Layered Clausal Resolution in the Multi-Modal Logic of Beliefs and Goals

Jamshid Bagherzadeh* and S. Arun-Kumar **

Indian Institute of Technology Delhi, 110016, New Delhi, India
jamshid@cse.iitd.ernet.in, sak@cse.iitd.ernet.in

Abstract. In this paper a proof technique for reasoning about the multi-modal logic of beliefs and goals is defined based on resolution at different levels of a tree of clauses. We have considered belief and goal as normal modal logic operators. The technique is inspired by that in [6, 7] and allows for a locality property to be satisfied. The main motivation for this work arises not as much from theorem-proving as from the notion of belief and goal revision under an assumption of consistency of the beliefs and goals of an agent. We also present proofs of soundness and completeness of the logic.
Keywords: multi-modal logic, multi-agent systems, resolution, proof method, belief revision.

1 Introduction

Modal logics are widely used for different purposes in computer science and mathematics. This class of logics extends classical logic with two main operators, necessity ($\Box$) and possibility ($\Diamond$) [4]. The semantics of these logics are usually defined in terms of Kripke structures [15]. Modal logics are used in the representation of knowledge, belief, goals and other mental attitudes of agents. Agents usually have three aspects:

- Informational aspects like Knowledge and Beliefs. The modal logics of S5, and KD45 are used usually for these aspects.
- Motivational aspects like Goals, Desires and Intentions. Modal logics of KD are used commonly for these aspects.
- Dynamic or temporal aspects. Linear time or branching time temporal logics are used for modeling these aspects.

In this paper we don't consider the dynamic aspects of agents and we only assume the informational and motivational issues. We use *Belief*, *Goal* and *Intention* for informational and motivational aspects respectively.

In some of the recent literature on agents, the mental state of an agent (in a system of many communicating agents each with incomplete knowledge of the global state of the system) is usually represented by data structures representing

* Supported by the ministry of science, research and technology, I.R. Iran.
** Partly supported by research grant F.26-1/2002-TS.V from MHRD, Govt. of India.

the beliefs, goals and intentions of the agents [21]. Two important issues arise in the context of execution of agent programs:

1. How does the mental state of an agent get revised when a new input arrives?
2. How does one reason about the mental state of an agent assuming that it has a finite base of beliefs, goals and intentions, even though their logical consequences may be infinite?

The two issues are closely linked since it is necessary to be able to reason about the mental state to ensure that its revision does not create any logical inconsistency. We will discuss these issues in this paper.

Assume Ψ_i is the mental state of agent i. Alchourron et.al. in [1] have proposed some postulates for belief expansion, contraction and revision which are well known as AGM postulates. The idea is to satisfy the AGM postulates when a new belief ϕ is observed and intended to be added to the belief state of the agent. For example one of the important issues in the revision is the consistency issue, i.e. if Ψ_i is consistent before addition of ϕ, then it should remain consistent after the addition also. This may result in the removal of some of the existing formulas from Ψ_i which contradict ϕ. There are different ways of defining functions for expansion, contraction and revision which satisfy AGM postulates (perhaps not all) in the belief sets[1]. We are not going to define these methods in detail.

We assume each agent has a belief base and a goal base consisting of a set of formulas. The structure of the formulas will be discussed in section 3. Although not common, we assume AGM postulates should hold for the goal base too. The consistency of the set of formulas is checked using the resolution method which will be discussed later. For the sake of completeness we define a simple procedure of revision (using expansion and contraction according to Levi [16]).

```
Function Revise(S, φ)     S is belief or goal base and φ is a new formula.
    S= S ∪ {φ};     expansion
    return (Contraction(S));     contraction of ¬φ
End Revise.

Function Contraction(S)
    S_0 = S;  i=0;
    while (S_i ⊨ false) do
        Find minimum F_i ⊆ S_i s.t. F_i ⊨ false;
        g_i = γ(F_i) ;     g_i is one of the formulas of F_i
        S_{i+1} = S_i − {g_i};     remove g_i from S_i
        i = i + 1;
    end while;
    return S_i;
End Contraction.
```

[1] A belief set is closed under logical consequence and so it is infinite but belief base is not closed under logical consequence and so it is finite

where γ is a function to select a formula from F_i (according to some criteria). In the function Contraction, F_i is one of the minimal subsets of S_i which implies **false**. To find F_i we start from the rule which has implied **false** and by backtracking the route which has resulted in **false**, we may find the subset of formulas which have implied it. If there are more subsets of formulas which imply **false**, they will be found in the next iterations.

Our contraction function is similar to *kernel base contraction* method. It has been shown in the literature [13] that kernel base contraction method satisfies the postulates of AGM (except recovery postulate for contraction[2]).

2 Beliefs and Goals

In this framework we consider n agents each of which has a belief base and a goal base for representing his mental state. We assume the modal operators B and G which stand for belief and goal respectively, satisfy the axioms of **KD45** and **KD**.

A crucial question is that how we can corporate the *intention* modality in such a framework since intentions are also an important part of any agent's mental state. Various authors [5, 20, 14] have given sound reasons that intention should be treated as non-normal modal operator. Therefor we assume intention is a derived operator in the spirit of [5], which may be defined as $I_i\phi \equiv G_i\phi \wedge B_i\neg\phi$.

Suppose $Ag = \{1, \ldots, n\}$ is a set of agents, and B_i and G_i (Belief and Goal respectively) for any $i \in Ag$, are called the *mental attitudes* for agent i. Let $\mathcal{O} = \{B, G\}$ be a set of symbols. Let $\mathbf{V}$ be the set $(\mathcal{O} \times Ag)^*$, i.e., the set of finite strings of the form $o_{1 i_1} \ldots o_{n i_n}$ with $o_k \in \mathcal{O}$ and $i_k \in Ag$. We call any $v \in \mathbf{V}$, a view. Intuitively, each view in $\mathbf{V}$ represents a possible nesting of mental attitudes. We may imagine the information store as a collection of n trees, such that the tree rooted at Ag_i consists of the information of agent i. Figure 1 shows a schematic information store of the multi-agent system and particularly that of agent i. Considering this structure, we assume any agent has a set of beliefs called the **belief base** and a set of goals called the **goal base**. We assume beliefs of an agent should be consistent. We also suppose the goals of an agent are consistent (set of goals is a subset of desires which are themselves consistent). These two sets are represented by Ψ_{B_i} and Ψ_{G_i} respectively. Each of these sets, contains formulas of a multi-modal logic called BG_n, which will be discussed below. Each formula of BG_n will be transformed to clauses, and clauses will be stored in different nodes (or views) of the tree. Then for reasoning about the system we use resolution inside any view or between two adjacent views.

The remaining part of this paper is organized as follows. In section 3 we define the syntax and semantics of the logic BG_n. Section 4 discusses the normal form NF_{BG} and the transformation of BG_n formulas to NF_{BG} clauses with a small example. Section 5 defines the resolution rules. Then we prove the soundness

[2] If we want to remove $p \vee q$ then we must remove p and q consequently, but after re-addition of $p \vee q$ it will imply neither p nor q.

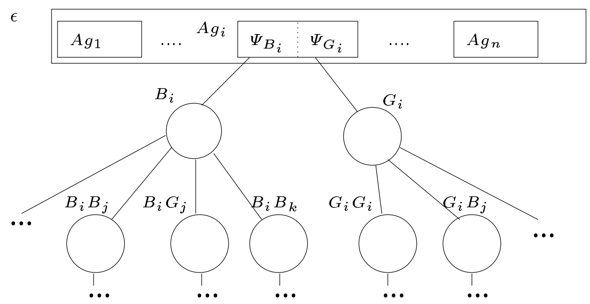

Fig. 1. Information store as a tree, with nodes representing views.

and completeness of the resolution system in section 6 and finally section 7 is the conclusion.

3 Syntax and Semantics of BG_n

As we said, any agent has two sets of BG_n formulas representing its beliefs, and goals. Formulas of BG_n are constructed from a set $\mathcal{P} = \{p, q, r, ...\}$ of atomic propositions, and the constants **true** and **false**. The language contains the standard propositional connectives $\neg$, $\wedge$, $\vee$, and unary modal connectives B_i and G_i ($i \in Ag$). Formally the set WFF_{BG} of well-formed formulas of BG_n, is defined as the smallest set such that

- any element of $\mathcal{P}$ is in WFF_{BG};
- **true** and **false** are in WFF_{BG};
- if F and G are in WFF_{BG} then so are
  $\neg F \quad , F \vee G \quad , F \wedge G \quad , B_iF \quad , G_iF \qquad$ where $i \in Ag$.

We use another binary operator $F \Rightarrow G$ which is an abbreviation of $\neg F \vee G$. We define some particular classes of formulas that will be useful later.

Definition 1. A **literal** l is either p or $\neg p$ where $p \in \mathcal{P}$. A **simple modal literal** is either O_il, or $\neg O_il$, where l is a literal, $i \in Ag$ and $O \in \{B, G\}$. A **modal literal** is a literal l or its negation $\neg l$ and if F is a modal literal then O_iF and $\neg O_iF$ also are modal literals, where $O \in \{B, G\}$.

Definition 2. A Model M is a structure $M = \langle S, L, S_0, B_1, ..., B_n,\ G_1, ..., G_n\rangle$, where S is a set of states ranged over by s and t and $\{\} \neq S_0 \subseteq S$ is a set of initial states. L is a state labeling function, i.e., $L : S \rightarrow 2^{\mathcal{P}}$. B_i, for all $i \in Ag$ is the agent belief accessibility relation over states, i.e., $B_i \subseteq S \times S$, where each B_i is transitive ($\forall s, s', s'' \in S$: if $(s, s') \in B_i$ and $(s', s'') \in B_i$ then $(s, s'') \in B_i$), serial ($\forall s \in S, \exists s' \in S$ s.t. $(s, s') \in B_i$), and euclidean ($\forall s, s', s'' \in S$, if $(s, s') \in B_i$ and $(s, s'') \in B_i$ then $(s', s'') \in B_i$). Finally G_i, for all $i \in Ag$ is the agent goal accessibility relation over states, i.e., $G_i \subseteq S \times S$, where each G_i is serial.

In Fig. 2 the semantics of the language is defined as the satisfaction relation $\models$ between the states of a model and BG_n formulas by induction on the structure of formulas. We note here that B_i satisfies the axioms of the modal logic **KD45** and

$(M,s) \models \mathbf{true}$		for any state s.
$(M,s) \models p$	iff	$p \in L(s)$ (where $p \in \mathcal{P}$).
$(M,s) \models \neg F$	iff	$(M,s) \not\models F$
$(M,s) \models F \wedge H$	iff	$(M,s) \models F$ and $(M,s) \models H$
$(M,s) \models F \vee H$	iff	$(M,s) \models F$ or $(M,s) \models H$
$(M,s) \models O_i F$	iff	$\forall t \in S$, if $(s,t) \in O_i$ then $(M,t) \models F$

Fig. 2. Semantics of BG_n.

G_i satisfies the axioms of the modal logic **KD**. These axioms for $O \in \{B, G\}$ are:

K: $\vdash O_i(F \Rightarrow H) \Rightarrow (O_i F \Rightarrow O_i H)$ **4**: $\vdash B_i F \Rightarrow B_i B_i F$
D: $\vdash O_i F \Rightarrow \neg O_i \neg F$ **5**: $\vdash \neg B_i F \Rightarrow B_i \neg B_i F$

4 A Normal Form for Formulas of BG_n

We first transform formulas of BG_n to a normal form called NF_{BG}. For this purpose we introduce a symbol **start** such that $(M, s_0) \models \mathbf{start}$ for any initial state s_0. Formulas in NF_{BG} are of the general form

$$\bigwedge_i v_i : C_i$$

where $v_i \in \mathbf{V}$ is a view and C_i is a *clause*. Clauses are of the following form:

$$\mathbf{start} \Rightarrow \bigvee_{a=1}^{r} l_a \quad \text{(an initial clause)}\ , \quad \mathbf{true} \Rightarrow \bigvee_{a=1}^{r} m_{Bi_a} \quad \text{(a } B_i \text{ clause)}$$
$$\mathbf{true} \Rightarrow \bigvee_{a=1}^{r} l_a \quad \text{(a literal clause)}\ , \quad \mathbf{true} \Rightarrow \bigvee_{a=1}^{r} m_{Gi_a} \quad \text{(a } G_i \text{ clause)}$$

Here l_a are literals, m_{Bi_a} are either literals or simple modal literals involving the B_i modality, and m_{Gi_a} are either literals or simple modal literals involving the G_i modality. For convenience the conjunction is dropped and we consider just the set of clauses of the form $v_i : C_i$.

4.1 Translation to Normal Form

Before the translation to normal form we replace formulas of the form $B_i B_i F$ and $B_i \neg B_i F$ by $B_i F$ and $\neg B_i F$ respectively. The translation to normal form requires a number of propositional variables $x,\ y, \ldots$ proportional to the number of modal operators and propositional connectives in the formula. In this section we define the process of translation of arbitrary BG_n formulas to the set of clauses in normal form. Consider a formula F of BG_n. The translation will be done in two steps by applying transformations τ_0 and τ_1 as described below (f is a new propositional variable).

$$\tau_0[F] \longrightarrow (\ \epsilon : \mathbf{start} \Rightarrow f\ ) \wedge \tau_1[\ \epsilon :\ f \Rightarrow F\ ]. \qquad (1)$$

ϵ denotes the initial view in the tree of clauses. Next, we define τ_1 as follows, assuming x is a proposition. If the main operator on the right side of the implication is either $\wedge$ or $\neg$ we remove it as follows.

$$\begin{array}{l}
\tau_1[v : x \Rightarrow (F \wedge H)] \longrightarrow \tau_1[v : x \Rightarrow F] \wedge \tau_1[v : x \Rightarrow H] \\
\tau_1[v : x \Rightarrow \neg(F \wedge H)] \longrightarrow \tau_1[v : x \Rightarrow (\neg F \vee \neg H)] \\
\tau_1[v : x \Rightarrow \neg(F \vee H)] \longrightarrow \tau_1[v : x \Rightarrow \neg F] \wedge \tau_1[v : x \Rightarrow \neg H] \\
\tau_1[v : x \Rightarrow \neg\neg F] \longrightarrow \tau_1[v : x \Rightarrow F].
\end{array}$$

Complex sub-formulas that appear within the scope of any modal operator, are transformed as follows (where y is a new proposition, $O_i \in \{B_i, G_i\}$, and F is not a literal).

$$\begin{array}{l}
\tau_1[v : x \Rightarrow O_i F] \longrightarrow \tau_1[v : x \Rightarrow O_i y] \wedge \tau_1[vO_i : y \Rightarrow F] \\
\tau_1[v : x \Rightarrow \neg O_i F] \longrightarrow \tau_1[v : x \Rightarrow \neg O_i \neg y] \wedge \tau_1[vO_i : y \Rightarrow \neg F]
\end{array}$$

Next, we use renaming on formulas whose right hand side has disjunction as its main operator but may not be in the correct form (y is a new proposition and D is a disjunction of formulas which are not necessarily in the normal form).

$\tau_1[v : x \Rightarrow D \vee F] \longrightarrow \tau_1[v : x \Rightarrow D \vee y] \wedge \tau_1[v : y \Rightarrow F]$
where F is neither a literal nor a simple modal literal, nor a disjunction of literals and simple modal literals. (For example F could be a conjunction of formulas)
$\tau_1[v : x \Rightarrow D \vee O_i F] \longrightarrow \tau_1[v : x \Rightarrow D \vee y] \wedge \tau_1[v : y \Rightarrow O_i F]$,
where D contains a disjunct of the form O'_j or $\neg O'_j$ and $O \neq O'$ or $i \neq j$.
$\tau_1[v : x \Rightarrow D \vee \neg O_i F] \longrightarrow \tau_1[v : x \Rightarrow D \vee y] \wedge \tau_1[v : y \Rightarrow \neg O_i F]$,
where D contains a disjunct of the form O'_j or $\neg O'_j$ and $O \neq O'$ or $i \neq j$.
$\tau_1[v : x \Rightarrow D \vee O_i F] \longrightarrow \tau_1[v : x \Rightarrow D \vee O_i y] \wedge \tau_1[vO_i : y \Rightarrow F]$,
where F is not a literal and D contains only the modality O_i.
$\tau_1[v : x \Rightarrow D \vee \neg O_i F] \longrightarrow \tau_1[v : x \Rightarrow D \vee \neg O_i \neg y] \wedge \tau_1[vO_i : y \Rightarrow \neg F]$,
where F is not a literal and D contains only the modality O_i.

According to the definition of NF_{BG}, each modal clause may contain simple modal literals involving only one modal operator. Thus clause $\textbf{true} \Rightarrow B_1 x \vee y \vee \neg B_1 z$ is allowed, but $\textbf{true} \Rightarrow B_1 x \vee y \vee B_2 z$ and $\textbf{true} \Rightarrow B_1 x \vee y \vee G_1 z$ are not allowed, as they contain more than one modality. The above transformations will make sure that each modal clause contains only one modal operator on the right hand side. Finally we transform the formulas whose right hand side is a disjunction of literals or simple modal literals of the same type: (D is a disjunction of literals and simple modal literals only involving one modal operator.)

$$\tau_1[v : x \Rightarrow D] \longrightarrow v : \textbf{true} \Rightarrow \neg x \vee D$$

After the above transformations we will have a set of clauses in NF_{BG} normal form. Figure 3 shows the different steps in the transformation of the formula $F = B_i(p \vee \neg B_j(q \vee \neg t))$ into normal form.

Proposition 1. *τ_0 transforms every $\varphi \in WFF_{BG}$ into normal form in $O(m+p)$ steps with an extra $O(m+p)$ new propositional variables, where m is the number of modal operators and p is the number of propositional connectives.*

$$\begin{tikzcd}[column sep=tiny]
 & \tau_0[B_i(p \vee \neg B_j(q \vee \neg t))] = (\epsilon : \mathbf{start} \Rightarrow f) \wedge \tau_1[\epsilon : f \Rightarrow B_i(p \vee \neg B_j(q \vee \neg t))] \arrow[d] & \\
 & \tau_1[\epsilon : f \Rightarrow B_i x_1] \wedge \tau_1[B_i : x_1 \Rightarrow p \vee \neg B_j(q \vee \neg t))] \arrow[dl] \arrow[d] & \\
\epsilon : \mathbf{true} \Rightarrow \neg f \vee B_i x_1 & \tau_1[B_i : x_1 \Rightarrow p \vee \neg B_j \neg x_2] \wedge \tau_1[B_i B_j : x_2 \Rightarrow \neg(q \vee \neg t)] \arrow[d] \arrow[dr] & \\
 & B_i : \mathbf{true} \Rightarrow \neg x_1 \vee p \vee \neg B_j \neg x_2 & \tau_1[B_i B_j : x_2 \Rightarrow \neg q \wedge t] \arrow[d] \\
 & & \tau_1[B_i B_j : x_2 \Rightarrow \neg q] \wedge \tau_1[B_i B_j : x_2 \Rightarrow t] \arrow[dl] \arrow[d] \\
 & B_i B_j : \mathbf{true} \Rightarrow \neg x_2 \vee \neg q & B_i B_j : \mathbf{true} \Rightarrow \neg x_2 \vee t
\end{tikzcd}$$

Fig. 3. Transformation of $F = B_i(p \vee \neg B_j(q \vee \neg t))$ into normal form

5 Resolution for NF_{BG} Normal Form Formulas

In this section we define the resolution rules for inferring a formula from the information store. Assuming F and H are disjunctions of literals, the initial rules are:

$$\text{[IRES1]} \quad \frac{\begin{array}{l}\epsilon : \mathbf{true} \Rightarrow (F \vee l) \\ \epsilon : \mathbf{start} \Rightarrow (H \vee \neg l)\end{array}}{\epsilon : \mathbf{start} \Rightarrow (F \vee H)} \qquad \text{[IRES2]} \quad \frac{\begin{array}{l}\epsilon : \mathbf{start} \Rightarrow (F \vee l) \\ \epsilon : \mathbf{start} \Rightarrow (H \vee \neg l)\end{array}}{\epsilon : \mathbf{start} \Rightarrow (F \vee H)}$$

Next we define modal resolution rules which are used to resolve two simple modal literals in the same view (MRES1, MRES2), or to resolve two clauses in adjacent views (MRES3, MRES4).

$$\text{[MRES1]} \quad \frac{\begin{array}{l}v : \mathbf{true} \Rightarrow D \vee m \\ v : \mathbf{true} \Rightarrow D' \vee \neg m\end{array}}{v : \mathbf{true} \Rightarrow D \vee D'} \qquad \text{[MRES3]} \quad \frac{\begin{array}{l}v : \mathbf{true} \Rightarrow D \vee \neg O_i l \\ vO_i : \mathbf{true} \Rightarrow D' \vee l\end{array}}{v : \mathbf{true} \Rightarrow D \vee mod_{O_i}(D')}$$

$$\text{[MRES2]} \quad \frac{\begin{array}{l}v : \mathbf{true} \Rightarrow D \vee O_i l \\ v : \mathbf{true} \Rightarrow D' \vee O_i \neg l\end{array}}{v : \mathbf{true} \Rightarrow D \vee D'} \qquad \text{[MRES4]} \quad \frac{\begin{array}{l}v : \mathbf{true} \Rightarrow D \vee O_i l \\ vO_i : \mathbf{true} \Rightarrow D' \vee \neg l\end{array}}{v : \mathbf{true} \Rightarrow D \vee mod_{O_i}(D')}$$

where $mod_{O_i}(D')$ is defined below. In MRES1 and MRES2, D and D' have the same kind of modal operators, i.e. if D has a simple modal literal $O_i l$ then all other simple modal literals of D and all simple modal literals of D' must involve O_i only. In MRES3 and MRES4 if D has belief modality operator, say B_i, then D' must not have any simple modal literals involving O_j such that $O_j \neq B_i$, i.e. all the simple modal literals of D and D' must have B_i only, otherwise we may obtain a resolvent containing a modal literal which has a nesting of two modal operators like $B_i O_j l'$ which is not in the normal form. In this case, if D' has a simple modal literal $O_j l'$ $(O_j \neq B_i)$, it must be resolved with another clause already. In MRES3 and MRES4 if $O_i = G_i$, then D' must be a disjunction of literals only, to avoid the problem of nested modal operators.

Definition 3. The function $mod_{O_i}(D)$, is defined on the disjunction of literals or simple modal literals as follows: $(O_i \in \{B_i, G_i\})$

$$mod_{O_i}(l) = \neg O_i \neg l \quad , \quad mod_{B_i}(B_i l) = B_i l$$
$$mod_{O_i}(F \vee H) = mod_{O_i}(F) \vee mod_{O_i}(H) \quad , \quad mod_{B_i}(\neg B_i l) = \neg B_i l$$

Note that $mod_{B_i}(O_j l)$ where $O_j \neq B_i$ and $mod_{G_i}(O_j l)$ are not defined, as these cases will not occur in the resolution process. Let us justify MRES3, assuming $O_i = B_i$; the same argument holds for G_i. The first clause $v : \mathbf{true} \Rightarrow D \vee \neg B_i l$ is from view v, and the second clause $vB_i : \mathbf{true} \Rightarrow D' \vee l$ is from view vB_i. In resolution rule MRES3, the second clause can be written as $vB_i : \neg D' \Rightarrow l$ and after distributing B_i it will be $v : B_i(\neg D' \Rightarrow l)$, which implies $v : B_i \neg D' \Rightarrow B_i l$. As D' is a disjunction of simple modal literals involving only B_i, i.e., $D' = m_1 \vee ... \vee m_k$ then $\neg D' = \neg m_1 \wedge ... \wedge \neg m_k$, and so $B_i \neg D' = B_i \neg m_1 \wedge ... \wedge B_i \neg m_k$. Finally we obtain the clause $v : \mathbf{true} \Rightarrow \neg B_i \neg m_1 \vee ... \vee \neg B_i \neg m_k \vee B_i l$. Now we can resolve two clauses $v : \mathbf{true} \Rightarrow D \vee \neg B_i l$ and $v : \mathbf{true} \Rightarrow \neg B_i \neg m_1 \vee ... \neg B_i \neg m_k \vee B_i l$, which will yield a new clause $v : \mathbf{true} \Rightarrow D \vee \neg B_i \neg m_1 \vee ... \neg B_i \neg m_k$. If m_i is a simple modal literal then according to a theorem of the logic KD45 [6] which says $\neg B_i \neg B_i \neg F \Leftrightarrow B_i \neg F$, we can remove $\neg B_i \neg$ from the simple modal literals and if $m_i = l'$ it will remain in the form $\neg B_i \neg l'$. In the case of goals, D' is just a disjunction of literals, that is because we don't have the equivalence $\neg G_i \neg G_i \neg F \Leftrightarrow G_i \neg F$ in the logic KD.

Example. Suppose agent i has the **belief base:** $B_i(\neg p \vee B_j q)$, $B_i B_j \neg q$. The question is, whether $B_i \neg p$ is implied by the belief base. We add $\neg B_i \neg p$ to the belief base and check if the resolution process results in the clause $\epsilon : \mathbf{start} \Rightarrow \mathbf{false}$ (see Fig. 4).

Clauses :

$B_i(\neg p \vee B_j q)$	$B_i B_j \neg q$	$\neg B_i \neg p$
1. $\epsilon : \mathbf{start} \Rightarrow f$	4. $\epsilon : \mathbf{start} \Rightarrow f$	7. $\epsilon : \mathbf{start} \Rightarrow f$
2. $\epsilon : \mathbf{true} \Rightarrow \neg f \vee B_i x_1$	5. $\epsilon : \mathbf{true} \Rightarrow \neg f \vee B_i y_1$	8. $\epsilon : \mathbf{true} \Rightarrow \neg f \vee \neg B_i \neg p$
3. $B_i : \mathbf{true} \Rightarrow \neg x_1 \vee \neg p \vee B_j q$	6. $B_i : \mathbf{true} \Rightarrow \neg y_1 \vee B_j \neg q$	

Resolution :

$\overset{3}{\underset{6}{\xrightarrow{\text{MRES2}}}}$ 9. $B_i : \mathbf{true} \Rightarrow \neg x_1 \vee \neg p \vee \neg y_1$ $\overset{8}{\xrightarrow{\text{MRES3}}}$ 10. $\epsilon : \mathbf{true} \Rightarrow \neg f \vee \neg B_i x_1 \vee \neg B_i y_1$

$\overset{2}{\underset{10}{\xrightarrow{\text{MRES1}}}}$ 11. $\epsilon : \mathbf{true} \Rightarrow \neg f \vee \neg B_i y_1$ $\overset{5}{\xrightarrow{\text{MRES1}}}$ 12. $\epsilon : \mathbf{true} \Rightarrow \neg f$ $\overset{1}{\xrightarrow{\text{IRES1}}}$ $\epsilon : \mathbf{start} \Rightarrow \mathbf{false}$

Fig. 4. Clausal form and resolution process of belief base of the example.

6 Soundness and Completeness

We will prove that the transformation into NF_{BG} preserves satisfiability. Assume $M = \langle S, L, S_0, B_1, \ldots, B_n, G_1, \ldots, G_n \rangle$ is a Kripke structure. We say s' is accessible from s via relation O_i if $(s, s') \in O_i$. Moreover if s'' is accessible from s' via v' and s' is accessible from s via v, then s'' is accessible from s via vv' where vv' is concatenation of v and v'. If state s is accessible from an initial state s_0 via v, we say s is at **level** v of the Kripke structure M. We say $M, s \models v : F$ iff for any state s' accessible from s via v, $M, s' \models F$. The following proposition shows that the transformation τ_0 preserves satisfiability and unsatisfiability.

Proposition 2. *Assume M is a Kripke structure and s_0 is an initial state,*
1. $(M, s_0) \models \tau_1[v : x \Rightarrow F]$ *implies* $(M, s_0) \models (v : x \Rightarrow F)$.
2. $(M, s_0) \models \tau_0[F]$ *implies* $(M, s_0) \models F$.
3. *If there is a model M, such that $(M, s_0) \models \epsilon : x \Rightarrow F$, then there is a model M' s.t. $(M', s'_0) \models \tau_1[\epsilon : x \Rightarrow F]$.*
4. *For any model M of F, there exists a model M_0 of $\tau_0[F]$.*

From the above proposition we have:

Theorem 1. A BG_n formula A is satisfiable if and only if $\tau_0[A]$ is satisfiable.

Theorem 2 (Soundness). Let T be a set of NF_{BG} clauses. Let the clause set R be obtained from T by applying one of the resolution rules. Then T is satisfiable if and only if R is satisfiable.

Sketch of the proof. We prove the above theorem by considering any rule and assuming its premises are satisfiable, then we prove its conclusion (or resolvent) is also satisfiable. For reverse direction, if T is unsatisfiable then after adding the new clause to obtain R, still R is unsatisfiable. $\square$

Theorem 3 (Termination). The resolution process (repeated applications of the rules in section 5) in a set of NF_{BG} clauses always terminates.

Sketch of the proof. As the resolution rules don't create new views, so the resolution process terminates after some steps, because there are a finite number of propositions, views and modal operators. $\square$

The completeness proof is based on the construction of a behavior graph [7, 6]. We construct a graph of NF_{BG} clauses which has belief and goal relations for any agent i. We will show that the set of resolution rules presented here is complete, and there is a refutation by resolution if the set of clauses is unsatisfiable. Note that we use the word 'view' when we refer to a subset of clauses (we say clauses of view v) and we use the word 'level' when we refer to some subset of states in the graph (we say states of level v).

Definition 4. The depth of a modal literal F is the number of modal operators applied to a literal. Depth of literal l or its negation $\neg l$ is 0. Depth of O_iF or $\neg O_iF$ is $1 + \mathit{depth}(F)$.

Definition 5. Given a set of NF_{BG} clauses, the view $v = O_{i_1} \dots O_{i_k}$ is called a *deepest view* if there are clauses in view v, but no clause in view $vO_{i_{k+1}}$ for any $O \in \{B, G\}$ and $i_1, \dots, i_{k+1} \in Ag$.

It is possible to have more than one deepest view in a set of clauses.

Definition 6. Let $C = v : \mathbf{true} \Rightarrow \phi$ be a NF_{BG} clause, we define $lset(C) = \{l, \neg l \mid l \textit{ is an atomic proposition in } \phi\}$ and $mset(C) = \{m, \neg m \mid m \textit{ is a simple modal literal in } \phi\}$. If v is a view and $C_1, \dots, C_n$ are all of the clauses of the form $v : \mathbf{true} \Rightarrow \phi$ then we define $cl(v) = \{C_1, \dots, C_n\}$ which is the set of clauses contained in view v. Moreover we define $lset(v) = lset(C_1) \cup \dots \cup lset(C_n)$ and $mset(v) = mset(C_1) \cup \dots \cup mset(C_n)$.

For example if $C_1 \equiv B_j : \mathbf{true} \Rightarrow \neg x \vee B_i p \vee q$ and $C_2 \equiv B_j : \mathbf{true} \Rightarrow y \vee B_k t \vee p$ then $lset(C_1) = \{x, p, q, \neg x, \neg p, \neg q\}$ and $mset(C) = \{B_i p, \neg B_i p\}$. If view B_j contains only clauses C_1 and C_2, then $lset(B_j) = \{x, p, q, y, t, \neg x, \neg p, \neg q, \neg y, \neg t\}$, and $mset(B_j) = \{B_i p, B_k t, \neg B_i p, \neg B_k t\}$.

Graph construction Assume as before $Ag = \{1, ..., n\}$ is a set of agents and T is a set of clauses. For any set $S = \{f_1, \dots, f_n\}$ of modal literals (cf. Def. 1), $i \in Ag$ and $O \in \{B, G\}$ **Application** of O_i to S is represented as $O_i.S$ and defined as $O_i.S = \{O_i f_1, \dots, O_i f_n\}$. We start with the clauses in the deepest views. Assume $v = O_1 \dots O_k$ is a deepest view. Let $\Delta_k = lset(v) \cup mset(v)$. We take $\Delta_{k-1} = lset(O_1 \dots O_{k-1}) \cup mset(O_1 \dots O_{k-1}) \cup O_k.\Delta_k \cup \{\neg f \mid f \in O_k.\Delta_k\}$. Now we will do the same process for obtaining elements of Δ_{k-2} ($\Delta_{k-2} = lset(O_1 \dots O_{k-2}) \cup mset(O_1 \dots O_{k-2}) \cup O_{k-1}.\Delta_{k-1} \cup \{\neg f \mid f \in O_{k-1}.\Delta_{k-1}\}$). We repeat the same process till we get the set Δ_0. We define $\Delta_v = \Delta_0 \cup \dots \cup \Delta_k$.[3] Now consider all other deepest views v' and do the same for v' to obtain other sets $\Delta_{v'}$. Finally we define $\Delta = \bigcup_v \Delta_v$ where v is a deepest view.

Definition 7. Let F and H be modal literals. We define the relation $\mathbf{F} \Rightarrow \mathbf{H}$ as: $F \Rightarrow F$, $O_i F \Rightarrow O_i H$ iff $F \Rightarrow H$, $O_i F \Rightarrow \neg O_i \neg H$ iff $F \Rightarrow H$. Moreover a pair of formulas is **complementary** if they are of one of the following forms (assume $F \Rightarrow H$): F and $\neg F$, $O_i F$ and $\neg O_i H$, $O_i F$ and $O_i \neg H$.

For example $B_i p$ and $B_i \neg p$, $G_i G_j p$ and $G_i \neg G_j p$, $B_i G_j B_k p$ and $\neg B_i \neg G_j \neg B_k p$ are all complementary pairs.

Graph $G = (S, B_1, ...B_n,\ G_1, ..., G_n)$ is constructed as follows. The set of states S is constructed by considering all possible maximal subsets of Δ (which is defined earlier as $\bigcup_v \Delta_v$) which are consistent. δ is a maximal consistent subset of Δ if we can not add any more element from Δ to δ, otherwise it will be inconsistent. δ is consistent if it doesn't have a complementary pair.

Definition 8. For each maximal consistent subset δ of Δ, we will have a corresponding state $s \in S$ and will say δ is the **label** of s, and we write $label(s) = \delta$.

[3] Note that Δ_v has modal literals of depth at most $k + 1$.

Let us consider a simple example. Consider only one clause B_j : **true** $\Rightarrow p$. Then Δ includes $\{p, \neg p, B_j p, B_j \neg p, \neg B_j p, \neg B_j \neg p\}$ and it has six maximal subsets:

$$S = \{\ \{p, B_j p, \neg B_j \neg p\},\ \{p, B_j \neg p, \neg B_j p\},\ \{p, \neg B_j p, \neg B_j \neg p\},$$
$$\{\neg p, B_j p, \neg B_j \neg p\},\ \{\neg p, B_j \neg p, \neg B_j p\},\ \{\neg p, \neg B_j p, \neg B_j \neg p\}\ \}$$

(Note that there are exactly some clauses in the view ϵ but we haven't considered them in this example.) So far we have considered all the possible states of a Kripke structure. We must check which states satisfy the clauses in different views. Let $C = vO_i$: **true** $\Rightarrow F$ be a clause, where $F = f_1 \vee \cdots \vee f_n$ and each f_i is a modal literal.

1. We move all but one of the disjuncts of F to the left of $\Rightarrow$ in clause C. Without loss of generality assume f_1 remains in the right hand side. Thus: $C = vO_i : \neg f_2 \wedge \cdots \wedge \neg f_n \Rightarrow f_1$.
2. We apply O_i to the clause and we obtain $C = v : O_i(\neg f_2 \wedge \cdots \wedge \neg f_n \Rightarrow f_1)$
3. Based on axiom **K** we have $v : O_i(\neg f_2 \wedge \cdots \wedge \neg f_n) \Rightarrow O_i f_1$
4. This in turn implies $v : O_i \neg f_2 \wedge \cdots \wedge O_i \neg f_n \Rightarrow O_i f_1$.
5. We again move formulas from left of $\Rightarrow$ to its right side, v : **true** $\Rightarrow O_i f_1 \vee \neg O_i \neg f_2 \vee \cdots \vee \neg O_i \neg f_n$.

The clause of step 5 is called a **pushed** clause (we have pushed O_i into clause) and it is a pushed clause in the view v. If $O_i = B_i$ and $f_j = B_i g_j$, with $j \neq 1$, then from the equivalence $\neg B_i \neg B_i f \Leftrightarrow B_i f$ of the logic KD45 we obtain $\neg B_i \neg B_i g_j = B_i g_j = f_j$ (so $\neg B_i \neg f_j = f_j$). But if $f_j = l$ is a literal, then it will remain $\neg B_i \neg f_j$. Similarly for $B_i f_1$. The reader can see that if in step 1 we keep f_j, $j \neq 1$, on the right side we obtain another pushed clause. In summary we have the following definition.

Definition 9. Let $C = vO_i$: **true** $\Rightarrow F$ be a clause of view vO_i. We define $C^{\rightarrow} = \{v$: **true** $\Rightarrow F'\}$ to be a set of clauses obtained after **pushing** O_i to clause C. F' is obtained from O_i and F using the above algorithm. If vO_i is a view such that $S = cl(vO_i) = \{C_1, \ldots, C_n\}$, then $S^{\rightarrow} = C_1^{\rightarrow} \cup \cdots \cup C_n^{\rightarrow}$ is the set of pushed clauses (in the view v) after pushing O_i.

For example suppose $C = vO_k$: **true** $\Rightarrow l$. Then $C^{\rightarrow} = \{v$: **true** $\Rightarrow O_k l\}$ has only one element. If $C = vO_k$: **true** $\Rightarrow l_1 \vee l_2$. Then $C^{\rightarrow} = \{v$: **true** $\Rightarrow O_k l_1 \vee \neg O_k \neg l_2$, v : **true** $\Rightarrow \neg O_k \neg l_1 \vee O_k l_2\}$. Consider $C = vB_i$: **true** $\Rightarrow l_1 \vee l_2 \vee B_i l_3$, then $C^{\rightarrow} = \{v$: **true** $\Rightarrow B_i l_1 \vee \neg B_i \neg l_2 \vee B_i l_3$, v : **true** $\Rightarrow \neg B_i \neg l_1 \vee B_i l_2 \vee B_i l_3\}$. Here v : **true** $\Rightarrow \neg B_i \neg l_1 \vee \neg B_i \neg l_2 \vee B_i l_3$ is a pushed clause also, but we may ignore it as it is implied by the first and second clauses. As a final example suppose $C = vG_i$: **true** $\Rightarrow l_1 \vee B_i l_2$, then $C^{\rightarrow} = \{v$: **true** $\Rightarrow G_i l_1 \vee \neg G_i \neg B_i l_2$, v : **true** $\Rightarrow \neg G_i \neg l_1 \vee G_i B_i l_2\}$.

Definition 10. Let $\omega_n = O_1 \ldots O_n$ be a sequence of n modal operators ($|\omega_n| = n$). Let $v\omega_n$ be a view with the set of clauses $S = cl(v\omega_n)$ then $S^{\rightarrow_n} = S^{\rightarrow}(\ldots(S^{\rightarrow}))$ where $^{\rightarrow}$ is applied n times, is a set of pushed clauses of the form v : **true** $\Rightarrow \phi$. This intuitively means all modal operators of ω_n consecutively are pushed to clauses of S. Generally if $\lambda_v = \{v\omega \mid v\omega \textit{ is a view with a nonempty set of clauses}\}$ is a set of all nonempty views which include subview v, then we define the **entire set of pushed clauses of** v as $pcl(v) = \bigcup_{v\omega \in \lambda_v} cl(v\omega)^{\rightarrow|\omega|}$. Intuitively $pcl(v)$ contains all pushed clauses which are in view v.

Now we go back to the graph and find the states which satisfy clauses of view v for any v. We assign v to state $s \in S$ if $label(s) \models cl(v) \wedge pcl(v)$, i.e. s can be in level v (it is accessible from one of the initial states via relation v) of the graph if it satisfies the clauses and pushed clauses of v. If state s is assigned more than one level (for example s is assigned v and v') then we make one copy of s for each combination of these levels and we assign that combination to the labeling of the corresponding copy of s. For example suppose s is assigned v and v', then we consider four copies of s as: s^1, s^2, s^3, s^4, which are assigned by $\{v\}$, $\{v'\}$, $\{v, v'\}$, and $\{\}$ respectively. The reason for this will become clear in the proof of theorem 4. For a sketch intuition behind this, assume $s \in S$ belongs to levels v_1 and v_2. Assume there is a state $t \in S$ which belongs to level v_1B_i but is not a member of level v_2B_i. As we will discuss below this means we can not make a B_i transition from s to t as t is not in level v_2B_i, although t is in level v_1B_i. For solving this problem we make various copies of s with different levels assigned to them. For example the copy of s which is assigned only by v_1 has a transition to t. Finally we define level(v) to be the set of all states assigned v as $\textbf{level}(\mathbf{v}) = \{s \in S \mid s \textit{ is } \textbf{assigned } v\}$.

Definition 11. For any agent i and set of modal literals X, $O_i\_set(X) = \{\ F \mid O_iF \in X\}$.

Now the set S of states is ready. The initial states are $\{s \in S \mid s \in level(\epsilon) \textit{ and } label(s) \models f\}$ where f is defined in the transformation process. We will find the accessibility relations B_i and G_i for any agent i. In the behavior graph we show each relation by edges between states labeled by the name of the relation. We add an edge from s to s' labeled by B_i iff the three following conditions hold:

a. If $V = \{v_1, \ldots, v_k\}$ is the set of levels assigned to s, then $VB = \{v_1B_i, \ldots, v_kB_i\}$ is the set of levels assigned to s' s.t. if vB_i is a view with an empty set of clauses, then vB_i is omitted from VB. Also $v_jB_iB_i = v_jB_i$.

b. $label(s') \models B_i\_set(label(s))$ which means if $label(s) \models B_iF$ then $label(s') \models F$ for any F.

c. $B_iF \in label(s)$ iff $B_iF \in label(s')$ and $\neg B_iF \in label(s)$ iff $\neg B_iF \in label(s')$, which means s and s' have the same set of beliefs involving B_i. This rule guarantees B_i to be euclidean and transitive.

To find G_i relations for state s we will find all states s' which satisfy only conditions **a.** and **b.** replacing B_i with G_i. Now we will delete those states which can not be a state in any model. If v is a view with a nonempty set of clauses, but $level(v)$ is empty, i.e. $\neg\exists s \in S : label(s) \models cl(v) \wedge pcl(v)$, then the set of clauses does not have any model. In this case we will delete all the states of graph, and we say graph is **empty**. Otherwise for any v with a nonempty set of clauses, $level(v) \neq \emptyset$. Now the graph is constructed. We can show the relations B_i are *serial, transitive, and euclidean* and the relations G_i are *serial*.

Proposition 3. *1. The relations G_i in the behavior graph are serial.*
2. The relations B_i in the behavior graph are serial, transitive and euclidean.

We could also prove the following lemma to ensure consistency between adjacent levels.

Lemma 1. Let T be a set of NF_{BG} clauses, and G be the behavior graph constructed by the above process. For any node s of the graph, if $\neg O_i f \in label(s)$ then there is a node s', s.t. $(s, s') \in O_i$ and $\neg f \in label(s')$.

The Above lemma and proposition show that the constructed graph is a Kripke structure for the set of clauses. But there is a point which must be cleared here. In the construction process of the graph, for each state of the graph in level v, we checked if it satisfies clauses of view v ($cl(v)$) and pushed clauses of view v ($pcl(v)$). The following lemma shows it is not possible that the set of original normal form clauses to be satisfiable while the set of clauses obtained after pushing the modalities is unsatisfiable.

Lemma 2. Let T be a set of clauses including a clause C in view vO_i. Let $R = T \cup C^{\rightarrow}$ be the set of clauses of T and the pushed clauses obtained from C. T is satisfiable if and only if R is satisfiable.

This lemma shows that pushing modalities into clauses preserves satisfiability. Finally we can prove that, for an unsatisfiable set of clauses, the constructed graph is empty, and thus there is no model.

Theorem 4. The set of clauses T is unsatisfiable iff its behavior graph G is empty.

Now we can prove the completeness of the method. The resolution rules are complete if they can detect the emptiness of the graph. The graph is empty if some level v (with nonempty set of clauses in view v) is empty. A level v is empty if the clauses and pushed clauses of view v imply **false**. In the following we will prove that our resolution calculus is complete.

Before proving the next theorem we will define two new resolution rules and later we will prove that they can be eliminated. Assume F and H are modal literals then resolution rules MRESC1 and MRESC2 are defined as:

$$[\text{MRESC1}]\ \frac{\begin{array}{l} v : \textbf{true} \Rightarrow D \vee O_i F \\ v : \textbf{true} \Rightarrow D' \vee O_i H \end{array}}{v : \textbf{true} \Rightarrow D \vee D'} \qquad [\text{MRESC2}]\ \frac{\begin{array}{l} v : \textbf{true} \Rightarrow D \vee O_i F \\ v : \textbf{true} \Rightarrow D' \vee \neg O_i H \end{array}}{v : \textbf{true} \Rightarrow D \vee D'}$$

where in MRESC1, F and H are complementary, and in MRESC2, $F \Rightarrow H$

Theorem 5 (Completeness). Let T be a set of clauses and their pushed clauses. Then T is unsatisfiable iff there is a refutation by resolution rules IRES1, IRES2, MRES1, MRES2, MRESC1 and MRESC2.

Next we will prove that rules MRESC1 and MRESC2 are not necessary.

Definition 12. If F, F_1 and F_2 are modal literals and $O_i \in \{B_i, G_i\}$ then *Rev* is defined as:
1. $Rev(\neg O_i F) = \neg F$ 2. $Rev(O_i F) = F$ 3. $Rev(B_i l) = B_i l$
4. $Rev(\neg B_i l) = \neg B_i l$ 5. $Rev(F_1 \vee F_2) = Rev(F_1) \vee Rev(F_2)$.

Relation $Rev(F)$ is the reverse of pushing modal operators ($C^{\rightarrow}$) into clauses. Any pushed clause P in view v has a corresponding original clause C in view $v\omega$ s.t. P is obtained by pushing modalities of ω into C. Relation *Rev* takes P and computes C. Note that for modal literals $B_i l$ and $\neg B_i l$ there might be two reverses (depending on the other disjunct). For example, if we have clause $v : B_i B_j p \vee \neg B_i q$ then its reverse can be either $vB_i : B_j p \vee \neg B_i q$ or $vB_i : B_j p \vee \neg q$, but the first one is not possible as it has two different modal operators and second one is the correct reverse. Using the relation *Rev*, we can prove the following lemma which completes the proof of completeness.

Lemma 3. If two clauses of view v can be resolved with resolution rules MRESC1 and MRESC2, then their corresponding original clauses can be resolved with resolution rules MRES1, MRES2, MRES3 and MRES4.

7 Conclusion and Future Work

In this paper we have defined a framework for belief and goal bases and a resolution based proof method for reasoning about them. We have also proved the soundness, termination and completeness of the method.

There do exist tableau based methods for various modal logics in the literature (notably [12, 17, 3]). For certain modal logics such as S4, S5 and T resolution methods exist [8]. Our method closely follows that of [6, 7]. However we have advanced their work to include an additional **KD** modality while dropping the temporal operators.

Our motivation however is not just to provide a proof system but instead to tackle the problem of revision of an information store organized hierarchically. The main feature of our method is the "locality" property enjoyed by our rules. We have shown that it is necessary to consider complementary pairs of clauses only at the same or between adjacent levels. This we believe considerably simplifies the tasks of belief and goal revision in order to keep the information store consistent on fresh inputs. Secondly, it is no longer necessary to translate the formulas into classical logic as is recommended by some authors [18, 22, 19, 9]. However, even though we have combined the logics of **KD45** and **KD**, we have not defined any interactions between them as it gets complicated to manage using resolution rules. This is a subject of future research.

The idea of hierarchical structure for information store is taken (in some sense) from Benerecetti et.al. [2]. More details of hierarchical structures and the proposed logic can be found in [11, 10].

Acknowledgment We thank the anonymous referees for some very insightful comments.

References

1. C. Alchourron, P. Gardenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, 1985.
2. M. Benerecetti, F. Giunchiglia, and L. Serafini. Model checking multi-agent systems. *Journal of Logic and Computation*, 8(3):401–424, 1998.
3. B. Bennett, C. Dixon, M. Fisher, U. Hustadt, E. Franconi, I. Horrocks, and M. De Rijke. Combinations of modal logics. *Artif. Intell. Rev.*, 17(1):1–20, 2002.
4. Brian F. Chellas. *Modal Logic: An Introduction.* Cambridge University Press, 1980.
5. P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.
6. Clare Dixon, Michael Fisher, and Alexander Bolotov. Clausal resolution in a logic of rational agency. *Artif. Intell.*, 139(1):47–89, 2002.
7. M. Fisher, C. Dixon, and M. Peim. Clausal temporal resolution. *ACM Trans. Coput. Logic*, 2(1):12–56, 2001.
8. M. Fitting. Proof methods for modal and intuitionistic logics. In *volume 169 of Synthese Library. D. Reidel, Dordrecht.* 1983.
9. A. M. Frisch and R. B. Scherl. A general framework for modal deduction. In *proccedings of KR*, pages 196–207. Morgan Kaufmann, 1991.
10. C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality+compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.
11. F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics or: How we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, 1994.
12. R. Gore. Tableau methods for modal and temporal logics. In M. D'Agostino, D. Gabbay, R. Haehnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.
13. Sven Ove Hansson. *A Textbook of Belief Dynamics.* Kluwer Academic Press, 1999.
14. A. Herzig and D. Longin. A logic of intention with cooperation principles and with assertive speech acts as communication primitives. In *Proc. first Int. joint Conf. on Autonomous agents and multiagent systems*, pages 920–927. ACM Press, 2002.
15. S. A. Kripke. Semantical considerations on modal logic. In *A Colloquium on Modal and Many-Valued Logics*, Helsinki, 1962.
16. I. Levi. Subjunctive, dispositions and chances. *Synthese*, 34:423–455, 1977.
17. F. Massacci. Single step tableaux for modal logics. *Journal of Automated Reasoning*, 24(3):319–364, 2000.
18. Robert C. Moore. A formal theory of knowledge and action. In J. R. Hobbs and R. C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358, Ablex, Norwood NJ, 1985.
19. H. J. Ohlbach. Semantics-based translation methods for modal logics. *Journal of Logic and Computation*, 1(5):691746, 1991.
20. M. D. Sadek. A study in the logic of intention. In B. Nebel, C. Rich, and W. Swartout, editors, *Proc. Third Int. Conf. on Principles of knowledge Representation and Reasoning (KR'92)*, pages 462–473. Morgan Kaufmann Publisher, 1992.
21. Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, March 1993.
22. R. M. Smullyan. A generalization of intuitionistic and modal logics. In H. Leblanc, editor, *Truth, Syntax and Modality*, pages 274–293, Amsterdam, 1973.