Resolution-Based Proof for Multi-Modal Temporal Logics of Knowledge

Clare Dixon and Michael Fisher

Department of Computing and Mathematics
Manchester Metropolitan University
Manchester M1 5GD U.K.

EMAIL: {C,Dixon,M.Fisher}@doc.mmu.ac.uk

Abstract

Temporal logics of knowledge are useful in order to specify complex systems in which agents are both dynamic and have information about their surroundings. We present a resolution method for propositional temporal logic combined with multi-modal S5 and illustrate its use on examples. This paper corrects a previous proposal for resolution in multi-modal temporal logics of knowledge.

Keywords: temporal and modal logics, non-classical resolution, theorem-proving

1 Introduction

Combinations of logics have been useful for specifying and reasoning about complex situations, for example multi-agent systems [21, 24], accident analysis [15], and security protocols [18]. For example, logics to formalise multi-agent systems often incorporate a dynamic component representing change of over time; an informational component to capture the agent's knowledge or beliefs; and a motivational component for notions such as goals, wishes, desires or intentions. Often temporal or dynamic logic is used for the dynamic aspects, the modal logics S5 and KD45 to represent knowledge and belief, and the modal logic KD for desires, intentions and wishes. Indeed the area of combining logics is a focus for much current research interest and activity, see for example [1, 2].

To be able to verify properties of specifications in these combined logics we need proof methods for them. However, comparatively little effort has been devoted to proof methods for combined logics. A resolution based method for linear time temporal logic with finite past and infinite future (PTL) plus a single copy of the modal logic S5 is given in [6]. A tableau algorithm for the multi-modal version of this logic and PTL combined with multi-modal KD45 is presented in [25]. Tableau based proof methods for Belief Desire Intention (BDI) Logics (combining linear or branching time temporal logics with the modal logics KD45 and KD) can be found in [20, 22]. Halpern, Vardi et al. provide complete axiomatisations for temporal logics of knowledge in [10] and consider complexity issues in [11].

In this paper we outline a resolution based proof method for propositional linear time temporal logic (PTL) with multi-modal logic S5. This work is an extension of the work described in [6]. At the end of [6] a proposal is made concerning an extension to its multi-modal version. In fact this proposal does not achieve this. In this paper we show how the multi-modal extension can be provided. This is carried out by separating out information concerning PTL and each modal S5 dimension. As no interactions between each sub logic take place, resolution in each component is carried out independently, with formulae involving only disjunctions of literals carrying information between components.

Proof in temporal logics such as PTL is hard due to the interaction between $\square$ (at every future moment) and $\bigcirc$ (in the next moment in time) encoding a simple for of induction, i.e.

$$(\varphi \wedge \square(\varphi \Rightarrow \bigcirc\varphi)) \Rightarrow \square\varphi$$

is valid in PTL. The complex temporal resolution rule described in §4.6 is required to deal with this. The complexity of satisfiability in PTL is PSPACE [23] as is PTL combined with multi-modal S5 (not allowing interactions between sub-logics) [11].

The structure of the paper is as follows. In §2 the syntax and semantics of the multi-modal temporal logic of knowledge we consider is given. A normal form for this logic is presented in §3 and the relevant set of resolution rules is given in §4. In §5 we illustrate the problems with the version previously presented in [6]. Finally, correctness arguments are outlined in §6 with related work and conclusions being given in §7 and §8

2 A Temporal Logic of Knowledge

In this section, we give the syntax and semantics of a logic KL_n a *temporal logic of knowledge* where the modal relation K_i is restricted to be an equivalence relation. This logic KL_n is the same as that used by Halpern and Vardi in, for example, [11].

2.1 Syntax

Formulae are constructed from a set $\mathcal{P} = \{p, q, r, \ldots\}$ of *primitive propositions.* The language KL_n contains the standard propositional connectives $\neg$ (not), $\vee$ (or), $\wedge$ (and) and $\Rightarrow$ (implies). For knowledge we assume a set of agents $Ag = \{1, \ldots n\}$ and we introduce a set of unary modal connectives K_i, for $i \in Ag$, where a formula $K_i\phi$ is read as "agent i knows ϕ". For the temporal dimension we take the usual set of future-time connectives $\bigcirc$ (*next*),$\Diamond$ (*sometime* or *eventually*), $\Box$ (*always*), $\mathcal{U}$ (*until*) and $\mathcal{W}$ (*unless* or *weak until*). We interpret these connectives over a discrete linear model of time with finite past, and infinite future; an obvious choice for such a flow of time is $(\mathbb{N}, <)$, i.e., the natural numbers ordered by the usual 'less than' relation. To assist readers who may be unfamiliar with the semantics of the temporal operators we introduce, in the next section, all operators as basic. Alternatively we could have provided the syntax and semantics of just a subset of the operators and introduced the remainder as abbreviations.

The formulae of KL_n are constructed using the following connectives and proposition symbols:

- a set $\mathcal{P}$ of proposition symbols;
- the constants **false** and **true**;
- the propositional connectives $\neg$, $\vee$, $\wedge$, $\Rightarrow$;
- the future-time temporal connectives, $\bigcirc$, $\Diamond$, $\Box$, $\mathcal{U}$ and $\mathcal{W}$;
- the modal connectives K_i (where $i \in Ag$).

The set of well-formed formulae of KL_n, WFF_K is defined by the following rules:

- any element of $\mathcal{P}$ is in WFF_K;
- **false** and **true** are in WFF_K;
- if A and B are in WFF_K then so are

$$\begin{array}{lllll} \neg A & A \vee B & A \wedge B & A \Rightarrow B & K_i A \\ \Diamond A & \Box A & A\,\mathcal{U}\,B & A\,\mathcal{W}\,B & \bigcirc A \end{array}$$

  where $i \in Ag$.

We define some particular classes of formulae that will be useful later.

Definition 1 *A* literal *is either r, or $\neg r$ where r is a proposition.*

Definition 2 *A* modal literal *is either $K_i l$ or $\neg K_i l$ where l is a literal.*

2.2 Semantics

First, we assume that the world may be in any of a set, S, of *states.*

Definition 3 *A* timeline, *t, is an infinitely long, linear, discrete sequence of states, indexed by the natural numbers.*

Note that timelines correspond to the *runs* of Halpern and Vardi [11]. Let *TLines* be the set of all timelines.

Definition 4 *A point, p, is a pair $p = (t, u)$, where $t \in TLines$ is a timeline and $u \in \mathbb{N}$ is a temporal index into t.*

Let *Points* be the set of all points.

Definition 5 *A* valuation, *π, is a function $\pi : Points \times \mathcal{P} \rightarrow \{T, F\}$.*

Definition 6 *A* model, *M, for KL_n is a structure $M = \langle TL, R_1, \ldots, R_n, \pi\rangle$, where:*

- *$TL \subseteq TLines$ is a set of timelines, with a distinguished timeline t_0;*
- *R_i, for all $i \in Ag$ is the agent accessibility relation over Points, i.e., $R_i \subseteq Points \times Points$ where each R_i is an equivalence relation;*
- *π is a valuation.*

As usual, we define the semantics of the language via the satisfaction relation '$\models$'. This relation holds between pairs of the form $\langle M, p\rangle$ (where M is a model and $p \in Points$), and KL_n-formulae. The rules defining the satisfaction relation are given below.

$$
\begin{array}{lll}
\langle M,(t,u)\rangle \models \mathbf{true} & & \\
\langle M,(t,u)\rangle \not\models \mathbf{false} & & \\
\langle M,(t,u)\rangle \models q & \text{iff} & \pi((t,u),q) = T \\
 & & \quad \text{(where } q \in \mathcal{P}\text{)} \\
\langle M,(t,u)\rangle \models \neg\phi & \text{iff} & \langle M,(t,u)\rangle \not\models \phi \\
\langle M,(t,u)\rangle \models \phi \vee \psi & \text{iff} & \langle M,(t,u)\rangle \models \phi \text{ or} \\
 & & \quad \langle M,(t,u)\rangle \models \psi \\
\langle M,(t,u)\rangle \models \phi \wedge \psi & \text{iff} & \langle M,(t,u)\rangle \models \phi \text{ and} \\
 & & \quad \langle M,(t,u)\rangle \models \psi \\
\langle M,(t,u)\rangle \models \phi \Rightarrow \psi & \text{iff} & \langle M,(t,u)\rangle \not\models \phi \text{ or} \\
 & & \quad \langle M,(t,u)\rangle \models \psi \\
\langle M,(t,u)\rangle \models \bigcirc\phi & \text{iff} & \langle M,(t,u+1)\rangle \models \phi \\
\langle M,(t,u)\rangle \models \Box\phi & \text{iff} & \forall v \in \mathbb{N},\ \text{if } (u \leq v) \\
 & & \quad \text{then } \langle M,(t,v)\rangle \models \phi \\
\langle M,(t,u)\rangle \models \Diamond\phi & \text{iff} & \exists v \in \mathbb{N},\ \text{if } (u \leq v) \\
 & & \quad \text{then } \langle M,(t,v)\rangle \models \phi \\
\langle M,(t,u)\rangle \models \phi\,\mathcal{U}\,\psi & \text{iff} & \exists v \in \mathbb{N} \text{ such that} \\
 & & \quad (u \leq v) \text{ and} \\
 & & \quad \langle M,(t,v)\rangle \models \psi, \text{and} \\
 & & \quad \forall w \in \mathbb{N},\ \text{if} \\
 & & \quad (u \leq w < v) \\
 & & \quad \text{then } \langle M,(t,w)\rangle \models \phi \\
\langle M,(t,u)\rangle \models \phi\,\mathcal{W}\,\psi & \text{iff} & \langle M,(t,u)\rangle \models \phi\,\mathcal{U}\,\psi \text{ or} \\
 & & \quad \langle M,(t,u)\rangle \models \Box\phi \\
\langle M,(t,u)\rangle \models K_i\phi & \text{iff} & \forall t' \in TL\ \forall v \in \mathbb{N} \\
 & & \quad \text{if } ((t,u),(t',v)) \in R_i \\
 & & \quad \text{then } \langle M,(t',v)\rangle \models \phi
\end{array}
$$

Satisfiability and validity in KL_n are defined in the usual way.

As agent accessibility relations in KL_n models are equivalence relations, the axioms of the normal modal system S5 are valid in KL_n models. They are

$$
\begin{array}{llrcl}
K: & \vdash & K_i(\phi \Rightarrow \psi) & \Rightarrow & (K_i\phi \Rightarrow K_i\psi) \\
T: & \vdash & K_i\phi & \Rightarrow & \phi \\
D: & \vdash & K_i\phi & \Rightarrow & \neg K_i\neg\phi \\
4: & \vdash & K_i\phi & \Rightarrow & K_iK_i\phi \\
5: & \vdash & \neg K_i\neg\phi & \Rightarrow & K_i\neg K_i\neg\phi
\end{array}
$$

The system S5 is widely recognised as the logic of idealised *knowledge*, and for this reason KL_n is often termed a *temporal logic of knowledge*.

In the following, l are literals, m are literals or modal literals and D are disjunctions of literals or modal literals.

3 A Normal Form for Temporal Logic of Knowledge

Formulae in KL_n can be transformed to a normal form, which we call Separated Normal Form for KL_n (SNF_K), which is the basis of the resolution method used in this paper. SNF for linear-time temporal logics was introduced first in [7] and has been extended to both first-order temporal logic [8] and branching-time temporal logic [3].

To assist in the definition of the normal form we introduce a further (nullary) connective **start**, that holds only at the beginning of time, i.e.

$$\langle M,(t,u)\rangle \models \mathbf{start} \quad \text{iff} \quad t = t_0 \text{ and } u = 0.$$

This allows the general form of the (rules of the) normal form to be implications. An alternative would be to allow disjunctions of literals as part of the normal form representing the clauses holding at the beginning of time.

Formulae in SNF_K are of the general form

$$\Box^* \bigwedge_i T_i$$

where each T_i is known as a *rule* and must be of a particular form and $\Box^*$ is the universal relation (which can be defined in terms of the operators "everyone knows", E and "common knowledge", C) as follows. We define E by

$$E\phi \;\Leftrightarrow\; \bigwedge_{i \in Ag} K_i\phi\,.$$

The common knowledge operator, C, is then defined as the maximal fixpoint of the formula

$$C\phi \;\Leftrightarrow\; E(\phi \wedge C\phi)\,.$$

Finally, the $\Box^*$ operator is defined as the maximal fixpoint of

$$\Box^*\phi \;\Leftrightarrow\; \Box(\phi \wedge C\,\Box^*\phi).$$

Each rule is of the following form.

$$
\begin{array}{lcll}
\mathbf{start} & \Rightarrow & \displaystyle\bigvee_{b=1}^{r} l_b & \text{(an \textit{initial} rule)} \\
\displaystyle\bigwedge_{a=1}^{g} k_a & \Rightarrow & \bigcirc \displaystyle\bigvee_{b=1}^{r} l_b & \text{(a \textit{step} rule)}
\end{array}
$$

$$\bigwedge_{a=1}^{g} k_a \Rightarrow \Diamond l \qquad \text{(a } \textit{sometime} \text{ rule)}$$

$$\textbf{true} \Rightarrow \bigvee_{b=1}^{r} m_{1_b} \qquad \text{(a } K_1\text{–rule)}$$

$$\ldots \Rightarrow \ldots$$

$$\textbf{true} \Rightarrow \bigvee_{b=1}^{r} m_{n_b} \qquad \text{(a } K_n\text{–rule)}$$

$$\textbf{true} \Rightarrow \bigvee_{b=1}^{r} l_b \qquad \text{(a literal rule)}$$

Here k_a, l_b, and l are literals and m_{i_b} are either literals, or modal literals involving the modal operator K_i. Further, each K_i rule has at least one disjunct that is a modal literal. K_i rules are sometimes known as *modal rules.* The outer '$\Box^*$' operator that surrounds the conjunction of rules is usually omitted. Similarly, for convenience the conjunction is dropped and we consider just the set of rules T_i.

3.1 Translation into SNF$_K$

The translation to SNF$_K$ uses the renaming technique [19] where complex subformulae are replaced by new propositions and then the truth value of these propositions are linked to the formulae they replaced in all states. Temporal operators are replaced by their fixpoint definitions reducing the number of different temporal operators down to a core set. See [6] for more details on how to translate into the normal form.

Note however [6] allowed modal rules with disjuncts containing different modal operators, for example

$$\textbf{true} \Rightarrow K_1 l \vee \neg K_2 p.$$

To rewrite into the more restricted normal form presented here we require additional renaming steps. For example, considering the rule given above, by renaming the formula $\neg K_2 p$ by a new proposition t we obtain two rules

$$\begin{array}{lcl} \textbf{true} & \Rightarrow & K_1 l \vee t \\ \textbf{true} & \Rightarrow & \neg t \vee \neg K_2 p \end{array}$$

thus ensuring each modal rule only contains modal literals for one K_i operator.

3.2 Merging

To apply the temporal resolution rule (see §4.6), one or more step rules may need to be combined. Consequently, a variant on SNF$_K$ called *merged-SNF (mSNF$_K$)* [7], is also defined. Given a set of rule in SNF, any rule in SNF is also a rule in mSNF$_K$. Any two rules in mSNF$_K$ may be combined to produce a rule in mSNF$_K$ as follows.

$$\begin{array}{rcl} A & \Rightarrow & \bigcirc C \\ B & \Rightarrow & \bigcirc D \\ \hline (A \wedge B) & \Rightarrow & \bigcirc (C \wedge D) \end{array}$$

Thus, any possible conjunctive combination of SNF$_K$ rules can be represented in mSNF$_K$.

4 Resolution for Temporal Logics of Knowledge

The resolution rules presented are split into four groups, initial resolution, modal resolution, step resolution and temporal resolution. The first three types of resolution are variants of classical resolution. Temporal resolution, however, is an extension allowing the resolution between formulae such as $\Box r$ with $\Diamond \neg r$.

4.1 Initial Resolution

A literal rule may be resolved with an initial rule as follows

$$[\text{IRES1}] \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & (A \vee r) \\ \textbf{start} & \Rightarrow & (B \vee \neg r) \\ \hline \textbf{start} & \Rightarrow & (A \vee B) \end{array}$$

where A is a disjunction of literals. Similarly, two initial rules may be resolved together

$$[\text{IRES2}] \quad \begin{array}{rcl} \textbf{start} & \Rightarrow & (A \vee r) \\ \textbf{start} & \Rightarrow & (B \vee \neg r) \\ \hline \textbf{start} & \Rightarrow & (A \vee B) \end{array}$$

4.2 Modal Resolution

During modal resolution we apply the following rules which are based on the modal resolution system introduced by Mints [16]. In general we may only apply a (modal) resolution rule between two literal rules, a modal and a literal rule, or between two modal rules relating to the same modal operator eg. two K_1 rules. Firstly we are allowed to resolve a literal or modal literal and its negation (where if both are K_i rules they apply to the same i).

$$[\text{MRES1}] \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & D \vee m \\ \textbf{true} & \Rightarrow & D' \vee \neg m \\ \hline \textbf{true} & \Rightarrow & D \vee D' \end{array}$$

Secondly we can resolve the formulae $K_i l$ and $K_i \neg l$ as we cannot know something and know its negation. Essentially applying the D axiom to $K_i l$ (respectively

$K_i\neg l$) we obtain $\neg K_i \neg l$ (respectively $\neg K_i l$) which is the negation of $K_i \neg l$ (respectively $K_i l$).

$$[\text{MRES2}] \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & D \vee K_i l \\ \textbf{true} & \Rightarrow & D' \vee K_i \neg l \\ \hline \textbf{true} & \Rightarrow & D \vee D' \end{array}$$

Next, as we have the T axiom $K_i p \Rightarrow p$, we can resolve between formulae such as $K_i l$ and $\neg l$ giving the following rule (where either both are K_i rules for the same i or the second premise is a literal rule).

$$[\text{MRES3}] \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & D \vee K_i l \\ \textbf{true} & \Rightarrow & D' \vee \neg l \\ \hline \textbf{true} & \Rightarrow & D \vee D' \end{array}$$

Finally, we have the following rules which involve pushing the external K_i operator into one of the rules to allow us to resolve, for example, $\neg K_i l$ with l (where either both are K_i rules for the same i or the second premise is a literal rule)

$$[\text{MRES4}] \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & D \vee \neg K_i l \\ \textbf{true} & \Rightarrow & D' \vee l \\ \hline \textbf{true} & \Rightarrow & D \vee \text{mod}\,(D') \end{array}$$

where $\text{mod}\,(D')$ is defined below.

Definition 7 *We define a function* $\text{mod}\,(D)$*, defined on disjunctions of literals or modal literals* D*, as follows.*

$$\begin{array}{rcl} \text{mod}\,(A \vee B) & = & \text{mod}\,(A) \vee \text{mod}\,(B) \\ \text{mod}\,(K_i l) & = & K_i l \\ \text{mod}\,(\neg K_i l) & = & \neg K_i l \\ \text{mod}\,(l) & = & \neg K_i \neg l \end{array}$$

The last resolution rule requires explanation. Recall, there is an implicit K_i operator surrounding each rule from the universal operator $\Box^*$. We are resolving the first rule in MRES4, as it is, with the second rule having distributed the external K_i over the implication. Thus, when we resolve $\neg K_i l$ with $K_i l$, we must adjust the other disjuncts of the second rule to show that K_i has been distributed. In more detail we consider the right hand sides of the rules given, i.e. $D \vee \neg K_i l$ and $D' \vee l$. Rewriting as implications we have $\neg D \Rightarrow \neg K_i l$ and $\neg D' \Rightarrow l$. Recall that each of these modal rules is surrounded by an implicit K_i operator therefore the second rule can be rewritten as $K_i(\neg D' \Rightarrow l)$ or $K_i \neg D' \Rightarrow K_i l$. Now D' is a disjunction of modal literals or literals i.e. $D' = m_1 \vee m_2 \vee \ldots$ so $K_i \neg D' = K_i \neg m_1 \wedge K_i \neg m_2 \wedge \ldots$. Now we can resolve the $\neg K_i l$ and $K_i l$ on the right hand side of the implication obtaining $\neg D \wedge (K_i \neg m_1 \wedge K_i \neg m_2 \wedge \ldots) \Rightarrow \textbf{false}$. Rewriting as a disjunction we have $D \vee \neg K_i \neg m_1 \vee \neg K_i \neg m_2 \vee \ldots$.

Since, in S5, we have the theorems

$$\begin{array}{rcl} \neg K_i \neg p & \Leftrightarrow & \neg K_i K_i \neg p \\ K_i p & \Leftrightarrow & \neg K_i \neg K_i p \end{array}$$

we can delete $\neg K_i \neg$ from any of the disjuncts m_i that are modal literals and obtain the required resolvent[1].

Finally we require the following rewrite rule to allow us to obtain the most comprehensive set of literal rules for use during step and temporal resolution

$$[\text{MRES5}] \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & L \vee K_i l_1 \vee K_i l_2 \vee \ldots \\ \hline \textbf{true} & \Rightarrow & L \vee l_1 \vee l_2 \vee \ldots \end{array}$$

where L is a disjunction of literals.

4.3 Step Resolution

'Step' resolution consists of the application of standard classical resolution to formulae representing constraints at a particular moment in time, together with simplification rules for transferring contradictions within states to constraints on previous states. Simplification and subsumption rules are also applied.

Pairs of step rules may be resolved using the following (step resolution) rule.

$$[\text{SRES1}] \quad \begin{array}{rcl} P & \Rightarrow & \bigcirc(A \vee r) \\ Q & \Rightarrow & \bigcirc(B \vee \neg r) \\ \hline (P \wedge Q) & \Rightarrow & \bigcirc(A \vee B) \end{array}$$

A literal rule may be resolved with a step rule as follows.

$$[\text{SRES2}] \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & (A \vee r) \\ Q & \Rightarrow & \bigcirc(B \vee \neg r) \\ \hline Q & \Rightarrow & \bigcirc(A \vee B) \end{array}$$

Once a contradiction within a state is found, the following rule can be used to generate extra global constraints.

$$[\text{SRES3}] \quad \begin{array}{rcl} P & \Rightarrow & \bigcirc\textbf{false} \\ \hline \textbf{true} & \Rightarrow & \neg P \end{array}$$

This rule states that if, by satisfying P in the last moment in time a contradiction is produced, then P must never be satisfied in *any* moment in time. The new constraint therefore represents $\Box \neg P$

[1] In [6] this rule was called MRES4a and we defined a similar rule MRES4b. MRES4b has since been found to be redundant and is omitted here. By applying MRES4b to rules P1 and P2 obtaining R1, for any rule P3 we can resolve with R1 to obtain R2, we can derive R2' where R2' $\Rightarrow$ R2 by resolving P3 with P1 and P2.

4.4 Simplification and Subsumption

The left and right hand sides of SNF_K rules can be simplified as follows

$$\begin{array}{rclll} \bigvee_{b=1}^{r} m_b \vee m \vee m' & \rightarrow & \bigvee_{b=1}^{r} m_b \vee m & \text{iff} & \vdash m' \Rightarrow m \\ \bigwedge_{b=1}^{r} l_b \wedge l \wedge l' & \rightarrow & \bigwedge_{b=1}^{r} l_b \wedge l' & \text{iff} & \vdash l' \Rightarrow l \end{array}$$

The following SNF_K rules can be removed during simplification as they represent valid subformulae and therefore cannot contribute to the generation of a contradiction.

$$\begin{array}{rcl} \textbf{false} & \Rightarrow & \bigcirc A \\ A & \Rightarrow & \bigcirc \textbf{true} \end{array}$$

The first rule is valid as **false** can never be satisfied, and the second is valid as $\bigcirc$**true** is always satisfied.

Subsumption also forms part of the step resolution process. Here, as in classical resolution, a rule may be removed from the rule-set if it is subsumed by another rule already present. Subsumption may be expressed as the following operation.

$$\left\{ \begin{array}{rcl} C & \Rightarrow & A \\ D & \Rightarrow & B \end{array} \right\} \xrightarrow{\vdash C \Rightarrow D \quad \vdash B \Rightarrow A} \{D \Rightarrow B\}$$

The side conditions $\vdash C \Rightarrow D$ and $\vdash B \Rightarrow A$ must hold before this subsumption step can be applied and, in this case, the rule $C \Rightarrow A$ can be deleted without losing information.

4.5 Termination

Each cycle of initial, modal or step resolution terminates when either no new resolvents are derived, or **false** is derived in the form of **start** $\Rightarrow$ **false**.

4.6 Temporal Resolution

During temporal resolution the aim is to resolve a $\Diamond$–rule, $Q \Rightarrow \Diamond l$, with a set of rules that together imply $\Box\neg l$, for example a set of rules that together have the effect of $A \Rightarrow \bigcirc\Box\neg l$. So, the general temporal resolution operation, written as an inference rule, becomes

$$\begin{array}{rcl} A & \Rightarrow & \bigcirc\Box\neg l \\ Q & \Rightarrow & \Diamond l \\ \hline Q & \Rightarrow & (\neg A)\,\mathcal{W}\, l \end{array}$$

The intuition behind the resolvent is that, once Q has occurred then A must not be satisfied until l has occurred (i.e. the eventuality has been satisfied). That is we can not allow A to be satisfied once Q has been satisfied as this would ensure that l is false at all moments in the future. (Note that the generation of $Q \Rightarrow (\neg A)\,\mathcal{U}\, l$ as a resolvent is sound. However as $(\neg A)\,\mathcal{U}\, l \equiv ((\neg A)\,\mathcal{W}\, l) \wedge \Diamond l$ the resolvent would be equivalent to the pair of resolvents $Q \Rightarrow (\neg A)\,\mathcal{W}\, l$ and $Q \Rightarrow \Diamond l$. The latter is subsumed by the sometime PTL rule we have resolved with leaving the '$\mathcal{W}$' formula.) The resolvent must be translated into SNF_K.

However the interaction between the '$\bigcirc$' and '$\Box$' operators in KL_n makes the definition of such a rule non-trivial and further the translation from KL_n to SNF_K will have removed all but the outer level of $\Box$–operators. So, resolution will be between a $\Diamond$–rule and a *set* of merged SNK_K rules that together imply an $\Box$–formula which will contradict the $\Diamond$–rule. Thus, given a set of rules in SNF_K, then for every rule of the form $Q \Rightarrow \Diamond l$ temporal resolution may be applied between this sometime rule and a set of merged step rules, which taken together force $\neg l$ to always be satisfied.

The temporal resolution rule is given by

$$\text{[TRES]} \quad \begin{array}{rcl} A_0 & \Rightarrow & \bigcirc F_0 \\ \cdots & & \cdots \\ A_n & \Rightarrow & \bigcirc F_n \\ Q & \Rightarrow & \Diamond l \\ \hline Q & \Rightarrow & (\bigwedge_{i=0}^{n} \neg A_i)\,\mathcal{W}\, l \end{array}$$

with side conditions

$$\left\{ \begin{array}{rcl} \text{for all } 0 \leq i \leq n & \vdash & F_i \Rightarrow \neg l \\ \text{and} & \vdash & F_i \Rightarrow \bigvee_{j=0}^{n} A_j \end{array} \right\}$$

These side conditions ensure that the set of rules $A_i \Rightarrow \bigcirc F_i$ together imply $\bigcirc\Box\neg l$. In particular the first side condition ensures that each rule, $A_i \Rightarrow \bigcirc F_i$, makes $\neg l$ true in the next moment if A_i is satisfied. The second side condition ensures that the right hand side of each rule, $A_i \Rightarrow \bigcirc F_i$, means that the left hand side of one of the rules in the set will be satisfied. So once the left hand side of one of these rules is satisfied, i.e. if A_i is satisfied for some i in the last moment in time, then $\neg l$ will hold now and the left hand side of another rule will also be satisfied. Thus at the next moment in time again $\neg l$ holds and the left hand side of another rule is satisfied and so on. So if any of the A_i are satisfied then $\neg l$ will be *always* be satisfied, i.e.,

$$\bigvee_{k=0}^{n} A_k \Rightarrow \bigcirc\Box\neg l.$$

Such a set of rules are known as a *loop* in $\neg l$. The search for such a set of rules is non-trivial and an algorithm to detect such sets of rules are given in [5].

As we usually work with rules in the normal form we translate the resolvent from TRES into SNF_K obtaining the following rules for each i from 0 to n where t is a new proposition.

$$\begin{array}{rcl} \mathbf{true} & \Rightarrow & \neg Q \vee l \vee \neg A_i \\ \mathbf{true} & \Rightarrow & \neg Q \vee l \vee t \\ t & \Rightarrow & \bigcirc(l \vee \neg A_i) \\ t & \Rightarrow & \bigcirc(l \vee t) \end{array}$$

4.7 An Algorithm for the Temporal Resolution Method

Given any KL_n formula φ, to be tested for unsatisfiability, the following steps are performed.

1. Translate φ into SNF_K, giving φ_s.
2. Perform initial, modal and step resolution (including simplification and subsumption) on φ_s until either
   (a) **start** $\Rightarrow$ **false** is derived — terminate noting that φ is unsatisfiable; or
   (b) no new resolvents are generated — continue to step (3).
3. Select an eventuality from the right-hand side of a sometime clause within φ_s. Search for a set of clauses with which one of the temporal resolution rules can be applied.
4. If the resolvent is new (i.e. is not subsumed by previously detected resolvents) translate into SNF_K and go to step (2). Otherwise if no new resolvents have been found for any eventuality, terminate declaring φ satisfiable, else go to step (3).

4.8 Example

First we consider the following unsatisfiable, purely modal, formula

$$K_1 p \wedge \neg K_1 \neg K_2 \neg p$$

for later comparison with the resolution system suggested in [6]. We begin by illustrating why the formula is unsatisfiable. Take an initial world s_0 containing $K_1 p$ and $\neg K_1 \neg K_2 \neg p$. There must be a 1-edge to a world s_1 containing $K_2 \neg p$ and p. As models of S5 are reflexive there must be a 2-edge from s_1 to itself. Thus from $K_2 \neg p$, the world s_1 must also contain $\neg p$ giving a contradiction.

To show the unsatisfiablilty of the formula we first translate to the normal form, separating the relevant rules, we obtain

$$\begin{array}{rrcl} 1. & \mathbf{start} & \Rightarrow & x \\ 2. & \mathbf{true} & \Rightarrow & \neg x \vee K_1 p \\ 3. & \mathbf{true} & \Rightarrow & \neg x \vee \neg K_1 \neg y \\ 4. & \mathbf{true} & \Rightarrow & \neg y \vee K_2 \neg p \end{array}$$

The proof continues as follows.

$$\begin{array}{rrcll} 5. & \mathbf{true} & \Rightarrow & \neg y \vee \neg p & [4\ \mathit{MRES}5] \\ 6. & \mathbf{true} & \Rightarrow & \neg x \vee \neg K_1 p & [3,5\ \mathit{MRES}4] \\ 7. & \mathbf{true} & \Rightarrow & \neg x & [2,6\ \mathit{MRES}1] \\ 8. & \mathbf{start} & \Rightarrow & \mathbf{false} & [1,7\ \mathit{IRES}] \end{array}$$

We have obtained a contradiction showing that the original formula is unsatisfiable.

Secondly we show

$$K_1 \Diamond \phi \Rightarrow \neg \square K_2 \neg \phi$$

is valid by negating and translating into SNF_K.

$$\begin{array}{rrcl} 1. & \mathbf{start} & \Rightarrow & x \\ 2. & \mathbf{true} & \Rightarrow & \neg x \vee K_1 y \\ 3. & y & \Rightarrow & \Diamond \phi \\ 4. & \mathbf{true} & \Rightarrow & \neg x \vee z \\ 5. & \mathbf{true} & \Rightarrow & \neg x \vee t \\ 6. & t & \Rightarrow & \bigcirc t \\ 7. & t & \Rightarrow & \bigcirc z \\ 8. & \mathbf{true} & \Rightarrow & \neg z \vee K_2 \neg \phi \end{array}$$

Resolution commences as follows.

$$\begin{array}{rrcll} 9. & \mathbf{true} & \Rightarrow & \neg z \vee \neg \phi & [8\ \mathit{MRES}5] \\ 10. & t & \Rightarrow & \bigcirc \neg \phi & [7,9\ \mathit{SRES}2] \end{array}$$

Thus by merging rules 6 and 10 we obtain

$$t \Rightarrow \bigcirc(t \wedge \neg \phi)$$

for resolution with rule 3.

$$\begin{array}{rrcll} 11. & y & \Rightarrow & \neg t\, \mathcal{W}\, \phi & [3,6,10\ \mathit{TRES}] \\ 12. & \mathbf{true} & \Rightarrow & \neg y \vee \neg t \vee \phi & [11\ \mathit{SNF_K}] \\ 13. & \mathbf{true} & \Rightarrow & \neg y \vee \neg t \vee \neg z & [8,12\ \mathit{MRES}3] \\ 14. & \mathbf{true} & \Rightarrow & \neg x \vee \neg t \vee \neg z & [2,13\ \mathit{MRES}3] \\ 15. & \mathbf{true} & \Rightarrow & \neg x \vee \neg t & [4,14\ \mathit{MRES}1] \\ 16. & \mathbf{true} & \Rightarrow & \neg x & [5,15\ \mathit{MRES}1] \\ 17. & \mathbf{start} & \Rightarrow & \mathbf{false} & [1,16\ \mathit{IRES}1] \end{array}$$

5 Comparison with Previous Version

The version suggested in [6] allowed modal rules that possibly contained modal literals for more than one modal operator, for example $\mathbf{true} \Rightarrow l \vee K_1 r \vee \neg K_2 q$. The modal rule MRES1 could be applied between any literal or modal literal and its negation. An extended version of MRES2 was also allowed

$$\text{[MRES2']} \quad \frac{\begin{array}{lcl}\textbf{true} & \Rightarrow & D \vee K_i l \\ \textbf{true} & \Rightarrow & D' \vee K_j \neg l\end{array}}{\begin{array}{lcl}\textbf{true} & \Rightarrow & D \vee D'\end{array}}$$

Here i may or may not equal j. The rule is sound due to the presence of the T axiom. Again MRES3 could be applied between any modal or literal rules. The following rule is suggested as for the multi-modal extension of MRES4.

$$\text{MRES4'} \quad \frac{\begin{array}{lcl}\textbf{true} & \Rightarrow & D \vee \neg K_i l \\ \textbf{true} & \Rightarrow & D' \vee l\end{array}}{\begin{array}{lcl}\textbf{true} & \Rightarrow & D \vee \neg K_i \neg x \\ \textbf{true} & \Rightarrow & \neg x \vee D'\end{array}}$$

However it is obvious that this achieves nothing as the resolvent is just a copy of the original rules with l being replaced by $\neg x$. Similarly we have an extended version of the MRES5 rule

$$\text{[MRES5']} \quad \frac{\textbf{true} \Rightarrow L \vee K_i l_1 \vee K_j l_2 \vee \ldots}{\textbf{true} \Rightarrow L \vee l_1 \vee l_2 \vee \ldots}$$

Now we try resolution on the modal formula given in §4.8. First we translate to the normal form obtaining the same normal form as previously.

$$\begin{array}{rlcl}
1. & \textbf{start} & \Rightarrow & x \\
2. & \textbf{true} & \Rightarrow & \neg x \vee K_1 p \\
3. & \textbf{true} & \Rightarrow & \neg x \vee \neg K_1 \neg y \\
4. & \textbf{true} & \Rightarrow & \neg y \vee K_2 \neg p
\end{array}$$

Applying the rule MRES4' doesn't lead to a contradiction. That is we could try resolve rules 3 and 4 and introduce a new variable z as follows.

$$\begin{array}{rlcll}
5. & \textbf{true} & \Rightarrow & \neg x \vee \neg K_1 \neg z & [3,4\ \mathit{MRES4'}] \\
6. & \textbf{true} & \Rightarrow & \neg z \vee K_2 \neg p & [3,4\ \mathit{MRES4'}]
\end{array}$$

Next we could potentially apply MRES4' again to 5 and 6 continuing for ever. Alternatively we could continue as follows

$$\begin{array}{rlcll}
7. & \textbf{true} & \Rightarrow & \neg z \vee \neg x & [2,6\ \mathit{MRES2'}] \\
8. & \textbf{true} & \Rightarrow & \neg x \vee \neg K_1 x & [5,7\ \mathit{MRES4}] \\
9. & \textbf{true} & \Rightarrow & \neg K_1 x & [\text{SIMP } \neg x \Rightarrow \neg K_1 x]
\end{array}$$

without achieving a contradiction. If we don't allow the use of the original MRES4 rule the only option we have is to resolve $K_1 p$ and $K_2 \neg p$ (or apply MRES5' to one or both followed by either MRES3 or MRES1).

$$\begin{array}{rlcll}
5'. & \textbf{true} & \Rightarrow & \neg x \vee \neg y & [2,4\ \mathit{MRES2'}] \\
6'. & \textbf{true} & \Rightarrow & \neg x \vee \neg K_1 x & [3,5'\ \mathit{MRES4}] \\
7'. & \textbf{true} & \Rightarrow & \neg K_1 x & [\text{SIMP } \neg x \Rightarrow \neg K_1 x]
\end{array}$$

Again we don't obtain a contradiction.

So if we don't allow the original MRES4 rule this example demonstrates incompleteness of the suggested set of rules. If the original MRES4 rule is allowed in addition to MRES4' we can reach a proof by following the steps in §4.8. However the use of MRES4' produces resolvents to which we can re-apply MRES4', that is MRES4' can be applied forever and we have termination problems.

6 Correctness

Firstly we can show that the transformation into SNF_K preserves satisfiability.

Theorem 1 *A KL_n formula A is satisfiable if, and only if, $\tau_0[A]$ is satisfiable (where τ_0 is the translation into SNF_K).*

Proofs analogous to those in [6, 9] will suffice.

Theorem 2 *(Soundness) Let S be a satisfiable set of SNF_K rules and T be the set of rules obtained from S by an application of one of the resolution rules. Then T is also satisfiable.*

This can be shown by showing an application of each resolution rule preserves satisfiability.

Theorem 3 *(Completeness) If a set of SNF_K rules is unsatisfiable then it has a refutation by the temporal resolution procedure given in this paper.*

Completeness is shown by constructing a graph to represent all possible models for the set of rules. There are $n+1$ types of edges, n for each of the n different modal operators, the other representing the temporal dimension. Each node in the graph is a pair (V, E). V is the consistent union of a member of the following set for each proposition p and modal operator K_i

$$\{\{K_i p, \neg K_i \neg p, p\}, \{K_i \neg p, \neg K_i p, \neg p\}, \\ \{\neg K_i p, \neg K_i \neg p, p\}, \{\neg K_i p, \neg K_i \neg p, \neg p\}\}.$$

E contains unsatisfied eventualities from the temporal dimension. Deletions in the graph represent the application of of the resolution rules. An empty graph corresponds with the generation of false. A proof for the single modal case is given in [6]

Theorem 4 *(Termination) The temporal resolution procedure described above applied to a set of SNF_K rules terminates.*

Essentially given a set of SNF_K we only need one new proposition for each different right hand side of a sometime rule [9]. Thus the set of possible SNF_K rules that may be generated by applying resolution rules is finite and the resolution algorithm terminates when nothing new is derived.

7 Related Work

Translation based methods have been successfully used, to reason about combinations of modal logics. To prove a modal formula valid, instead of reasoning directly about the formulae in the modal logic it is translated into (a subclass of) first-order classical logic and reasoning takes place in the first-order logic framework. The translation based approach for multi-modal logics logics that are extensions of K_m by the axioms 4, 5, B, D, T can be found in [17, 4]. The translation based approach to modal theorem proving has been combined by the temporal reasoning part of the resolution method described in this paper in [13].

Rao and Georgeff consider at tableau based proof methods for BDI logics in [20, 22]. Here they combine either linear or branching-time temporal logics (CTL and CTL*) with the modal logics with the modal logics KD45 for belief and KD for desire and intention. In [22] interactions between the modal components are also considered. Tableau methods for PTL combined with multi-modal S5 or multi-modal KD45 are also considered in [25]. Tableaux methods for description logics (essentially combinations of modal logics) have been described and implemented in [12].

Halpern, Vardi et al. consider the combination of propositional linear and branching time temporal logics with multi-modal S5 allowing a variety of interactions in [11, 10]. Rather than theorem proving this series of papers concentrates on the complexity of the satisfiability problem and obtaining complete axiomatisations for various systems. For example adding the axiom

$$K_i \bigcirc \varphi \Rightarrow \bigcirc K_i \varphi$$

to the axioms of PTL and S5 gives a complete axiomatisation for systems of synchrony and perfect recall. A system is said to have perfect recall if the set of executions an agent considers possible stays the same or decreases over time. Adding this interaction to the single agent case increases the complexity of validity to double exponential time and the to multi-agent case to non-elementary time [11, 10].

Theorem proving in reified temporal logics of belief are considered in [14]. The branching time temporal logic considered in [14] does not include the until operator we allow but more importantly does not allow the induction axiom mentioned in §1. Further, in this paper we give a decision procedure for the logic KL_n whereas for the method mentioned in [14] temination and its correctness with respect to the original modal and temporal logic semantics is also not discussed.

Further issues relating to combining logics are considered in in [2, 1].

8 Conclusions

We have presented a resolution method for a multi-modal temporal logic of knowledge. This extends the single modal temporal logic of knowledge presented in [6]. Further, it corrects the suggestion of how to extend the single modal version to a multi-modal version given at the end of the paper. In the single modal version, formulae involving temporal operators and those involving modal operators were separated and resolution rules applied to each component. Information was passed between the two logics in the form of disjunctions of literals (literal rules). In the multi-modal extension presented here we do the same, i.e. separate out the temporal formula and separate out formulae associated with each modal operator. Again resolution rules are applied on each component separately.

This work is useful for the verification of complex systems such as agent specification languages which involve both a dynamic and an informational component and has been motivated in part by collaboration with the developers of KARO [24]. Our current work involves implementing this system and adding resolution rules to deal with interactions between the temporal and modal logics. Further we are using KL_n to specify problems from a core of the KARO framework and from the domain of accident analysis and attempting to prove properties of the specifications using resolution.

Acknowledgements

The authors would like to thank Ullrich Hustadt and Weibe van der Hoek for their helpful comments and suggestions about this work.

References

[1] B. Bennett, C. Dixon, M. Fisher, E. Franconi, I. Horrocks, U. Hustadt, and M. de Rijke. Combinations of Modal Logics. (Submitted).

[2] P. Blackburn and M. de Rijke. Why Combine Logics? *Studia Logica*, 59:5–27, 1997.

[3] A. Bolotov and M. Fisher. A Clausal Resolution Method for CTL Branching Time Temporal Logic. *Journal of Experimental and Theoretical Artificial Intelligence*, 11:77–93, 1999.

[4] H. de Nivelle. Translation of S4 into GF and 2VAR. Manuscript, May 1999.

[5] C. Dixon. Temporal Resolution using a Breadth-First Search Algorithm. *Annals of Mathematics and Artificial Intelligence*, 22:87–115, 1998.

[6] C. Dixon, M. Fisher, and M. Wooldridge. Resolution for Temporal Logics of Knowledge. *Journal of Logic and Computation*, 8(3):345–372, 1998.

[7] M. Fisher. A Resolution Method for Temporal Logic. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 99–104, Sydney, Australia, August 1991. Morgan Kaufman.

[8] M. Fisher. A Normal Form for First-Order Temporal Formulae. In *Proceedings of Eleventh International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Computer Science*, pages 370–384, Saratoga Springs, New York, June 1992. Springer-Verlag.

[9] M. Fisher, C. Dixon, and M. Peim. Clausal Temporal Resolution. (Submitted), 2000.

[10] J. Y. Halpern, R. van der Meyden, and M. Vardi. Complete axiomatizations for reasoning about knowledge and time. Submitted for publication, 1999.

[11] J. Y. Halpern and M. Y. Vardi. The Complexity of Reasoning about Knowledge and Time. I Lower Bounds. *Journal of Computer and System Sciences*, 38:195–237, 1989.

[12] I. Horrocks. The FaCT System. In *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference (Tableaux'98)*, volume 1397 of *LNCS*, pages 307–312. Springer-Verlag, May 1998.

[13] U. Hustadt, C. Dixon, R. A. Schmidt, and M. Fisher. Normal forms and proofs in combined modal and temporal logics. In *Proceedings of the International Workshop on Frontiers of Combining Systems (FroCoS'2000)*, LNAI. Springer, 2000. To appear.

[14] H. Reichgelt J. S. Aitken and N. Shadbolt. Resolution theorem proving in reified modal logics. *Journal of Automated Reasoning*, 12:103–129, 1994.

[15] C.W. Johnson. The application of user modeling techn iques to reason about the human contribution to major accidents. In Judy Kay, editor, *Proceedings of the Seventh International User Modelling Conference (UM'99)*, pages 13–22, New York, USA, 1999. Springer Verlag.

[16] G. Mints. Gentzen-Type Systems and Resolution Rules, Part I: Propositional Logic. *Lecture Notes in Computer Science*, 417:198–231, 1990.

[17] H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. To appear in the *Logic Journal of the IGPL*.

[18] L. C. Paulson. Mechanized proofs of security protocols: Needham-schroeder with public keys. Technical Report 413, Computer Laboratory, University of Cambridge, January 1997.

[19] D. A. Plaisted and S. A. Greenbaum. A Structure-Preserving Clause Form Translation. *Journal of Symbolic Computation*, 2(3):293–304, September 1986.

[20] A. S. Rao. Decision Procedures for Propositional Linear-Time Belief-Desire-Intention Logics. In M. Wooldridge, K. Fischer, P. Gmytrasiewicz, N. R. Jennings, J. P. Müller, and M. Tambe, editors, *IJCAI-95 Workshop on Agent Theories, Architectures, and Languages*, pages 102–118, Montréal, Canada, August 1995.

[21] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, pages 473–484. Morgan-Kaufmann, April 1991.

[22] A. S. Rao and M. P. Georgeff. Formal Models and Decision Procedures for Multi-Agent Systems. Technical Report 61, Australian Artificial Intelligence Institute, June 1995.

[23] A. P. Sistla and E. M. Clarke. Complexity of Propositional Linear Temporal Logics. *ACM Journal*, 32(3):733–749, July 1985.

[24] B. van Linder, W. van der Hoek, and J. J. Ch. Meyer. Formalising motivational attitudes of agents: On preferences, goals and commitments. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents II (LNAI 1037)*, pages 17–32. Springer-Verlag, 1996.

[25] M. Wooldridge, C. Dixon, and M. Fisher. A Tableau-Based Proof Method for Temporal Logics of Knowledge and Belief. *Journal of Applied Non-Classical Logics*, 8(3):225–258, 1998.