Resolution for Temporal Logics of Knowledge

Clare Dixon, Michael Fisher
Dept. of Computing
Manchester Metropolitan University
Manchester M1 5GD
United Kingdom

Michael Wooldridge
Dept. of Electronic Engineering
Queen Mary & Westfield College
London E1 4NS
United Kingdom

Journal of Logic and Computation 8(3):345–372, 1998

Abstract

A resolution based proof system for a temporal logic of knowledge is presented and shown to be correct. Such logics are useful for proving properties of distributed and multi-agent systems. Examples are given to illustrate the proof system. An extension of the basic system to the multi-modal case is given and illustrated using the 'muddy children problem'.

1 Introduction

Temporal logics have been shown to have many applications in computer science and artificial intelligence. For example, they are used in the specification and verification of reactive systems [28], in temporal query languages [8], executable logics [18] and for reasoning about action [36]. For some applications, however, logics containing connectives that operate over just the one modal dimension of time do not provide sufficient expressive power. For such applications, it is necessary to provide connectives that allow us to represent the properties of different modal dimensions *in the same logic.*

In this paper, we consider such a *multi-modal* logic, called KL_n, containing connectives for representing both time and *knowledge*. Thus, in addition to the usual connectives of linear discrete temporal logic [12], KL_n contains an indexed set of unary modal connectives that allow us to represent the information possessed by a group of agents. These connectives satisfy analogues of the axioms of the modal system S5 [7], which is widely recognized as a logic of *idealized knowledge* [14]. It is for this reason that we call KL_n a temporal logic of knowledge (formal properties of KL_n-like logics are studied in [25, 14]).

While logics such as KL_n have been studied for some time (see, e.g., [22, 27, 25, 15, 14]), relatively little effort has been directed at developing proof methods for such logics [41]. This is perhaps because of the complexity of the problem: it is shown in [25, 14] that even for comparatively simple temporal logics of knowledge, the decision problem for validity is PSPACE complete. For more complex variants, the problem is undecidable even in the propositional case. However, recent advances in proof methods for the underlying temporal logic (for which the decision problem is also PSPACE complete [37]) indicate that practical theorem provers for such complex logics may still be possible [17, 10]. In this paper, we extend the proof method for purely temporal logics described in [17] to deal with KL_n by using modal resolution rules similar to those devised by Mints [29]. Specifically, we present a clausal resolution method for KL_1, an instantiation of KL_n containing only one knowledge operator, provide soundness and completeness arguments and show how this approach can be extended to KL_n in general.

The structure of this paper is as follows. In §2, we formally define our temporal logic of knowledge and, in §3, we provide a normal form for formulae of this logic. In §4, we provide a resolution system for this logic, based upon the normal form, and provide simple examples of its use in §5. The correctness of the system is considered in §6, while the extension of the method to a multi-agent context is addressed in §7. Related work is examined in §8 and, in §9, some conclusions and areas for further work are outlined.

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.

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), $\Rightarrow$ (implies) and $\Leftrightarrow$ (if, and only if). 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.

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 $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$;
- the future-time temporal connectives, **start**, $\bigcirc$, $\Diamond$, $\Box$, $\mathcal{U}$ 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**, **true**, and **start** are in WFF_K;
- if A and B are in WFF_K then so are

$$\begin{array}{llllll} \neg A & A \lor B & A \land B & A \Rightarrow B & A \Leftrightarrow B & \\ \Diamond A & \Box A & A \mathcal{U} B & A \mathcal{W} B & \bigcirc A & K_i A \end{array}$$

  where $i \in Ag$.

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

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

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

Definition 3 The *knowledge set* for agent i for a set of literals or modal literals X is defined as

$$K_i\text{\_set}\,(X) = \{l \mid K_i l \in X\}.$$

2.2 Semantics

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

Definition 4 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 [25]. Let *TLines* be the set of all timelines.

Definition 5 A point, p, is a pair $p = (t, u)$, where $t \in \mathit{TLines}$ is a timeline and $u \in \mathbb{N}$ is a temporal index into t.

Let *Points* be the set of all points.

Definition 6 A *valuation*, π, is a function $\pi : \mathit{Points} \times \Phi \rightarrow \{T, F\}$.

Definition 7 A *model*, M, for KL_n is a structure $M = \langle TL, R_1, \ldots, R_n, \pi \rangle$, where:

- $TL \subseteq \mathit{TLines}$ 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., $R_i \subseteq \mathit{Points} \times \mathit{Points}$ 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 '$\models$'. For KL_n. This relation holds between pairs of the form $\langle M, p \rangle$ (where M is a model and $p \in \mathit{Points}$), and KL_n-formulae. The rules defining the satisfaction relation are given below. We omit most of the propositional connectives as they are standard.

$\langle M, (t_0, 0)\rangle \models \mathbf{start}$		
$\langle M, (t, u)\rangle \models \mathbf{true}$		
$\langle M, (t, u)\rangle \models p$	iff	$\pi((t, u), p) = T$ (where $p \in \mathcal{P}$)
$\langle M, (t, u)\rangle \models \neg A$	iff	$\langle M, (t, u)\rangle \not\models A$
$\langle M, (t, u)\rangle \models A \vee B$	iff	$\langle M, (t, u)\rangle \models A$ or $\langle M, (t, u)\rangle \models B$
$\langle M, (t, u)\rangle \models \bigcirc A$	iff	$\langle M, (t, u+1)\rangle \models A$
$\langle M, (t, u)\rangle \models \square A$	iff	$\forall u' \in \mathbb{N}$, if $(u \leq u')$ then $\langle M, (t, u')\rangle \models A$
$\langle M, (t, u)\rangle \models \lozenge A$	iff	$\exists u' \in \mathbb{N}$, if $(u \leq u')$ then $\langle M, (t, u')\rangle \models A$
$\langle M, (t, u)\rangle \models A \mathcal{U} B$	iff	$\exists u' \in \mathbb{N}$ such that $(u' \geq u)$ and $\langle M, (t, u')\rangle \models B$, and $\forall u'' \in \mathbb{N}$, if $(u \leq u'' < u')$ then $\langle M, (t, u'')\rangle \models A$
$\langle M, (t, u)\rangle \models A \mathcal{W} B$	iff	$\langle M, (t, u)\rangle \models A \mathcal{U} B$ or $\langle M, (t, u)\rangle \models \square A$
$\langle M, (t, u)\rangle \models K_i A$	iff	$\forall t' \in TL.\ \forall u' \in \mathbb{N}$. if $((t, u), (t', u')) \in R_i$ then $\langle M, (t', u')\rangle \models A$

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. 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*. (Our logic KL_n in fact corresponds exactly to Halpern and Vardi's logic $KL_{(n)}$ [25], hence the name.)

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 [17] and has been extended to both first-order temporal logic [19] and branching-time temporal logic [4].

The translation to SNF$_K$ uses the renaming technique [33] 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. So, to be able to define the normal form, we must first define the $\square^*$ operator, which is in turn defined in terms of the C (or *common knowledge*) and E operators. 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

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

To illustrate the properties of this operator, we must formalise the notion of *reachability*.

Definition 8 Let M be a KL_n-model and $(t,u),(t',u')$ be points in M. Then (t',u') is *reachable* from (t,u) iff either: (i) $t = t'$ and $u' \geq u$; (ii) $((t,u),(t',u')) \in R_i$ for some agent $i \in Ag$; or (iii) there exists some point (t'',u'') in M such that (t'',u'') is reachable from (t,u) and (t',u') is reachable from (t'',u'').

The important property of the $\Box^*$ operator can now be stated.

Theorem 9 Let M be a KL_n-model and p, p' be points in M such that $\langle M,p\rangle \models \Box^*\phi$. Then $\langle M,p'\rangle \models \phi$ if p' is reachable from p.

Proof Assume $\langle M,p\rangle \models \Box^*\phi$ and that p' is reachable from p. Then there must be some sequence of states $p_1,\ldots,p_k$ such that $p_1 = p$ and $p_k = p'$, such that $\forall v \in \{1,\ldots,k-1\}$, we have either (i) p_v and p_{v+1} are on the same time line but p_{v+1} occurs later than p_v, or (ii) $(p_v,p_{v+1}) \in R_i$ for some $i \in Ag$. An induction shows that $\langle M,p_v\rangle \models \Box^*\phi$ implies $\langle M,p_{v+1}\rangle \models \Box^*\phi$, for all $v \in \{1,\ldots,k-1\}$. The base follows from the fact that $\Box^*\phi$ is defined to be $\Box(\phi \wedge C\,\Box^*\phi)$ and that $\models C\phi \Rightarrow \phi$. For the inductive step, assume $\langle M,p_w\rangle \models \Box^*\phi$. Then either (i) p_{w+1} occurs on the same timeline as p_w but later, in which case from the definition of $\Box^*$ we know that $\langle M,p_{w+1}\rangle \models \Box^*\phi$, or else (ii) $(p_w,p_{w+1}) \in R_i$, in which case from the definition of $\Box^*$ and C, we also know that $\langle M,p_{w+1}\rangle \models \Box^*\phi$. Hence $\langle M,p_k\rangle \models \Box^*\phi$, so $\langle M,p'\rangle \models \Box^*\phi$, and so from the definition of $\Box^*$, we have $\langle M,p'\rangle \models \phi$.

Thus we reason about reachable points from the initial point in the distinguished timeline t_0 (where **start** is satisfiable), i.e. the points we require in the proof. As renaming [33] is central to the transformation of arbitrary formulae to SNF, we must be sure that its properties carry over to KL_n. Now we have defined the $\Box^*$ operator then renaming of a formula such as $\Diamond(\phi\,\mathcal{U}\,\psi)$, where ψ is a complex subformula, produces $\Diamond(\phi\,\mathcal{U}\,p) \wedge \Box^*(p \Leftrightarrow \psi)$. This theorem therefore guarantees that renaming carried out within the context of the $\Box^*$ operator will preserve satisfiability.

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 one of the following forms.

$$
\begin{array}{rcll}
\textbf{start} & \Rightarrow & \bigvee_{b=1}^{r} l_b & \text{(an \textit{initial} rule)} \\
\bigwedge_{a=1}^{g} k_a & \Rightarrow & \bigcirc \bigvee_{b=1}^{r} l_b & \text{(a \textit{global} rule)} \\
\bigwedge_{a=1}^{g} k_a & \Rightarrow & \Diamond l & \text{(a \textit{sometime} rule)} \\
\textbf{true} & \Rightarrow & \bigvee_{b=1}^{r} m_b & \text{(a } K\text{–rule)}
\end{array}
$$

Here k_a, l_b, and l are literals and m_b are either literals or modal literals. 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. In the following discussion we further split the K-rules into two types, *literal rules* and *modal rules*. Literal rules are K-rules where the right hand side consists of a disjunction of literals. Modal rules are K-rules where at least one of the disjuncts is a modal literal.

3.1 Translation into $\mathbf{SNF}_K$

In this section, we consider the translation of an arbitrary KL_n formula into the normal form. We will describe the individual transformations in detail, but note that many of them are similar to those used in translating purely temporal logic to SNF [19].

Take any formula A of KL_n and translate into SNF_K as follows (where f is a new proposition).

$$
A \longrightarrow \left\{ \begin{array}{rcl} \textbf{start} & \Rightarrow & f \\ f & \Rightarrow & A \end{array} \right\}
$$

If the main operator on the right of the implication is a classical operator remove it as follows.

$$\{x \Rightarrow (A \wedge B)\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & A \\ x & \Rightarrow & B \end{array} \right\}$$

$$\{x \Rightarrow (A \Rightarrow B)\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \neg A \vee B \end{array} \right\}$$

$$\{x \Rightarrow (A \Leftrightarrow B)\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \neg A \vee B \\ x & \Rightarrow & \neg B \vee A \end{array} \right\}$$

$$\{x \Rightarrow \neg(A \wedge B)\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \neg A \vee \neg B \end{array} \right\}$$

$$\{x \Rightarrow \neg(A \Rightarrow B)\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & A \\ x & \Rightarrow & \neg B \end{array} \right\}$$

$$\{x \Rightarrow \neg(A \vee B)\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \neg A \\ x & \Rightarrow & \neg B \end{array} \right\}$$

$$\{x \Rightarrow \neg(A \Leftrightarrow B)\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & ((A \wedge \neg B) \vee (B \wedge \neg A)) \end{array} \right\}$$

Rename complex subformulae enclosed in a modal operator as follows where y is a new proposition.

$$\{x \Rightarrow K_i A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & K_i y \\ y & \Rightarrow & A \end{array} \right\} \quad \text{For any A that is not a literal.}$$

$$\{x \Rightarrow \neg K_i A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \neg K_i \neg y \\ y & \Rightarrow & \neg A \end{array} \right\} \quad \text{For any A that is not a literal.}$$

Rename complex subformulae enclosed in any temporal operators as follows where y is a new proposition.

$$\{x \Rightarrow \bigcirc A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \bigcirc y \\ y & \Rightarrow & A \end{array} \right\} \quad \text{For any A that is not a disjunction of literals.}$$

$$\{x \Rightarrow \neg \bigcirc A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \bigcirc y \\ y & \Rightarrow & \neg A \end{array} \right\}$$

$$\{x \Rightarrow \square A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \square y \\ y & \Rightarrow & A \end{array} \right\} \quad \text{For any A that is not a literal.}$$

$$\{x \Rightarrow \neg \square A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \diamondsuit y \\ y & \Rightarrow & \neg A \end{array} \right\}$$

$$\{x \Rightarrow \diamondsuit A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \diamondsuit y \\ y & \Rightarrow & A \end{array} \right\} \quad \text{For any A that is not a literal.}$$

$$\{x \Rightarrow \neg \diamondsuit A\} \longrightarrow \left\{ \begin{array}{lll} x & \Rightarrow & \square y \\ y & \Rightarrow & \neg A \end{array} \right\}$$

$$\{x \Rightarrow A\,\mathcal{U}\,B\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & y\,\mathcal{U}\,B \\ y & \Rightarrow & A \end{array}\right\} \qquad \text{For any A that is not a literal.}$$

$$\{x \Rightarrow A\,\mathcal{U}\,B\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & A\,\mathcal{U}\,y \\ y & \Rightarrow & B \end{array}\right\} \qquad \text{For any B that is not a literal.}$$

$$\{x \Rightarrow \neg(A\,\mathcal{U}\,B)\} \longrightarrow \left\{\begin{array}{lll} y & \Rightarrow & (\neg B)\,\mathcal{W}\,(\neg A \wedge \neg B) \end{array}\right\}$$

$$\{x \Rightarrow A\,\mathcal{W}\,B\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & y\,\mathcal{W}\,B \\ y & \Rightarrow & A \end{array}\right\} \qquad \text{For any A that is not a literal.}$$

$$\{x \Rightarrow A\,\mathcal{W}\,B\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & A\,\mathcal{W}\,y \\ y & \Rightarrow & B \end{array}\right\} \qquad \text{For any B that is not a literal.}$$

$$\{x \Rightarrow \neg(A\,\mathcal{W}\,B)\} \longrightarrow \left\{\begin{array}{lll} y & \Rightarrow & (\neg B)\,\mathcal{U}\,(\neg A \wedge \neg B) \end{array}\right\}$$

Then any temporal operators, applied to literals, that do not occur in the normal form are removed as follows (where y is a new proposition).

$$\{x \Rightarrow \Box A\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & A \\ x & \Rightarrow & y \\ y & \Rightarrow & \bigcirc A \\ y & \Rightarrow & \bigcirc y \end{array}\right\} \qquad \text{For A a literal.}$$

$$\{x \Rightarrow A\,\mathcal{U}\,B\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & \Diamond B \\ x & \Rightarrow & A \vee B \\ x & \Rightarrow & y \vee B \\ y & \Rightarrow & \bigcirc(A \vee B) \\ y & \Rightarrow & \bigcirc(y \vee B) \end{array}\right\} \qquad \text{For A, B that are literals.}$$

$$\{x \Rightarrow A\,\mathcal{W}\,B\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & A \vee B \\ x & \Rightarrow & y \vee B \\ y & \Rightarrow & \bigcirc(A \vee B) \\ y & \Rightarrow & \bigcirc(y \vee B) \end{array}\right\} \qquad \text{For A, B that are literals.}$$

Next we use renaming on formulae whose right hand side has disjunction as its main operator but may not be in the correct form where y is a new proposition and D is a disjunction of formulae.

$$\{x \Rightarrow D \vee A\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & D \vee y \\ y & \Rightarrow & A \end{array}\right\} \qquad \text{For any A that is not a literal, or whose main operator is not } K_i \text{ or } \neg K_i.$$

$$\{x \Rightarrow D \vee K_i A\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & D \vee K_i y \\ y & \Rightarrow & A \end{array}\right\} \qquad \text{For any A that is not a literal.}$$

$$\{x \Rightarrow D \vee \neg K_i A\} \longrightarrow \left\{\begin{array}{lll} x & \Rightarrow & D \vee \neg K_i \neg y \\ y & \Rightarrow & \neg A \end{array}\right\} \qquad \text{For any A that is not a literal.}$$

Finally rewrite formulae containing no temporal operators whose right hand side is a disjunction of literals or modal literals into rule form.

$$\{x \Rightarrow D\} \quad \longrightarrow \quad \{\ \mathbf{true} \ \Rightarrow \ \neg x \vee D\ \} \quad \text{For any proposition } x.$$

Thus, the above transformations are applied until the formula is in the normal form (see [19] for further details).

4 Resolution for Temporal Logics of Knowledge

Here we consider the resolution rules for the temporal logic of knowledge $KL_{(1)}$. To simplify notation we shall write the single modal operator K_1 as K. The extension of this system into its multi-modal version is considered in Section 7.

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 $\Box p$ with $\Diamond \neg p$.

4.1 Initial Resolution

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

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

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

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

4.2 Modal Resolution

During modal resolution we apply the following rules which are based on the modal resolution system introduced by Mints [29]. Firstly we are allowed to resolve a literal or modal literal and its negation.

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

Secondly we can resolve the formulae Kl and $K\neg l$ as we cannot know something and know its negation.

$$\text{[MRES2]} \qquad \begin{array}{rcl} \mathbf{true} & \Rightarrow & D \vee Kl \\ \mathbf{true} & \Rightarrow & D' \vee K\neg l \\ \hline \mathbf{true} & \Rightarrow & D \vee D' \end{array}$$

Next, as we have the T axiom $Kp \Rightarrow p$, we can resolve between formulae such as Kl and $\neg l$ giving the following rule.

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

Finally, we have the following rules which involve pushing the external K operator into one of the rules to allow us to resolve, for example, $\neg Kl$ with l

$$\text{[MRES4a]} \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & D \vee \neg Kl \\ \textbf{true} & \Rightarrow & D' \vee l \\ \hline \textbf{true} & \Rightarrow & D \vee \mathrm{mod}\,(D') \end{array}$$

$$\text{[MRES4b]} \quad \begin{array}{rcl} \textbf{true} & \Rightarrow & D \vee K\neg l_1 \\ \textbf{true} & \Rightarrow & D' \vee l_1 \vee l_2 \\ \hline \textbf{true} & \Rightarrow & D \vee \mathrm{mod}\,(D') \vee Kl_2 \end{array}$$

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

Definition 10 We define a function $\mathrm{mod}\,(D)$, defined on disjunctions of literals or modal literals D, as follows.

$$\begin{array}{rcl} \mathrm{mod}\,(A \vee B) & = & \mathrm{mod}\,(A) \vee \mathrm{mod}\,(B) \\ \mathrm{mod}\,(Kl) & = & Kl \\ \mathrm{mod}\,(\neg Kl) & = & \neg Kl \\ \mathrm{mod}\,(l) & = & \neg K \neg l \end{array}$$

These last two resolution rules require explanation. We explain the rule MRES4a below. The justification for MRES4b is similar. Recall, there is an implicit K operator surrounding each rule. We are resolving the first rule in MRES4a, as it is, with the second rule having distributed the external K over the implication. Thus, when we resolve $\neg Kl$ with Kl, we must adjust the other disjuncts of the second rule to show that K has been distributed. In more detail we consider the right hand sides of the rules given, i.e. $D \vee \neg Kl$ and $D' \vee l$. Rewriting as implications we have $\neg D \Rightarrow \neg Kl$ and $\neg D' \Rightarrow l$. Recall that each of these global rules is surrounded by an implicit K operator therefore the second rule can be rewritten as $K(\neg D' \Rightarrow l)$ or $K\neg D' \Rightarrow Kl$. Now D' is a disjunction of modal literals or literals i.e. $D' = m_1 \vee m_2 \vee \ldots$ so $K\neg D' = K\neg m_1 \wedge K\neg m_2 \wedge \ldots$. Now we can resolve the $\neg Kl$ and Kl on the right hand side of the implication obtaining $\neg D \wedge (K\neg m_1 \wedge K\neg m_2 \wedge \ldots) \Rightarrow \textbf{false}$. Rewriting as a disjunction we have $D \vee \neg K\neg m_1 \vee \neg K \neg m_2 \vee \ldots$.

Since, in S5, we have the theorems

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

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

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

$$[\text{MRES5}] \quad \frac{\textbf{true} \Rightarrow L \vee Kl_1 \vee Kl_2 \vee \ldots}{\textbf{true} \Rightarrow L \vee l_1 \vee l_2 \vee \ldots}$$

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

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

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

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

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

$$[\text{SRES3}] \quad \frac{P \Rightarrow \bigcirc\textbf{false}}{\textbf{true} \Rightarrow \neg P}$$

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 $\Box\neg P$

4.4 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 one of the following rules.

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

4.5 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 $A \Rightarrow \bigcirc\Box\neg l$. 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 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 global rules, which taken together force $\neg l$ to always be satisfied.

The temporal resolution rule is given by

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

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 $\neg l$.

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.

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

4.6 The Temporal Resolution Algorithm

Given any temporal formula ψ to be shown unsatisfiable the following steps are performed.

1. Translate ψ into a set of SNF_K rules ψ_s.

2. Perform initial resolution until either
   (a) false is derived - terminate noting ψ unsatisfiable; or
   (b) no new resolvents are generated - continue at step 3.
3. Perform modal and step resolution (including simplification and subsumption) until either
   (a) false is derived - terminate noting ψ unsatisfiable; or
   (b) no new resolvents are generated - continue to step 4.
4. Select an eventuality from the right hand side of a $\Diamond$–rule within ψ_s, for example $\Diamond l$. Search for loops in $\neg l$ and generate the appropriate resolvents. If there are no eventualities remaining unchecked, go to step 6.
5. If any new formulae have been generated, translate the resolvents into SNF_K, add them to the rule-set and go to step 2, otherwise continue to step 4.
6. Terminate declaring ψ satisfiable.

5 Examples

1. First we prove the purely modal formula $K(p \Rightarrow q) \Rightarrow (Kp \Rightarrow Kq)$, the K axiom for normal modal logics. We negate obtaining $K(p \Rightarrow q) \wedge Kp \wedge \neg Kq$ and rename obtaining the following.

1.	**start**	$\Rightarrow$	f	
2.	**true**	$\Rightarrow$	$\neg f \vee Ka$	
3.	**true**	$\Rightarrow$	$\neg f \vee Kp$	
4.	**true**	$\Rightarrow$	$\neg f \vee \neg Kq$	
5.	**true**	$\Rightarrow$	$\neg a \vee \neg p \vee q$	
6.	**true**	$\Rightarrow$	$\neg f \vee \neg Ka \vee \neg Kp$	[4, 5 *MRES4a*]
7.	**true**	$\Rightarrow$	$\neg f \vee \neg Kp$	[2, 6 *MRES1*]
8.	**true**	$\Rightarrow$	$\neg f$	[3, 7 *MRES1*]
9.	**start**	$\Rightarrow$	**false**	[1, 6 *IRES1*]

2. Next we prove a purely temporal formula that involves the application of the temporal resolution rule, namely $\Box p \Rightarrow \neg\Diamond\neg p$. We first negate obtaining $\Box p \wedge \Diamond\neg p$ and rename to give the following.

1.	**start**	$\Rightarrow$	f	
2.	**true**	$\Rightarrow$	$\neg f \vee p$	
3.	**true**	$\Rightarrow$	$\neg f \vee t$	
4.	t	$\Rightarrow$	$\bigcirc t$	
5.	t	$\Rightarrow$	$\bigcirc p$	
6.	f	$\Rightarrow$	$\Diamond\neg p$	
7.	f	$\Rightarrow$	$\neg t \,\mathcal{W}\, \neg p$	[4, 5, 6 *TRES*]
8.	**true**	$\Rightarrow$	$\neg f \vee \neg t \vee \neg p$	[7 SNF_K]
9.	**true**	$\Rightarrow$	$\neg f \vee \neg p$	[3, 8 *MRES1*]
10.	**true**	$\Rightarrow$	$\neg f$	[2, 9 *MRES1*]

11.	**start**	$\Rightarrow$	**false**	$[1, 10$ *IRES1*$]$

3. Finally we prove a formula that involves both modal and temporal resolution $\Box K\neg p \Rightarrow \neg K\neg \Box\neg p$. We first negate obtaining $\Box K\neg p \wedge K\neg \Box\neg p$ and rename obtaining the following.

1.	**start**	$\Rightarrow$	f	
2.	**true**	$\Rightarrow$	$\neg x \vee K\neg p$	
3.	**true**	$\Rightarrow$	$\neg f \vee x$	
4.	**true**	$\Rightarrow$	$\neg f \vee t$	
5.	**true**	$\Rightarrow$	$\neg f \vee Ky$	
6.	t	$\Rightarrow$	$\bigcirc t$	
7.	t	$\Rightarrow$	$\bigcirc x$	
8.	y	$\Rightarrow$	$\Diamond p$	
9.	**true**	$\Rightarrow$	$\neg x \vee \neg p$	$[2$ *MRES5*$]$
10.	t	$\Rightarrow$	$\bigcirc \neg p$	$[7, 9$ *SRES2*$]$
11.	y	$\Rightarrow$	$\neg t \mathcal{W} p$	$[6, 8, 10$ *TRES*$]$
12.	**true**	$\Rightarrow$	$\neg y \vee p \vee \neg t$	$[11$ $SNF_K]$
13.	**true**	$\Rightarrow$	$\neg f \vee p \vee \neg t$	$[5, 12$ *MRES3*$]$
14.	**true**	$\Rightarrow$	$\neg f \vee p$	$[4, 13$ *MRES1*$]$
15.	**true**	$\Rightarrow$	$\neg f \vee \neg x$	$[2, 14$ *MRES3*$]$
16.	**true**	$\Rightarrow$	$\neg f$	$[3, 15$ *MRES1*$]$
17.	**start**	$\Rightarrow$	**false**	$[1, 16$ *IRES1*$]$

6 Correctness

6.1 The Normal Form

We can show that the normal form preserves satisfiability so that detecting unsatisfiability in the set of rules implies the original formula is unsatisfiable.

Theorem 11 Let ϕ be a well formed formula in KL_n and $\tau(\phi) = \Box^* \bigwedge_i T_i$ where T_i is the set of rules translated into SNF_k. If ϕ is satisfiable so is $\tau(\phi)$.

This result can be established in a similar manner to the way the normal form theorem in standard temporal logic is proved [19].

6.2 Soundness

Theorem 12 (Soundness) If T, a set of rules in SNF_K, has a refutation by the procedure described above, then it is unsatisfiable.

Soundness can easily be established by showing that given a satisfiable set of formulae, applying each resolution rule preserves satisfiability.

6.3 Completeness

The proof of completeness is based on that given in [32]. We construct a graph of the set of SNF_K rules that has two types of edge representing the modal and temporal dimensions. We show that an empty graph corresponds to an unsatisfiable set of rules and then that an unsatisfiable set of rules has a refutation by the resolution method presented in this paper.

New Propositions

As a technical device we add any new variables required for temporal resolution into the rule-set at the start of the proof to avoid the problem of adding new variables during the proof. Thus for each sometime rule of the form $A \Rightarrow \Diamond l$ we add the new variable w_l (meaning informally *waiting for* l) and the two rules from temporal resolution

$$\begin{array}{rcl} \mathbf{true} & \Rightarrow & \neg A \lor w_l \lor l \\ w_l & \Rightarrow & \bigcirc(w_l \lor l) \end{array}$$

that do not involve the loop. The full justification of this is is not given here but can be found in [32]. Basically by adding these rules at the start of the proof the set of rules obtained has a model (where w_l has the meaning we require, i.e. $\Box(w_l \Leftrightarrow \bigcirc\Diamond l)$) if and only if the original rule-set does.

Graph Construction

Let T be a set of SNF_K rules. Given T, we construct a finite directed graph $G = (N, E_K, E_T)$, for T where N is the set of nodes, E_K is the set of modal edges, representing knowledge, and E_T is the set of temporal edges. A node, $n = (V, Y)$, in G is a pair where V and Y are constructed as follows. In [32] V was a valuation of all the propositions in T. Here, however we have modal as well as temporal formulae and as we want to be able to construct edges representing the modal dimension, for any proposition p occurring in T we allow V to contain (consistent) subsets of all the literals or modal literals that can be constructed from p namely the formulae $\{p, \neg p, Kp, \neg Kp, K\neg p, \neg K \neg p\}$. Thus we construct all possible sets of formulae containing p or its negation, Kp or its negation and $K\neg p$ or its negation. Next we reduce the number of these sets that we must consider by using the axioms of S5. Thus we cannot have a set containing Kp and $\neg p$, or $K\neg p$ and p from the T axiom or Kp and $K\neg p$ as we cannot both know a proposition and its negation. For each proposition $p \in T$ this leaves the following four sets

$$V_p = \{\{Kp, \neg K \neg p, p\}, \{K\neg p, \neg Kp, \neg p\}, \{\neg K \neg p, \neg Kp, p\}, \{\neg K \neg p, \neg Kp, \neg p\}\}.$$

To construct V we take the union of a member of each V_p for each proposition p in T, i.e.

$$V = \cup_{p \in T} \; a \in V_p.$$

Nodes are pairs (V, Y) where Y is a subset of the literals that occur on the right hand side of sometime rules in T.

Delete any node $n = (V, Y)$ such that for some K-rule of the form

$$\mathbf{true} \Rightarrow \bigvee_i m_i$$

then there is no m_i such that $m_i \in V$. Informally this step deletes any nodes that do not *immediately* satisfy the set of K-rules. Recall that rules have an implicit K

operator surrounding them from the definition of $\square^*$ so when we construct edges in the graph representing the knowledge at each node we can only draw edges to nodes that satisfy the set of K-rules. Hence we delete those nodes unsatisfied by this set of rules.

Next we push the external K operator into each rule and delete nodes that do not satisfy the new set of rules. Consider any literal rule whose right hand side consists of a single literal, for example $\textbf{true} \Rightarrow l$. By pushing in the external K-operator this rule is equivalent to $\textbf{true} \Rightarrow Kl$ so we delete any nodes where Kl is not satisfied. Next consider any modal rule whose right hand side consists of a single literal disjoined with one or more modal literals, for example $\textbf{true} \Rightarrow l \lor Ka \lor \neg Kb \lor \neg Kc$. By pushing in the external K-operator this rule is equivalent to $\textbf{true} \Rightarrow Kl \lor Ka \lor \neg Kb \lor \neg Kc$ so any node that does not satisfy this rule is deleted (recall in S5 $\neg KK\neg p \Leftrightarrow \neg K\neg p$ and $Kp \Leftrightarrow \neg K\neg Kp$). Note we could also obtain $\textbf{true} \Rightarrow \neg K\neg l \lor Ka \lor \neg Kb \lor \neg Kc$ but as the original rule implies this rule (i.e. the original rule subsumes this rule) we ignore it. Finally consider a K-rule with more than one literal disjoined on the right hand side, for example $\textbf{true} \Rightarrow l_1 \lor l_2 \lor Ka$. We must consider all possible ways of pushing in the K operator into this rule obtaining $\textbf{true} \Rightarrow Kl_1 \lor \neg K\neg l_2 \lor Ka$ or $\textbf{true} \Rightarrow \neg K\neg l_1 \lor Kl_2 \lor Ka$. Nodes that don't satisfy these additional rules are deleted.

Next delete any nodes (V, Y) such that for any sometime rule $A \Rightarrow \Diamond l$ it is not the case that if $V \models A$ then $l \in Y$. Informally if the left hand side of a sometime rule is satisfied then the eventuality must be contained in the set of eventualities in that node.

Next we construct the knowledge edges (E_K edges) between the undeleted nodes as follows. Given node, $n = (V, Y)$, we construct E_K edges to any node $n' = (V', Y')$ as follows:

1. if $Kl \in V$ then $V' \models l$; and
2. $Kl \in V \Leftrightarrow Kl \in V'$ and $\neg Kl \in V \Leftrightarrow \neg Kl \in V'$ for each literal l.

Step 1 ensures that for any modal literal in V of the form Kl, l is satisfied in V' and in step 2 that the set of modal literals in both nodes remains the same.

Now we construct the temporal edges. Given a node (V, Y), let B' be the largest subset of the global rules such that V satisfies the literals on the left hand sides of each rule. Let C be the set of clauses on the right hand side of the rules in B' and V' be a set of literals and modal literals from a node in G that satisfies these clauses. Let $Y' \subseteq Y$ be the set of literals not satisfied by V. Let Y'' be the set of literals obtained from the right hand side of the sometimes rules where V' satisfies the left hand side. Edges are constructed from (V, Y) to (V', Y''') for each V' and $Y''' = Y' \cup Y''$ and these are the only edges out of (V, Y).

The set of *initial nodes* is identified by those nodes (V, Y) where the V satisfies the set of clauses on the right hand side of the set of initial rules and Y is the set of literals from the right hand side of the largest set of sometime rules whose left hand side are satisfied by V. The *behaviour graph* for a set of SNF rules T is the set of nodes and edges reachable from the initial nodes by either E_K or E_T edges.

Given a behaviour graph for a set of rules T carry out the following deletions. Delete any node $n = (V, Y)$ and any edges into or out of n as follows.

- If a node has no temporal edges leading from it delete this node and all edges into it.
- If a node (V, Y) contains an eventuality $l \in Y$ and l is neither satisfied by V nor is there a node reachable from (V, Y) by following temporal (E_T) edges only whose valuation satisfies l then (V, Y) is deleted.

The resulting graph is known as the *reduced behaviour graph* for T.

Lemma 13 The sets of nodes disallowed during the construction of V_p, namely $\{Kp, \neg K\neg p, \neg p\}$, $\{Kp, K\neg p, p\}$, $\{Kp, K\neg p, \neg p\}$ and $\{\neg Kp, K\neg p, p\}$ are unsatisfiable.

Proof Sets containing both Kp and $\neg p$ (respectively $K\neg p$ and p) are unsatisfiable because applying the T axiom $Kp \Rightarrow p$ (respectively $K\neg p \Rightarrow \neg p$) to Kp (respectively $K\neg p$) we can infer p (respectively $\neg p$) which contradicts with $\neg p$ (respectively p).

Lemma 14 If $G = (N, E_K, E_T)$ is a reduced behaviour graph for a set of rules then the set of edges E_K form an equivalence relation, i.e., are reflexive, transitive and symmetric.

Proof First note that E_K must be reflexive as any nodes that do not have a reflexive edge have been deleted during construction of the behaviour graph and by disallowing nodes containing both Kl and $\neg l$. To show transitivity take any nodes $n = (V, Y) \in N$ and $n' = (V', Y') \in N$ where $(n, n') \in E_K$. From condition 2 of adding edges between nodes we know that the set of modal literals in V and V' are the same, so the knowledge set for V and V' must also be the same. As each node is reflexive, $V \models K\_\mathrm{set}(V)$. Let $n'' = (V'', Y'')$ be any node with an edge from n' to n'' i.e. $(n', n'') \in E_K$. Now as the set of modal literals in V' and V'' are the same by the construction of the graph the knowledge set for V' and V'' must also be the same. Hence the set of modal literals in V and V'' are also be the same, as are the knowledge sets for V and V''. As n'' is also reflexive $V'' \models K\_\mathrm{set}(V'')$ so as $V'' \models K\_\mathrm{set}(V)$ and the sets of modal literals for V and V'' are the same there must be an edge between n and n'' also so the subgraph is transitive. Symmetry is similar as the sets of modal literals for two nodes $n = (V, Y)$ and $n' = (V', Y')$ such that $(n, n') \in E_K$ are the same so the knowledge sets for V and V' are the same. Hence $V \models K\_\mathrm{set}(V)$ and also $V \models K\_\mathrm{set}(V')$ so for any edge $(n, n') \in E_K$ there must be an edge $(n', n) \in E_K$. Hence the sets of nodes reachable via the relations in E_K are reflexive, transitive and symmetric, i.e. an equivalence relation and can be used to construct the R relation.

Proposition 15 A set of rules T in SNF_K is unsatisfiable if and only if its reduced behaviour graph is empty.

Proof We start by showing the if part. The construction of nodes in the behaviour graph generate all possible states the system may be in and any nodes reachable from the initial nodes in the reduced behaviour graph can be used to construct a model for T by unwinding through the temporal edges to construct timelines and using modal edges to reconstruct the equivalence relations.

We begin by justifying our choice of sets of literals and modal literals for each proposition $p \in T$ for each node (V, Y). ¿From Lemma 13 any nodes disallowed during the construction of V_p for each p are unsatisfiable so could not form part of a model. Further, each node must satisfy the clause on the right hand side of each K-rule so the deletion of nodes that do not satisfy the K-rules does not remove any models. Finally to take account of the external K operator we push K into each rule and ensure that each node satisfies this set of rules.

Secondly, infinite paths unwinding through the temporal edges starting from an initial node give a sequence of propositional valuations for our timelines by extracting the literals from each V. By construction of the graph, this sequence satisfies the conditions for constructing timelines for T apart from the conditions concerning the satisfaction of eventualities (and none infinite paths). The reconstruction of the R relation from G will construct equivalence classes from the construction of the E_K edges in G.

If the unreduced behaviour graph is empty then there are no nodes that directly satisfy the set of rules T, i.e. without considering the satisfaction of eventualities (or none infinite paths). There must be no reachable nodes from the set of initial nodes and as we have tried to construct every possible state for the set of rules T then T must be unsatisfiable.

If the unreduced behaviour graph is not empty however not all sets of nodes reachable from the initial nodes can be used to construct models of T. If a node n has no temporal successors then there are no infinite paths through that node. So any models of T must arise from a path through the graph with n deleted. Also if n contains an eventuality l then any path through that node which is to yield a model of T must satisfy l either at n or somewhere later in the path, i.e. by unwinding through the temporal edges. Hence we can apply the second deletion criterion without discarding any potential models. The "if" part follows.

We now show the only if part. Assume that the reduced behaviour graph is non-empty. We know the set of initial nodes is non-empty because the reduced behaviour graph is defined to be the set of nodes reachable from the initial nodes. We will construct a model of T. We can construct a model by unwinding through the temporal edges to obtain timelines and then reconstruct the modal edges between points in timelines by relating to nodes in the reduced behaviour graph.

For the modal dimension we must show that for any $Kl \in n$ in the reduced behaviour graph all nodes $n' = (V', Y')$ such that $(n, n') \in E_K$ satisfy $V' \models l$ and for each $\neg Kl \in n$ there exists some $n' = (V', Y')$ such that $(n, n') \in E_K$ and $V' \models \neg l$ where the edges between all nodes reachable from n form an equivalence relation. We ensure that for any modal literal $Kl \in n$, l is satisfied in all nodes $(n, n') \in E_K$ by construction of the knowledge edges E_K. Further recall that the set of K-rules in T have an external K and every node must lead to one that satisfies these rules. This is achieved by deleting nodes that do not immediately satisfy these rules during the construction of the behaviour graph.

Finally we must check that from each node containing $\neg Kl$ we can reach a node containing $\neg l$. Assume G contains a node $n = (V, E) \in N$ that contains a formula $\neg Kl$, i.e. $\neg Kl \in V$ and no node $n' = (V', E')$ is reachable from n (via E_K edges) such that $\neg l \in V'$. Firstly note that V must contain l because if V contained $\neg l$, as the E_K relation is reflexive, then $\neg Kl$ can be satisfied by V itself. Next note that V cannot contain Kl as it contains $\neg Kl$ and nodes containing Kl and $\neg Kl$ are not permitted as they are unsatisfiable. By construction of the graph each node must satisfy each K-rule in the rule-set, plus each rule with the external K pushed into it. Assume first that the literals and modal literals in V without l, i.e. $V\backslash\{l\}$, satisfy the set of K-rules with the K-operator pushed into each rule. Hence there must be a node $n' = (V', E')$ such that V' is the same as V except it contains $\neg l$ rather than l. As V' contains the same modal literals as V there must be an edge from n to n'. Hence $\neg Kl$ can be satisfied in a reachable node and our original assumption was wrong. Next we assume that l must be in V to satisfy a rule in T. We consider three cases.

- Assume that l is in V as T contains the rule **true** $\Rightarrow l$. To satisfy this rule every node in G contains l. By pushing in the external K the set of rules including the pushed rules must contain the rule **true** $\Rightarrow Kl$ so each node in G must also contain Kl. No node can contain both Kl and $\neg Kl$ hence l cannot exist because of a rule **true** $\Rightarrow l$.

- Assume that l is in V from satisfying a rule **true** $\Rightarrow D \vee l$ where D is a disjunction of modal literals and where V does not satisfy D. By pushing in the external K operator we obtain the rule **true** $\Rightarrow D \vee Kl$ hence n must also contain Kl. As no node can contain both Kl and $\neg Kl$ the node n cannot exist because of satisfying l in a rule **true** $\Rightarrow D \vee l$. Note if V satisfied D the case is as described above — a node $n' = (V', E')$ must exist such that V' is the same as V except it contains $\neg l$ rather than l and hence $\neg Kl$ is satisfiable.

- Assume that l is in V from satisfying a rule **true** $\Rightarrow D \vee l$ where D is a disjunction of literals or modal literals and where n does not satisfy D. We let $D = L \vee M$ where L is a disjunction of literals and M is a disjunction of modal literals. As V does not satisfy D, V must satisfy $\neg D$ therefore V satisfies $\neg L$ and $\neg M$. Now for any node $n' = (V', E')$ such that $(n, n') \in E_K$, $\neg Kl \in V'$ and V' must satisfy $\neg M$ by construction of the graph. However if V' satisfies L (and therefore D) and $\neg l$ then the rule **true** $\Rightarrow D \vee l$ is satisfied. Further $\neg Kl$ is satisfied as required. The only situation when this does not occur is when $(\neg Kl \wedge \neg M) \Rightarrow \neg L$, i.e., to satisfy the rule, **true** $\Rightarrow L \vee M \vee l$, l must hold in each node when both $\neg M$ and $\neg Kl$ do. Consider any node n' such that $(n, n') \in E_K$. The node n' must satisfy both $\neg Kl$ and $\neg M$, by construction of the graph, and therefore must also contain l. Hence there is no node reachable from n by E_K edges containing $\neg l$. However having pushed the K-operator into the rules each node must satisfy **true** $\Rightarrow Kl \vee M \vee \neg K \neg L$. Here no reachable node contains L so $\neg K \neg L$ is unsatisfiable, M is unsatisfiable by assumption, so n must contain Kl. However we have assumed n contains $\neg Kl$ so n does not satisfy the set of pushed rules and must be deleted.

Next we check the E_K edges form an equivalence relation. This is shown in Lemma 14.

Then we unwind through the temporal dimension. We unwind through the temporal dimension starting at an initial node n_0 and selecting a path that satisfies each eventuality in the initial node in turn ignoring any eventualities that have been satisfied on the way. This must be possible because if each eventuality was not able to be satisfied in a reachable node then the node must have been deleted by the second deletion criterion. Once all the eventualities from the initial node have been satisfied at some node, let n_1 be a successor of this node. There must be a successor as we have deleted any terminal nodes. The path through n_1 is extended until each eventuality in n_1 is satisfied one by one. Again take a successor node at this point and call it n_2. This construction continues until we reach a successor node $n_i = n_j$ for some $i > j$ that we have reached before. This must eventually happen as the graph is finite. Let Q be the path obtained from the unwinding above between n_i and n_j. Then the path obtained from unwinding up to the node n_i followed by an infinite cycle of the path Q has the property that for each node in the path each eventuality is satisfied by some node later in the path. Recall from the construction of the graph if an eventuality e has not been satisfied in a node then it must be contained in the set of eventualities in any successor nodes. Hence if any eventuality e has not been satisfied by the time we reach some n_k it either must be satisfied in n_k or must appear in the set of eventualities in n_k and be satisfied in the next portion of path.

Thus we can construct infinite sequences of states where all eventualities are satisfied through the temporal edges and the modal edges form equivalence relations so the construction of timelines and reconstruction of the agent accessibility relation means that from the construction of a non-empty behaviour graph we can construct a model for T, i.e. T is satisfiable.

Lemma 16 Let T be a set of SNF$_K$ rules and let T' be obtained from T by adding some initial, global or K-rules whose propositions are already in T. Then the behaviour graph of T' is a subgraph of the behaviour graph of T.

Proof We note that any node in the behaviour graph for T' will also be in the behaviour graph for T. Take any node n in T'. Then n has to immediately satisfy the set of rules T plus some extra rules. As it (immediately) satisfies the rules in T it must also occur in T.

Lemma 17 If the unreduced behaviour graph for a set of SNF$_K$ rules T is empty then a contradiction can be obtained by applying resolution rules IRES1, IRES2, MRES1, MRES2 MRES3, MRES4a or MRES4b to rules in or derived from T.

Proof If the behaviour graph is empty then by Proposition 15 the set of rules T is unsatisfiable.

Assume that T' is the set of rules T plus the result of pushing the external K-operator into any K-rule. Then the set of rules T' is unsatisfiable using classical resolution between literals or modal literals and their negations, modal resolution between modal literals Kl and $K\neg l$ or modal resolution between Kl and $\neg l$. That is, by applying the resolution rules IRES1, IRES2, MRES1 (that resolves a formula and its negation) MRES2 or MRES3 we can detect a contradiction. By conjoining the set of K-rules and rewriting the right hand side in DNF

$$\textbf{true} \Rightarrow \bigvee_i D_i$$

each node must satisfy D_i for some i. If D_i contains a literal or modal literal and its negation, $K\neg l$ and Kl, or Kl and $\neg l$ by applying resolution using rules MRES1, MRES2 or MRES3 we can add new rules and exclude this disjunct. Otherwise each D_i must cause a contradiction with the right hand sides of the initial rules and must be excluded. Thus we can use the resolution rules IRES1, IRES2, MRES1, MRES2 and MRES3 to derive a contradiction.

However in the resolution system the external K-operator is not pushed into the rules so we must make sure that any resolvents generated after K has been pushed into each modal rule can be produced by applying MRES1, MRES2, MRES3, MRES4a or MRES4b. First consider the single literal $\mathbf{true} \Rightarrow l$. Pushing K into this rule we obtain $\mathbf{true} \Rightarrow Kl$. This can be resolved with the rules $\mathbf{true} \Rightarrow \neg l \vee D$, $\mathbf{true} \Rightarrow K\neg l \vee D$ or $\mathbf{true} \Rightarrow \neg Kl \vee D$ obtaining the resolvent $\mathbf{true} \Rightarrow D$ by applying MRES3, MRES2 or MRES1 respectively. The rule $\mathbf{true} \Rightarrow l$ can also be resolved with the same three rules and produce the same resolvent by applying MRES1, MRES3 and MRES4a respectively.

Next we examine modal rules containing a single literal, for example $\mathbf{true} \Rightarrow l \vee D'$ where D' is a disjunction of modal literals. Pushing K into this rule and obtaining $\mathbf{true} \Rightarrow Kl \vee D'$ we may resolve it with $\mathbf{true} \Rightarrow \neg l \vee D$, $\mathbf{true} \Rightarrow K\neg l \vee D$ or $\mathbf{true} \Rightarrow \neg Kl \vee D$ where D is a disjunction of literals or modal literals by applying rules MRES3, MRES2 or MRES1 and obtaining the resolvent $\mathbf{true} \Rightarrow D \vee D'$. The original rule $\mathbf{true} \Rightarrow l \vee D'$ may also be resolved with each of the rules using the resolution rules MRES1, MRES3, MRES4a respectively to produce the same resolvent.

Finally we consider the case where there is more than one literal in a K-rule. Without loss of generality we consider a rule with two literals $\mathbf{true} \Rightarrow l_1 \vee l_2 \vee D'$ where D' is a disjunction of modal literals (or **false**). Pushing in a K operator we obtain two rules $\mathbf{true} \Rightarrow Kl_1 \vee \neg K\neg l_2 \vee D'$ and $\mathbf{true} \Rightarrow \neg K\neg l_1 \vee Kl_2 \vee D'$. Now at least one of these rules can be resolved with the rules $\mathbf{true} \Rightarrow \neg l_i \vee D$, $\mathbf{true} \Rightarrow K\neg l_i \vee D$ or $\mathbf{true} \Rightarrow \neg Kl_i \vee D$ for $i = 1, 2$ where D is a disjunction of literals or modal literals by applying rules MRES3, MRES2 or MRES1. We consider the resolvents for resolving with $\mathbf{true} \Rightarrow Kl_1 \vee \neg K\neg l_2 \vee D'$; the other case is similar. This rule may be resolved with $\mathbf{true} \Rightarrow \neg l_1 \vee D$, $\mathbf{true} \Rightarrow K\neg l_1 \vee D$, $\mathbf{true} \Rightarrow \neg Kl_1 \vee D$ or $\mathbf{true} \Rightarrow K\neg l_2 \vee D$. The resolvents obtained are $\mathbf{true} \Rightarrow D \vee \neg K\neg l_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee \neg K\neg l_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee \neg K\neg l_2 \vee D'$ and $\mathbf{true} \Rightarrow Kl_1 \vee D \vee D'$ respectively. These resolvents, or resolvents that imply these resolvents can be generated from resolving the original rule with each of these rules to obtain $\mathbf{true} \Rightarrow D \vee l_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee l_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee \neg K\neg l_2 \vee D'$ and $\mathbf{true} \Rightarrow Kl_1 \vee D \vee D'$ using the resolution rules MRES1, MRES3, MRES4a and MRES4b respectively.

Theorem 18 (Completeness) If T a set of rules in SNF_K is unsatisfiable then it has a refutation by the procedure described above.

Let T be an unsatisfiable set of SNF_K rules. The proof proceeds by induction on the number of nodes in the behaviour graph of T. If the (unreduced) behaviour graph is empty then by by Lemma 17 we can obtain a contradiction by applying resolution rules IRES1, IRES2, MRES1, MRES2, MRES3, MRES4a or MRES4b.

Now suppose the behaviour graph G is non-empty. By Proposition 15 the reduced behaviour graph must be empty so there must be a node than can be deleted from G as described above.

If a terminal node (V, Y) exists, consider the set of global rules B' whose left hand side satisfy the valuation V. Then from the construction of the graph the set of clauses from the right hand side of B' must be unsatisfiable. In the resolution system this represents a series of applications of step resolution (SRES1 or SRES2) between global rules or global and literal rules (the rule MRES5 may need to be first applied) which lead to a rule with **false** on the right hand side, i.e. $X \Rightarrow \bigcirc\mathbf{false}$ where $V \models X$. This is rewritten as

$$\mathbf{true} \quad \Rightarrow \quad \neg X$$

by applying the rule SRES3. Adding this rule to T giving T' and constructing the behaviour graph for T', no edges will be incident on (V, Y) because we have added the rule $\mathbf{true} \Rightarrow \neg X$. So any edges out of any node must lead to a node that satisfies $\neg X$. As $V \models X$ there can be no edges into (V, Y) in T'. So (V, Y) becomes unreachable and the graph for T' is a strict subset of the graph for T. By induction we assume that T' has a refutation and so must T.

Otherwise, if no terminal node exists there must be a node n that contains an eventuality l, where l is not satisfied in any node reachable from n. If N' is the set of nodes reachable from n then any edges out of a node in N' lead to a node that is also in N'. For each node $n = (V, Y)$ in N' the set of global rules or literal rules (having applied MRES5) whose left hand side is satisfied by V are combined to give $A_n \Rightarrow \bigcirc B_n$ for $n \in N'$. To show this is a loop in $\neg l$ we must check two conditions.

- For each $n \in N'$ we must have $\models B_n \Rightarrow \neg l$. Let V be a set of literals and modal literals from a node in G that satisfies B_n. By the construction of the behaviour graph there is a temporal successor of n in G of the form (V, Y). By assumption this node is also in N' and therefore $V \not\models l$. So $\neg l$ is a logical consequence of B_n and $B_n \Rightarrow \neg l$.

- For each $n \in N'$ we must have $\models B_n \Rightarrow \bigvee_{n' \in N'} A_{n'}$. Let V be a set of literals and modal literals from a node in G that satisfies B_n. By the construction of G since $V \models B_n$ there is an edge from n to a node $n' \in N'$ whose valuation is V. Since $n' \in N'$ by assumption $V \models A_{n'}$. Hence $\models B_n \Rightarrow \bigvee_{n' \in N'} A_{n'}$ as required.

We can use the set of rules $A_n \Rightarrow \bigcirc B_n$ for resolution with each eventuality l occurring in T.

Consider any node $n = (V, Y) \in N'$ which contains l as an eventuality. Let L be defined as

$$L = \bigvee_{n \in N'} A_n$$

Note that $V \models L$, $V \not\models l$ and $V \models w_l$. Either there is an edge $e \in E_T$ from some node (V', Y') into n such that $l \in Y'$ and $V' \not\models l$ or T contains a rule $A \Rightarrow \Diamond l$ where $V \models A$ and $V \not\models l$. For the former we must have $V' \models w_l$. Applying the temporal resolution rule adds the resolvent $w_l \Rightarrow \bigcirc(l \vee \neg L)$ to the set of global rules in T. Now V' satisfies the left hand side of this rule, i.e. $V' \models w_l$ but V does not satisfy the disjunction on the right hand side $(l \vee \neg L)$ so the resulting behaviour graph does nor contain e. Otherwise for the latter we have $A \Rightarrow \Diamond l$, $V \models A$ and $V \not\models l$. Then resolving the eventuality rule with the loop we obtain $\mathbf{true} \Rightarrow \neg A \vee l \vee \neg L$ as one of the resolvents. Now n doesn't satisfy this resolvent and so n must be deleted from G. In either case n either becomes unreachable or is deleted. So, the behaviour graph for T' is a strict subset of that for T and by induction we assume that as T' has a refutation so must T.

7 Resolution in a Multi-Agent Context

In this section we consider the extension of the resolution proof rules from KL_1 to KL_n. We only consider the modal resolution rules as the rules for initial, step and temporal resolution contain no modal operators.

The rule MRES1 holds as long as the modal literal $K_i l$ and its negation $\neg K_i l$ being resolved refer to the same K_i. The rule MRES3 is easily extended as we resolve a modal literal K_i with a literal $\neg l$. MRES5 is can be extended so that the disjunction of modal literals $Kl_1 \vee Kl_2 \vee \ldots$ in the hypothesis is $K_i l_1 \vee K_j l_2 \vee \ldots$ for any $i, j \in Ag$.

For MRES2 either we can extend the given rule to the following or allow a more flexible version of MRES5.

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

This is sound as the first rule implies $\textbf{true} \Rightarrow D \vee l$ and the second implies $\textbf{true} \Rightarrow D' \vee \neg l$. Resolving these we obtain $\textbf{true} \Rightarrow D \vee D'$ as required. As an alternative a more generally applied version of MRES5 can be used followed by an application of MRES1. The more general version of MRES5 is given below where D is a disjunction of literals or modal literals.

$$\text{MRES5'} \quad \begin{array}{lcl} \textbf{true} & \Rightarrow & D \vee K_i l \\ \hline \textbf{true} & \Rightarrow & D \vee l \end{array}$$

Soundness follows from applying the T axiom to the hypothesis.

Considering MRES4a we need a more complex resolvent in the multi-agent setting where x is a new proposition.

$$\text{MRES4a'} \quad \begin{array}{lcl} \textbf{true} & \Rightarrow & D \vee \neg K_i l \\ \textbf{true} & \Rightarrow & D' \vee l \\ \hline \textbf{true} & \Rightarrow & D \vee \neg K_i \neg x \\ \textbf{true} & \Rightarrow & \neg x \vee D' \end{array}$$

A similar rule is defined for MRES4b. The justification is as before, i.e. we need to push a K_i operator into the second of the hypotheses to resolve $\neg K_i l$ with $K_i l$. However, in the multi-agent setting the formulae $\neg K_j \neg p \Leftrightarrow \neg K_i K_j \neg p$ and $K_j p \Leftrightarrow \neg K_i \neg K_j p$ are not theorems so we must introduce a new variable x to rename D' in the resolvents. Note if D' is just a disjunction of literals the original MRES4a rule can be used to avoid the introduction of new variables. For an illustration of this see the use of the original MRES4a rule in the muddy children example given below.

For the completeness proof we construct a behaviour graph with a set of edges, E_{K_i} for each modal operator i, that is $G = (N, E_{K_1}, E_{K_2}, \ldots, E_T)$. Instead of constructing nodes from the union a member of each V_p for each p in T we use the union a member of each V_{p_i} (where consistent) for each p in T and $i \in Ag$. The set V_{p_i} is just V_p where each K is replaced by K_i. The use of MRES2' is justified as nodes containing $K_i l$ and $K_j \neg l$ are disallowed as they must contain both l and $\neg l$ and are therefore inconsistent. Otherwise the proof of completeness is similar to the above.

7.1 Muddy Children Example

To illustrate the resolution system in the multi-agent case we consider the *muddy children problem* a well known problem in reasoning about knowledge. We use a version taken from [14] page 4. A variant on this problem, known as the *wisest man puzzle*, is given in [26]. A tableau based proof for this variant is given in [41].

> Imagine n children playing together. .... Now it happens during their play that some of the children, say k of them, get mud on their foreheads. Each can see the mud on others but not on his own forehead. Along comes the father, who says, "At least one of you has mud on your forehead," thus expressing a fact known to each of them before he spoke (if $k > 1$). The father then asks the following question, over and over: "Does any of you know whether you have mud on your own forehead?" Assuming that all the children are perceptive, intelligent, truthful, and that they answer simultaneously, what will happen?
> There is a "proof" that the first $k-1$ times he asks the question, they will all say "No," but then the k^{th} time the children with muddy foreheads will all answer "Yes."

We consider the two person case and use m_1 to show that child one has a muddy forehead and m_2 to show that child two has a muddy forehead. The following rules show that if a child's head is muddy it stays muddy and if a child's head is not muddy it stays not muddy.

$$\begin{array}{rrcl} 1. & m_1 & \Rightarrow & \bigcirc m_1 \\ 2. & m_2 & \Rightarrow & \bigcirc m_2 \\ 3. & \neg m_1 & \Rightarrow & \bigcirc \neg m_1 \\ 4. & \neg m_2 & \Rightarrow & \bigcirc \neg m_2 \end{array}$$

Next, if a child has a muddy forehead then the other children know it is muddy and if a child has a forehead that is not muddy all the other children know it is not muddy.

$$\begin{array}{rlcl} 5. & \mathbf{true} & \Rightarrow & \neg m_1 \lor K_2 m_1 \\ 6. & \mathbf{true} & \Rightarrow & \neg m_2 \lor K_1 m_2 \\ 7. & \mathbf{true} & \Rightarrow & m_1 \lor K_2 \neg m_1 \\ 8. & \mathbf{true} & \Rightarrow & m_2 \lor K_1 \neg m_2 \end{array}$$

The father announces that at least one of the children's foreheads is muddy. Thus the first child knows this and the second child knows this and the first knows the second knows etc. As our SNF_K rules hold in each accessible state this is the same as just saying

$$9. \quad \mathbf{true} \quad \Rightarrow \quad m_1 \lor m_2$$

If a child knows his head is muddy then at all future moments he knows his head is muddy. This can be written into SNF_K as follows where the new proposition a represents $K_1 m_1$ and b represents $K_2 m_2$.

$$\begin{array}{rrcl} 10. & a & \Rightarrow & \bigcirc a \\ 11. & b & \Rightarrow & \bigcirc b \\ 12. & \mathbf{true} & \Rightarrow & \neg a \lor K_1 m_1 \\ 13. & \mathbf{true} & \Rightarrow & a \lor \neg K_1 m_1 \\ 14. & \mathbf{true} & \Rightarrow & \neg b \lor K_2 m_2 \end{array}$$

$$15.\quad \textbf{true} \Rightarrow b \vee \neg K_2 m_2$$

We take the case where both children's foreheads are initially muddy.

$$\begin{array}{lll} 16. & \textbf{start} & \Rightarrow & m_1 \\ 17. & \textbf{start} & \Rightarrow & m_2 \end{array}$$

We use the new variables x, y and z to denote times 0, 1 and 2.

$$\begin{array}{llll} 18. & \textbf{start} & \Rightarrow & x \\ 19. & x & \Rightarrow & \bigcirc y \\ 20. & y & \Rightarrow & \bigcirc z \end{array}$$

As each child speaks at the same time to answer whether he knows the colour of his spot we need rules to denote that, for example, at time 1 (where y holds) each child knows it is time 1 (i.e. knows y).

$$\begin{array}{llll} 21. & \textbf{true} & \Rightarrow & \neg x \vee K_1 x \\ 22. & \textbf{true} & \Rightarrow & \neg x \vee K_2 x \\ 23. & \textbf{true} & \Rightarrow & \neg y \vee K_1 y \\ 24. & \textbf{true} & \Rightarrow & \neg y \vee K_2 y \\ 25. & \textbf{true} & \Rightarrow & \neg z \vee K_1 z \\ 26. & \textbf{true} & \Rightarrow & \neg z \vee K_2 z \end{array}$$

At time 1 each child does not the the colour of his spot and will answer "No."

$$\begin{array}{llll} 27. & \textbf{true} & \Rightarrow & \neg y \vee \neg K_1 m_1 \\ 28. & \textbf{true} & \Rightarrow & \neg y \vee \neg K_2 m_2 \end{array}$$

Finally we are trying to prove that at time 2 (when z holds) both children know the colour of their spots. To obtain a contradiction we must add the negation of this to the set of rules.

$$29.\quad \textbf{true} \Rightarrow \neg z \vee \neg K_1 m_1 \vee \neg K_2 m_2$$

The proof commences as follows.

30.	**true**	$\Rightarrow$	$\neg y \vee \neg K_2 \neg m_1$	[9, 28 MRES4a]
31.	**true**	$\Rightarrow$	$\neg y \vee m_1$	[7, 30 MRES1]
32.	**true**	$\Rightarrow$	$\neg y \vee \neg K_1 \neg m_2$	[9, 27 MRES4a]
33.	**true**	$\Rightarrow$	$\neg y \vee m_2$	[8, 32 MRES1]
34.	**true**	$\Rightarrow$	$\neg z \vee \neg a \vee \neg K_2 m_2$	[12, 29 MRES1]
35.	**true**	$\Rightarrow$	$\neg z \vee \neg a \vee \neg b$	[14, 34 MRES1]
36.	a	$\Rightarrow$	$\bigcirc(\neg z \vee \neg b)$	[10, 35 SRES2]
37.	$(a \wedge b)$	$\Rightarrow$	$\bigcirc \neg z$	[11, 36 SRES1]
38.	$(a \wedge b \wedge y)$	$\Rightarrow$	$\bigcirc$**false**	[20, 37 SRES1]
39.	**true**	$\Rightarrow$	$\neg a \vee \neg b \vee \neg y$	[38 SRES3]
40.	**true**	$\Rightarrow$	$\neg K_1 m_1 \vee \neg b \vee \neg y$	[13, 39 MRES1]
41.	**true**	$\Rightarrow$	$\neg K_1 m_1 \vee \neg K_2 m_2 \vee \neg y$	[15, 40 MRES1]
42.	**true**	$\Rightarrow$	$\neg K_1 y \vee \neg K_2 m_2 \vee \neg y$	[31, 41 MRES4a]
43.	**true**	$\Rightarrow$	$\neg K_1 y \vee \neg K_2 y \vee \neg y$	[33, 42 MRES4a]
44.	**true**	$\Rightarrow$	$\neg K_2 y \vee \neg y$	[23, 43 MRES1]
45.	**true**	$\Rightarrow$	$\neg y$	[24, 44 MRES1]
46.	x	$\Rightarrow$	$\bigcirc$**false**	[19, 45 SRES2]
47.	**true**	$\Rightarrow$	$\neg x$	[46 SRES3]
48.	**start**	$\Rightarrow$	**false**	[18, 47 IRES1]

8 Related Work

The work we have presented is a resolution method for a temporal logic of knowledge. Although resolution methods have been described for both modal logics [2, 3, 9, 6, 13, 16, 29, 30, 31] and temporal logics [1, 5, 39] the only method for logics with both dimensions we know about is that in [20]. This work has the same mechanism for the temporal dimension as presented here but differs elsewhere. The normal form in [20] allows both temporal and modal operators in the same rules while the approach here is to separate the two types of rules so that interaction between the two dimensions is via rules containing only disjunctions of literals. Further the modal resolution system given in [20] emphasises the addition of rules (and new variables) corresponding with the application of modal axioms. Here we provide a set of resolution rules that incorporate the S5 axioms (for example being allowed to resolve Kl with $\neg l$ relating to the axiom T). So we trade the easy application of rewrite-style rules that may potentially generate many new rules with the application of resolution-style rules that are more difficult to apply but produce only one resolvent with no new variables. However [20] admits temporal belief logics as well as the temporal logics of knowledge we have described here. We anticipate this will also be possible here if we amend the modal resolution rules to correspond with the modal logic we use for belief (KD45).

The temporal component of the resolution mentioned here was originally introduced in [17]. Subsequent work involved providing efficient algorithms to apply the complex temporal resolution rule [10], developing strategies to guide the search [11] as well as extending the approach to other logics [4]. Other resolution approaches for temporal logics can be found in [1, 5, 39].

Resolution for modal systems are given in [2, 3, 9, 6, 13, 16, 29, 30, 31]. These fall into two main groups, those that work in the modal logic directly [2, 29] or those that use a translation into predicate logic for example [30, 31]. Our system follows the former route and is based on that for propositional S5 modal logic given by Mints [29]. The use of new variables to represent subformulae whilst translating into the normal form and then linking this new proposition with the subformula it represents everywhere is essentially the renaming approach used in the transformation to the normal form for temporal logics [33].

Other proof methods for such logics have been based on tableau methods, for example the work on proof methods for BDI-logics given in [34, 35].

Here we combine a modal logic with a temporal logic to obtain a temporal logic of knowledge. The theoretical properties of temporal logics of knowledge have been studied extensively in [24, 25, 38]. Work has also been carried out into combining arbitrary logics, see for example the work on *fibring* in [21].

9 Conclusions and Future Work

We have presented a set of resolution proof rules for temporal logics of knowledge. We feel this is an improvement on that presented previously as the proofs for both the temporal and modal dimensions remain separate and we utilise particular resolution rules for S5 rather than potentially generating many new formulae using rewrite rules [20].

We are at present extending this system to deal with the evolution of knowledge over time which requires interaction between the temporal and modal components. In fact when time is incorporated into the muddy children problem it can be viewed as a system with synchrony and perfect recall [14] as the puzzle proceeds in rounds or steps (synchrony) as the children all answer "Yes" or "No" simultaneously and they can remember what has happened in previous steps (perfect recall). The introduction of resolution rules to incorporate the axiom for synchrony and perfect recall, $K_i \bigcirc \varphi \Rightarrow \bigcirc K_i \varphi$, means that (the slightly artificial) rules 21–26 can be dispensed with. The complexity of axiom systems for several such interactions have been studied in [24, 25, 38], and we note that such interaction increases the complexity of the logics in many cases and makes the problem undecidable in others.

A prototype version of the approach described in this paper has been implemented, based upon an extension of our earlier Prolog system [10]. An improved version is beginning to be developed, based on a C++ implementation and, with this, we expect to be able to test the approach on significantly larger examples.

References

[1] M. Abadi. *Temporal-Logic Theorem Proving.* PhD thesis, Department of Computer Science, Stanford University, March 1987.

[2] M. Abadi and Z. Manna. Modal Theorem Proving. *Lecture Notes in Computer Science*, 230:172–189, 1986.

[3] Y. Auffray and P. Enjalbert. Modal Theorem Proving: An Equational Viewpoint. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 441–445, Detroit, USA, 1989. Morgan Kaufmann.

[4] A. Bolotov and M. Fisher. A Resolution Method for CTL Branching-Time Temporal Logic. In *Proceedings of TIME-97 the Fourth International Workshop on Temporal Representation and Reasoning*, Daytona Beach, Florida, May 1997. IEEE Computer Society Press.

[5] A. Cavalli and L. Fariñas del Cerro. A Decision Method for Linear Temporal Logic. In R. E. Shostak, editor, *Proceedings of the 7th International Conference on Automated Deduction*, volume 170 of *Lecture Notes in Computer Science*, pages 113–127. Springer-Verlag, 1984.

[6] M-C. Chan. The Recursive Resolution Method for Modal Logic. *New Generation Computing*, 5:155–183, 1987.

[7] B. Chellas. *Modal Logic: An Introduction.* Cambridge University Press: Cambridge, England, 1980.

[8] J. Chomicki and D. Niwinski. On the Feasibility of Checking Temporal Integrity Constraints. *Journal of Computer and System Sciences*, 51(3):523–535, December 1995.

[9] M. Cialdea. Resolution for some First-Order Modal Systems. *Theoretical Computer Science*, 85:213–229, 1991.

[10] C. Dixon. Search Strategies for Resolution in Temporal Logics. In M. A. McRobbie and J. K. Slaney, editors, *Proceedings of the Thirteenth International Conference on Automated Deduction (CADE)*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 672–687, New Brunswick, New Jersey, July/August 1996. Springer-Verlag.

[11] C. Dixon. Temporal Resolution: Removing Irrelevant Information. In *Proceedings of TIME-97 the Fourth International Workshop on Temporal Representation and Reasoning*, Daytona Beach, Florida, May 1997. IEEE Computer Society Press.

[12] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1990.

[13] P. Enjalbert and L. Fariñas del Cerro. Modal Resolution in Clausal Form. *Theoretical Computer Science*, 65:1–33, 1989.

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

[15] R. Fagin, J. Y. Halpern, and M. Y. Vardi. What can machines know? on the properties of knowledge in distributed systems. *Journal of the ACM*, 39(2):328–376, 1992.

[16] L. Fariñas del Cerro. Resolution Modal Logics. In *Proceedings of Advanced NATO Study Institute on Logics and Models for Verification and Specification of Concurrent Systems*, pages 46–78, La Colle-sur-Loup, France, October 1984.

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

[18] M. Fisher. An Introduction to Executable Temporal Logics. *Knowledge Engineering Review*, 11(1):43–56, March 1996.

[19] M. Fisher. A Normal Form for Temporal Logic and its Application in Theorem-Proving and Execution. *Journal of Logic and Computation*, 7(4), August 1997.

[20] M. Fisher, M. Wooldridge, and C. Dixon. A resolution-based proof method for temporal logics of knowledge and belief. In *Proceedings of the International Conference on Formal and Applied Practical Reasoning (FAPR-96)*, Bonn, Germany, June 1996.

[21] D. M. Gabbay. Fibred-Semantics and the Weaving of Logics Part 1: Modal and Intuitionistic Logics. *Journal of Symbolic Logic*, 61(4):1057–1120, December 1996.

[22] J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2:37–68, 1987.

[23] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[24] J. Y. Halpern and M. Y. Vardi. The Complexity of Reasoning about Knowledge and Time: Synchronous Systems. Technical Report RJ 6097, IBM Almaden Research Center, San Jose, California, April 1988.

[25] 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.

[26] K. Konolige. A Deduction Model of Belief. Pitman/Morgan Kaufmann, 1986.

[27] S. Kraus and D. Lehmann. Knowledge, belief and time. *Theoretical Computer Science*, 58:155–174, 1988.

[28] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer-Verlag, New York, 1992.

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

[30] A. Nonnengart. Resolution-Based Calculi for Modal and Temporal Logics. In M. A. McRobbie and J. K. Slaney, editors, *Proceedings of the Thirteenth International Conference on Automated Deduction (CADE)*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 598–612, New Brunswick, New Jersey, July/August 1996. Springer-Verlag.

[31] H-J. Ohlbach. A Resolution Calculus for Modal Logics. *Lecture Notes in Computer Science*, 310:500–516, May 1988.

[32] M. Peim. Propositional Temporal Resolution Over Labelled Transition Systems. Unpublished Technical Note, Department of Computer Science, University of Manchester, 1994.

[33] D. A. Plaisted and S. A. Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, 2(3):293–304, September 1986.

[34] 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.

[35] 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.

[36] Y. Shoham. *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence.* The MIT Press, 1988.

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

[38] R. van der Meyden. Axioms for Knowledge and Time in Distributed Systems with Perfect Recall. In *Proceedings of the Ninth IEEE Symposium on Logic in Computer Science*, pages 448–457, 1994.

[39] G. Venkatesh. A Decision Method for Temporal Logic based on Resolution. *Lecture Notes in Computer Science*, 206:272–289, 1986.

[40] M. Wooldridge. Agent-Based Software Engineering. *IEE Proceedings on Software Engineering*, February 1997.

[41] M. Wooldridge and M. Fisher. A decision procedure for a temporal belief logic. In D. M. Gabbay and H. J. Ohlbach, editors, *Temporal Logic — Proceedings of the First International Conference (LNAI Volume 827)*, pages 317–331. Springer-Verlag: Heidelberg, Germany, July 1994.