A Resolution-Based Proof Method for Temporal Logics of Knowledge and Belief*

Michael Fisher   Michael Wooldridge   Clare Dixon

Department of Computing
Manchester Metropolitan University
Manchester M1 5GD
United Kingdom

{M.Fisher,M.Wooldridge,C.Dixon}@doc.mmu.ac.uk
http://www.doc.mmu.ac.uk/

Abstract. In this paper we define two logics, KL_n and BL_n, and present resolution-based proof methods for both. KL_n is a *temporal logic of knowledge*. Thus, in addition to the usual connectives of linear discrete temporal logic, it contains a set of unary modal connectives for representing the *knowledge* possessed by *agents*. The logic BL_n is somewhat similar: it is a temporal logic that contains connectives for representing the *beliefs* of agents. The proof methods we present for these logics involve two key steps. First, a formula to be tested for unsatisfiability is translated into a normal form. Secondly, a family of resolution rules are used, to deal with the interactions between the various operators of the logics. In addition to a description of the normal form and the proof methods, we present some short worked examples and proposals for future work.

1 Introduction

This paper presents two logics, called KL_n and BL_n respectively, and gives resolution-based proof methods for both. The logic KL_n is a *temporal logic of knowledge*. That is, in addition to the usual connectives of linear discrete temporal logic [4], 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 [2], which is widely recognized as a logic of *idealized knowledge* [10]. It is for this reason that we call KL_n a temporal logic of knowledge. (The properties of KL_n-like logics are studied in [11, 5], where a version of KL_n is used that contains only future time connectives.) Syntactically, the logic BL_n is identical to KL_n. It is also a temporal logic that contains connectives for representing the information possessed by a group of agents. However, the *agent modalities* in BL_n satisfy analogues of the modal axioms KD45 — a system widely accepted as a logic of *idealized belief* [10]. For this reason, we say that BL_n is a *temporal logic of belief*.

Logics such as KL_n and BL_n have been studied in both AI and mainstream computer science for some time (see, e.g., [9, 12, 11, 6, 5]). However, very little effort has been directed at developing proof methods for such logics [14]. This is perhaps

* Work partially supported under EPSRC Research Grant GR/K57282.

because of the complexity of the problem: it is proved in [11, 5] 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) indicate that practical theorem provers for such complex logics may be possible [7, 3]. In this paper, we extend the proof method for purely temporal logics described in [7] to deal with KL_n and BL_n. Specifically, we present a clausal resolution method for BL_n, which we show to be sound and refutation complete. A simple extension to this method gives a sound and refutation complete proof method for KL_n. This work represents the first attempt to extend this resolution method to multi-modal logics, and is the first resolution method for temporal logics of knowledge and belief.

The remainder of this paper is structured as follows. Section 2 gives complete formal definitions of the two logics. Section 3 defines SNF*, a normal form for KL_n and BL_n. The proof method itself is presented in section 4. Worked examples, illustrating the proof method, are presented in section 5, and some concluding remarks appear in section 6.

Notation: If $\mathcal{L}$ is a logical language, then we write $Form(\mathcal{L})$ for the set of (well-formed) formulae of $\mathcal{L}$. We use the lowercase Greek letters φ, ψ, and χ as meta-variables ranging over formulae of the logical languages we consider.

2 Temporal Logics of Knowledge and Belief

In this section, we formally present the syntax and semantics of two logics: BL_n is a *temporal logic of belief*, and KL_n is a *temporal logic of knowledge*. These logics actually share a common syntax, which we shall call the language $\mathcal{L}$. Note that due to space restrictions, our presentation of the syntax and semantics of the language, though complete, is of necessity somewhat terse.

First, note that $\mathcal{L}$ is not a quantified language. We shall thus build formulae from a set $\Phi = \{p, q, r, \ldots\}$ of *primitive propositions*. In fact, the language $\mathcal{L}$ generalizes classical propositional logic, and thus it contains the standard propositional connectives $\neg$ (not) and $\vee$ (or); the remaining connectives ($\wedge$ (and), $\Rightarrow$, (implies), and $\Leftrightarrow$ (if, and only if)) are assumed to be introduced as abbreviations in the usual way. We use temporal connectives that can refer both to the *past* and to the *future*. With respect to the future, we use two basic connectives: $\bigcirc$ (for 'next'), and $\mathcal{U}$ (for 'until'). With respect to the past, we use '$\odot$' (for 'last') and '$\mathcal{S}$' (for 'since'). We explain these connectives in detail below. The temporal connectives are interpreted over a *flow of time* that is linear, discrete, bounded in the past, and infinite in the future. An obvious choice for such a flow of time is $(I\!N, <)$, i.e., the natural numbers ordered by the usual 'less than' relation.

With respect to belief/knowledge connectives, we assume a set $Ag = \{1, \ldots, n\}$ of *agents*. We then build an indexed set of unary modal connectives $\{[i] \mid i \in Ag\}$, where a formula $[i]\varphi$ is to be read (in BL_n) as 'agent i believes that φ', or (in KL_n) as 'agent i knows that φ'. In both cases, $\varphi \in Form(\mathcal{L})$.

2.1 Syntax

Definition 1. The set $Form(\mathcal{L})$ of (well-formed) formulae of $\mathcal{L}$ is defined by the following rules:

1. (Primitive propositions are formulae): if $p \in \Phi$ then $p \in Form(\mathcal{L})$;
2. (Nullary connectives): **false** $\in Form(\mathcal{L})$, **true** $\in Form(\mathcal{L})$;
3. (Unary connectives): if $\varphi \in Form(\mathcal{L})$ then $\neg\varphi \in Form(\mathcal{L})$, $\bigcirc\varphi \in Form(\mathcal{L})$, $\circledcirc\varphi \in Form(\mathcal{L})$, and $(\varphi) \in Form(\mathcal{L})$;
4. (Binary connectives): if $\varphi, \psi \in Form(\mathcal{L})$, then $\varphi \vee \psi \in Form(\mathcal{L})$, $\varphi\,\mathcal{U}\,\psi \in Form(\mathcal{L})$, and $\varphi\,\mathcal{S}\,\psi \in Form(\mathcal{L})$;
5. (Agent modalities): if $\varphi \in Form(\mathcal{L})$ and $i \in Ag$ then $[i]\varphi \in Form(\mathcal{L})$.

Definition 2. If $p \in \Phi$ then both p and $\neg p$ are *literals*. If l is a literal and $i \in Ag$, then $[i]l$ and $\neg[i]l$ are *agent literals*.

2.2 Semantics

Definition 3. It is assumed that the world may be in any of a set S of *states*. We generally use s (with annotations, e.g., s_0, s', ...) to denote a state.

Definition 4. A *timeline*, l, is an infinitely long, linear, discrete sequence of states, indexed by the natural numbers. For convenience, we define a timeline l to be a function $l : I\!N \to S$. Let *TLines* be the set of all timelines.

Note that timelines correspond to the *runs* of Halpern *et al* [11, 5].

Definition 5. A point, p, is a pair $p = (l, u)$, where $l \in TLines$ is a time line and $u \in I\!N$ is a temporal index into l. Let the set of all points (over S) be *Points*.

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

Definition 7. A *model*, M, for $\mathcal{L}$ is a structure $M = \langle TL, R_1, \ldots, R_n, \pi \rangle$, where: $TL \subseteq TLines$ is a set of timelines; R_i, for all $i \in Ag$, is an agent accessibility relation over *Points*, i.e., $R_i \subseteq Points \times Points$; and $\pi : Points \times \Phi \to \{T, F\}$ is a valuation.

As usual, we define the semantics of the language via the satisfaction relation '$\models$'. For $\mathcal{L}$, this relation holds between pairs of the form $\langle M, p \rangle$ (where M is a model and $p \in Points$), and $\mathcal{L}$-formulae. The rules defining the satisfaction relation are given in Figure 1. Satisfiability and validity in $\mathcal{L}$ are defined in the usual way.

Knowledge and belief models: We shall now define two classes of $\mathcal{L}$-models: KL_n-models are models of *knowledge*, and BL_n-models are models of *belief*.

Definition 8. An $\mathcal{L}$-model $M = \langle TL, R_1, \ldots, R_n, \pi \rangle$ is a KL_n-model iff R_i is an equivalence relation, for all $i \in Ag$.

$\langle M,(l,u)\rangle \models \mathbf{true}$		
$\langle M,(l,u)\rangle \models p$	iff	$\pi((l,u),p) = T$ (where $p \in \Phi$)
$\langle M,(l,u)\rangle \models \neg\varphi$	iff	$\langle M,(l,u)\rangle \not\models \varphi$
$\langle M,(l,u)\rangle \models \varphi \vee \psi$	iff	$\langle M,(l,u)\rangle \models \varphi$ or $\langle M,(l,u)\rangle \models \psi$
$\langle M,(l,u)\rangle \models [i]\varphi$	iff	$\forall l' \in TL, \forall v \in I\!N,$ if $((l,u),(l',v)) \in R_i$, then $\langle M,(l',v)\rangle \models \varphi$
$\langle M,(l,u)\rangle \models \bigcirc\varphi$	iff	$\langle M,(l,u+1)\rangle \models \varphi$
$\langle M,(l,u)\rangle \models \odot\varphi$	iff	$(u > 0)$ and $\langle M,(l,u-1)\rangle \models \varphi$
$\langle M,(l,u)\rangle \models \varphi\, \mathcal{U}\, \psi$	iff	$\exists v \in I\!N$ such that $(v \geq u)$ and $\langle M,(l,v)\rangle \models \psi$, and $\forall w \in I\!N,$ if $(u \leq w < v)$ then $\langle M,(l,w)\rangle \models \varphi$
$\langle M,(l,u)\rangle \models \varphi\, \mathcal{S}\, \psi$	iff	$\exists v \in I\!N$ such that $(v < u)$ and $\langle M,(l,v)\rangle \models \psi$, and $\forall w \in I\!N,$ if $(v < w < u)$ then $\langle M,(l,w)\rangle \models \varphi$

Fig. 1. Semantics of $\mathcal{L}$

It should be clear that as agent accessibility relations in KL_n models are equivalence relations, the axioms of the normal modal system S5 are valid in the class of KL_n models.

Theorem 9.

$$\models_{KL_n} [i](\varphi \Rightarrow \psi) \Rightarrow ([i]\varphi \Rightarrow [i]\psi) \tag{1}$$
$$\models_{KL_n} [i]\varphi \Rightarrow \neg[i]\neg\varphi \tag{2}$$
$$\models_{KL_n} [i]\varphi \Rightarrow \varphi \tag{3}$$
$$\models_{KL_n} [i]\varphi \Rightarrow [i][i]\varphi \tag{4}$$
$$\models_{KL_n} \neg[i]\varphi \Rightarrow [i]\neg[i]\varphi \tag{5}$$

These axioms are called K, D, T, 4, and 5, respectively. The system S5 is widely recognized as the logic of idealized *knowledge*, and for this reason we say KL_n is a *temporal logic of knowledge*. (The future-time component of KL_n corresponds to Halpern and Vardi's logic $KL_{(m)}$ [11], also known as $S5_n^U$ in [5, p283], where a complete axiomatization is given.) We now define *belief models*.

Definition 10. An $\mathcal{L}$-model $M = \langle TL, R_1, \ldots, R_n, \pi\rangle$ is a BL_n-model iff for all $i \in Ag$, we have:

1. (Euclidean:) $\forall p, p', p'' \in Points$, if $(p,p') \in R_i$ and $(p,p'') \in R_i$, then $(p',p'') \in R_i$;
2. (Serial:) $\forall p \in Points$, $\exists p' \in Points$ such that $(p,p') \in R_i$; and
3. (Transitive:) $\forall p, p', p'' \in Points$, if $(p,p') \in R_i$ and $(p',p'') \in R_i$, then $(p,p'') \in R_i$.

It is well-known that the axioms K, D, 4, and 5 from normal modal logic are valid in models whose accessibility relations satisfy properties (1)–(3) of Definition 10. However, axiom T (formula 3 above) is not. Axiom T is generally taken to be the axiom that distinguishes knowledge from belief: it says that if an agent knows φ, then φ is true. As this axiom is not BL_n-valid, we say that BL_n is a *temporal logic of belief*.

Derived Temporal Connectives Other standard temporal connectives are introduced as abbreviations, in terms of $\mathcal{S}$, $\mathcal{U}$ and $\odot$:

$$
\begin{array}{rclrcl}
\Diamond\varphi & \stackrel{\text{def}}{=} & \mathbf{true}\,\mathcal{U}\varphi & \blacklozenge\varphi & \stackrel{\text{def}}{=} & \mathbf{true}\,\mathcal{S}\psi \\
\Box\varphi & \stackrel{\text{def}}{=} & \neg\Diamond\neg\varphi & \blacksquare\varphi & \stackrel{\text{def}}{=} & \neg\blacklozenge\neg\varphi \\
\varphi\mathcal{W}\psi & \stackrel{\text{def}}{=} & \varphi\mathcal{U}\psi \vee \Box\varphi & \varphi\mathcal{Z}\psi & \stackrel{\text{def}}{=} & \varphi\mathcal{S}\psi \vee \blacksquare\varphi \\
\bullet\varphi & \stackrel{\text{def}}{=} & \neg\odot\neg\varphi & \mathbf{start} & \stackrel{\text{def}}{=} & \bullet\mathbf{false}
\end{array}
$$

We now informally consider the meaning of the temporal connectives. First, consider the two basic future-time connectives: $\bigcirc$ and $\mathcal{U}$. The $\bigcirc$ connective means ‘at the next time’. Thus $\bigcirc\varphi$ will be satisfied at some time if φ is satisfied at the *next* time. The $\mathcal{U}$ connective means ‘until’. Thus $\varphi\mathcal{U}\psi$ will be satisfied at some time if ψ is satisfied at that time or some time in the future, and φ is satisfied at all times until the time that ψ is satisfied. Of the derived connectives, $\Diamond$ means ‘either now, or at some time in the future’. Thus $\Diamond\varphi$ will be satisfied at some time if either φ is satisfied at that time, or some later time. The $\Box$ connective means ‘now, and at all future times’. Thus $\Box\varphi$ will be satisfied at some time if φ is satisfied at that time and at all later times. The binary $\mathcal{W}$ connective means ‘unless’. Thus $\varphi\mathcal{W}\psi$ will be satisfied at some time if either φ is satisfied until such time as ψ is satisfied, or else φ is always satisfied. Note that $\mathcal{W}$ is similar to, but weaker than, the $\mathcal{U}$ connective; for this reason it is sometimes called ‘weak until’.

The past-time connectives are similar: $\odot$ and $\bullet$ are true at some moment if their arguments were true at the previous moment. The difference between them is that, since the model of time underlying the logic is bounded in the past, the beginning of time is a special case: $\odot\varphi$ will always be false when interpreted at the beginning of time, whereas $\bullet\varphi$ will always be true at the beginning of time. The $\blacklozenge$ connective is a past-time version of $\Diamond$. Thus $\blacklozenge\varphi$ will be true at some time if φ was true at some previous moment in time. The $\blacksquare$ connective is a past-time version of $\Box$. Thus $\blacksquare\varphi$ will be true at some time if φ was true at all previous moments in time. The $\mathcal{S}$ connective mirrors $\mathcal{U}$, and so $\varphi\mathcal{S}\psi$ will be true now if ψ was true at some previous moment in time, and φ has been true since then; $\mathcal{Z}$ is the same, but allowing for the possibility that the second argument was never true. Thus $\mathcal{Z}$ mirrors $\mathcal{W}$. Finally, a temporal operator that takes no arguments can be defined which is true only at the first moment in time: this operator is ‘**start**’.

3 A Normal Form for KL_n and BL_n

The proof methods we present in section 4 depend on formulae being rewritten into a normal form, which we call SNF*(after the *separated normal form* of [7, 8]). In this section, we define SNF*, and outline the procedure by which an arbitrary $\mathcal{L}$-formula may be rewritten in this form. The translation depends upon the use of *renaming* [13] to simplify formulae. We therefore begin, in the following section, by describing how renaming works.

3.1 Renaming

The basic idea of renaming is to simplify a formula φ by replacing sub-formulae of φ by new proposition symbols that act, in effect, as abbreviations for the sub-formulae they replace. In order to preserve satisfiability during this process, we must link the truth-value of a new proposition to the truth-value of the sub-formula it replaced. Enforcing this link in modal logic is complicated somewhat by the fact that a formula can be interpreted in many different states: we must ensure that the link is maintained in all states that can play a part in the interpretation of a formula. In temporal logic, we can do this by carrying out the renaming within the scope of a '$\Box$' connective. For example, in temporal logic, renaming can be used to replace a formulae such as $\Diamond(\varphi\,\mathcal{U}\,\psi)$ by $\Diamond(\varphi\,\mathcal{U}\,p) \wedge \Box(p \Leftrightarrow \psi)$, where p is a new proposition symbol. In temporal logic, the operator '$\Box$' accesses all reachable states, thus p is defined as an abbreviation for ψ in every state, and so satisfiability is preserved.

Unfortunately, in our logics, the situation is complicated yet further by the presence of two kinds of modal links: temporal ones, and those given by each agent's accessibility relation R_i. We must therefore introduce a derived operator $\Box^*$, such that $\Box^*\varphi$ means φ is satisfied in every *reachable* state — intuitively, every state that can play a part in interpreting a formula. Renaming is then carried out within the context of the $\Box^*$ operator, and thus the link between a new proposition and the sub-formula it replaces is forced across all reachable states.

In order to define the operator $\Box^*$, we must first define an operator C to capture the notion of *common knowledge* (or, in BL_n, mutual belief). This, in turn requires an operator E to capture the idea of *every agent* knowing (believing) a formula. We define E by

$$E\varphi \Leftrightarrow \bigwedge_{i\in Ag} [i]\varphi .$$

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

$$C\varphi \Leftrightarrow E(\varphi \wedge C\varphi) .$$

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

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

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

Definition 11. Let M be an $\mathcal{L}$-model and $(l,u),(l',v)$ be points in M. Then (l',v) is *reachable* from (l,u) iff either: (i) $l = l'$ and $v \geq u$; (ii) $((l,u),(l',v)) \in R_i$ for some agent $i \in Ag$; or (iii) there exists some point (l'',w) in M such that (l'',w) is reachable from (l,u) and (l',v) is reachable from (l'',w).

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

Theorem 12. *Let M be an $\mathcal{L}$-model and p,p' be points in M such that $\langle M,p\rangle \models \square^*\varphi$. Then $\langle M,p'\rangle \models \varphi$ if p' is reachable from p.*

Now, renaming of a formula such as $\diamondsuit(\varphi\,\mathcal{U}\,\psi)$ produces $\diamondsuit(\varphi\,\mathcal{U}\,p) \wedge \square^*(p \Leftrightarrow \psi)$. This theorem therefore guarantees that renaming carried out within the context of the $\square^*$ operator will preserve satisfiability.

3.2 SNF*: A Normal Form for $\mathcal{L}$

We now describe SNF*, the normal form that we use in the proof methods described in Section 4. An $\mathcal{L}$-formula in SNF* is of the form:

$$\square^* \bigwedge_{i=1}^{n} (\varphi_i \Rightarrow \psi_i)$$

where each of the '$\varphi_i \Rightarrow \psi_i$' (called *rules*) is one of the following

$$\begin{array}{lcll}
\mathbf{start} & \Rightarrow & \bigvee_{k=1}^{r} l_k & \text{(an } \textit{initial } \square\text{-rule)} \\
\bigcirc \bigwedge_{j=1}^{q} m_j & \Rightarrow & \bigvee_{k=1}^{r} l_k & \text{(a } \textit{global } \square\text{-rule)} \\
\mathbf{true} & \Rightarrow & \bigvee_{k=1}^{r} l_k & \text{(a } \textit{global} \text{ renaming rule)} \\
\mathbf{start} & \Rightarrow & \diamondsuit l & \text{(an } \textit{initial } \diamondsuit\text{-rule)} \\
\bigcirc \bigwedge_{j=1}^{q} m_j & \Rightarrow & \diamondsuit l & \text{(a } \textit{global } \diamondsuit\text{-rule)}
\end{array}$$

where each m_j, or l is a literal, and l_k is either a literal or an agent literal.

Theorem 13. *There exists a translation function $\tau : Form(\mathcal{L}) \rightarrow Form(\mathrm{SNF}^*)$ such that for any $\varphi \in Form(\mathcal{L})$, we have $\tau(\varphi)$ is satisfiable just in case φ is satisfiable.*

Proof. Full details of the translation process are rather complex, and so we simply sketch the main steps. Note that the translation is similar to that described in [8], where more details can be found. The main steps are:

1. 'Pushing' negations (converting into negation normal form).
   This involves applying transformations such as

$$\begin{aligned}\neg(\varphi \wedge \psi) &\longrightarrow \neg\varphi \vee \neg\psi \\ \neg\bigcirc\varphi &\longrightarrow \bigcirc\neg\varphi \\ \neg\Box\varphi &\longrightarrow \Diamond\neg\varphi\end{aligned}$$

   both inside and outside (but not *across*) agent modalities. This operation preserves validity.
2. Dealing with axiom K:

$$[i](\varphi \vee \psi) \;\longrightarrow\; \neg[i]\neg\varphi \vee [i]\psi$$

   Preservation of validity is obvious.
3. Removing derived temporal operators.
   This simply involves the replacement of various temporal operators by their definitions, for example

$$\varphi\,\mathcal{W}\,\psi \;\longrightarrow\; \varphi\,\mathcal{U}\,\psi \vee \Box\varphi\,.$$

   Again, this operation preserves validity.
4. Renaming embedded non-literal formulae.
   Here, the renaming procedure described above is applied exhaustively to removed embedded formulae, for example

$$\begin{aligned}\varphi\,\mathcal{U}\,\psi &\longrightarrow x\,\mathcal{U}\,y \wedge \Box^*(x \Leftrightarrow \varphi) \wedge \Box^*(y \Leftrightarrow \psi) \\ [i]\varphi &\longrightarrow [i]z \wedge \Box^*(z \Leftrightarrow \varphi)\,.\end{aligned}$$

   where φ and ψ are non-literal formulae.
   Once such transformations have been applied, each literal is guaranteed to be within the scope of at most one modal or temporal operator (apart from '$\Box^*$').
5. Removing maximal fixpoint operators, e.g. '$\Box$'.
   This involves unwinding fixpoint operators defined in temporal logic, for example

$$\Box\varphi \longrightarrow \varphi \wedge x \wedge \Box^*(\bullet x \Leftrightarrow (\varphi \wedge x))$$

   (For more detail, see [8].) Again, this step preserves satisfiability.
6. Rewriting into SNF* (effectively an extension of CNF).
   This final phase involves the use of classical transformations, analogous to those used to produce clausal form, to produce a set of SNF* rules from the formula produced so far.
   This step is a variation on the standard transformation into CNF, and so preserves validity.

Thus, the proof reduces to showing that the renaming process itself preserves satisfiability, which can be shown by observing that

- the new propositions introduced are defined in every state (see Theorem 12);
- the original formula is satisfied iff the renamed formula is satisfied (which follows from the structure of the formula resulting from renaming).

In order to ensure that renaming rules are available everywhere, formulae such as

$$\square^*(x \Leftrightarrow \varphi)$$

are represented by SNF* rules

$$\mathbf{true} \Rightarrow \neg x \vee \varphi$$
$$\mathbf{true} \Rightarrow \neg \varphi \vee x$$

which again appear within the context of an '$\square^*$' operator. Note that such rules (with '**true**' on their left-hand side) are only used for renaming in this way.

4 Resolution-Based Proof Methods for KL_n and BL_n

Before describing the resolution method in detail, we give an overview of our approach. First, recall the two basic problems associated with extending resolution to modal and multi-modal logics such as those considered in this paper:

1. Literals cannot generally be moved in and out of modal or temporal contexts. In particular, if M is a modal or temporal operator, p and $M\neg p$ cannot be resolved. Thus, the only inferences that can be made occur in particular modal or temporal contexts. For example, both p and $\neg p$ can be resolved, as, for certain types of modal operator, can Mp and $M\neg p$.
2. In many non-classical logics, particularly multi-modal logics, the operators interact in complex ways. For example, the logics considered here have transitive accessibility relations and so the axiom $M\varphi \Rightarrow MM\varphi$ holds. Thus, in addition to the problem of reasoning in the presence of modal and temporal contexts, the proof process must take into account this interaction between operators.

The resolution method described in this paper tackles these problems by:

– using the normal form SNF*, which separates out complex formulae from their contexts through the use of renaming (as described above); and
– utilizing additional translation, resolution, and simplification rules in order to represent the interaction between operators.

The translation of a formula into a normal form is particularly important. In removing formulae from their contexts and replacing them by new propositions (effectively, abbreviations), we are able to manipulate these formulae using what is essentially classical resolution, only returning the results to their contexts under specific conditions.

We can now describe the proof method. This method extends that of the underlying temporal logic [7] and consists of the following steps. To determine whether a formula $\varphi \in Form(\mathcal{L})$ is unsatisfiable:

1. Rewrite φ into SNF*.
2. Repeat
   (a) apply step resolution (effectively a form of classical resolution)
   (b) rewrite new resolvents into SNF*

(c) apply simplification and subsumption
(d) apply temporal resolution
(e) rewrite new resolvents into SNF*

until either **false** is derived or no more rules can be applied.

The process of applying temporal and step resolution rules to a set of formulae in SNF* eventually terminates. On termination, either **false** has been derived (showing that the formula is unsatisfiable) or no more rules can be applied (showing that the formula is satisfiable). As in classical resolution, simplification and subsumption procedures are applied throughout the process.

Since agent literals can never occur either on the left-hand side of an SNF* rule, or within a $\Diamond$-formula, rules containing such literals can not directly participate in temporal resolution steps. Thus, in extending the temporal resolution method of [7] to BL_n and KL_n, we need only consider the additional step resolution rules and simplification rules for handling formulae containing modal operators.

4.1 Step Resolution

The step resolution rule is simply a version of the classical resolution rule rewritten as follows.

$$\frac{\begin{array}{r}\varphi_1 \Rightarrow \psi_1 \vee l \\ \varphi_2 \Rightarrow \psi_2 \vee \neg l\end{array}}{\varphi_1 \wedge \varphi_2 \Rightarrow \psi_1 \vee \psi_2} \qquad \text{(SRES)}$$

Now, however, the 'l' above is now an atom. In particular, we can resolve agent literals via the following rule:

$$\frac{\begin{array}{r}\varphi_1 \Rightarrow \psi_1 \vee [i]m \\ \varphi_2 \Rightarrow \psi_2 \vee \neg[i]m\end{array}}{\varphi_1 \wedge \varphi_2 \Rightarrow \psi_1 \vee \psi_2} \qquad \text{(SRESa)}$$

In addition to this general rule, we add a more specific resolution rule for resolving within modal contexts:

$$\frac{\begin{array}{r}\varphi_1 \Rightarrow \psi_1 \vee [i]m \\ \varphi_2 \Rightarrow \psi_2 \vee [i]\neg m\end{array}}{\varphi_1 \wedge \varphi_2 \Rightarrow \psi_1 \vee \psi_2} \qquad \text{(MRES)}$$

This resolution rule derives from axiom (2) of the logic, (which in turn corresponds to axiom D of normal modal logic).

4.2 Simplification

The simplification rules used are similar to the temporal case, which are, in turn, similar to the classical case, consisting of both simplification and subsumption rules. The major rule required in the temporal resolution method is used when a contradiction in a state is produced:

$$\frac{\bigcirc\varphi \Rightarrow \textbf{false}}{\textbf{true} \Rightarrow \neg\varphi} \qquad \text{(SIMP1)}$$

¿From the properties of the '$\square^*$' operator, which implicitly surrounds all SNF* rules, we are able to derive the following rule:

$$\frac{\textbf{true} \Rightarrow \varphi}{\textbf{true} \Rightarrow [i]\varphi} \qquad \text{(SIMP2)}$$

where $[i]\varphi$ must again be rewritten into SNF*.

Additional simplification rules are provided by axioms (3), (4), and (5), corresponding to normal modal axioms T, 4, and 5 respectively. The first, $(SIMP3)$, is derived from axiom T.

$$\frac{\psi_1 \Rightarrow \psi_2 \ \lor \ [i]\varphi}{\psi_1 \Rightarrow \psi_2 \ \lor \ \varphi} \qquad \text{(SIMP3)}$$

Note that this rule is not used in the BL_n proof method, as axiom (3) is not valid in BL_n: it is *only* used in KL_n. The use of this rule represents the only difference between the proof method for KL_n and that for BL_n. The next simplification rule is derived from axiom (4). It can be used at most once per rule within SNF*.

$$\frac{\psi_1 \Rightarrow \psi_2 \ \lor \ [i]\varphi}{\begin{array}{l}\psi_1 \Rightarrow \psi_2 \ \lor \ [i]x \\ x \Leftrightarrow [i]\varphi\end{array}} \qquad \text{(SIMP4)}$$

This final rule, $(SIMP5)$, is derived from axiom 5. Like $(SIMP4)$, this rule is also used at most once per formula in SNF*.

$$\frac{\psi_1 \Rightarrow \psi_2 \ \lor \ \neg[i]\varphi}{\begin{array}{l}\psi_1 \Rightarrow \psi_2 \ \lor \ [i]y \\ y \Leftrightarrow \neg[i]\varphi\end{array}} \qquad \text{(SIMP5)}$$

4.3 Temporal Resolution

Rather than describe the temporal resolution rule in detail, we refer the interested reader to [7]. The basic idea is to resolve one $\Diamond$-rule with a *set* of $\square$-rules as follows[2].

$$\left.\frac{\begin{array}{c}\bigcirc\varphi_1 \Rightarrow \psi_1 \\ \bigcirc\varphi_2 \Rightarrow \psi_2 \\ \vdots \\ \bigcirc\varphi_n \Rightarrow \psi_n \\ \bigcirc\chi \Rightarrow \Diamond l\end{array}}{\begin{array}{c}\textbf{true} \Rightarrow \neg(\bigvee_{i=1}^{n} \varphi_i \land \chi) \\ \bigcirc\chi \Rightarrow (\neg \bigvee_{i=1}^{n} \varphi_i)\, \mathcal{W} l\end{array}}\right\} \text{ where } \bigvee_{i=1}^{n} \varphi_i \ \Rightarrow \ \square\neg l \qquad \text{(TRES)}$$

The resolvents produced must again be translated into SNF*.

[2] A similar rule resolving with a $\Diamond$-rule of the form **start** $\Rightarrow \Diamond l$ is also used.

4.4 Termination

Finally, if any of the following rules are produced, the original formula is unsatisfiable and the resolution process terminates.

$$\mathbf{start} \Rightarrow \mathbf{false} \qquad (\text{NULL1})$$

$$\bigcirc\mathbf{true} \Rightarrow \mathbf{false} \qquad (\text{NULL2})$$

$$\mathbf{true} \Rightarrow \mathbf{false} \qquad (\text{NULL3})$$

4.5 Correctness

Theorem 14. *The resolution method is sound, i.e., if a contradiction is derived using the resolution method, then the original formula is unsatisfiable.*

Theorem 15. *The resolution method is refutation complete, i.e., if a formula is unsatisfiable, then the resolution method will eventually derive a contradiction when applied to that formula.*

5 Worked Examples

In this section, we illustrate the proof methods for KL_n and BL_n through a number of short worked examples. (Note that, for brevity, we omit many of the rules derived but not crucial to the refutation.)

Example 1. Consider the formula

$$([1]\,\Box p) \wedge \Box\neg p$$

which is BL_n-satisfiable, but KL_n-unsatisfiable. We can transform this into SNF*, giving:

1. $\mathbf{start} \Rightarrow [1]a$
2. $\mathbf{true} \Rightarrow \neg a \vee p$
3. $\mathbf{true} \Rightarrow \neg a \vee b$
4. $\bigcirc b \Rightarrow p$
5. $\bigcirc b \Rightarrow b$
6. $\mathbf{start} \Rightarrow n$
7. $\mathbf{start} \Rightarrow \neg p$
8. $\bigcirc n \Rightarrow n$
9. $\bigcirc n \Rightarrow \neg p$

Given this, resolution can proceed as follows.

10. $\mathbf{start} \Rightarrow \neg a$ [2, 7, SRES]

At this point the refutation stalls for BL_n, as no further resolution or simplification rules can be applied. Thus the formula is BL_n-satisfiable. However, in KL_n, $(SIMP3)$ can be used as a simplification rule, producing:

11. **start** $\Rightarrow a$ [1, SIMP3]
12. **start** $\Rightarrow$ **false** [10, 11, SRES]

As $(NULL1)$ has been derived, representing **false**, the formula is KL_n-unsatisfiable.

Example 2. Consider the following formula, illustrating the inductive nature of the underlying temporal logic. This formula is valid in both BL_n and KL_n.

$$(\Diamond[i]a \wedge \Box([i]a \;\Rightarrow\; \bigcirc[i]a)) \;\Rightarrow\; \Diamond\Box[i]a$$

If we negate this, and transform it into a set of SNF* rules, we get the following (note that m is an abbreviation for $[i]a$).

1. **start** $\Rightarrow \Diamond m$
2. $\boldsymbol{\bigcirc} m \Rightarrow m$
3. **start** $\Rightarrow s$
4. $\boldsymbol{\bigcirc} s \Rightarrow s$
5. $\boldsymbol{\bigcirc}(s \wedge m) \Rightarrow \Diamond\neg m$

The resolution process then proceeds as follows.

6. $\boldsymbol{\bigcirc}(s \wedge m) \Rightarrow \neg m$ [2, 5, TRES]
7. **true** $\Rightarrow \neg s \vee \neg m$ [2, 5, TRES]
8. $\boldsymbol{\bigcirc} s \Rightarrow \neg m$ [4, 7, SRES]
9. **start** $\Rightarrow m \vee \neg s$ [1, 4, 8, TRES]
10. **start** $\Rightarrow \neg s$ [7, 9, SRES]
11. **start** $\Rightarrow$ **false** [3, 10, SRES]

Example 3. To show

$$[i]\bigcirc p \wedge [i]\bigcirc\neg p$$

is BL_n-unsatisfiable, we translate into SNF*, giving:

1. **start** $\Rightarrow [i]x$
2. **start** $\Rightarrow [i]y$
3. **true** $\Rightarrow \neg x \vee a$
4. **true** $\Rightarrow \neg y \vee b$
5. $\boldsymbol{\bigcirc} a \Rightarrow p$
6. $\boldsymbol{\bigcirc} b \Rightarrow \neg p$

A contradiction can be derived as follows.

7. $\boldsymbol{\bigcirc}(a \wedge b) \Rightarrow$ **false** [5, 6, SRES]
8. **true** $\Rightarrow \neg a \vee \neg b$ [7, SIMP1]
9. **true** $\Rightarrow \neg x \vee \neg b$ [3, 8, SRES]
10. **true** $\Rightarrow \neg x \vee \neg y$ [4, 9, SRES]
11. **true** $\Rightarrow \neg[i]x \vee [i]\neg y$ [10, SIMP2]
12. **start** $\Rightarrow [i]\neg y$ [1, 11, SRESa]
13. **start** $\Rightarrow$ **false** [2, 12, MRES]

Example 4. Finally, we consider a purely modal example:

$$\neg[i]\neg[i]p \wedge \neg p$$

which is unsatisfiable in KL_n. We first translate this into SNF*, giving

1. **start** $\Rightarrow \neg p$
2. **start** $\Rightarrow \neg[i]x$
3. **true** $\Rightarrow \neg x \vee \neg[i]p$
4. **true** $\Rightarrow x \vee [i]p$

A contradiction can be derived as follows.

5. **true** $\Rightarrow x \vee p$	[4, SIMP3]
6. **start** $\Rightarrow x$	[1, 5, SRES]
7. **start** $\Rightarrow \neg[i]p$	[3, 6, SRES]
8. **start** $\Rightarrow [i]y$	[7, SIMP5]
9. **true** $\Rightarrow \neg y \vee \neg[i]p$	[7, SIMP5]
10. **true** $\Rightarrow y \vee [i]p$	[7, SIMP5]
11. **true** $\Rightarrow \neg y \vee x$	[4, 9, SRES]
12. **true** $\Rightarrow \neg[i]y \vee [i]x$	[11, SIMP2]
13. **start** $\Rightarrow [i]x$	[8, 12, SRESa]
14. **start** $\Rightarrow$ **false**	[2, 13, SRESa]

Space precludes the presentation of further larger examples.

6 Summary

While the complexity of the decision problem for the temporal logics of knowledge and belief presented here has been studied [11], few proof methods have been developed for these logics [14]. Proof methods for multi-modal logics (e.g., [1]) have been developed, but have generally been based on tableaux methods and, moreover, have not been extended to modal *and* temporal combinations. Our work therefore represents an important step towards the mechanization of a class of logics with a wide variety of applications in distributed systems and distributed AI.

In the future, we hope to extend the proof method to deal with (limited) quantification, to implement the method, and to apply the method to the specification and verification of distributed intelligent systems. We also hope to evaluate the resolution method against tableaux proof methods for such logics [14].

References

1. L. Catach. TABLEAUX: A general theorem prover for modal logics. *Journal of Automated Reasoning*, 7:489–510, 1991.
2. B. Chellas. *Modal Logic: An Introduction.* Cambridge University Press: Cambridge, England, 1980.

3. C. Dixon, M. Fisher, and H. Barringer. A graph-based approach to temporal resolution. In D. M. Gabbay and H. J. Ohlbach, editors, *Temporal Logic — Proceedings of the First International Conference (LNAI Volume 827)*, pages 415–429, July 1994.
4. 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.
5. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. The MIT Press: Cambridge, MA, 1995.
6. 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.
7. M. Fisher. A resolution method for temporal logic. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI)*, Sydney, Australia, August 1991.
8. M. Fisher. A normal form for first-order temporal formulae. In *Proceedings of Eleventh International Conference on Automated Deduction (CADE)*. Springer-Verlag: Heidelberg, Germany, June 1992.
9. J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2:37–68, 1987.
10. 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.
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. S. Kraus and D. Lehmann. Knowledge, belief and time. *Theoretical Computer Science*, 58:155–174, 1988.
13. D. A. Plaisted and S. A. Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, 2(3):293–304, September 1986.
14. 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.

This article was processed using the LaTeX macro package with LLNCS style