A Decision Procedure for a Temporal Belief Logic

Michael Wooldridge and **Michael Fisher**

Department of Computing
Manchester Metropolitan University
Chester Street, Manchester M1 5GD
United Kingdom
{M.Wooldridge, M.Fisher}@mmu.ac.uk

Abstract. This paper presents a *temporal belief logic* called $\mathcal{L}_{TB}$. In addition to the usual connectives of linear discrete temporal logic, $\mathcal{L}_{TB}$ contains an indexed set of modal *belief* connectives, via which it is possible to represent the belief systems of resource-bounded reasoning agents. The applications of $\mathcal{L}_{TB}$ in general, and its use for representing the dynamic properties of multi-agent AI systems in particular, are discussed in detail. A tableau-based decision procedure for $\mathcal{L}_{TB}$ is then described, and some examples of its use are presented. The paper concludes with a discussion and future work proposals.

1 Introduction

Temporal logics have been shown to have many applications, in a variety of disciplines. For example: in computer science, temporal logics are used in the specification and verification of reactive systems [16]; in artificial intelligence, they are used as knowledge representation formalisms, and have proved to be a valuable tool in tackling such problems as reasoning about action [18]. 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*. Logics which contain more than one different type of modality are called *multi-modal* logics [3]. In this paper, we consider a multi-modal logic which contains connectives for representing both time and *belief*.

The obvious approach to defining the semantics of a temporal belief logic involves adapting possible worlds semantics for belief [11]: one might define a world to be a sequence of time points, so that a belief accessibility relation holds between alternative histories (cf. tensed modal logics [20]). Although such an approach is undoubtedly simple, it suffers from at least two drawbacks. The first is that, in common with all normal modal formalisations of belief, it implies that agent's beliefs are closed under logical consequence; this is the logical omniscience problem [17]. While logical omniscience is acceptable in the study of theoretically perfect believers, it is clearly at odds with any reasonable understanding of how belief works in resource-bounded reasoners. The second problem is that belief and time would *interact* in such a way as as to make the development of an automatic proof method awkward [3].

In this paper, we develop a temporal belief logic called $\mathcal{L}_{TB}$, in which the semantics of belief are not based on possible worlds, but on a simple new model of belief which is outlined in §2. The logic $\mathcal{L}_{TB}$ is then developed in §3, which also includes a discussion of its applications. Since time and belief do not interact directly in $\mathcal{L}_{TB}$, it is possible to develop a tableau-based decision procedure for $\mathcal{L}_{TB}$ as a generalisation of the temporal tableau method. Such a decision procedure is presented in §4. Some worked examples, illustrating the decision procedure, are given in §4.1. The paper closes with some comments and future work proposals.

Notational Conventions: 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, and the uppercase Greek letters Δ and Γ as meta-variables ranging over sets of formulae. To give the reader some visual clues, we generally use Δ to denote a set of beliefs, and Γ to stand for an arbitrary set of formulae. We use a VDM-style notation for manipulating sets and functions [13], and use $\emptyset$ for the empty set.

2 Belief Models

In this section we develop the formal framework which will be used in $\mathcal{L}_{TB}$ to give a semantics to belief[1]. This new framework may be used to represent the belief systems of resource bounded reasoning agents, although it is sufficiently rich that it can also be used to represent, for example, the perfect reasoners of possible worlds semantics. In the space available, we can do no more than sketch the properties of the model; for details, see [23].

The structures we use to represent belief systems are called *belief models*. A belief model representing an agent i's belief system is a pair. The first component of this pair is a set of observations that have been made about i's beliefs. These observations are expressed in some *internal language*; throughout this paper we shall call this internal language $\mathcal{L}$. In general, the internal language may be one of rules, frames, semantic nets, or some other kind of KR formalism but, for simplicity, we shall assume that $\mathcal{L}$ is a *logical* language. Thus, the first component of i's belief model is a set of $\mathcal{L}$-formulae representing observations that have been made about i's beliefs. The second component is a relation, which holds between sets of $\mathcal{L}$-formulae and $\mathcal{L}$-formulae. This relation is called a *belief extension* relation, (hereafter abbreviated to 'b.e. relation'), and it is intended to model i's reasoning ability. Let BE_i be the b.e. relation for agent i. Then the way we interpret BE_i is:

$$\textbf{if } i \text{ believes } \Delta \textbf{ and } (\Delta, \varphi) \in BE_i \textbf{ then } i \text{ also believes } \varphi.$$

It is via i's b.e. relation that we are able to make deductions about what other beliefs i has. In [23], we show how a b.e. relation that correctly describes the behaviour of an agent's belief system may be derived in a principled way. We now formally define belief models.

[1] Note that *human* belief is *not* the object of study in this paper, and in particular, no claims are made about the validity or usefulness of the model for representing human believers.

Definition 1. A *belief model*, b, is a pair $b = (\Delta, BE)$ where

– $\Delta \subseteq Form(\mathcal{L})$; and
– $BE \subseteq (powerset(Form(\mathcal{L})) \times Form(\mathcal{L}))$ is a countable non-empty binary relation between sets of $\mathcal{L}$-formulae and $\mathcal{L}$-formulae, which must satisfy the following requirements:
  1. *Reflexivity*: if $(\Delta, \varphi) \in BE$, then $\forall \psi \in \Delta, (\Delta, \psi) \in BE$;
  2. *Monotonicity*: if $(\Delta, \varphi) \in BE$, $(\Delta', \psi) \in BE$, and $\Delta \subseteq \Delta'$, then $(\Delta', \varphi) \in BE$;
  3. *Transitivity*: if $(\Delta, \varphi) \in BE$ and $(\{\varphi\}, \psi) \in BE$, then $(\Delta, \psi) \in BE$.

If $b = (\Delta, BE)$ is a belief model, then Δ is said to be its *base set*, and BE its *belief extension relation*. We now define a function *bel* which takes as its sole argument a belief model, and returns the set of $\mathcal{L}$-formulae representing the *belief set* of that model.

Definition 2.

$$bel((\Delta, BE)) \stackrel{\text{def}}{=} \{\varphi \mid (\Delta, \varphi) \in BE\}$$

Suppose b_i is a belief model which represents agent i's belief system. Then the interpretation of 'belief' in this paper is as follows:

$\varphi \in bel(b_i)$ — i believes φ	$\neg\varphi \in bel(b_i)$ — i believes $\neg\varphi$
$\varphi \notin bel(b_i)$ — i doesn't believe φ	$\neg\varphi \notin bel(b_i)$ — i doesn't believe $\neg\varphi$.

3 The Temporal Belief Logic $\mathcal{L}_{TB}$

In this section, we develop the new temporal belief logic $\mathcal{L}_{TB}$. This logic is essentially a standard linear discrete temporal logic enriched by the addition of a set of unary modal belief connectives, with a semantics given in terms of belief models, as described in the preceding section.

We let time be linear, discrete, bounded in the past, and infinite in the future, giving the temporal model $(\mathbb{N}, <)$. We take as primitive just two temporal connectives: $\bigcirc$ ('next'), and $\mathcal{U}$ ('until'). The remaining standard connectives of linear discrete future temporal logic may be derived from these.

3.1 Syntax

In the interests of simplicity, we shall restrict our attention in this paper to propositional languages. We assume an underlying classical propositional language, which we shall call $\mathcal{L}_0$. This language is defined over a set Φ of primitive propositions, and is closed under the unary connective '$\neg$' (not), and the binary connective '$\vee$' (or). The remaining connectives of classical logic ('$\wedge$' (and), '$\Rightarrow$' (implies), and '$\Leftrightarrow$' (iff)) are assumed to be introduced as abbreviations, in the standard way. $\mathcal{L}_0$ is also assumed to contain the logical constants **true** and **false**, and the usual punctuation symbols ')' and '('. Finally, note that $\mathcal{L}_{TB}$ is to be used for representing beliefs expressed in the internal language, $\mathcal{L}$. It follows that $\mathcal{L}$ must appear in $\mathcal{L}_{TB}$ somewhere. For simplicity, we shall assume that $\mathcal{L} = \mathcal{L}_{TB}$, i.e., agents are capable of having beliefs about beliefs, and about how beliefs change over time.

Definition 3. The language $\mathcal{L}_{TB}$ contains the following symbols:

1. All symbols of $\mathcal{L}_0$;
2. The set $Ag = \{1, \ldots, n\}$ of agent names;
3. The symbols ‘]’ and ‘[’;
4. The unary temporal connective $\bigcirc$, and binary temporal connective $\mathcal{U}$.

Definition 4. The set $Form(\mathcal{L}_{TB})$ of (well-formed) formulae of $\mathcal{L}_{TB}$ is defined by the following rules:

1. If $\varphi \in Form(\mathcal{L}_0)$ then $\varphi \in Form(\mathcal{L}_{TB})$;
2. If $\varphi \in Form(\mathcal{L}_{TB})$ and $i \in Ag$ then $[i]\varphi \in Form(\mathcal{L}_{TB})$;
3. If $\varphi \in Form(\mathcal{L}_{TB})$ then $\neg\varphi, \bigcirc\varphi, (\varphi) \in Form(\mathcal{L}_{TB})$;
4. If $\varphi, \psi \in Form(\mathcal{L}_{TB})$, then $\varphi \vee \psi, \varphi \mathcal{U} \psi \in Form(\mathcal{L}_{TB})$.

3.2 Semantics

Before we define the semantics of $\mathcal{L}_{TB}$, we must make a number of assumptions plain. First, we assume that an agent’s beliefs can change over time. (If we assumed that beliefs were fixed, then there would be little point in having a temporal component in the language.) Secondly, we assume that an agent’s reasoning ability, as represented in its b.e. relation, does not change with time. Although dropping this assumption would be relatively simple in terms of semantics, it would complicate the proof theory of the language considerably, and we do not consider it necessary in practice.

Models for $\mathcal{L}_{TB}$ include a valuation function, giving the truth of each primitive proposition at each time; additionally, they include a function which assigns each agent a base set of beliefs at each moment in time, and an indexed set of b.e. relations.

Definition 5. A model, M, for $\mathcal{L}_{TB}$ is a triple $M = \langle \pi, a, \{BE_i\} \rangle$, where

- $\pi : \mathbb{N} \times \Phi \to \{T, F\}$ interprets propositions at each time point;
- $a : \mathbb{N} \times Ag \to powerset(Form(\mathcal{L}_{TB}))$ assigns each agent a base set of beliefs at each time; and
- $\{BE_i\}$ is an indexed set of b.e. relations, one for each agent $i \in Ag$.

As usual, we define the semantics of the language via the satisfaction relation ‘$\models$’. For $\mathcal{L}_{TB}$, this relation holds between pairs of the form $\langle M, u \rangle$, (where M is a model and $u \in \mathbb{N}$ is a temporal index into M), and $\mathcal{L}_{TB}$-formulae. The rules defining the satisfaction relation are given in Fig. 1. Satisfiability and validity for $\mathcal{L}_{TB}$ are defined as follows: if $\varphi \in Form(\mathcal{L}_{TB})$ and there is some $\langle M, u \rangle$ such that $\langle M, u \rangle \models \varphi$, then φ is said to be *satisfiable*, otherwise φ is said to be *unsatisfiable*. If $\neg\varphi$ is unsatisfiable, then φ is *valid* (notation $\models \varphi$).

The remaining temporal connectives of $\mathcal{L}_{TB}$ are introduced as abbreviations.

$$
\begin{aligned}
\Diamond\varphi &\stackrel{\text{def}}{=} \mathbf{true}\,\mathcal{U}\,\varphi \\
\Box\varphi &\stackrel{\text{def}}{=} \neg\Diamond\neg\varphi \\
\varphi\,\mathcal{W}\,\psi &\stackrel{\text{def}}{=} \varphi\,\mathcal{U}\,\psi \vee \Box\varphi
\end{aligned}
$$

$\langle M, u\rangle \models \mathbf{true}$		
$\langle M, u\rangle \models p$	iff	$\pi(u, p) = T$ (where $p \in \Phi$)
$\langle M, u\rangle \models \neg\varphi$	iff	$\langle M, u\rangle \not\models \varphi$
$\langle M, u\rangle \models \varphi \vee \psi$	iff	$\langle M, u\rangle \models \varphi$ or $\langle M, u\rangle \models \psi$
$\langle M, u\rangle \models [i]\varphi$	iff	$\varphi \in bel((a(u, i), BE_i))$
$\langle M, u\rangle \models \bigcirc\varphi$	iff	$\langle M, u+1\rangle \models \varphi$
$\langle M, u\rangle \models \varphi \mathcal{U} \psi$	iff	$\exists v \in \mathbb{N}$ s.t. $(v \geq u)$ and $\langle M, v\rangle \models \psi$, and $\forall w \in \mathbb{N}$, if $(u \leq w < v)$ then $\langle M, w\rangle \models \varphi$

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

We now informally consider the meaning of the connectives. The formula $[i]\varphi$ is read 'agent i believes φ'; it will be satisfied if φ is present in i's belief set at the current time. The $\bigcirc$ connective means 'at the next time'. Thus $\bigcirc\varphi$ will be satisfied at some time point if φ is satisfied at the *next* time point. 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 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'.

3.3 Properties of $\mathcal{L}_{TB}$

Since the propositional connectives of $\mathcal{L}_{TB}$ have standard semantics, all propositional tautologies will be valid; additionally, the inference rule *modus ponens* will preserve validity. In short, we can use all propositional modes of reasoning in $\mathcal{L}_{TB}$. The new logic also inherits the axioms and inference rules associated with its temporal component (see, e.g., [4] for discussion). However, $\mathcal{L}_{TB}$ has some additional properties. To illustrate this, we first establish an analogue of Konolige's attachment lemma [14, pp34–35].

Theorem 6. *The set* $\{[i]\Delta, \neg[i]\Delta'\}$ *is unsatisfiable iff* $\exists\varphi \in \Delta'$ *such that* $(\Delta, \varphi) \in BE_i$.

Proof. This, and all remaining proofs, are omitted due to space restrictions; full details may be found in the associated technical report [23].

This theorem allows us to derive a number of results; for example:

Theorem 7. $\models [i]\varphi_1 \wedge \cdots \wedge [i]\varphi_n \Rightarrow [i]\varphi$, *where* $(\{\varphi_1, \ldots, \varphi_n\}, \varphi) \in BE_i$.

This theorem represents the basic mechanism for reasoning about belief systems: if it is known that i believes $\{\varphi_1, \ldots, \varphi_n\}$, and that $(\{\varphi_1, \ldots, \varphi_n\}, \varphi) \in BE_i$, then this implies that i also believes φ. Note that axiom K and the necessitation rule from classical modal logic do *not* in general hold for belief modalities in $\mathcal{L}_{TB}$, and thus $\mathcal{L}_{TB}$ does not fall prey to logical omniscience. However, $\mathcal{L}_{TB}$ *is* capable of representing logically omniscient believers [23].

3.4 Applications of $\mathcal{L}_{TB}$

Temporal belief logics such as $\mathcal{L}_{TB}$ have a number of applications. For example: formalisms for representing the time-varying properties of multi-agent systems are essential in the emerging discipline of Distributed Artificial Intelligence (DAI) [2]; epistemic temporal logics have been used by researchers in computer science to reason about distributed systems [12]; temporal belief logics have recently found a role in the specification and verification of DAI systems [22, 7]; and ultimately, temporal belief logics may even be *executed*, as in [6]. In the remainder of this section, we consider various properties of agents that may be expressed using $\mathcal{L}_{TB}$ (note that we do not consider the 'standard' axioms of belief — KD45 — in this paper; we are concerned instead with axioms in which time and belief interact).

First, consider the *persistence* of belief. Suppose that at some time an agent believes φ, then how long might the agent persist in this belief? An extreme case is that in which, when an agent comes to believe something, it always believes it:

$$[i]\varphi \Rightarrow \Box[i]\varphi. \tag{1}$$

Agents with property (1) are not very interesting from the point of view of AI. A more reasonable assumption is that beliefs persist until a contradictory belief is held:

$$(([i]\varphi) \wedge (\neg\bigcirc[i]\neg\varphi)) \Rightarrow \bigcirc[i]\varphi. \tag{2}$$

Property (2) may also be expressed as:

$$[i]\varphi \Rightarrow (([i]\varphi)\,\mathcal{W}\,([i]\neg\varphi)). \tag{3}$$

We might also like to state that if an agent believes that φ will be true at some point in the future, then at some point in the future the agent will believe φ. This gives the following three axioms.

$$[i]\bigcirc\varphi \Rightarrow \bigcirc[i]\varphi \tag{4}$$

$$[i]\Diamond\varphi \Rightarrow \Diamond[i]\varphi \tag{5}$$

$$[i]\varphi\,\mathcal{U}\,\psi \Rightarrow \Diamond[i]\psi \tag{6}$$

Kraus-Lehmann suggest that (4)–(6) describe a notion of belief closer to the sense of 'religious' belief than the everyday notion of belief as 'readiness to bet' [15, pp166-168]. This weaker notion cannot easily be axiomatized, but (7)–(9) seem reasonable properties.

$$[i]\bigcirc\varphi \Rightarrow [i]\bigcirc[i]\varphi \tag{7}$$

$$[i]\Diamond\varphi \Rightarrow [i]\Diamond[i]\varphi \tag{8}$$

$$[i]\varphi\mathcal{U}\,\psi \Rightarrow [i]\Diamond[i]\psi \tag{9}$$

Finally, consider the class of agents with the ability to affect the future, perhaps by acting in the world. Divide the set Φ of primitive propositions into two disjoint sets, Φ_e (environment propositions) and Φ_a (agent propositions), and let $\Phi_a = \Phi_{a_1} \cup \cdots \cup \Phi_{a_n}$. The idea is that Φ_e is the set of propositions whose truth or falsity is controlled by the world — propositions in this set are not affected by the actions of agents. The set Φ_a contains propositions under the control of individual agents; Φ_{a_i} is the set of the propositions under the control of agent i. Now consider the following axioms:

$$\left.\begin{array}{l}[i]\bigcirc\varphi \Rightarrow \bigcirc\varphi\\ [i]\Diamond\varphi \Rightarrow \Diamond\varphi\end{array}\right\} \qquad \text{where } \varphi \in \Phi_{a_i}. \tag{10}$$

The axioms in (10), which are related to axiom T from classical modal logic, state that an agent's beliefs about the future state of the propositions under its control are reflected in the actual future state of those propositions. Axioms like this have been used to reason about the behaviour of systems in the 'imperative future' paradigm [8, 6]; see [7, 22] for details.

4 A Decision Procedure for $\mathcal{L}_{TB}$

In this section, we present a tableau-based decision procedure for $\mathcal{L}_{TB}$. The procedure consists of two functions: a main function *structure*, and an auxiliary function *tableau*. The function *structure* takes as its input an $\mathcal{L}_{TB}$ formula φ, and systematically searches for a model of φ. If φ is satisfiable, then *structure* returns a graph from which a model for φ can be extracted; if φ is unsatisfiable, then *structure* returns an empty graph. The function *tableau* is used to check the internal consistency of states during graph generation. The algorithm draws on the tableau methods for temporal logic described by Wolper [21], Gough [10], and Ben-Ari [1, pp216–228], and in fact generalises the basic temporal tableau method. Note that the algorithm assumes that we have the belief extension relation of each agent available.

As with all tableau-based decision procedures, our procedure relies upon *alpha* and *beta* equivalences; these equivalences are defined in Fig. 2. (We assume that all input formulae are rewritten into a normal form in which negations are only applied to primitive propositions and belief modalities; see [10].)

Definition 8. If φ is an alpha-formula, (notation *is-alpha*(φ)), then its components are given by $\alpha_1(\varphi)$ and $\alpha_2(\varphi)$ respectively. If φ is a beta-formula (notation *is-beta*(φ)), then its components are given by $\beta_1(\varphi)$ and $\beta_2(\varphi)$ respectively. The *alpha closure* of a formula is given by the function α^*, which has the signature

$$\alpha^* : \mathit{Form}(\mathcal{L}_{TB}) \rightarrow \mathit{powerset}(\mathit{Form}(\mathcal{L}_{TB}))$$

α	α_1	α_2
$\varphi \wedge \psi$	φ	ψ
$\Box\varphi$	φ	$\bigcirc\Box\varphi$

β	β_1	β_2
$\varphi \vee \psi$	φ	ψ
$\Diamond\varphi$	φ	$\bigcirc\Diamond\varphi$
$\varphi\,\mathcal{U}\,\psi$	ψ	$\varphi \wedge \bigcirc(\varphi\,\mathcal{U}\,\psi)$
$\varphi\,\mathcal{W}\,\psi$	$\Box\varphi$	$\varphi\,\mathcal{U}\,\psi$

Fig. 2. Alpha and Beta Equivalences

and which is defined by

$$\alpha^*(\varphi) \stackrel{\text{def}}{=} \begin{cases} \alpha^*(\alpha_1(\varphi)) \cup \alpha^*(\alpha_2(\varphi)) & \text{if } \textit{is-alpha}(\varphi) \\ \{\varphi\} & \text{otherwise.} \end{cases}$$

The function α^* is extended to sets of formulae in an obvious way.

The internal consistency of nodes during tableau generation is established by checking whether they are *proper*.

Definition 9. If $\Gamma \subseteq Form(\mathcal{L}_{TB})$ then Γ is *proper*, (notation *proper*(Γ)) iff:

1. **false** $\notin \Gamma$;
2. If $\varphi \in \Gamma$, then $\neg\varphi \notin \Gamma$;
3. If $\{[i]\varphi_1, \ldots, [i]\varphi_n, \neg[i]\varphi\} \subseteq \Gamma$, then $(\{\varphi_1, \ldots, \varphi_n\}, \varphi) \notin BE_i$.

Note that proper sets may be unsatisfiable, but improper sets are never satisfiable; the only non-obvious part is (3), which is given by Theorem 6. We now define tableau structures (cf. [19]). Nodes are drawn from some arbitrary set *Node*.

Definition 10. A *tableau*, Υ, is a quad $\Upsilon = (N, T, l_1, l_2)$, where

– $N \subseteq Node$ is a set of nodes;
– $T \subseteq N \times N$ is a binary tree over N;
– $l_1 : N \to powerset(Form(\mathcal{L}_{TB}))$ labels each node with a set of $\mathcal{L}_{TB}$-formulae; and
– $l_2 : N \to \{o, c\}$ labels each node in N with either o (open) or c (closed).

Let *Tableaux* be the set of all tableaux. If $\Upsilon = (N, T, l_1, l_2)$ is a tableau, then let *leaves*(Υ) denote the subset of N containing the leaves of T.

The function *tableau*, which is used to check the internal consistency of states during graph generation, is given in Fig. 3. This function has one important property:

Lemma 11. *If tableau*($\{\varphi\}$) *returns a tableau with no open leaves, then* φ *is unsatisfiable.*

Definition 12. If $\Upsilon = (N, T, l_1, l_2)$ is a tableau and $n \in leaves(\Upsilon)$, then denote by *walk*(Υ, n) the sequence of nodes obtained by walking from the root n_0 of Υ to n. If *walk*(Υ, n) $= (n_0, \ldots, n_k)$ then let *walk-set*(Υ, n) denote the set $l_1(n_0) \cup \cdots \cup l_1(n_k)$.

```
function tableau(Γ : powerset(Form(L_TB))) : (N, T, l1, l2) : Tableaux
vars            n, n', n'' : Node
                flag : B
begin (* initialise *)
      create new node n
      N := {n}                        (* root *)
      T := ∅                          (* empty tree *)
      l1 := {n ↦ Γ}                   (* label root with Γ *)
      if proper(Γ) then l2 := {n ↦ o}
                    else l2 := {n ↦ c}
      repeat (* main loop *)
            flag := false             (* no leaves created *)
            for each n ∈ leaves((N, T, l1, l2)) s.t. l2(n) = o do
(* α *)           l1 := l1 † {n ↦ α*(l1(n))}
(* β *)           for each φ ∈ l1(n) s.t. is-beta(φ) do
                        create new nodes n' and n''
                        N := N ∪ {n', n''}
                        l1 := l1 † {n' ↦ (l1(n) − {φ}) ∪ {β1(φ)}}
                        l1 := l1 † {n'' ↦ (l1(n) − {φ}) ∪ {β2(φ)}}
                        T := T ∪ {(n, n'), (n, n'')}
                        if proper(l1(n')) then l2 := l2 † {n' ↦ o}
                                          else l2 := l2 † {n' ↦ c}
                        if proper(l1(n'')) then l2 := l2 † {n'' ↦ o}
                                           else l2 := l2 † {n'' ↦ c}
                        if l2(n') = o or l2(n'') = o then flag := true
                  end-for
            end-for
      until ¬flag
end-function
```

Fig. 3. Function *tableau*

We now move on to the model-like graph structures that will be generated by the procedure; states are drawn from some arbitrary set *State*.

Definition 13. A *structure*, H, is a triple $H = (S, R, L)$, where

- $S \subseteq State$ is a set of states;
- $R \subseteq S \times S$ is a binary relation on S; and
- $L : S \to powerset(Form(\mathcal{L}_{TB}))$ labels each state with a set of $\mathcal{L}_{TB}$-formulae.

Let *Structures* be the set of all structures.

Definition 14. If $\varphi \in Form(\mathcal{L}_{TB})$ is of the form $\chi \mathcal{U} \psi$ or $\Diamond \psi$ then φ is said to have *eventuality* ψ. If (S, R, L) is a structure, $s \in S$ is a state, R^* is the reflexive transitive closure of R, and $\varphi \in Form(\mathcal{L}_{TB})$, then φ is said to be *resolvable* in (S, R, L) from

s, (notation *resolvable*$(\varphi, s, (S, R, L))$), iff if φ has eventuality ψ, then $\exists s' \in S$ s.t. $(s, s') \in R^*$ and $\psi \in L(s')$.

Definition 15. If $\Gamma \subseteq Form(\mathcal{L}_{TB})$, then $next(\Gamma)$ is defined:

$$next(\Gamma) \stackrel{\text{def}}{=} \{\varphi \mid \bigcirc\varphi \in \Gamma\}.$$

The decision procedure is then given by the function *structure*, which is presented in Fig. 4. The following two theorems describe the key properties of this function.

Theorem 16. *If* $\varphi \in Form(\mathcal{L}_{TB})$*, then* φ *is satisfiable iff structure*(φ) *returns* (S, R, L)*, and* $\exists s \in S$ *s.t.* $\alpha^*(\varphi) \subseteq L(s)$*.*

Theorem 17. *If* $\varphi \in Form(\mathcal{L}_{TB})$*, then structure*$(\varphi)$ *terminates.*

4.1 Examples

Example 1: The first example is a purely temporal formula taken from [10].

$$(\Diamond p \wedge \Box(p \Rightarrow \bigcirc p)) \Rightarrow \Diamond\Box p \tag{11}$$

After negating and rewriting into normal form, (11) becomes:

$$\Diamond p \wedge \Diamond\neg p \wedge (\neg p \vee \bigcirc p) \wedge \bigcirc\Box(\neg p \vee \bigcirc p) \wedge \bigcirc\Box\Diamond\neg p. \tag{12}$$

The graph generation stage of *structure* terminates after generating a graph containing seven states, s_1–s_7, labelled as follows.

$$\begin{aligned}
s_1 &= \{p, \bigcirc\Diamond\neg p, \bigcirc p, \bigcirc\Box(\neg p \vee \bigcirc p), \bigcirc\Box\Diamond\neg p, \neg p \vee \bigcirc p, \Diamond\neg p, \Diamond p\}\\
s_2 &= \{\bigcirc\Diamond p, \neg p, \bigcirc\Box(\neg p \vee \bigcirc p), \bigcirc\Box\Diamond\neg p, \neg p \vee \bigcirc p, \Diamond\neg p, \Diamond p\}\\
s_3 &= \{\bigcirc\Diamond p, \neg p, \bigcirc p, \bigcirc\Box(\neg p \vee \bigcirc p), \bigcirc\Box\Diamond\neg p, \neg p \vee \bigcirc p, \Diamond\neg p, \Diamond p\}\\
s_4 &= \{\bigcirc\Diamond p, \bigcirc\Diamond\neg p, \neg p, \bigcirc\Box(\neg p \vee \bigcirc p), \bigcirc\Box\Diamond\neg p, \neg p \vee \bigcirc p, \Diamond\neg p, \Diamond p\}\\
s_5 &= \{\bigcirc\Diamond p, \bigcirc\Diamond\neg p, \bigcirc p, \bigcirc\Box(\neg p \vee \bigcirc p), \bigcirc\Box\Diamond\neg p, \neg p \vee \bigcirc p, \Diamond\neg p, \Diamond p\}\\
s_6 &= \{\bigcirc\Diamond\neg p, p, \bigcirc p, \bigcirc\Box(\neg p \vee \bigcirc p), \bigcirc\Box\Diamond\neg p, \neg p \vee \bigcirc p, \Diamond\neg p\}\\
s_7 &= \{\bigcirc\Diamond p, p, \bigcirc p, \bigcirc\Box(\neg p \vee \bigcirc p), \bigcirc\Diamond\neg p, \bigcirc\Box\Diamond\neg p, \Diamond\neg p, \neg p \vee \bigcirc p, \Diamond p\}
\end{aligned}$$

The final state of R is summarised in the following adjacency matrix.

From …

To …	s_1	s_2	s_3	s_4	s_5	s_6	s_7
s_1		×	×	×	×		×
s_2		×		×			
s_3		×		×			
s_4		×		×			
s_5		×		×			
s_6	×					×	
s_7			×		×		×

Graph contraction then begins; as all states contain unresolved eventualities, they are all deleted. The graph that is returned is therefore empty, so (12) is unsatisfiable, meaning that (11) is valid.

```
function structure(φ : Form(ℒ_TB)) : (S, R, L) : Structures
vars        (N, T, l1, l2) : Tableaux
            n : Node
            s, s′ : State
            flag : 𝔹
begin                 (* stage 1: initialise *)
      S := R := L := ∅
      (N, T, l1, l2) := tableau({φ})
      for each n ∈ leaves((N, T, l1, l2)) s.t. l2(n) = o do
            create new state s
            S := S ∪ {s}
            L := L † {s ↦ walk-set((N, T, l1, l2), n)}
      end-for
      repeat          (* stage 2: create graph *)
            flag := false
            for each s ∈ S do
                  (N, T, l1, l2) := tableau(next(L(s)))
                  for each n ∈ leaves((N, T, l1, l2)) s.t. l2(n) = o do
                        if ∃s′ ∈ S s.t. walk-set((N, T, l1, l2), n) = L(s′) then
                              R := R ∪ {(s, s′)}
                        else
                              create new state s′
                              S := S ∪ {s′}
                              L := L † {s′ ↦ walk-set((N, T, l1, l2), n)}
                              R := R ∪ {(s, s′)}
                              flag := true
                  end-for
            end-for
      until ¬flag
      repeat          (* stage 3: contract graph *)
            flag := false
            for each s ∈ S do
                  if ∃ψ ∈ L(s) s.t. ¬resolvable(ψ, s, (S, R, L)) or
                     ∃ψ ∈ L(s) s.t. ψ is of the form ◯χ
                        and ¬∃s′ ∈ S s.t. (s, s′) ∈ R and χ ∈ L(s′)
                  then
                        S := S − {s}
                        flag := true
            end-for
      until ¬flag
end-function
```

Fig. 4. Function *structure*

Example 2: The formula used in the previous example contained only classical and temporal connectives; it contained no belief modalities. In this section, we present an example in which belief and time interact. Imagine an agent i that is a perfect propositional reasoner; that is, whenever i believes Δ, then i will also believe φ if $\Delta \vdash_{\mathcal{L}_0} \varphi$ (i.e., if there is a proof of φ from Δ in $\mathcal{L}_0$). Thus $(\Delta, \varphi) \in BE_i$ iff $\Delta \vdash_{\mathcal{L}_0} \varphi$. Now consider the following formula.

$$\Box[i](p \wedge (p \Rightarrow q)) \;\wedge\; \Diamond\neg[i]q \tag{13}$$

Intuitively, it is easy to see that (13) is unsatisfiable: it is obvious that $(p \wedge (p \Rightarrow q)) \vdash_{\mathcal{L}_0} q$, and so if i always believes $(p \wedge (p \Rightarrow q))$, then i also always believes q. Hence i can never not believe q. We use the decision procedure to show this formally.

The alpha closure of (13) is:

$$\{[i](p \wedge (p \Rightarrow q)), \bigcirc\Box[i](p \wedge (p \Rightarrow q)), \Diamond\neg[i]q\}.$$

The tableau for this set has two leaves, n_1 and n_2, labelled thus:

$$\begin{aligned} n_1 &= \{[i](p \wedge (p \Rightarrow q)), \bigcirc\Box[i](p \wedge (p \Rightarrow q)), \neg[i]q\} \\ n_2 &= \{[i](p \wedge (p \Rightarrow q)), \bigcirc\Box[i](p \wedge (p \Rightarrow q)), \bigcirc\Diamond\neg[i]q\}. \end{aligned}$$

Node n_1 is closed: it contains both $[i](p \wedge (p \Rightarrow q))$ and $\neg[i]q$, and $(\{p \wedge (p \Rightarrow q)\}, q) \in BE_i$, since $p \wedge (p \Rightarrow q) \vdash_{\mathcal{L}_0} q$. (See Definition 9(3)). Node n_2 is open, so a state s_1 is created in the graph, labelled:

$$\{[i](p \wedge (p \Rightarrow q)), \bigcirc\Box[i](p \wedge (p \Rightarrow q)), \bigcirc\Diamond\neg[i]q, \Diamond\neg[i]q\}.$$

However, the next time formulae of this set correspond to the alpha closure of the input formula, so we need not build another tableau; we simply make a link from s_1 to itself. Graph generation then ends, and graph contraction begins: state s_1 is deleted, as it contains an unresolved eventuality ($\Diamond\neg[i]q$). The structure returned is thus empty, and (13) is therefore unsatisfiable.

Example 3: The *wisest man puzzle* is a classic problem in reasoning about knowledge and belief that is widely used as a benchmark against which formalisms for representing these notions are evaluated. We have used the decision procedure for $\mathcal{L}_{TB}$ to solve a variant of the problem, which involves an element of time. The variant we used, in its most general form, may be stated as follows (see [14, p58] for the original):

> A king wishes to know which of his n advisors is the wisest. He arranges them in a circle, so that they can both see and hear each other, and tells them that he will paint either a white or black dot on each of their foreheads, but that at least one dot will be white. He offers his favour to the one that can correctly identify the colour of his spot. At time 1 he asks advisor 1 if he knows the colour of his spot; the advisor does not know. At time 2 he asks advisor 2 if he knows the colour of his spot; he does not know. The king continues in this way, until at time n he asks advisor n, who correctly identifies that his spot is white.

A solution to the problem (without a temporal component) may be found in [14, pp57–61]; another interesting solution, involving common knowledge and time, and given as a proof in a Hilbert-style axiom system appears in [15, pp168–174][2]. We now give an axiomatisation of the puzzle. We write $w(i)$ for 'agent i's dot is white'. First, we state that every dot is white.

$$\Box \bigwedge_{i=1}^{n} w(i) \tag{14}$$

Next, we need to state that it is *mutually believed* that at least one dot is white. Since $\mathcal{L}_{TB}$ does not contain a mutual belief operator, we define one: we write $[M]\varphi$ if $[i]\varphi$, and $[i][j]\varphi$ and $[i][j][k]\varphi$, and so on.

$$[M]\varphi \stackrel{\text{def}}{=} ([i_1]\varphi) \wedge ([i_1][i_2]\varphi) \wedge \cdots \wedge ([i_1][i_2]\cdots[i_n]\varphi) \qquad \text{for all } i_1, \ldots, i_n \in \{1, \ldots, n\}$$

The mutual belief that at least one dot is white is represented by the following axiom.

$$\Box [M] \bigvee_{j=1}^{n} w(j) \tag{15}$$

The following *observation axioms* state that each advisor can see everyone else's dot.

$$\left.\begin{array}{l} \Box w(i) \Rightarrow [j]w(i) \\ \Box \neg w(i) \Rightarrow [j]\neg w(i) \end{array}\right\} \qquad \text{for all } i, j \in \{1, \ldots, n\} \text{ s.t. } i \neq j \tag{16}$$

The axioms in (16) are mutually believed.

$$\left.\begin{array}{l} \Box [M]w(i) \Rightarrow [j]w(i) \\ \Box [M]\neg w(i) \Rightarrow [j]\neg w(i) \end{array}\right\} \qquad \text{for all } i, j \in \{1, \ldots, n\} \text{ s.t. } i \neq j \tag{17}$$

Advisors are at least partially consistent: if they believe that some advisor's dot is white, then they do not believe it is not white.

$$\Box [i]w(j) \Rightarrow \neg [i]\neg w(j) \qquad \text{for all } i, j \in \{1, \ldots, n\} \tag{18}$$

Axiom (18) is mutually believed.

$$\Box [M](([i]w(j)) \Rightarrow \neg [i]\neg w(j)) \qquad \text{for all } i, j \in \{1, \ldots, n\} \tag{19}$$

At time $u \in \{1, \ldots, n-1\}$, advisor u reveals that he does not know the colour of his dot, making this mutually believed:

$$\bigcirc^{u} \Box [M]\neg [u]w(u) \qquad \text{for all } u \in \{1, \ldots, n-1\} \tag{20}$$

where $\bigcirc^{u}$, for $u \in \mathbb{N}$, means $\bigcirc$ iterated u times. Finally, the aim of the puzzle is to show that at time n, advisor n knows that his spot is white.

$$\bigcirc^{n} [n]w(n) \tag{21}$$

[2] However, the logic used in [15] is not given a semantics.

We have used the $\mathcal{L}_{TB}$ decision procedure to solve the 2-advisor version of this problem; full details are included in the associated technical report [23]. Extensions to the general *n*-advisor case are not problematic.

An interesting aspect of the decision procedure, when applied to this and many other problems, is that showing that a set of $\mathcal{L}_{TB}$-formulae containing belief modalities is improper, (and thus unsatisfiable), involves a recursive call on the decision procedure, to show that a simpler set of formulae (essentially the original belief formulae with the outermost belief modality stripped off) is improper (cf. [9]). While this has obvious implications with respect to efficiency, it has the advantage of being conceptually a very simple way of dealing with belief modalities.

5 Concluding Remarks

In this paper, we have developed a temporal belief logic called $\mathcal{L}_{TB}$. By using this logic, it is possible to represent the time-varying properties of systems containing multiple resource-bounded reasoning agents. We have discussed both the applications of the logic, and some properties that might be represented in it, and have presented a tableau-based decision procedure for it.

Future work will focus on the following areas: extensions of the decision procedure to restricted first-order logics; implementing and improving the efficiency of the decision procedure; decision procedures for normal modal temporal belief logics, (and ultimately many-dimensional modal logics in general); and resolution-style calculi for temporal belief logics, (perhaps based on [5]).

References

1. M. Ben-Ari. *Mathematical Logic for Computer Science*. Prentice Hall, 1993.
2. A. H. Bond and L. Gasser, editors. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., 1988.
3. L. Catach. Normal multimodal logics. In *Proceedings of the National Conference on Artificial Intelligence (AAAI '88)*, St. Paul, MN, 1988.
4. E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier, 1990.
5. M. Fisher. A resolution method for temporal logic. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI '91)*, Sydney, Australia, August 1991.
6. M. Fisher and M. Wooldridge. Executable temporal logic for distributed A.I. In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, Hidden Valley, PA, May 1993.
7. M. Fisher and M. Wooldridge. Specifying and verifying distributed intelligent systems. In M. Filgueiras and L. Damas, editors, *Progress in Artificial Intelligence — Sixth Portuguese Conference on Artificial Intelligence (LNAI Volume 727)*, pages 13–28. Springer-Verlag, October 1993.
8. D. Gabbay. Declarative past and imperative future. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Proceedings of the Colloquium on Temporal Logic in Specification (LNCS Volume 398)*, pages 402–450. Springer-Verlag, 1989.

9. C. Geissler and K. Konolige. A resolution method for quantified modal logics of knowledge and belief. In J. Y. Halpern, editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 309–324. Morgan Kaufmann Publishers, Inc., 1986.
10. G. D. Gough. Decision procedures for temporal logic. Master's thesis, Department of Computer Science, Manchester University, Oxford Rd., Manchester M13 9PL, UK, October 1984.
11. 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.
12. 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.
13. C. B. Jones. *Systematic Software Development using VDM (second edition)*. Prentice Hall, 1990.
14. K. Konolige. *A Deduction Model of Belief*. Pitman/Morgan Kaufmann, 1986.
15. S. Kraus and D. Lehmann. Knowledge, belief and time. *Theoretical Computer Science*, 58:155–174, 1988.
16. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag, 1992.
17. H. Reichgelt. Logics for reasoning about knowledge and belief. *Knowledge Engineering Review*, 4(2), 1989.
18. Y. Shoham. *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence*. The MIT Press, 1988.
19. R. M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.
20. R. H. Thomason. Combinations of tense and modality. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic Volume II — Extensions of Classical Logic*, pages 135–166. D. Reidel Publishing Company, 1984. (Synthese library Volume 164).
21. P. Wolper. The tableau method for temporal logic: An overview. *Logique et Analyse*, 110–111, 1985.
22. M. Wooldridge. *The Logical Modelling of Computational Multi-Agent Systems*. PhD thesis, Department of Computation, UMIST, Manchester, UK, October 1992.
23. M. Wooldridge. A temporal belief logic. Technical report, Department of Computing, Manchester Metropolitan University, Chester St., Manchester M1 5GD, UK, 1994.

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