcal{R}_2 \Delta_2$ and also $\Delta_1 \mathcal{B} \Delta_2$. The second condition is like the first, with the roles of the two frames switched around. Suppose $\Gamma_2, \Delta_2 \in \mathcal{G}_2$ with $\Gamma_2 \mathcal{R}_2 \Delta_2$, and suppose $\Gamma_1 \in \mathcal{G}_1$ with $\Gamma_1 \mathcal{B} \Gamma_2$. Then there is some $\Delta_1 \in \mathcal{G}_1$ such that $\Gamma_1 \mathcal{R}_1 \Delta_1$ and also $\Delta_1 \mathcal{B} \Delta_2$. This is sometimes loosely described by saying, for each move in either frame there is some corresponding move in the other.

Two Kripke models that are based on bisimilar frames are *bisimilar models* if possible worlds that correspond under the bisimilarity have the same propositional letters true at them. The key result is that under any model bisimulation, two bisimilar models will evaluate the same formulas to be true at worlds that are related under the bisimulation. This implies that no modal formula can distinguish between bisimilar models, and so bisimilarity provides a tool for examining the expressivity of modal languages.

Now we extend the notion of bisimulation to the multi-agent setting. As usual, we take $\mathcal{A}$ to be a finite set of agents, and use $\mathcal{P}(\mathcal{A})$ as a space of truth values.

DEFINITION 8.1. We say $\mathcal{B}$ is *a* $\mathcal{P}(\mathcal{A})$ *valued relation between sets* S_1 *and* S_2 *provided* $\mathcal{B} : S_1 \times S_2 \longrightarrow \mathcal{P}(\mathcal{A})$. For such a relation, and for each $c \in \mathcal{A}$, by the *c slice of* $\mathcal{B}$ we mean the classical relation $\mathcal{B}_c \subseteq S_1 \times S_2$ such that $s_1 \mathcal{B}_c s_2$ just in case $c \in \mathcal{B}(s_1, s_2)$.

We also make use of the notion of c slice of a frame, from Definition 4.1, where the definition of the accessibility relation is really a special case of that above.

DEFINITION 8.2. Let $\mathcal{B}$ be a $\mathcal{P}(\mathcal{A})$ valued relation between sets $\mathcal{G}'$ and $\mathcal{G}''$.

1. Let $\mathcal{F}' = \langle \mathcal{G}', \mathcal{R}' \rangle$ and $\mathcal{F}'' = \langle \mathcal{G}'', \mathcal{R}'' \rangle$ be two $\mathcal{P}(\mathcal{A})$ valued frames. We say $\mathcal{B}$ is *a* $\mathcal{P}(\mathcal{A})$ *valued frame bisimulation* between $\mathcal{F}'$ and $\mathcal{F}''$ provided, for each agent $c \in \mathcal{A}$, the c slice of $\mathcal{B}$ is a conventional frame bisimulation between the c slice of $\mathcal{F}'$ and the c slice of $\mathcal{F}''$.
2. Let $\mathcal{M}' = \langle \mathcal{G}', \mathcal{R}', v' \rangle$ and $\mathcal{M}'' = \langle \mathcal{G}'', \mathcal{R}'', v'' \rangle$ be two $\mathcal{P}(\mathcal{A})$ valued modal models. We say $\mathcal{B}$ is *a* $\mathcal{P}(\mathcal{A})$ *valued model bisimulation* between $\mathcal{M}'$ and $\mathcal{M}''$ provided, for each agent $c \in \mathcal{A}$, the c slice of $\mathcal{B}$ is a conventional model bisimulation between the c slice of $\mathcal{M}'$ and the c slice of $\mathcal{M}''$.

We don't actually make use of part 2 of the definition above, but we give it for completeness sake. We give an example of a frame bisimulation, in

Figure 3. In it, $\mathcal{A} = \{1, 2, 3\}$. On the left of the figure is shown the $\mathcal{P}(\mathcal{A})$ valued frame $\mathcal{F}'$, and on the right a second, $\mathcal{F}''$. The bisimulation relation is shown using dotted lines with labels. Dotted lines whose label would be $\emptyset$ are omitted.

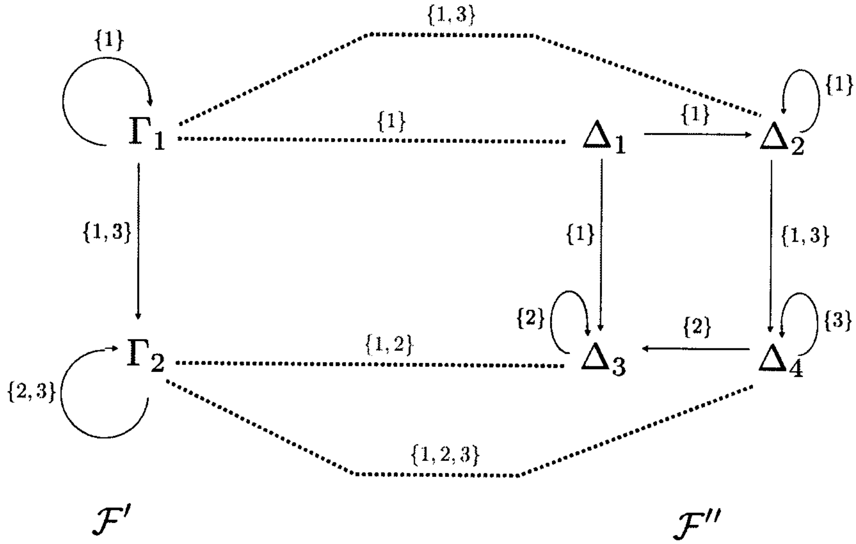


Figure 3. Frame Bisimulation Example

In order to verify that what is shown in Figure 3 is a $\mathcal{P}(\mathcal{A})$ valued frame bisimulation, we must check that we have conventional frame bisimulations for each of the three slices. We give a diagram corresponding to the slice for agent 3, in Figure 4, and leave it to you to verify that it depicts a conventional bisimulation. We omit the other two cases.

Basic results concerning bisimulations extend to the multi-agent case, but this is not what concerns us here. The question we address is: is there any simple, mechanical test for determining whether or not we have a bisimulation, in the generalized sense we are considering? The answer is affirmative, but it requires consideration of Boolean valued matrices, more general than the two-valued Boolean matrices that one usually sees.

In a two valued setting, an accessibility relation can be represented naturally by a *transition matrix*—a standard Kripke frame is a graph, after all. An obvious generalization of transition matrix makes sense in the multi-agent setting, too. Consider the frame $\mathcal{F}'$, from Figure 3. We can represent its multi-agent accessibility relation using a Boolean valued transition matrix, which we call $[\mathcal{F}']$. The matrix $[\mathcal{F}']$ is 2×2 because $\mathcal{F}' = \langle \mathcal{G}', \mathcal{R}' \rangle$

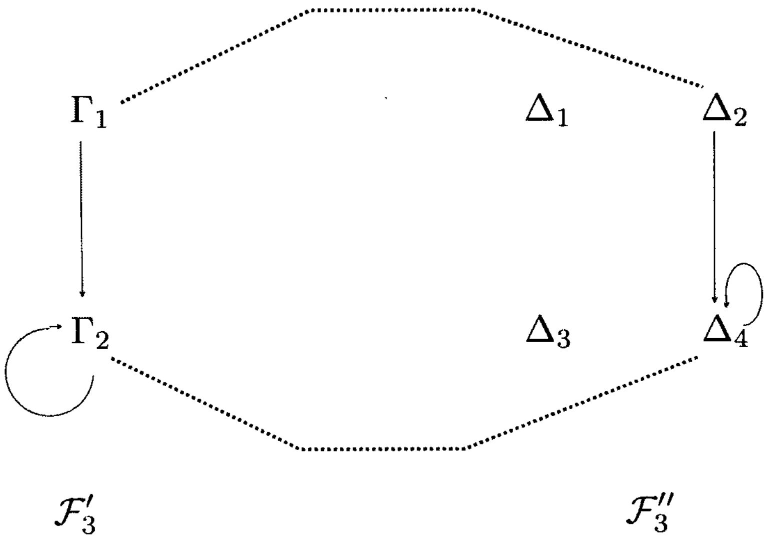


Figure 4. Agent 3 Slice Bisimulation

and $\mathcal{G}' = \{\Gamma_1, \Gamma_2\}$ has two possible worlds. In $[\mathcal{F}']$, set the entry in row i, column j to be $\mathcal{R}'(\Gamma_i, \Gamma_j)$. This gives us the following.

$$[\mathcal{F}'] = \begin{bmatrix} \{1\} & \{1,3\} \\ \emptyset & \{2,3\} \end{bmatrix}$$

This matrix can easily be made to act as a possibility operator, but we do not follow up on this point now.

As we did with the first frame, $\mathcal{F}''$ can also be represented by a $\mathcal{P}(\mathcal{A})$ valued transition matrix.

$$[\mathcal{F}''] = \begin{bmatrix} \emptyset & \{1\} & \{1\} & \emptyset \\ \emptyset & \{1\} & \emptyset & \{1,3\} \\ \emptyset & \emptyset & \{2\} & \emptyset \\ \emptyset & \emptyset & \{2\} & \{3\} \end{bmatrix}$$

Suppose we represent the relation $\mathcal{B}$ using the following matrix, which is not really a transition matrix in the sense that $[\mathcal{F}']$ and $[\mathcal{F}'']$ are.

$$[\mathcal{B}] = \begin{bmatrix} \{1\} & \emptyset \\ \{1,3\} & \emptyset \\ \emptyset & \{1,2\} \\ \emptyset & \{1,2,3\} \end{bmatrix}$$

This matrix is 4×2, where 4 is the size of $\mathcal{F}''$ and 2 is the size of $\mathcal{F}$. The entry in $[\mathcal{B}]$ in row i, column j is $\mathcal{B}(\Gamma_j, \Delta_i)$.

With matrices involved, matrix operations can be introduced. We understand matrix multiplication to be computed in the long-familiar way, but with the role of number addition replaced by the join operation ($\cup$ in $\mathcal{P}(\mathcal{A})$), and multiplication by the meet operation ($\cap$ in $\mathcal{P}(\mathcal{A})$). With this operation we easily calculate the following, for instance.

$$[\mathcal{B}][\mathcal{F}'] = \begin{bmatrix} \{1\} & \{1\} \\ \{1\} & \{1,3\} \\ \emptyset & \{2\} \\ \emptyset & \{2,3\} \end{bmatrix} \qquad [\mathcal{F}''][\mathcal{B}] = \begin{bmatrix} \{1\} & \{1\} \\ \{1\} & \{1,3\} \\ \emptyset & \{2\} \\ \emptyset & \{2,3\} \end{bmatrix}$$

For two $\mathcal{P}(\mathcal{A})$ valued matrices M and N with the same number of rows and the same number of columns, we write $M \leq N$ to mean that each entry of M is a subset of the corresponding entry of N. In the example we have been considering, we have that $[\mathcal{B}][\mathcal{F}'] \leq [\mathcal{F}''][\mathcal{B}]$.

We write M^T for the *transpose* of matrix M, in the usual way. We leave it to you to check that $[\mathcal{B}]^T[\mathcal{F}''] \leq [\mathcal{F}'][\mathcal{B}]^T$ as well. It is the combination of these two inequalities that guarantees we have a frame bisimulation.

THEOREM 8.3. *Let $\mathcal{A}$ be a finite set of agents. Let $\mathcal{F}'$ and $\mathcal{F}''$ be $\mathcal{P}(\mathcal{A})$ valued frames, and let $\mathcal{B}$ be a $\mathcal{P}(\mathcal{A})$ valued relation between $\mathcal{F}'$ and $\mathcal{F}''$. Then the following two items are equivalent.*

1. *$\mathcal{B}$ is a $\mathcal{P}(\mathcal{A})$ valued frame bisimulation between $\mathcal{F}'$ and $\mathcal{F}''$ (Definition 8.2);*
2. *$[\mathcal{B}][\mathcal{F}'] \leq [\mathcal{F}''][\mathcal{B}]$ and $[\mathcal{B}]^T[\mathcal{F}''] \leq [\mathcal{F}'][\mathcal{B}]^T$.*

There are some unstated assumptions in the theorem above. Strictly speaking, we have matrix representations for frame accessibility relations and a bisimulation relation, only if the frames involved are finite. And even then, how the possible worlds are numbered ($\Gamma_1, \Gamma_2, \ldots$) affects the details of the matrices. But an enumeration of possible worlds can be arbitrarily chosen, and the theorem will hold. Further, the set of possible worlds can also be denumerable—an obvious extension of matrix operations to encompass matrices with countably many rows, or columns, or both, can be made, and the theorem still holds. The result can be further extended from frames to models. Details can be found in [9], but this is far enough to develop things here.

9. What More?

We have argued for the usefulness of allowing more general Boolean algebras than just the usual $\{false, true\}$ as truth value spaces. Doing so allows one to treat the knowledge of multiple agents uniformly, combining multiple knowledge operators into one. This makes one suspect that an approach to knowledge puzzles that is uniform across varying numbers of agents may be possible. We also saw that allowing more general Boolean algebras leads to an interesting computational approach to bisimulation. In fact, the material discussed here is presented more generally elsewhere. We say something about that, and close with suggestions for further research.

The material in Sections 2 through 5 is drawn from [4, 5], and that in Sections 6 and 7 from [7]. In this paper we considered agents that were independent, but in the earlier papers there could be dependencies between agents. For example, agent 2 might accept the truth (at each possible world) of anything agent 1 accepted (the match might not be exact—agent 2 might accept more truths than agent 1). If such dependencies are allowed then Boolean algebras are no longer the appropriate truth value spaces, instead *Heyting algebras* are. The resulting logic has an intuitionistic + modal flavor. The tableau rules given here are actually presented for Heyting algebras in [7].

In this paper we took $\Box$ as primitive and $\Diamond$ as defined. If Heyting algebras are brought into the picture this is no longer appropriate. $\Box$ is still understood semantically more-or-less as it is here.

$$v(\Gamma, \Box X) = \bigcap_{\Delta \in \mathcal{G}} [\mathcal{R}(\Gamma, \Delta) \Rightarrow v(\Delta, X)]$$

This looks like the definition given in (3), but now $\Rightarrow$ is the implication of a Heyting algebra, which is not definable from complement and join. Then the evaluation of $\Diamond X$, given at the end of Section 3, no longer goes through. Instead it is made into an independent definition.

$$v(\Diamond X) = \bigcup_{\Delta \in \mathcal{G}} [\mathcal{R}(\Gamma, \Delta) \cap v(\Delta, X)]$$

With dependent agents, using Heyting algebras, $\Box$ and $\Diamond$ are not dual operators, as they are classically.

The matrix approach to bisimulations, from Section 8, originates in [9]. Not just frame, but also model bisimulations are investigated, as part of a general operator approach to the subject. If one considers dependent agents, bringing Heyting algebras into the picture, propositional connectives have an intuitionistic flavor, and $\Box$ and $\Diamond$ are no longer dual operators, as noted above, so it would seem that the notion of a bisimulation must become more complex than in Section 8. This is investigated in [3], where two different notions of bisimulation are introduced, *strong bisimulation* and *weak t bisimulation.* Although it was developed independently of [9], the work in [3] seems to show that weak t bisimulation is the Heyting algebra counterpart of the Boolean version considered here, though this has not been fully checked yet.

Boolean/Heyting algebra methodology even extends to treat multi-agent versions of non monotonic modal logics. Investigation on this began in [6] and was extensively continued in [15, 17, 18].

What remains to be done? Quite a bit, actually. We briefly sketch some issues, with the hope of encouraging others to work on them.

We saw in Section 7 that it is sometimes unclear how to represent knowledge using the multi-agent approach. In particular, this affects knowledge of negative facts, and common knowledge. To address the issue of negative knowledge, it was suggested that an appropriate $\not\supset$ operator might be introduced. This needs investigation. As to common knowledge, we noted that for the muddy children puzzle only positive common knowledge is needed. That is, one makes use of common knowledge but one is never asked to conclude that it has been achieved. This implies that any common knowledge operators arising in tableau arguments will appear in positive locations, and this simplifies things in the classical two-valued setting. But in the present approach further work is clearly needed. There are other puzzles that have been considered in the literature, including variations on muddy children. Most of these (perhaps all of them) have the same positive-only aspect where common knowledge is concerned. Thus what is needed is a uniform way of incorporating positive common knowledge into the multi-agent logics presented here. This must involve an appropriate semantics, tableau rules, and soundness and completeness results.

In section 8 we saw that there was a nice computational test for whether something was a bisimulation, involving matrices. We noted above that the Boolean algebra approach generalizes naturally to a Heyting algebra version, with an intuition involving dependent agents. We also noted that there has been work on bisimulations in the Heyting algebra setting, [3]. It is not clear that matrix methods carry over to Heyting algebras, particularly in the light of the independence of $\Box$ and $\Diamond$. Nonetheless, it is possible that some simple computational test can be devised. This needs further exploration.

Bilattices are natural generalizations of the well-known four-valued logic of Belnap, [1]. A philosophically oriented presentation can be found in [10], though bilattices have also been applied in the area of logic programming semantics, and in modeling certain aspects of natural language. Bilattices have two partial orderings, one on degree of truth and one on degree of information. The truth ordering of bilattices can be used analogously to the way we used Boolean algebras above, to provide interpretations of modal formulas in bilattice valued Kripke models. There is some work on this subject, beginning with [12, 13], up to the present, [20], but fundamental work remains to be done. In particular, what relationships connect the information ordering with bisimulations seems quite an open issue.

Finally, we have taken sets of finitely many agents as truth values. This gives us finite Boolean algebras. One might also consider infinite Boolean algebras, something that is done in one approach to forcing arguments in set theory. Complete tableau systems become problematic then, though some of the bisimulation work extends reasonably well. It is not clear what the purpose of such a generalization would be if applications to human knowledge and reasoning are intended, but the mathematics that results might be of interest for its own sake, and is worth exploring for that reason.

We hope enough has been said to show the utility and interest to be found in working with sets of agents as truth values. And we hope some readers will be motivated to continue with the development of the area.

Acknowledgements. This research was partly supported by CUNY Collaborative Grant CIRG R14.

References

[1] Belnap, Jr., N. D., 'A useful four-valued logic', in J. M. Dunn and G. Epstein (eds.), *Modern Uses of Multiple-Valued Logic*, D. Reidel, 1977.

[2] Blackburn, P., M. de Rijke, and Y. Venema, *Modal Logic*, Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, UK, 2001.

[3] Eleftheriou, P. E., C. D. Koutras, and C. Nomikos, 'Notions of bisimulation for Heyting-valued modal languages', in A. Beekman, C. Dimitracopulos, and B. Loewe (eds.), *Logic and Theory of Algorithms, Proceedings of Computability in Europe 2008*, Athens University Press, 2008, pp. 117–126.

[4] Fitting, M. C., 'Many-valued modal logics', *Fundamenta Informaticae* 15:235–254, 1991.

[5] Fitting, M. C., 'Many-valued modal logics, II', *Fundamenta Informaticae* 17:55–73, 1992.

[6] Fitting, M. C., 'Many-valued non-monotonic modal logics', in A. Nerode and M. Taitslin (eds.), *Logical Foundations of Computer Science — Tver '92*, Springer Lecture Notes in Computer Science, 620, 1992, pp. 139–150.

[7] FITTING, M. C., 'Tableaus for many-valued modal logic', *Studia Logica* 55:63–87, 1995.

[8] FITTING, M. C., *First-Order Logic and Automated Theorem Proving*, Springer-Verlag, 1996, First edition, 1990.
Errata at *http://comet.lehman.cuny.edu/fitting/errata/errata.html.*

[9] FITTING, M. C., 'Bisimulations and Boolean vectors', in P. Balbiani, N.-Y. Suzuki, and M. Zakharyaschev (eds.), *Advances in Modal Logic*, volume 4, King's College Publications, 2003, pp. 97–125.

[10] FITTING, M. C., 'Bilattices are nice things', in T. Bolander, V. Hendricks, and S. A. Pedersen (eds.), *Self-Reference*, Center for the Study of Language and Information, 2006, chapter 3, pp. 53–77.

[11] FITTING, M. C., 'Modal proof theory', in P. Blackburn, J. van Benthem, and F. Wolter (eds.), *Handbook of Modal Logic*, chapter 2, pp. 85–138, Elsevier, 2007.

[12] GINSBERG, M. L., 'Bilattices and modal operators', in *Proceedings of the Third Conference on Theoretical Aspects of Reasoning about Knowledge*, San Francisco, CA, TARK, Morgan Kaufmann, 1990, pp. 273–287.

[13] GINSBERG, M. L., 'Bilattices and modal operators', *Journal of Logic and Computation* 1(1):41–69, 1990.

[14] KOUTRAS, C. D., 'A catalog of weak many-valued modal axioms and their corresponding frame classes', *Journal of Applied Non-Classical Logics* 13(1):47–73, 2003.

[15] KOUTRAS, C. D., G. KOLETSOS, and S. ZACHOS, 'Many-valued modal non-monotonic reasoning: sequential stable sets and logics with linear truth spaces', *Fundamenta Informaticae* 38(3):281 – 324, 1999.

[16] KOUTRAS, C. D., C. NOMIKOS, and P. PEPPAS, 'Canonicity and completeness results for many-valued modal logics', *Journal of Applied Non-Classical Logics* 12(1):7–42, 2002.

[17] KOUTRAS, C. D., and P. PEPPAS, 'Weaker axioms, more ranges', *Fundamenta Informaticae*, 51(3):297 – 310, 2002.

[18] KOUTRAS, C. D., and S. ZACHOS, 'Many-valued reflexive autoepistemic logic', *Logic Journal of IGPL* 8(1):33–54, 2000.

[19] MONK, J. D., and R. BONNET (eds.), *Handbook of Boolean Algebras*, North-Holland, 1989, (Three volumes).

[20] SIM, K. M., 'Beliefs and bilattices', in *Methodologies for Intelligent Systems*, Lecture Notes in Computer Science, volume 869, Springer, Berlin/Heidelberg, 1994, pp. 594–603.

[21] SMULLYAN, R. M., *First-Order Logic*, Springer-Verlag, Berlin, 1968, Revised Edition, Dover Press, New York, 1994.

MELVIN FITTING
Lehman College
City University of New York
250 Bedford Park Boulevard West
Bronx, NY 10468-1589, USA
melvin.fitting@lehman.cuny.edu