ELSEVIER

Artificial Intelligence 139 (2002) 47–89

Artificial Intelligence

www.elsevier.com/locate/artint

Clausal resolution in a logic of rational agency

Clare Dixon [a,*], Michael Fisher [a], Alexander Bolotov [b]

[a] *Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK*
[b] *Harrow School of Computer Science, University of Westminster, Harrow HA1 3TP, UK*

Received 16 February 2001

Abstract

A resolution based proof system for a Temporal Logic of Possible Belief is presented. This logic is the combination of the branching-time temporal logic CTL (representing change over time) with the modal logic KD45 (representing belief). Such combinations of temporal or dynamic logics and modal logics are useful for specifying complex properties of multi-agent systems. Proof methods are important for developing verification techniques for these complex multi-modal logics. Soundness, completeness and termination of the proof method are shown and simple examples illustrating its use are given. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Verification; Rational agents; Automated deduction; Resolution; Temporal logics; Modal logics

1. Introduction

The use of *agents* is now seen as an essential tool in representing, understanding and implementing complex software systems. In particular, the characterisation of complex components as *intelligent* or *rational* agents allows the system designer to analyse applications at a much higher level of abstraction. In order to reason about such agents, a number of theories of rational agency have been developed, such as the BDI [42] and KARO [44] frameworks. These frameworks are usually represented as complex multi-modal logics. In addition to their use in agent theories, where the basic representation of agency and rationality is explored, these logics form the basis for agent-based formal methods [24]. In both these uses, the notion of proof is important. In agent theories, proof allows us to examine properties of the overall theory and, in some cases, to characterise

* Corresponding author.
E-mail address: C.Dixon@csc.liv.ac.uk (C. Dixon).

0004-3702/02/$ – see front matter
PII: S0004-3702(02)00196-0

computation within that theory. In agent-based formal methods, proof is clearly important in developing verification techniques.

The leading agent theories and formal methods in this area all share similar logical properties. In particular, the logics used have:

- an *informational* component, such as being able to represent an agent's beliefs or knowledge,
- a *dynamic* component, allowing the representation of dynamic activity, and
- a *motivational* component, often representing the agent's desires, intentions or goals.

These aspects are typically represented as follows:

- *Information*—modal logic of belief (KD45) or knowledge (S5);
- *Dynamism*—temporal or dynamic logic;
- *Motivation*—modal logic of intention (KD) or desire (KD).

Thus, the predominant approaches use relevant combinations, e.g.,

- *Moore* [36] dynamic logic + knowledge (S5);
- *BDI* [42] branching temporal logic (CTL or CTL*) or linear time temporal logic (PTL) + belief (KD45) + desire (KD) + intention (KD);
- *KARO* [44] dynamic logic (PDL) + belief (KD45) (or knowledge (S5)) + wishes (KD).

Unfortunately, many of these combinations, particularly those using dynamic logic, become too complex (not only undecidable, but incomplete) to use in practical situations. Thus, much current research activity is centred around developing simpler combinations of non-classical logics that can express many of the same properties as the more complex combinations, yet are simpler to mechanise. For example, some of our work in this area has involved developing a simpler logical basis for BDI-like agents [22], while others have developed temporal logics of knowledge [21].

The aim of this paper is to examine proof methods for one particular logic that is being developed in this way. This is based on the KARO framework [44] but, rather than using a very complex combination of logics we, in collaboration with the KARO developers, have identified a simpler logic just combining a branching-time temporal aspect with a modal information aspect and have shown how this can be used to successfully represent many of the core elements of KARO [32,33]. This logic is essentially the branching-time temporal logic, Computational Tree Logic (CTL) combined with the modal logic KD45. CTL was first described in [11] and can be distinguished from the variety of branching time temporal logics proposed in the literature, as every temporal operator, for example '$\bigcirc$' (in the next moment) must be preceded immediately by a path operator, for example **A** (on all paths). Thus an example of a CTL formula is $\mathbf{A}\bigcirc\varphi$, meaning 'in all possible futures, in the next moment φ holds'.

In previous work, we have developed a proof method based upon clausal resolution for CTL [9]. We here consider the extension of this approach to the combination of CTL with

the KD45 modal logic. The method involves the translation to a normal form which reduces the number of operators to a core set and separates temporal and belief components. Resolution rules are defined for each component and information between components is passed via propositional formulae. The completeness proof (which differs from that given in [9]) for the resulting method involves the construction of a graph representing the normal form formulae. This graph is shown to be empty if and only if the set of normal form formulae is unsatisfiable. Deletions of parts of the graph are shown to correspond with resolution rules.

Thus, the contribution of the paper is to provide a resolution based proof method for the fusion of CTL and KD45 which is proved to be sound, complete and terminating. As mentioned above this particular combination of logics was chosen to verify properties from a core of the KARO language [32,33]. This is a first step towards our goal of verification of a larger subset of the full KARO language and we hope to extend the framework we have, to deal with, for example, interaction between the combined logics. Finally, rather than being viewed in isolation, this work is part of a longer term effort concerning how to combine a range of modal and temporal logics and provide proof methods for them.

In Section 2 we give details of this Temporal Logic of Possible Belief, BB_n, the fusion of CTL and multi-modal KD45. A normal form for this logic and details of how to translate BB_n formulae into the normal form is given in Sections 3 and 4. The resolution rules, i.e., initial, modal, step and temporal resolution rules for the single modal version of this logic, BB_1 are introduced in Section 5. We consider examples in Section 6 and prove soundness, completeness and termination for the resolution system for BB_1 in Section 7. In Section 8 we discuss the extension of the resolution rules to the multi-modal version of this logic and in Sections 9 and 10 we consider related work and conclusions.

2. A temporal logic of possible belief

In this section, we give the syntax and semantics of a logic BB_n, the fusion of CTL and multi-modal KD45, a *branching-time temporal logic of belief*. This is the smallest logic containing both CTL and multi-modal KD45 but allowing no axioms containing interactions between the two logics.

2.1. Syntax

Formulae are constructed from a set $\mathcal{P} = \{p, q, r, \ldots\}$ of *primitive propositions*. The language BB_n contains the standard propositional connectives $\neg$ (not), $\vee$ (or), $\wedge$ (and) and $\Rightarrow$ (implies). For belief we assume a set of agents $Ag = \{1, \ldots, n\}$ and we introduce a set of unary modal connectives B_i, for $i \in Ag$, where a formula $B_i\phi$ is read as "agent i believes ϕ". For the temporal dimension we take the path operators **A** and **E** in conjunction with 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, branching model of time with finite past, and infinite future. The formulae of BB_n are constructed using the following connectives and proposition symbols:

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

The set of well-formed formulae of BB_n, WFF_B, is defined by the following rules:

- any element of $\mathcal{P}$ is in WFF_B;
- **false** and **true** are in WFF_B;
- if F and G are in WFF_B then so are

$$\begin{array}{lllll} \neg F & F \vee G & F \wedge G & F \Rightarrow G & B_i F \\ \mathbf{A}\Diamond F & \mathbf{A}\Box F & \mathbf{A}(F\,\mathcal{U}\,G) & \mathbf{A}(F\,\mathcal{W}\,G) & \mathbf{A}\bigcirc F \\ \mathbf{E}\Diamond F & \mathbf{E}\Box F & \mathbf{E}(F\,\mathcal{U}\,G) & \mathbf{E}(F\,\mathcal{W}\,G) & \mathbf{E}\bigcirc F \end{array}$$

  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 $B_i l$ or $\neg B_i l$ where l is a literal and $i \in Ag$.

Definition 3. The *belief set* for agent i for a set of literals or modal literals X is defined as $B_i\_\text{set}(X) = \{l \mid B_i l \in X\}$. In the single agent case this is abbreviated to B_set and is defined as $B\_\text{set}(X) = \{l \mid Bl \in X\}$.

2.2. Semantics

We follow closely the presentation of semantics given in [28]. First, we assume that the world may be in any of a set, S, of *states*.

Definition 4. A *tree* is a structure (S, η), where S is the set of states and $\eta \subseteq S \times S$ is a relation between states such that

- $s_0 \in S$ is a unique root node (i.e., $\neg\exists s_i \in S$ such that $(s_i, s_0) \in \eta$);
- for each $s_i \in S$ there exists $s_j \in S$ such that $(s_i, s_j) \in \eta$;
- for all $s_i, s_j, s_k \in S$ if $(s_j, s_i) \in \eta$ and $(s_k, s_i) \in \eta$ then $s_j = s_k$;
- for all $s_i \in S$, $(s_0, s_i) \in \eta^*$ where η^* is the reflexive, transitive, closure of η.

Let T be the set of all trees.

Definition 5. 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 [28,29]. Given a set of trees T, the set of timelines can be extracted by taking the union of the infinite branches of each tree in T. Let *TLines* be the set of all timelines in T.

Definition 6. A point, p, is a pair $p = (t, u)$, where $t \in \mathit{TLines}$ is a timeline and $u \in \mathbf{N}$ is a temporal index into t.

Let *Points* be the set of all points.

Definition 7. Given T, a set of trees, let *TLines* be the set of timelines constructed from T. We say that two timelines t and t' coincide up to point (t, n) if, and only if, $(t, n') = (t', n')$ for all $n' \leqslant n$. A timeline t' extends (t, n) if, and only if, t and t' coincide up to (t, n).

Definition 8. Given a timeline, t, a finite sub-sequence of points $(t, i), (t, i+1), \ldots, (t, n)$ is abbreviated by $[(t, i), (t, n)]$ and an infinite sub-sequence of states from (t, i) onwards $(t, i), (t, i+1), (t, i+2), \ldots$, known as a *suffix* of (t, i), is abbreviated by $[(t, i), \ldots]$.

Definition 9. A set of all timelines, *TLines*, is *limit closed* if it satisfies the following condition. For any timeline $t \in \mathit{TLines}$ and any timelines $t_1, t_2, t_3, \ldots \in \mathit{TLines}$ such that t and t_1 coincide up to a point (t, i), t_1 and t_2 coincide up to a point (t_1, j), t_2 and t_3 coincide up to a point (t_2, k), etc., the following holds: there exists a timeline t' as a limit of the prefixes $[(t, 0), (t, i)]$, $[(t_1, i), (t_1, j)]$, $[(t_2, j), (t_2, k)]$ ....

In the following we assume that the set of all timelines, *TLines*, is *limit closed*. It is easy to see that the set of timelines defined as above also satisfies *suffix* and *fusion* closure properties [18]. Therefore, according to [18], the underlying tree models can be characterized as Kripke structures. Although capturing the limit closure property complicates translation into normal form (see Section 3), resulting, in particular, in the introduction of labels, on the other hand, this property significantly simplifies the resolution procedure (see Section 5).

Definition 10. A *valuation*, π, is a function $\pi : \mathit{Points} \times \mathcal{P} \to \{T, F\}$.

Definition 11. A *model*, M, for BB_n is a structure $M = \langle T, R_1, \ldots, R_n, \pi \rangle$, where:

- T is a set of trees, with a distinguished tree r_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 transitive, serial ($\forall q \in \mathit{Points}, \exists q' \in \mathit{Points}$ s.t. $(q, q') \in R_i$) and Euclidean ($\forall q, q', q'' \in \mathit{Points}$ if $(q, q') \in R_i$ and $(q, q'') \in R_i$ then $(q', q'') \in R_i$);
- π is a valuation function, as above.

$\langle M,(t,u)\rangle \models \mathbf{true}$		
$\langle M,(t,u)\rangle \models p$	iff	$\pi((t,u),p) = T$ (where $p \in \mathcal{P}$)
$\langle M,(t,u)\rangle \models \neg F$	iff	$\langle M,(t,u)\rangle \not\models F$
$\langle M,(t,u)\rangle \models F \vee G$	iff	$\langle M,(t,u)\rangle \models F$ or $\langle M,(t,u)\rangle \models G$
$\langle M,(t,u)\rangle \models \mathbf{A}F$	iff	$\langle M,(t',u)\rangle \models F$ for all timelines t' extending (t,u)
$\langle M,(t,u)\rangle \models \mathbf{E}F$	iff	$\langle M,(t',u)\rangle \models F$ for some timeline t' extending (t,u)
$\langle M,(t,u)\rangle \models \bigcirc F$	iff	$\langle M,(t,u+1)\rangle \models F$
$\langle M,(t,u)\rangle \models \Box F$	iff	$\forall u' \in \mathbf{N}$, if $(u \leqslant u')$ then $\langle M,(t,u')\rangle \models F$
$\langle M,(t,u)\rangle \models \Diamond F$	iff	$\exists u' \in \mathbf{N}$, such that $(u \leqslant u')$ and $\langle M,(t,u')\rangle \models F$
$\langle M,(t,u)\rangle \models F\,\mathcal{U}\,G$	iff	$\exists u' \in \mathbf{N}$ such that $(u' \geqslant u)$ and $\langle M,(t,u')\rangle \models G$, and $\forall u'' \in \mathbf{N}$, if $(u \leqslant u'' < u')$ then $\langle M,(t,u'')\rangle \models F$
$\langle M,(t,u)\rangle \models F\,\mathcal{W}\,G$	iff	$\langle M,(t,u)\rangle \models F\,\mathcal{U}\,G$ or $\langle M,(t,u)\rangle \models \Box F$
$\langle M,(t,u)\rangle \models B_i F$	iff	$\forall t' \in \mathit{TLines}.\ \forall u' \in \mathbf{N}$, if $((t,u),(t',u')) \in R_i$ then $\langle M,(t',u')\rangle \models F$

Fig. 1. Semantics of BB_n.

As usual, we define the semantics of the language via the satisfaction relation '$\models$'. For BB_n, this relation holds between pairs of the form $\langle M, p\rangle$ (where M is a model and $p \in \mathit{Points}$), and BB_n-formulae. The rules defining the satisfaction relation are given in Fig. 1. For any formula F, if there is some model M such that $\langle M,(t_0,0)\rangle \models F$, for any timeline t_0 extracted from the distinguished tree r_0, then F is said to be satisfiable. If $\langle M,(t_0,0)\rangle \models F$ for all models M, for any timeline t_0 extracted from the distinguished tree r_0, then F is said to be valid.

As agent accessibility relations in BB_n models are transitive, serial and Euclidean, the axioms of the normal modal system KD45 are valid in BB_n models. They are

$$\begin{aligned}
&\mathrm{K}: \quad \vdash B_i(F \Rightarrow G) \Rightarrow (B_i F \Rightarrow B_i G)\\
&\mathrm{D}: \quad \vdash B_i F \Rightarrow \neg B_i \neg F\\
&4: \quad \vdash B_i F \Rightarrow B_i B_i F\\
&5: \quad \vdash \neg B_i \neg F \Rightarrow B_i \neg B_i \neg F.
\end{aligned}$$

Thus the following are theorems of KD45

$$\vdash \neg B_i B_i F \Leftrightarrow \neg B_i F \tag{1}$$

$$\vdash \neg B_i \neg B_i \neg F \Leftrightarrow B_i \neg F. \tag{2}$$

To prove (1), first to show $\Rightarrow$ take the contrapositive of 4 to obtain $\neg B_i B_i F \Rightarrow \neg B_i F$. To show the other direction we have $\neg B_i F \Rightarrow B_i \neg B_i F$ from 5 and $B_i \neg B_i F \Rightarrow \neg B_i B_i F$ from D so $\neg B_i F \Rightarrow \neg B_i B_i F$. Similarly to prove (2) take the contrapositive of 5, i.e., $\neg B_i \neg B_i \neg F \Rightarrow B_i \neg F$ to show $\Rightarrow$ for the other direction we have $B_i \neg F \Rightarrow B_i B_i \neg F$ from 4 and $B_i B_i \neg F \Rightarrow \neg B_i \neg B_i \neg F$ from D so $B_i \neg F \Rightarrow \neg B_i \neg B_i \neg F$. The result follows. Theorems (1) and (2) are important as they show that any literal prefixed by two or more B_i operators, for some $i \in Ag$, is equivalent to a literal prefixed by at most one B_i operator.

Thus our normal form described in the subsequent section only requires at most one B_i operator applied to a literal.

To assist the understanding of the translation to the normal form given in Section 3 we list some equivalent BB_n formulae where **P** is either **A** or **E** (but is the same operator on both sides of the equivalence).

$$\begin{aligned}
\neg\mathbf{A}\bigcirc F &\equiv \mathbf{E}\bigcirc\neg F & \mathbf{P}\Diamond F &\equiv F \vee \mathbf{P}\bigcirc\mathbf{P}\Diamond F\\
\neg\mathbf{E}\bigcirc F &\equiv \mathbf{A}\bigcirc\neg F & \mathbf{P}(F\,\mathcal{U}\,G) &\equiv G \vee \big(F \wedge \mathbf{P}\bigcirc\mathbf{P}(F\,\mathcal{U}\,G)\big)\\
\neg\mathbf{A}\Box F &\equiv \mathbf{E}\Diamond\neg F & \neg\mathbf{A}(F\,\mathcal{U}\,G) &\equiv \mathbf{E}\big(\neg G\,\mathcal{W}\,(\neg F \wedge \neg G)\big)\\
\neg\mathbf{E}\Box F &\equiv \mathbf{A}\Diamond\neg F & \neg\mathbf{E}(F\,\mathcal{U}\,G) &\equiv \mathbf{A}\big(\neg G\,\mathcal{W}\,(\neg F \wedge \neg G)\big)\\
\neg\mathbf{A}\Diamond F &\equiv \mathbf{E}\Box\neg F & \mathbf{P}(F\,\mathcal{W}\,G) &\equiv G \vee \big(F \wedge \mathbf{P}\bigcirc\mathbf{P}(F\,\mathcal{W}\,G)\big)\\
\neg\mathbf{E}\Diamond F &\equiv \mathbf{A}\Box\neg F & \neg\mathbf{A}(F\,\mathcal{W}\,G) &\equiv \mathbf{E}\big(\neg G\,\mathcal{U}\,(\neg F \wedge \neg G)\big)\\
\mathbf{P}\Box F &\equiv F \wedge \mathbf{P}\bigcirc\mathbf{P}\Box F & \neg\mathbf{E}(F\,\mathcal{W}\,G) &\equiv \mathbf{A}\big(\neg G\,\mathcal{U}\,(\neg F \wedge \neg G)\big).
\end{aligned}$$

These are standard, see [18] for example.

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 possible belief

The normal form we use is known as SNF_B and is based on SNF (see for example [25]). For the purposes of the normal form we introduce a symbol **start** such that $\langle M, (t_0, 0)\rangle \models \mathbf{start}$ for any timeline t_0 extracted from the distinguished tree r_0. Formulae in SNF_B are of the general form

$$\mathbf{A}\Box^* \bigwedge_i L_i$$

where each L_i is known as a *clause* and must be one of the following forms and $\mathbf{A}\Box^*$ is the universal relation. $\mathbf{A}\Box^*$ can be defined by using the operators E, informally *everyone believes*

$$E\varphi \Leftrightarrow \bigwedge_{i\in Ag} B_i\varphi$$

and C, common belief $C\varphi \Leftrightarrow E(\varphi \wedge C\varphi)$ as follows $\mathbf{A}\Box^*\varphi \Leftrightarrow \mathbf{A}\Box(\varphi \wedge C\mathbf{A}\Box^*\varphi)$. Clauses are of the following form.

$$\mathbf{start} \Rightarrow \bigvee_{b=1}^{r} l_b \qquad \text{(an } \textit{initial} \text{ clause)}$$

$$\bigwedge_{a=1}^{g} k_a \Rightarrow \mathbf{A}\bigcirc \bigvee_{b=1}^{r} l_b \qquad \text{(an } \mathbf{A} \textit{ step} \text{ clause)}$$

$$\bigwedge_{a=1}^{g} k_a \Rightarrow \mathbf{E}\bigcirc \bigvee_{b=1}^{r} l_{b\langle \mathrm{cj}\rangle} \qquad \text{(an } \mathbf{E} \textit{ step} \text{ clause)}$$

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

$$\bigwedge_{a=1}^{g} k_a \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle} \qquad \text{(an } \mathbf{E} \textit{ sometime} \text{ clause)}$$

$$\mathbf{true} \Rightarrow \bigvee_{b=1}^{r} m_{1_b} \qquad \text{(a 1-}\textit{belief}\text{ clause)}$$

$$\ldots \Rightarrow \ldots$$

$$\mathbf{true} \Rightarrow \bigvee_{b=1}^{r} m_{n_b} \qquad \text{(an } n\textit{-belief}\text{ clause)}$$

$$\mathbf{true} \Rightarrow \bigvee_{b=1}^{r} l_b \qquad \text{(a } \textit{literal}\text{ clause).}$$

Here k_a, l_b, and l are literals, m_{i_b} are either literals or modal literals involving the B_i operator and $\langle \mathsf{c_j} \rangle$ is a path label that is present on **E** step and **E** sometime clauses. This label indicates a particular path and arises, for example, from the translation of formulae such as $\mathbf{E}(F \,\mathcal{U}\, G)$. During the translation to the normal form such formulae are translated into several **E** step clauses and an **E** sometime clause (which ensures that G must actually hold). To indicate that all these clauses refer to the same path they are annotated with a label.

The outer '$\mathbf{A}\,\Box^*$' operator that surrounds the conjunction of clauses is usually omitted. Similarly, for convenience the conjunction is dropped and we consider just the set of clauses L_i. Note that literal clauses are just a special case of belief clauses where the right hand side consists of a disjunction of literals.

3.1. Interpretation of labelled formulae

Once the translation to SNF_B has been carried out, all **E** formulae must be labelled with some label. Before presenting the semantics of labelled formulae let us recall the following important property of the underlying CTL tree structures. Without loss of generality we can assume that the number of successors in the underlying tree structures is bounded and known in advance (as we know the number of **E** quantifiers). This enables us to introduce labels to associate the **E** formulae at any point in a structure with particular timelines.

Let $\mathsf{LAB} = \{\mathsf{c_i}, \mathsf{c_j}, \ldots\}$ be the set of labels for a set S of SNF_B clauses. Thus, $\mathbf{E}A_{\langle \mathsf{c_j} \rangle}$ means that A holds along some timeline labelled as $\langle \mathsf{c_j} \rangle$. Dependent on the context of $\langle \mathsf{c_j} \rangle$, we interpret SNF_B clauses as follows.[1] A step clause

$$\mathbf{A}\,\Box^*(x \Rightarrow \mathbf{E}\bigcirc p_{\langle \mathsf{c_j} \rangle}),$$

[1] Note in [9] two types of labels were used. Here we use the same notation for labels associated with both step and sometime clauses as their context makes the type clear. That is, labels associated with sometime clauses are always limit closure type labels and limit closure type labels only appear on sometime clauses.

where the '**E**' quantifier is associated with the label $\langle \mathsf{c_j} \rangle$ is understood as follows. For any timeline t, and any point (t, u), if x is satisfied at a point (t, u) then p must be satisfied at the point $(t', u+1)$, along some timeline, t', such that t and t' coincide up to a point (t, u) and the suffix, $[(t', u), \ldots]$, of t' is associated with $\langle \mathsf{c_j} \rangle$.

A sometime clause

$$\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond p_{\langle \mathsf{c_j} \rangle})$$

is interpreted as follows. For any timeline, t, and any point (t, u), if x is satisfied at a point (t, u) then p must be satisfied at the point, say, (t', v) $(u \leqslant v)$, along some timeline, t', such that t and t' coincide up to point (t, u) and the suffix, $[(t', u), \ldots]$, of t' is associated with the limit closure of $\langle \mathsf{c_j} \rangle$.

Note the principal difference between two types of labels mentioned above. Given $\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\bigcirc p_{\langle \mathsf{c_j} \rangle})$, we reason about $\mathbf{E}\bigcirc p$ only 'one step ahead', ignoring the future after p has been satisfied. On the contrary, given $\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond p_{\langle \mathsf{c_j} \rangle})$, we must reason about $\mathbf{E}\Diamond p$ in the context of a longer period of time (until p is satisfied along some timeline t'). Now, taking into account that $\mathbf{E}\Diamond p = p \vee \mathbf{E}\bigcirc\mathbf{E}\Diamond p$ and associating the $\mathbf{E}\bigcirc$ with $\langle \mathsf{c_j} \rangle$ (as in the previous above case), assume that x is satisfied at a point (t, u) along some timeline t. If the point (t, u) does not satisfy p then it must satisfy $\mathbf{E}\bigcirc\mathbf{E}\Diamond p$. This means that there must be a point, say $(t_1, u+1)$, such that it is a successor of (t, u) along some timeline associated with $\langle \mathsf{c_j} \rangle$ and it satisfies $\mathbf{E}\Diamond p$, etc. Thus, due to the limit closure property, there is a timeline t' which satisfies the requirements of Definition 9. The construction of the timeline t' is represented by labelling $\mathbf{E}\Diamond p$ with $\langle \mathsf{c_j} \rangle$ and from its context, i.e., appearing with the $\mathbf{E}\Diamond$ operators, we conclude it is a "limit closure" timeline. More details can be found in [6].

3.2. Merged-SNF$_B$

To apply the temporal resolution rule (see Section 5.5), one or more step or literals clauses may need to be combined. Consequently, a variant on SNF$_B$ called merged-SNF$_B$, is also required. Given a set of SNF$_B$ clauses, any **A** or **E** step SNF$_B$ clause is also a merged-SNF$_B$ clause. Any literal clause of the form $\mathbf{true} \Rightarrow F$ is written into a merged-SNF$_B$ clause as $\mathbf{true} \Rightarrow \mathbf{A}\bigcirc F$. Any two merged-SNF$_B$ clauses may be combined to produce a merged-SNF$_B$ clause as follows.

$$\frac{P \Rightarrow \mathbf{A}\bigcirc F \quad Q \Rightarrow \mathbf{A}\bigcirc G}{(P \wedge Q) \Rightarrow \mathbf{A}\bigcirc(F \wedge G)} \qquad \frac{P \Rightarrow \mathbf{A}\bigcirc F \quad Q \Rightarrow \mathbf{E}\bigcirc G_{\langle \mathsf{c_j} \rangle}}{(P \wedge Q) \Rightarrow \mathbf{E}\bigcirc(F \wedge G)_{\langle \mathsf{c_j} \rangle}} \qquad \frac{P \Rightarrow \mathbf{E}\bigcirc F_{\langle \mathsf{c_j} \rangle} \quad Q \Rightarrow \mathbf{E}\bigcirc G_{\langle \mathsf{c_j} \rangle}}{(P \wedge Q) \Rightarrow \mathbf{E}\bigcirc(F \wedge G)_{\langle \mathsf{c_j} \rangle}}$$

Thus, as long as labels match, any possible conjunctive combination of SNF$_B$ step clauses or literal clauses rewritten as step clauses can be represented in merged-SNF$_B$.

4. Translation to normal form

In this section, we review the translation of an arbitrary BB_n formula into the normal form.

Take any formula F of BB_n and translate into SNF_B by applying the τ_0 and τ_1 transformations described below (where f is a new proposition).

$$\tau_0[F] \longrightarrow \mathbf{A}\Box^*(\mathbf{start} \Rightarrow f) \wedge \tau_1\big[\mathbf{A}\Box^*(f \Rightarrow F)\big].$$

Next, we give the τ_1 transformation where x is a proposition. If the main operator on the right of the implication is a classical operator remove it as follows.

$$\tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow (F \wedge G)\big)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow F)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow G)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow (F \Rightarrow G)\big)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow (\neg F \vee G)\big)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow \neg(F \wedge G)\big)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow (\neg F \vee \neg G)\big)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow \neg(F \Rightarrow G)\big)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow F)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg G)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow \neg(F \vee G)\big)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg F)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg G)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg\neg F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow F)\big].$$

Complex subformulae that appear within the scope of any temporal or modal operators are renamed as follows (where v, y and z are all new propositions and **P** is either **A** or **E** but is the same on both sides of the transformation).

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow B_i F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow B_i y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow F)\big] \quad F \text{ not a literal.}$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg B_i F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg B_i \neg y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg F)\big] \quad F \text{ not a literal.}$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}\bigcirc F)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{P}\bigcirc y) \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow F)\big] \quad F \text{ neither a literal nor a disjunction of literals.}$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{A}\bigcirc F)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\bigcirc y) \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg F)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{E}\bigcirc F)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{A}\bigcirc y) \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg F)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}\Box F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}\Box y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow F)\big] \quad F \text{ not a literal.}$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{A}\Box F)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond y) \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg F)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{E}\Box F)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{A}\Diamond y) \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg F)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}\Diamond F)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{P}\Diamond y) \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow F)\big] \quad F \text{ not a literal.}$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{A}\Diamond F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Box y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg F)\big]$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{E}\Diamond F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{A}\Box y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg F)\big]$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(F\,\mathcal{U}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(y\,\mathcal{U}\,G))] \wedge \tau_1[\mathbf{A}\Box^*(y \Rightarrow F)] \quad F \text{ not a literal.}$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(F\,\mathcal{U}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(F\,\mathcal{U}\,y))] \wedge \tau_1[\mathbf{A}\Box^*(y \Rightarrow G)] \quad G \text{ not a literal.}$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{A}(F\,\mathcal{U}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}(y\,\mathcal{W}\,v))] \wedge \tau_1[\Box(y \Rightarrow \neg G)] \wedge \tau_1[\mathbf{A}\Box^*(v \Rightarrow (y \wedge z))] \wedge \tau_1[\Box(z \Rightarrow \neg F)]$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{E}(F\,\mathcal{U}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{A}(y\,\mathcal{W}\,v))] \wedge \tau_1[\Box(y \Rightarrow \neg G)] \wedge \tau_1[\mathbf{A}\Box^*(v \Rightarrow (y \wedge z))] \wedge \tau_1[\Box(z \Rightarrow \neg F)]$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(F\,\mathcal{W}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(y\,\mathcal{W}\,G))] \wedge \tau_1[\mathbf{A}\Box^*(y \Rightarrow F)] \quad F \text{ not a literal.}$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(F\,\mathcal{W}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(F\,\mathcal{W}\,y))] \wedge \tau_1[\mathbf{A}\Box^*(y \Rightarrow G)] \quad G \text{ not a literal.}$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{A}(F\,\mathcal{W}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}(y\,\mathcal{U}\,v))] \wedge \tau_1[\Box(y \Rightarrow \neg G)] \wedge \tau_1[\mathbf{A}\Box^*(v \Rightarrow (y \wedge z))] \wedge \tau_1[\Box(z \Rightarrow \neg F)]$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \neg\mathbf{E}(F\,\mathcal{W}\,G))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{A}(y\,\mathcal{U}\,v))] \wedge \tau_1[\Box(y \Rightarrow \neg G)] \wedge \tau_1[\mathbf{A}\Box^*(v \Rightarrow (y \wedge z))] \wedge \tau_1[\Box(z \Rightarrow \neg F)].$$

The translations for $\neg\mathbf{P}(F\,\mathcal{U}\,G)$ and $\neg\mathbf{P}(F\,\mathcal{W}\,G)$ are obtained by applying the negation equivalences for these formulae given in Section 2, renaming $\neg G$ by y and $\neg F$ by z and the conjunction $y \wedge z$ by v. Then, any temporal operators that do not occur in the normal form, applied to literals, are removed as follows (where y and z are new propositions, l, m are literals and $\mathbf{P}$ is either $\mathbf{A}$ or $\mathbf{E}$ but is the same on both sides of the transformation).

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}\Box l)] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow (l \wedge y))] \wedge \tau_1[\mathbf{A}\Box^*(y \Rightarrow \mathbf{P}\bigcirc(y \wedge l))]$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{A}(l\,\mathcal{U}\,m))] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{A}\Diamond m) \wedge \tau_1[\mathbf{A}\Box^*(x \Rightarrow z)] \wedge \tau_1[\mathbf{A}\Box^*(z \Rightarrow (m \vee (l \wedge y)))] \wedge \mathbf{A}\Box^*(y \Rightarrow \mathbf{A}\bigcirc z)$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}(l\,\mathcal{U}\,m))] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond m_{\langle \mathsf{cj} \rangle}) \wedge \tau_1[\mathbf{A}\Box^*(x \Rightarrow z)] \wedge \tau_1[\mathbf{A}\Box^*(z \Rightarrow (m \vee (l \wedge y)))] \wedge \mathbf{A}\Box^*(y \Rightarrow \mathbf{E}\bigcirc z_{\langle \mathsf{cj} \rangle}) \quad \langle \mathsf{cj} \rangle \text{ is new.}$$

$$\tau_1[\mathbf{A}\Box^*(x \Rightarrow \mathbf{P}(l\,\mathcal{W}\,m))] \longrightarrow \tau_1[\mathbf{A}\Box^*(x \Rightarrow z)] \wedge \tau_1[\mathbf{A}\Box^*(z \Rightarrow (m \vee (l \wedge y)))] \wedge \mathbf{A}\Box^*(y \Rightarrow \mathbf{P}\bigcirc z)$$

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.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow F)\big]$$
F neither a literal nor a modal literal,
nor a disjunction of literals and modal literals.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee B_i F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow B_i F)\big]$$
D contains a disjunction of the form $B_j G$
or $\neg B_j G$ where $i \neq j$.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee \neg B_i F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow \neg B_i F)\big]$$
D contains a disjunction of the form $B_j G$
or $\neg B_j G$ where $i \neq j$.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee B_i F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee B_i y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow F)\big]$$
F not a literal.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee \neg B_i \neg F)\big] \longrightarrow \tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D \vee \neg B_i \neg y)\big] \wedge \tau_1\big[\mathbf{A}\Box^*(y \Rightarrow F)\big] \quad F \text{ is not a literal.}$$

Recall from the definition of SNF_B each belief clause may only contain modal literals involving one modal operator, i.e., $\mathbf{true} \Rightarrow B_1 x \vee y \vee \neg B_1 z$ is allowed as the modal literals only contain one modal operator B_1 but $\mathbf{true} \Rightarrow B_1 x \vee y \vee \neg B_2 z$ is not allowed as it contains both modal operators B_1 and B_2. The second transformation in the above group renames a disjunct $B_i F$ that involves a different modal operator to another disjunct in D (the third transformation is similar). The fourth transformation in the group renames F from disjunct $B_i F$ where F is not a literal (the fifth is similar). Finally, we rewrite formulae containing no temporal operators whose right hand side is a disjunction of literals or modal literals into clause form and stop applying the transformation to clauses already in the correct form.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow D)\big] \longrightarrow \mathbf{A}\Box^*\big(\mathbf{true} \Rightarrow (\neg x \vee D)\big)$$
D literal or modal literal or disjunction of literals and
modal literals only involving one modal operator.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{A}\Diamond l)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{A}\Diamond l) \quad l \text{ is a literal.}$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond l)\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle}) \quad l \text{ is a literal and } \langle \mathsf{c_j} \rangle \text{ is new.}$$

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle})\big] \longrightarrow \mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle}) \quad l \text{ is a literal.}$$

$$\tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow \mathbf{A}\bigcirc(l_1 \vee \cdots \vee l_n)\big)\big] \longrightarrow \mathbf{A}\Box^*\big(x \Rightarrow \mathbf{A}\bigcirc(l_1 \vee \cdots \vee l_n)\big)$$
each l_i is a literal.

$$\tau_1\big[\mathbf{A}\Box^*\big(x \Rightarrow \mathbf{E}\bigcirc(l_1 \vee \cdots \vee l_n)\big)\big] \longrightarrow \mathbf{A}\Box^*\big(x \Rightarrow \mathbf{E}\bigcirc(l_1 \vee \cdots \vee l_n)_{\langle \mathsf{c_j}\rangle}\big)$$
each l_i is a literal and $\langle \mathsf{c_j}\rangle$ is new.

$$\tau_1\big[\mathbf{A}\Box^*(x \Rightarrow \mathbf{E}\bigcirc(l_1 \vee \cdots \vee l_n)_{\langle \mathsf{c_j}\rangle})\big] \longrightarrow \mathbf{A}\Box^*\big(x \Rightarrow \mathbf{E}\bigcirc(l_1 \vee \cdots \vee l_n)_{\langle \mathsf{c_j}\rangle}\big)$$
each l_i is a literal.

Thus, the above transformations are applied until the formula is in the form

$$\bigwedge_i \mathbf{A}\Box^* F_i$$

where each F_i is one of the required formats. This, in turn, is equivalent to

$$\mathbf{A}\Box^* \bigwedge_i F_i.$$

5. Resolution for temporal logic of possible belief

Here we consider the resolution rules for the temporal logic of belief $BB_{(1)}$. To simplify notation we shall write the single modal operator B_1 as B. The extension of this system into its multi-modal version is considered in Section 8.

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 $\Diamond\neg p$ on the same path. The step and temporal resolution rules for CTL were presented in [9]. Here, as we allow literal clauses there are some minor differences.

5.1. Initial resolution

A literal clause may be resolved with an initial clause or two initial clauses may be resolved together as follows

$$\text{[IRES1]}\quad \frac{\mathbf{true} \Rightarrow (F \vee l) \qquad \mathbf{start} \Rightarrow (G \vee \neg l)}{\mathbf{start} \Rightarrow (F \vee G)} \qquad \text{[IRES2]}\quad \frac{\mathbf{start} \Rightarrow (F \vee l) \qquad \mathbf{start} \Rightarrow (G \vee \neg l)}{\mathbf{start} \Rightarrow (F \vee G)}$$

where F is a disjunction of literals.

5.2. Modal resolution

During modal resolution we apply the following rules which are based on the modal resolution system introduced by Mints [35]. Firstly we are allowed to resolve a literal or modal literal and its negation or we can resolve the formulae Bl and $B\neg l$ as we cannot believe something and believe its negation.

$$\text{[MRES1]}\quad \frac{\textbf{true} \Rightarrow D \vee m \qquad \textbf{true} \Rightarrow D' \vee \neg m}{\textbf{true} \Rightarrow D \vee D'} \qquad \text{[MRES2]}\quad \frac{\textbf{true} \Rightarrow D \vee Bl \qquad \textbf{true} \Rightarrow D' \vee B\neg l}{\textbf{true} \Rightarrow D \vee D'}$$

Also we have the following rules which involve pushing the external B operator into one of the clauses to allow us to resolve, for example, $\neg Bl$ with l

$$\text{[MRES4a]}\quad \frac{\textbf{true} \Rightarrow D \vee \neg Bl \qquad \textbf{true} \Rightarrow D' \vee l}{\textbf{true} \Rightarrow D \vee \mathrm{mod}(D')} \qquad \text{[MRES4b]}\quad \frac{\textbf{true} \Rightarrow D \vee Bl \qquad \textbf{true} \Rightarrow D' \vee \neg l}{\textbf{true} \Rightarrow D \vee \mathrm{mod}(D')}$$

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

Definition 12. The function $\mathrm{mod}(D)$, defined on disjunctions of literals or modal literals D, is given as follows.

$$\mathrm{mod}(F \vee G) = \mathrm{mod}(F) \vee \mathrm{mod}(G)$$

$$\mathrm{mod}(Bl) = Bl$$

$$\mathrm{mod}(\neg Bl) = \neg Bl$$

$$\mathrm{mod}(l) = \neg B\neg l.$$

These last two resolution operations require explanation. We explain the motivation behind MRES4a below; the justification for MRES4b is similar. Recall that there is an implicit B operator surrounding each clause as each clause is surrounded by $\mathbf{A}\Box^*$. We are resolving the first clause in MRES4a, as it is, with the second clause having distributed the external B over the implication. Thus, when we resolve $\neg Bl$ with Bl, we must adjust the other disjuncts of the second clause to show that B has been distributed. In more detail we consider the right hand sides of the clauses given, i.e., $D \vee \neg Bl$ and $D' \vee l$. Rewriting as implications we have $\neg D \Rightarrow \neg Bl$ and $\neg D' \Rightarrow l$. As each of these belief or literal clauses is surrounded by an implicit B operator, the second clause can be rewritten as $B(\neg D' \Rightarrow l)$ and, hence, $B\neg D' \Rightarrow Bl$. Now D' is a disjunction of modal literals or literals, i.e., $D' = m_1 \vee m_2 \vee \cdots$ so $B\neg D' = B\neg m_1 \wedge B\neg m_2 \wedge \cdots$. Thus we can resolve the $\neg Bl$ and Bl on the right hand side of the implication obtaining $\neg D \wedge (B\neg m_1 \wedge B\neg m_2 \wedge \cdots) \Rightarrow \textbf{false}$. Rewriting as a disjunction we have $D \vee \neg B\neg m_1 \vee \neg B\neg m_2 \vee \cdots$. Since in KD45 we have theorems (1) and (2) given in Section 2.2, we can delete $\neg B\neg$ from any of the disjuncts m_i that are modal literals and obtain the required resolvent. The equivalences (1) and (2) ensure that following translation to normal form the application of any resolution rules to a pair of clauses results in a clause where each disjunct contains at most one explicit modal operator.

5.3. 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 all states. Simplification and subsumption rules are also applied.

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

$$\text{[SRES1]}\quad \frac{P \Rightarrow \mathbf{A}\bigcirc(F \vee l) \qquad Q \Rightarrow \mathbf{A}\bigcirc(G \vee \neg l)}{(P \wedge Q) \Rightarrow \mathbf{A}\bigcirc(F \vee G)}$$

$$\text{[SRES2]}\quad \frac{P \Rightarrow \mathbf{E}\bigcirc(F \vee l)_{\langle \mathsf{c_j}\rangle} \qquad Q \Rightarrow \mathbf{A}\bigcirc(G \vee \neg l)}{(P \wedge Q) \Rightarrow \mathbf{E}\bigcirc(F \vee G)_{\langle \mathsf{c_j}\rangle}}$$

$$\text{[SRES3]}\quad \frac{P \Rightarrow \mathbf{E}\bigcirc(F \vee l)_{\langle \mathsf{c_j}\rangle} \qquad Q \Rightarrow \mathbf{E}\bigcirc(G \vee \neg l)_{\langle \mathsf{c_j}\rangle}}{(P \wedge Q) \Rightarrow \mathbf{E}\bigcirc(F \vee G)_{\langle \mathsf{c_j}\rangle}}$$

A step clause may be resolved with a literal clause (where G is a disjunction of literals) and any label is carried to the resolvent.

$$\text{[SRES4]}\quad \frac{P \Rightarrow \mathbf{A}\bigcirc(F \vee l) \qquad \mathbf{true} \Rightarrow (G \vee \neg l)}{P \Rightarrow \mathbf{A}\bigcirc(F \vee G)} \qquad \frac{P \Rightarrow \mathbf{E}\bigcirc(F \vee l)_{\langle \mathsf{c_j}\rangle} \qquad \mathbf{true} \Rightarrow (G \vee \neg l)}{P \Rightarrow \mathbf{E}\bigcirc(F \vee G)_{\langle \mathsf{c_j}\rangle}}$$

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

$$\text{[SRES5]}\quad \frac{Q \Rightarrow \mathbf{P}\bigcirc\mathbf{false}}{\mathbf{true} \Rightarrow \neg Q}$$

where **P** is either path operator (the clause in the antecedent being labelled if **P** was the **E** operator). 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 $\mathbf{A}\,\Box^*\neg Q$.

5.4. 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 either **start** $\Rightarrow$ **false** or **true** $\Rightarrow$ **false**.

5.5. Temporal resolution

During temporal resolution the aim is to resolve one of the sometime clauses, $Q \Rightarrow \mathbf{P}\Diamond l$, with a set of clauses that together imply $\Box\neg l$ *along the same path*, for example a set of clauses that together have the effect of $F \Rightarrow \mathbf{A}\bigcirc\mathbf{A}\,\Box\neg l$ or possibly $F \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\,\Box\neg l$. However the interaction between the '$\bigcirc$' and '$\Box$' operators in BB_n makes the definition of such a rule non-trivial and further the translation from BB_n to SNF_B will have removed all but the outer level of $\Box$-operators. So, resolution will be between a sometime clause

and a *set* of clauses that together imply an $\Box$-formula that occurs on the same path, which will contradict the $\Diamond$-clause.

$$\text{[TRES1]}\quad \frac{P \Rightarrow \mathbf{A}\bigcirc\mathbf{A}\Box\neg l \qquad Q \Rightarrow \mathbf{A}\Diamond l}{Q \Rightarrow \mathbf{A}(\neg P\,\mathcal{W}\,l)} \qquad \text{[TRES2]}\quad \frac{P \Rightarrow \mathbf{A}\bigcirc\mathbf{A}\Box\neg l \qquad Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j}\rangle}}{Q \Rightarrow \mathbf{E}(\neg P\,\mathcal{W}\,l)_{\langle \mathsf{c_j}\rangle}}$$

$$\text{[TRES3]}\quad \frac{P \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\Box\neg l_{\langle \mathsf{c_j}\rangle} \qquad Q \Rightarrow \mathbf{A}\Diamond l}{Q \Rightarrow \mathbf{A}(\neg P\,\mathcal{W}\,l)} \qquad \text{[TRES4]}\quad \frac{P \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\Box\neg l_{\langle \mathsf{c_j}\rangle} \qquad Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j}\rangle}}{Q \Rightarrow \mathbf{E}(\neg P\,\mathcal{W}\,l)_{\langle \mathsf{c_j}\rangle}}$$

In each case the resolvent ensures that once Q has been satisfied, meaning that the eventuality $\Diamond l$ must be satisfied on some or all paths, the conditions for triggering a $\Box$-formula are not allowed to occur, i.e., either P must be false at every future moment or must be false until the eventuality (l) has been satisfied. It may be surprising that resolving an **A**-formula with an **E**-formula in TRES3 results in an **A**-formula. This is because the eventuality l must appear on *all* paths so similarly the resolvent will also hold on all paths. Here the label $\langle \mathsf{c_j}\rangle$ on all formulae are limit closure labels. As we see below formulae of the form $\mathbf{E}\bigcirc\mathbf{E}\Box\neg l_{\langle \mathsf{c_j}\rangle}$ are constructed from one or more merged step clauses, for example $a \Rightarrow \mathbf{E}\bigcirc(a \wedge \neg l)_{\langle \mathsf{c_j}\rangle}$. The timeline that traces out $\mathbf{E}\bigcirc(a \wedge \neg l)$ in every next moment satisfies the definition of a limit closure timeline. Similarly the resolvent is rewritten into several step clauses labelled where appropriate by $\langle \mathsf{c_j}\rangle$.

As there are no clauses of the form $A \Rightarrow \mathbf{P}\bigcirc\mathbf{P}\Box l$ the full temporal resolution operation applies between a **P** sometime clause and a set of merged clauses that together imply $A \Rightarrow \mathbf{P}\bigcirc\mathbf{P}\Box\neg l$. For example the temporal resolution operation, in detail for TRES3, is

$$\frac{\begin{array}{l} F_0 \Rightarrow \mathbf{P}_0\bigcirc G_0 \\ \ldots \Rightarrow \ldots \\ F_n \Rightarrow \mathbf{P}_n\bigcirc G_n \\ Q \Rightarrow \mathbf{A}\Diamond l \end{array}}{Q \Rightarrow \mathbf{A}\Big[\bigwedge_{i=0}^{n}(\neg F_i)\Big]\mathcal{W}\,l}$$

with the side conditions that
$\vdash G_i \Rightarrow \neg l$ for all $i,\ 0 \leqslant i \leqslant n$; and
$\vdash G_i \Rightarrow \bigvee_{j=0}^{n} F_j$ for all $i,\ 0 \leqslant i \leqslant n$,

where each $\mathbf{P}_i$ is either **A** or **E** (where **E** step clauses are labelled) and at least one $\mathbf{P}_i$ is **E** (otherwise it is an **A** loop and we apply TRES1). The set of merged clauses $F_i \Rightarrow \mathbf{P}\bigcirc G_i$ that satisfy these side conditions are together known as an **E** *loop in* l. The disjunction of the left hand side of this set of clauses, i.e., $\bigvee_i F_i$ is known as an **E** *loop formula* for l. The most complex part of this approach is the search for the set of clauses to use in the application of the temporal resolution operation. Detailed explanation of techniques developed for this search is beyond the scope of this paper but is discussed at length for PTL in [12,13] and CTL in [7].

The resolvent must be translated into the normal form before any further resolution steps can be applied. A translation to the normal form is given below that avoids the renaming of the subformula $\bigwedge_{i=0}^{n}\neg F_i$.

$$\textbf{true} \Rightarrow (\neg Q \vee l \vee \neg F_i) \tag{3}$$

$$\textbf{true} \Rightarrow (\neg Q \vee l \vee t) \tag{4}$$

$$t \Rightarrow \mathbf{P}\bigcirc(l \vee \neg F_i) \tag{5}$$

$$t \Rightarrow \mathbf{P}\bigcirc(l \vee t). \tag{6}$$

Thus, we only introduce one new proposition symbol t which we see later (in Section 5.7) can be introduced at the start of the proof. This is important for the termination as once the relevant new propositions t have been introduced we require no new propositions to be introduced during the proof. In the above there are $n+1$ copies of the clauses (3) and (5), one for each F_i, $i = 0, \ldots, n$. Also **P** will be **A** for resolvents of TRES1 and TRES3 and **E** for resolvents of TRES2 and TRES4 (and the relevant clauses will be labelled with the same label as the **E** sometime clause in the premise). We note that only the resolvents (3) and (5) depend on the particular loop being resolved with, i.e., contain a reference to F_i.

5.6. Subsumption and simplification

In addition to normal classical simplification of the conjunctions of literals on the left hand sides of clauses and disjunctions of literals or modal literals on the right hand sides of clauses we can perform additional simplification due to the axioms of KD45 such as the following.

$$Bl \wedge F \wedge B\neg l \wedge G \rightarrow \textbf{false}$$

$$\neg Bl \vee F \vee \neg B\neg l \vee G \rightarrow \textbf{true}$$

$$Bl \vee F \vee \neg B\neg l \vee G \rightarrow \neg B\neg l \vee F \vee G$$

$$Bl \wedge F \wedge \neg B\neg l \wedge B \rightarrow Bl \wedge F \wedge G.$$

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

$$\left\{ \begin{array}{l} C \Rightarrow A \\ D \Rightarrow B \end{array} \right\} \xrightarrow{\vdash C \Rightarrow D \;\; \vdash B \Rightarrow A} \{D \Rightarrow B\}.$$

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

5.7. Augmentation

The introduction of new propositions, such as t in Section 5.5, during the proof, makes proofs about the resolution method difficult. Furthermore, if a resolution proof involves two temporal resolution inferences involving the same literal, we would introduce two new propositions where one would suffice. Thus for n different eventualities we only require n new propositions. Given an eventuality $\mathbf{P}\Diamond l$, the new proposition introduced is w_l (rather than t above) thought of as *waiting for* l. Thus we introduce a proposition w_l for each literal occurring on the right hand side of a sometime rule in a systematic

way, that is used in the translation of the resolvents from temporal resolution into SNF_B. We introduce these new propositions at the start of the proof by adding the resolvents that have no reference to the loop detected (i.e., the clauses above labelled (4) and (6) which do not contain $\neg F_i$) at the beginning and the rest of the clauses as the proof proceeds. The following definitions formalise this technique and is known as *augmentation*. Hence having translated to SNF_B and augmented, no new propositions or labels appear during the application of the resolution rules.

Definition 13 (*Augmented* SNF_B *clause sets*). Given a set, S, of SNF_B clauses, we construct an augmented set of clauses $Aug(S)$ as follows. For each literal l which occurs as an eventuality in S we introduce a new proposition, w_l, and record the correspondence between l and w_l. The variable w_l will be used to record the condition that we are *waiting* for l to occur.

Thus for each sometime clause $Q \Rightarrow \mathbf{A}\Diamond l$ the defining clauses for w_l are

$$w_l \Rightarrow \mathbf{A}\bigcirc(l \vee w_l) \tag{7}$$

$$\mathbf{true} \Rightarrow (\neg Q \vee l \vee w_l) \tag{8}$$

and for each sometime clause $Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle}$ the defining clauses for w_l are

$$w_l \Rightarrow \mathbf{E}\bigcirc(l \vee w_l)_{\langle \mathsf{c_j} \rangle} \tag{9}$$

$$\mathbf{true} \Rightarrow (\neg Q \vee l \vee w_l). \tag{10}$$

Definition 14. The *loop resolvents* for an **A** sometime clause $Q \Rightarrow \mathbf{A}\Diamond l$ and a loop formula $\bigvee_i F_i$ are

$$\mathbf{true} \Rightarrow (\neg Q \vee l \vee \neg F_i) \tag{11}$$

$$w_l \Rightarrow \mathbf{A}\bigcirc(l \vee \neg F_i) \tag{12}$$

and the *loop resolvents* for an **E** sometime clause $Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle}$ and a loop formula $\bigvee_i F_i$ are

$$\mathbf{true} \Rightarrow (\neg Q \vee l \vee \neg F_i) \tag{13}$$

$$w_l \Rightarrow \mathbf{E}\bigcirc(l \vee \neg F_i)_{\langle \mathsf{c_j} \rangle}. \tag{14}$$

In summary, to apply one of the temporal resolution rules we must search for a set of merged clauses $F_i \Rightarrow \mathbf{P}\bigcirc G_i$ such that $\bigvee_i F_i \Rightarrow \mathbf{P}\bigcirc\mathbf{P}\Box\neg l$ corresponding to the relevant TRES rule (where $\bigvee_i F_i$ is known as a *loop formula* for $\neg l$).

5.8. Resolution method algorithm

Given any BB_n formula, F, to be tested for unsatisfiability, the following steps are performed.

(1) Translate F into SNF_B, giving F_s.
(2) Augment F_s, giving $Aug(F_s)$.

(3) Perform initial, modal and step resolution (including simplification and subsumption) on $Aug(F_s)$ until either
   (a) **start** $\Rightarrow$ **false** is derived—terminate noting that F is unsatisfiable; or
   (b) no new resolvents are generated—continue to step (4).
(4) Select an eventuality from the right hand side of a sometime clause within $Aug(F_s)$, for example $\Diamond l$. Search for loop formulae for $\neg l$ with which we can apply a temporal resolution rule.
(5) Construct loop resolvents for the loop formulae detected and each relevant sometime clause with $\Diamond l$ on the right hand side. If any new formulae (i.e., that are not subsumed by SNF_B clauses already present) have been generated, go to step (3).
(6) If all eventualities have been resolved, i.e., no new formulae have been generated for any of the eventualities, terminate declaring F satisfiable; otherwise go to step (4).

6. Examples

We consider two simple examples that can be represented within this logic. The first one shows how actions, plans and goals can be represented, while the second exhibits a refutation derived for a specific scenario.

6.1. Representing aspects of rational agency

The key aspects of agent theories, such as KARO [44] are to be able to represent an agents beliefs and its actions. Representing beliefs in our framework is simple; representing actions is also relatively easy. For example, if a particular action, α, has a certain pre-requisite, *pre*, and an effect *post*, then we can represent the action by

$$pre \Rightarrow \mathbf{A}\bigcirc(done(\alpha) \Rightarrow post).$$

Thus, if *pre* is satisfied in a state, then in all successor states where α has been done, then *post* is satisfied. Similarly, we can represent the fact that an action can not be undertaken is its precondition is not satisfied:

$$\neg pre \Rightarrow \mathbf{A}\bigcirc\neg done(\alpha).$$

In order to simply state the planning problem, we could use

$$\mathbf{start} \Rightarrow \mathbf{E}\Diamond goal$$

or, more realistically, use the following which states that the goal can be reached by undertaking a sequence of actions (taken from a finite set).

$$\mathbf{start} \Rightarrow \mathbf{E}\big((\exists a.\, done(a))\,\mathcal{U}\, goal\big).$$

In addition, we can represent the fact that an agent has beliefs about the actions it can perform, for example

$$B_i\mathbf{E}\bigcirc done(\alpha).$$

Note while these examples have been written using first-order syntax, as long as the set of actions is finite they can be rewritten as propositional BB_n formulae. Many further

examples of this form can be given, and properties of specifications of rational agents can be given (see, for example, [23]).

6.2. Belief about possibilities

Consider the formula (partially translated into the normal form).

$$B\mathbf{E}\Box oxygen \wedge B\mathbf{A}\Diamond hydrogen \wedge \mathbf{A}\Box^{*}\left[\begin{array}{r} safe \Rightarrow \mathbf{A}\Box\neg explode \wedge \\ (hydrogen \wedge oxygen) \Rightarrow explode \end{array}\right]$$

characterising the statement

> "I believe that in some possible future there will always be oxygen and I believe that in all possible futures, hydrogen will occur sometime. If something is safe then it will never explode and if hydrogen and oxygen occur together then they will explode."

Now, we characterise this as a set of SNF_B clauses and show that these contradict with the belief of safety $Bsafe$. The set of SNF_B clauses generated is

1. $\mathbf{start} \Rightarrow a$
2. $\mathbf{true} \Rightarrow \neg a \vee Bsafe$
3. $\mathbf{true} \Rightarrow \neg safe \vee \neg explode$
4. $\mathbf{true} \Rightarrow \neg safe \vee x$
5. $x \Rightarrow \mathbf{A}\bigcirc\neg explode$
6. $x \Rightarrow \mathbf{A}\bigcirc x$
7. $\mathbf{true} \Rightarrow \neg a \vee Bz$
8. $\mathbf{true} \Rightarrow \neg z \vee oxygen$
9. $\mathbf{true} \Rightarrow \neg z \vee w$
10. $w \Rightarrow \mathbf{E}\bigcirc oxygen_{\langle c_i\rangle}$
11. $w \Rightarrow \mathbf{E}\bigcirc w_{\langle c_i\rangle}$
12. $\mathbf{true} \Rightarrow \neg a \vee Bb$
13. $b \Rightarrow \mathbf{A}\Diamond hydrogen$
14. $\mathbf{true} \Rightarrow (\neg hydrogen \vee \neg oxygen \vee explode)$.

Statements 1 and 2 evolve from the $Bsafe$ (a is a new proposition used to rename $Bsafe$, $B\mathbf{E}\Box oxygen$ and $B\mathbf{A}\Diamond hydrogen$), 3–6 are from $safe \Rightarrow \mathbf{A}\Box\neg explode$, 7–11 are from $B\mathbf{E}\Box oxygen$, 12 and 13 are from $B\mathbf{A}\Diamond hydrogen$ and 14 is from $(hydrogen \wedge oxygen) \Rightarrow explode$. Next we augment the set of clauses with the following:-

15. $w_{hydrogen} \Rightarrow \mathbf{A}\bigcirc(hydrogen \vee w_{hydrogen})$
16. $\mathbf{true} \Rightarrow (\neg b \vee hydrogen \vee w_{hydrogen})$.

The refutation now proceeds as follows

17. $x \Rightarrow \mathbf{A}\bigcirc(\neg hydrogen \vee \neg oxygen)$ [5, 14, SRES4]
18. $(w \wedge x) \Rightarrow \mathbf{E}\bigcirc\neg hydrogen_{\langle \mathsf{C_i} \rangle}$ [10, 17, SRES2].

Merging clauses 6, 11 and 18 we obtain

$$x \wedge w \Rightarrow \mathbf{E}\bigcirc(x \wedge w \wedge \neg hydrogen)_{\langle \mathsf{C_i} \rangle}$$

for resolution with clause 13 giving the following resolvent.

19. $b \Rightarrow \mathbf{A}(\neg(w \wedge x)\,\mathcal{W}\,hydrogen)$ [6, 11, 13, 18, TRES3].

In rewriting 19 into SNF_B, one of the clauses (i.e., the loop resolvent (11) from Section 5.7) we get is

20. **true** $\Rightarrow (\neg b \vee hydrogen \vee \neg w \vee \neg x)$ [19]

from which the refutation continues as follows.

21. **true** $\Rightarrow (\neg b \vee \neg oxygen \vee explode \vee \neg w \vee \neg x)$ [14, 20, MRES1]
22. **true** $\Rightarrow (\neg safe \vee \neg b \vee \neg oxygen \vee \neg w \vee \neg x)$ [3, 21, MRES1]
23. **true** $\Rightarrow (\neg safe \vee \neg b \vee \neg oxygen \vee \neg w)$ [4, 22, MRES1]
24. **true** $\Rightarrow (\neg safe \vee \neg z \vee \neg b \vee \neg w)$ [8, 23, MRES1]
25. **true** $\Rightarrow (\neg safe \vee \neg z \vee \neg b)$ [9, 24, MRES1]
26. **true** $\Rightarrow (\neg a \vee \neg B safe \vee \neg Bz)$ [12, 25, MRES4b]
27. **true** $\Rightarrow (\neg a \vee \neg B safe)$ [7, 26, MRES1]
28. **true** $\Rightarrow \neg a$ [2, 27, MRES1]
29. **start** $\Rightarrow$ **false** [1, 28, IRES1].

7. Soundness, completeness and termination

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

Theorem 15. *A BB_n formula A is satisfiable if, and only if, $\tau_0[A]$ is satisfiable.*

Proof. Proofs analogous to those in [6,17,25] will suffice. □

Next we show that augmenting the clause set preserves satisfiability. We will show that an augmented clause set has a *model* if and only if its underlying (non-augmented) clause set has a model.

Definition 16. Given a set, S, of SNF_B clauses, a *normal model* for the augmented clause set for S is a model which satisfies the formula

$$\Box\big(w_l \Leftrightarrow (\neg l \wedge \mathbf{A}\Diamond l)\big) \tag{15}$$

for each literal l which occurs as an eventuality in an **A** sometime clause in S and

$$\Box\big(w_l \Leftrightarrow (\neg l \wedge \mathbf{E}\Diamond l_{\langle \mathsf{c_j}\rangle})\big) \tag{16}$$

for each literal l which occurs as an eventuality in an **E** sometime clause in S labelled by $\langle \mathsf{c_j}\rangle$.

Definition 17. An augmented clause set is said to be *well-behaved* if it is either unsatisfiable or has a normal model.

Lemma 18 (Augmentation). *If S is a set of* SNF_B *clauses then*

(1) *$Aug(S)$ is well-behaved, and,*
(2) *$Aug(S)$ has a model if and only if S has a model.*

Proof. If $Aug(S)$ has a model then, ignoring the value of each w_l at each moment gives a model for S. Conversely, if S has a model M, then M can be extended to a model M' for $Aug(S)$ by giving w_l the same truth value as $\neg l \wedge \mathbf{A}\Diamond l$ in M in each state, and for each literal l on the right hand side of an **A** sometime clause and the same truth value as $\neg l \wedge \mathbf{E}\Diamond l_{\langle \mathsf{c_j}\rangle}$ in M in each state, and for each literal l on the right hand side of an **E** sometime clause labelled by $\langle \mathsf{c_j}\rangle$. The model M' clearly satisfies the formulae (7), (8), (9) and (10) from Section 5.7 and (15) and (16) above. The lemma follows easily from these two observations. □

Lemma 19 (Soundness—initial, step and modal resolution). *Let S be a set of* SNF_B *clauses. Let the clause set T be obtained from S by the application of an initial, step or modal resolution inference. Then S is satisfiable if and only if T is satisfiable.*

Proof. We prove soundness of the modal resolution rule MRES2. The other rules are similar. Assume the clauses $\mathbf{true} \Rightarrow D \vee Bl$ and $\mathbf{true} \Rightarrow D' \vee B\neg l$ are in S and that S is satisfiable. Let T be obtained by applying MRES2 to $\mathbf{true} \Rightarrow D \vee Bl$ and $\mathbf{true} \Rightarrow D' \vee B\neg l$, i.e., $T = S \cup \{\mathbf{true} \Rightarrow D \vee D'\}$. As S is satisfiable there must be a model M that satisfies S. We show M also satisfies T. Take any point (t, j) in M. Thus $\langle M, (t, j)\rangle \models D \vee Bl$ and $\langle M, (t, j)\rangle \models D' \vee B\neg l$. From the semantics of disjunction either $\langle M, (t, j)\rangle \models D$ or $\langle M, (t, j)\rangle \models Bl$. If the former then $\langle M, (t, j)\rangle \models D \vee D'$ and thus $\langle M, (t, j)\rangle \models \mathbf{true} \Rightarrow D \vee D'$ and we are done. Otherwise assume that $\langle M, (t, j)\rangle \not\models D$ and $\langle M, (t, j)\rangle \models Bl$. Hence $\langle M, (t, j)\rangle \models \neg B\neg l$ from axiom D. Thus from the semantics of negation $\langle M, (t, j)\rangle \not\models B\neg l$. Thus $\langle M, (t, j)\rangle \models D'$ and therefore from the semantics of disjunction $\langle M, (t, j)\rangle \models D \vee D'$ and $\langle M, (t, j)\rangle \models \mathbf{true} \Rightarrow D \vee D'$ as required. If S is unsatisfiable then by adding $\mathbf{true} \Rightarrow D \vee D'$ to obtain T, T is still unsatisfiable. □

Lemma 20 (Soundness—temporal resolution). *Let S be a well-behaved augmented clause set. Let the clause set T be obtained from S by application of the temporal resolution operation. Then*

(1) *T is well-behaved, and,*
(2) *S is satisfiable iff T is satisfiable.*

Proof. If S is satisfiable so S has a model, then by Definition 17 it has a normal model M. The side conditions for temporal resolution guarantee that the loop resolvents, i.e., formulae (11) and (12) (or (13) and (14) respectively) given in Section 5.7 hold in M, and so M is a (normal) model for T, i.e., T is satisfiable. If S is unsatisfiable then the addition of clauses to produce T is also unsatisfiable. Hence T is well-behaved. □

Theorem 21 (Soundness). *Given a satisfiable set of* SNF_B *clauses S, if T is obtained by applying a resolution rule then S is satisfiable iff T is satisfiable.*

Proof. Lemma 18 shows augmentation preserves satisfiability, i.e., S is satisfiable iff $Aug(S)$ is satisfiable. In the following assume S is augmented. Lemma 19 shows that S is satisfiable iff T is satisfiable if T is obtained from S by the application of an initial, modal or step resolution rule. Lemma 20 shows that S is satisfiable iff T is satisfiable if T is obtained from S by the application of a temporal resolution rule. Hence the theorem follows. □

Theorem 22 (Termination). *The resolution method applied to a set of* SNF_B *clauses will terminate.*

Proof. We do not need any new labels during the proof (these are fixed after the translation into SNF_B) and by augmenting the clause set similarly we do not require any new propositions during the proof. Hence, modulo ordering of the literals, there are a finite number of right and left hand sides of each type of clause. Hence either we terminate having derived a contradiction or we can generate no new clauses. □

7.1. Completeness

The proof of completeness is based on that given in [25]. We construct a graph of the set of SNF_B clauses that has two types of edges representing the modal and temporal dimensions. Temporal edges may be labelled with a set of labels to capture the movement from one state to another by satisfying a labelled **E** step clause. We show that an empty graph corresponds to an unsatisfiable set of clauses and then that an unsatisfiable set of clauses has a refutation by the resolution method presented in this paper.

Graph construction

Definition 23 (*Labelled behaviour graph*). Let T be a set of augmented SNF_B clauses and LAB be the set of labels labelling **E** sometime clauses and **E** step clauses in T.

Given T, we construct a finite directed graph $H = (N, E_B, E_T)$, for T where N is the set of nodes, $E_B \subseteq (N \times N)$ is the set of modal edges representing belief, and $E_T \subseteq (N \times \mathcal{P}(LAB) \times N)$ is the set of labelled temporal edges (where $\mathcal{P}(LAB)$ denotes the powerset of the set of labels). A node, $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$, in H is a triple where V, $Y_{\mathbf{A}}$ and $Y_{\mathbf{E}}$ are constructed as follows. For any proposition p occurring in T, V contains (consistent) subsets of all the literals or modal literals that can be constructed from p, namely the formulae $\{p, \neg p, Bp, \neg Bp, B\neg p, \neg B\neg p\}$. Thus we construct all possible sets of formulae containing p or its negation, Bp or its negation, and $B\neg p$ or its negation. Next we reduce the number of these sets that we must consider by using the axioms of KD45. Thus we cannot have a set containing Bp and $B\neg p$ as from the D axiom $Bp \Rightarrow \neg B\neg p$ which contradicts with $B\neg p$. For each proposition $p \in T$ this leaves the following six sets

$$V_p = \big\{\{Bp, \neg B\neg p, p\}, \{Bp, \neg B\neg p, \neg p\}, \{B\neg p, \neg Bp, p\}, \{B\neg p, \neg Bp, \neg p\}, \{\neg B\neg p, \neg Bp, p\}, \{\neg B\neg p, \neg Bp, \neg p\}\big\}.$$

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

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

Nodes are triples $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ where $Y_{\mathbf{A}}$ is a subset of the literals that occur on the right hand side of **A** sometime clauses in T and $Y_{\mathbf{E}}$ is a subset of $\{l_{\langle \mathsf{c_j} \rangle} \mid Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle} \in T\}$. That is $Y_{\mathbf{E}}$ contains labelled literals of the form $l_{\langle \mathsf{c_j} \rangle}$ where l is a literal that occurs on the right hand side of an **E** sometime clause in T and $\langle \mathsf{c_j} \rangle$ is the label of that clause. Informally $Y_{\mathbf{A}}$ contains currently unsatisfied eventualities of the **A** sometime clauses and $Y_{\mathbf{E}}$ contains unsatisfied eventualities of the **E** sometime clauses that have been labelled by the relevant label.

Delete any node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ such that for some belief clause of the form

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

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 belief clauses.

Next we push the external B operator (from the $\mathbf{A}\Box^*$ operator surrounding each clause) into each clause and delete nodes that do not satisfy the new set of clauses. Consider any literal clause whose right hand side consists of a single literal, for example $\mathbf{true} \Rightarrow l$. By pushing in the external B-operator this clause is equivalent to $\mathbf{true} \Rightarrow Bl$ so we delete any nodes where Bl is not satisfied. Next consider any belief clause whose right hand side consists of a single literal disjoined with one or more modal literals, for example $\mathbf{true} \Rightarrow l \vee Ba \vee \neg Bb \vee \neg Bc$. By pushing in the external B-operator this clause is equivalent to $\mathbf{true} \Rightarrow Bl \vee Ba \vee \neg Bb \vee \neg Bc$ so any node that does not satisfy this clause is deleted (recall in KD45 $\neg BB\neg p \Leftrightarrow \neg B\neg p$ and $Bp \Leftrightarrow \neg B\neg Bp$). Note we could also obtain $\mathbf{true} \Rightarrow \neg B\neg l \vee Ba \vee \neg Bb \vee \neg Bc$ but as the original clause implies this clause (i.e., the original clause subsumes this clause) we ignore it. Finally consider a belief clause with more than one literal disjoined on the right hand side, for example $\mathbf{true} \Rightarrow l_1 \vee l_2 \vee Ba$. We must consider all possible ways of pushing in the B operator into this clause obtaining

true $\Rightarrow Bl_1 \vee \neg B \neg l_2 \vee Ba$ or **true** $\Rightarrow \neg B \neg l_1 \vee Bl_2 \vee Ba$. Nodes that don't satisfy these additional clauses are deleted.

Next delete any nodes $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ such that for any **A** sometime clause $Q \Rightarrow \mathbf{A}\Diamond l$ it is not the case that if $V \models Q$ then $l \in Y_{\mathbf{A}}$. Similarly delete any nodes $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ such that for any **E** sometime clause $Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c_j} \rangle}$ it is not the case that if $V \models Q$ then $l_{\langle \mathsf{c_j} \rangle} \in Y_{\mathbf{E}}$. Informally if the left hand side of a **A** (respectively **E**) sometime clause is satisfied then the eventuality must be contained in the set of **A** (respectively **E**) eventualities in that node.

Next we construct the belief edges (E_B edges) between the undeleted nodes as follows. Given node, $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$, we construct E_B edges to any node $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ as follows:

(1) if $Bl \in V$ then $V' \models l$ (i.e., $V' \models B\_\text{set}(V)$); and
(2) $Bl \in V \Leftrightarrow Bl \in V'$ and $\neg Bl \in V \Leftrightarrow \neg Bl \in V'$ for each literal l.

Step 1 ensures that for any modal literal in V of the form Bl, 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 $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$, let X be the largest subset of the **A** step clauses such that V satisfies the literals on the left hand sides of each clause. Let C be the set of right hand sides of the clauses in X having deleted the $\mathbf{A}\bigcirc$ operators. Note if V does not satisfy the left hand side of any **A** step clauses (i.e., $X = \emptyset$) there are no constraints from the **A** step clauses on the next state so $C =$ **true**. Let $R^{\mathbf{E}}$ be the largest set of **E** step clauses of T such that V satisfies the left hand side. Let $R^{\mathbf{E}^{I'}} \subseteq R^{\mathbf{E}}$ be the set of **E** step clauses $\{F \Rightarrow \mathbf{E}\bigcirc G_{\langle \mathsf{c_j} \rangle} \in R^{\mathbf{E}} \mid \langle \mathsf{c_j} \rangle \in I'\}$. For each $I' \subseteq LAB$ let $C^{I'}$ be the set right hand sides of the clauses in $R^{\mathbf{E}^{I'}}$ having deleted the $\mathbf{E}\bigcirc$ operators. As with the **A** step clauses if this set is empty $C^{I'} =$ **true**.

Let V' be a set of literals and modal literals from a node in H that satisfies $C \cup C^{I'}$, i.e., V' satisfies the right hand sides of the clauses in $X \cup R^{\mathbf{E}^{I'}}$. Let $Y'_{\mathbf{A}} \subseteq Y_{\mathbf{A}}$ be the set of literals in $Y_{\mathbf{A}}$ not satisfied by V. Let $Y''_{\mathbf{A}}$ be the set of literals obtained from the right hand side of the **A** sometimes clauses where V' satisfies the left hand side. Let $Y'''_{\mathbf{A}} = Y'_{\mathbf{A}} \cup Y''_{\mathbf{A}}$. Informally $Y'''_{\mathbf{A}}$ keeps track of unsatisfied eventualities, i.e., $Y'_{\mathbf{A}}$, those remaining unsatisfied by V plus $Y''_{\mathbf{A}}$, new eventualities triggered by V'. Similarly let $Y'_{\mathbf{E}} \subseteq Y_{\mathbf{E}}$ be the set of labelled literals labelled by a member of I' not satisfied by V (i.e., $l_{\langle \mathsf{c_j} \rangle} \in Y'_{\mathbf{E}}$ iff $(l_{\langle \mathsf{c_j} \rangle} \in Y_{\mathbf{E}}) \wedge (\langle \mathsf{c_j} \rangle \in I') \wedge (l \notin V)$). Let $Y''_{\mathbf{E}}$ be the set of labelled literals obtained from the right hand side of **E** sometime clauses where V' satisfies the left hand side. As before let $Y'''_{\mathbf{E}} = Y'_{\mathbf{E}} \cup Y''_{\mathbf{E}}$ where $Y'''_{\mathbf{E}}$ keeps track of unsatisfied eventualities for a subset of labels I', i.e., $Y'_{\mathbf{E}}$, those labelled by a member of I' remaining unsatisfied by V plus $Y''_{\mathbf{E}}$, new sometime eventualities triggered by V'. Edges labelled by I' are constructed from $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ to $(V', Y'''_{\mathbf{A}}, Y'''_{\mathbf{E}})$ for each V', $Y'''_{\mathbf{A}}$, and $Y'''_{\mathbf{E}}$. Repeat for all $I' \subseteq LAB$. Delete any edge (n, I'', n') such that there exists an edge (n, I', n') and $I'' \subseteq I'$. These are the only edges out of $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$.

Let L_0 be the set of initial clauses of T. Let C_0 be the set of right hand sides of the clauses in L_0. For each valuation V which satisfies C_0, where $Y_{\mathbf{A}}$ is the set of literals occurring on the right hand sides of the **A** sometime clauses fired by V, i.e., for each $Q \Rightarrow \mathbf{A}\Diamond l \in T$ such that $V \models Q$, $l \in Y_{\mathbf{A}}$ and $Y_{\mathbf{E}}$ is the set of labelled literals $l_{\langle \mathsf{c_j} \rangle}$ such that

$Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c}_j \rangle} \in T$ and $V \models Q$. The node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ is designated as an *initial node* of H. The *labelled behaviour graph* for a set of SNF$_B$ clauses T is the set of nodes and edges reachable from the initial nodes by either E_B or E_T edges.

Lemma 24. *The sets of nodes disallowed during the construction of V_p, namely $\{Bp, B\neg p, p\}$ and $\{Bp, B\neg p, \neg p\}$ are unsatisfiable.*

Proof. Sets containing both Bp and $B\neg p$ are unsatisfiable due to the D axiom. Applying the D axiom $Bp \Rightarrow \neg B\neg p$ to Bp we can infer $\neg B\neg p$ which contradicts with $B\neg p$. □

Definition 25. Given a set of labels I, an I *labelled terminal subgraph* $(N', \emptyset, E'_T)$ of a graph (N, E_B, E_T) is one such that

- $N' \subseteq N$ and $E'_T \subseteq E_T$; and
- for any node $n \in N'$ if for any edge $(n, J, n') \in E_T$ and $I \cap J \neq \emptyset$ then $n' \in N'$ and $(n, J, n') \in E'_T$ and for all edges $(n, J', n'') \in E_T$ such that $I \cap J' = \emptyset$ then $(n, J', n'') \notin E'_T$.

Definition 26. Given a graph (N, E_B, E_T), a set of labels I and a node $n \in N$, a node $n' \in N$ is I *reachable* from n is one such that

- $(n, J, n') \in E_T$ is labelled by one or more elements of I, i.e., $I \cap J \neq \emptyset$; or
- $(n, J, n'') \in E_T$ is labelled by one or more elements of I, i.e., $I \cap J \neq \emptyset$ and n' is I reachable from n''; or

Lemma 27. *Let S be a set of* SNF$_B$ *clauses such that $Q \Rightarrow \mathbf{A}\Diamond l \in S$ and let $H = (N, E_B, E_T)$ be the labelled behaviour graph for S. For any node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}}) \in H$ if $l \in Y_{\mathbf{A}}$ and $V \not\models l$ then $V \models w_l$.*

Proof. From the construction of the labelled behaviour graph an edge may only be drawn from node $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ to a node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ such that $l \in Y_{\mathbf{A}}$ iff either $V \models Q$ or $l \in Y'_{\mathbf{A}}$. The proof is by induction on the length of the shortest path to n from a node $n'' = (V'', Y''_{\mathbf{A}}, Y''_{\mathbf{E}})$ such that $V'' \models Q$.

The base case is where the length is zero, i.e., $n = n''$ and therefore $V \models Q$ and $l \in Y_{\mathbf{A}}$ by construction. By augmentation $\mathbf{true} \Rightarrow \neg Q \vee l \vee w_l \in S$. By assumption $V \not\models l$, hence $V \models w_l$.

Otherwise assume the lemma holds for nodes n' distance m from a node n'' where $V'' \models Q$ and prove it holds for those $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ distance $m + 1$. Thus $l \in Y_{\mathbf{A}}$ and $V \not\models l$ and $V \not\models Q$ by assumption. From the inductive hypothesis we have $V' \models w_l$. By augmentation we have $w_l \Rightarrow \mathbf{A}\bigcirc(w_l \vee l) \in S$. Thus by construction we have $V \models w_l$. □

Lemma 28. *Let S be a set of* SNF$_B$ *clauses such that $Q \Rightarrow \mathbf{E}\Diamond l_{\langle \mathsf{c}_j \rangle} \in S$ and $H = (N, E_B, E_T)$ be the labelled behaviour graph for S. If for any node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}}) \in H$ such that $l_{\langle \mathsf{c}_j \rangle} \in Y_{\mathbf{E}}$ and $V \not\models l$ then $V \models w_l$.*

Proof. Similar to the above but we use the fact that we have augmented with $w_l \Rightarrow \mathbf{E}\bigcirc(w_l \vee l)_{\langle \mathrm{c_j} \rangle} \in S$. □

Lemma 29. *Let S be a set of* SNF_B *clauses and let T be the set of* SNF_B *clauses obtained from S by adding finitely many initial clauses or finitely many* **A***-step clauses or finitely many literal or belief clauses which only involve propositions occurring in S. Then the labelled behaviour graph of T ($H' = (N', E'_B, E'_T)$) contains a subset of nodes and edges from that for S ($H = (N, E_B, E_T)$), i.e., $N' \subseteq N$, $E'_B \subseteq E_B$, $E'_T \subseteq E_T$.*

Proof. This is established by induction on the length of the shortest path by following modal or temporal edges from an initial node to another node in the labelled behaviour graph of T. Let *len* be the length of the shortest path from an initial node to a node n. To show the base case we let $len = 0$ and show that any initial node in the labelled behaviour graph of T is an initial node in the labelled behaviour graph of S. Let $I \subseteq S$ be the initial clauses of S and $I' \subseteq T$ the initial clauses of T. Let $M \subseteq S$ be the literal or belief clauses of S and $M' \subseteq T$ be the literal or belief clauses of T. As T has been constructed by adding initial or **A**-step clauses and/or literal or belief clauses to S, $I \subseteq I'$ and $M \subseteq M'$. Take any initial node $n_0 = (V_0, Y_{\mathbf{A}_0}, Y_{\mathbf{E}_0})$ in the labelled behaviour graph for T. From the definition of the labelled behaviour graph V_0 must satisfy the right hand side of the clauses in I' and the right hand side of clauses in M'. As $I \subseteq I'$ and $M \subseteq M'$ then V_0 must also satisfy the right hand side of the clauses in both I and M. As the set of sometime clauses in S and T are unchanged, i.e., as V_0 satisfies the left hand side of the same sometime clauses in S and T the sets $Y_{\mathbf{A}_0}$ and $Y_{\mathbf{E}_0}$ will be the same in each graph for V_0 and thus the node $n_0 = (V_0, Y_{\mathbf{A}_0}, Y_{\mathbf{E}_0})$ is also in the labelled behaviour graph for S.

Next we assume that if any node n, where the length of the shortest path from an initial node to n is m, is in the labelled behaviour graph for T, it is also in the labelled behaviour graph for S. We show that any node n' in the labelled behaviour graph for T whose shortest path length from an initial node is $m + 1$, is also in the labelled behaviour graph for S. Let $J \subseteq S$ be the **A**-step clauses in S and $J' \subseteq T$ the **A**-step clauses in T. As above let $M \subseteq S$ be the literal or belief clauses of S and $M' \subseteq T$ be the literal or belief clauses of T. By assumption we have $J \subseteq J'$ and $M \subseteq M'$. Consider some node $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ in the labelled behaviour graph of T where the shortest path from an initial node to n' is $m + 1$. Let $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ be any node in the labelled behaviour graph for T such that there is an edge either in E'_B or E'_T from n to n' and the shortest path from an initial node to n is of length m. By the induction hypothesis, we assume that n is also in the labelled behaviour graph for S.

First assume (n, I, n') is in E'_T. Let $X' \subseteq J'$ be the set of **A**-step clauses in T such that the left hand sides are satisfied by V. By construction the right hand sides of the clauses in X' (without the $\mathbf{A}\bigcirc$ operators) are satisfied by V'. Let $X \subseteq J$ be the corresponding set of **A**-step clauses in S, i.e., where the left hand sides are satisfied by V and thus by construction the right hand sides (without the $\mathbf{A}\bigcirc$ operators) are satisfied by V'. As $J \subseteq J'$ we have $X \subseteq X'$ and as previously we have $M \subseteq M'$ so V' must also satisfy the right hand sides of the clauses in X and M. Furthermore as no change has been made to the set of sometime clauses any eventualities outstanding from n or triggered by n' will be the same in each graph. Thus n' is also present in the labelled behaviour graph for S.

Next assume (n, n') is in E'_B. Thus both V and V' must satisfy M' by definition and for all $Bl \in V$, $V' \models l$, and $Bl \in V$ iff $Bl \in V'$, and $\neg Bl \in V$ iff $\neg Bl \in V'$. We have $M \subseteq M'$ so V' satisfies M. Other criteria are unchanged so V' is also in H. □

Lemma 30. *Let S be a set of* SNF_B *clauses and let T be the set of* SNF_B *clauses obtained from S by adding an* **E***-step clause from performing a step resolution inference* (SRES2, SRES3, SRES4) *or from applying* TRES2 *or* TRES4*. Then the labelled behaviour graph of T $(H' = (N', E'_B, E'_T))$ contains a subset of nodes and edges from that for S $(H = (N, E_B, E_T))$, i.e., $N' \subseteq N$, $E'_B \subseteq E_B$, $E'_T \subseteq E_T$.*

Proof. Assume $P \Rightarrow \mathbf{E}\bigcirc(F \vee l)_{\langle \mathsf{c_j} \rangle}$ and $Q \Rightarrow \mathbf{A}\bigcirc(G \vee \neg l)$ are both in S and T and that $T = S \cup \{(P \wedge Q) \Rightarrow \mathbf{E}\bigcirc(F \vee G)_{\langle \mathsf{c_j} \rangle}\}$, i.e., following an application of SRES2. That is, the set T is the same as S apart from adding the above resolvent. Take any node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ in the labelled behaviour graph for T such that V satisfies both P and Q. Let $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ where (n, I, n') and $\langle \mathsf{c_j} \rangle \in I$. From the construction of the graph V' satisfies $(F \vee l)$, $(G \vee \neg l)$ and $(F \vee G)$. As V' satisfies $(F \vee l)$ and $(G \vee \neg l)$ then n' is also in S where $\langle \mathsf{c_j} \rangle \in I$ for (n, I, n'). The argument is similar when applying SRES3 and SRES4.

Finally take the addition of $w_l \Rightarrow \mathbf{E}\bigcirc(l \vee \neg F_i)_{\langle \mathsf{c_j} \rangle}$ to S to form T after temporal resolution between a sometime clause labelled by $\langle \mathsf{c_j} \rangle$ and a loop formula $\bigvee_i F_i$. From augmentation S (and therefore T) must contain $w_l \Rightarrow \mathbf{E}\bigcirc(l \vee w_l)_{\langle \mathsf{c_j} \rangle}$. Take a node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ in T such that V satisfies w_l. Let $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ be a node in the labelled behaviour graph for T such that $(n, I, n') \in E'_T$ and $\langle \mathsf{c_j} \rangle \in I$. Thus by construction V' satisfies $(l \vee \neg F_i)$ and $(l \vee w_l)$ so V' is also in the labelled behaviour graph for S. □

Definition 31 (*Reduced labelled behaviour graph*). Given a labelled behaviour graph for a set of clauses T carry out the following deletions. Delete any node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ and any edges into or out of n as follows.

(1) If a node has no temporal edges leading from it delete this node and all edges into it.
(2) If $R^{\mathbf{E}\{\langle c_i \rangle\}}$, for some node and label $\langle c_i \rangle$, is non-empty, where $R^{\mathbf{E}\{\langle c_i \rangle\}}$ is the largest set of **E** step clauses of T labelled by $\langle c_i \rangle$ such that V satisfies the left hand side of each clause in $R^{\mathbf{E}\{\langle c_i \rangle\}}$, and no temporal edges leading from this node are labelled by $\langle c_i \rangle$ delete this node and all edges into it.
(3) If a node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ contains an eventuality $l \in Y_{\mathbf{A}}$ and l is neither satisfied by V nor is there a node reachable from $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ by following temporal (E_T) edges only, whose valuation satisfies l then $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ is deleted.
(4) If a node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ contains an eventuality $l \in Y_{\mathbf{A}}$ and l is neither satisfied by V nor is there a node I reachable from $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$, for I some non-empty subset of labels occurring in T, whose valuation satisfies l then $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ is deleted.
(5) If a node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ contains a labelled eventuality $l_{\langle \mathsf{c_j} \rangle} \in Y_{\mathbf{E}}$ and l is neither satisfied by V nor is there a node reachable from $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ whose valuation satisfies l by following temporal (E_T) edges only then $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ is deleted.

(6) If a node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ contains a labelled eventuality $l_{\langle \mathsf{c_j} \rangle} \in Y_{\mathbf{E}}$ and l is neither satisfied by V nor is there a node $\{\langle \mathsf{c_j} \rangle\}$ reachable from $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ whose valuation satisfies l then $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ is deleted.

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

Lemma 32. *Let T be a set of* SNF_B *clauses and $H = (N, E_B, E_T)$ be the reduced labelled behaviour graph for T. If $\neg Bl \in V$ for some node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}}) \in N$ then there is a node $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}}) \in N$ such that $V \models \neg l$ where $(n, n') \in E_B$.*

Proof. Assume H contains a node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}}) \in N$ that contains a formula $\neg Bl$, i.e., $\neg Bl \in V$ and there is no node $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ such that $(n, n') \in E_B$ and $\neg l \in V'$. Note that V cannot contain Bl as it contains $\neg Bl$ and nodes containing Bl and $\neg Bl$ are not permitted as they are unsatisfiable. By construction of the graph each node must satisfy each belief clause in the clause set, plus each clause with the external B pushed into it. Assume first that the literals and modal literals in V without l, i.e., $V \backslash \{l\}$, satisfy the set of belief clauses with the B-operator pushed into each clause. Hence there must be a node $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ 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 $\neg Bl$ can be satisfied in a node n' where $(n, n') \in E_B$ and our original assumption was wrong. Next we assume that l must be in V to satisfy a clause in T. We consider three cases.

- Assume that l is in V as T contains the clause $\mathbf{true} \Rightarrow l$. To satisfy this clause every node in H contains l. By pushing in the external B the set of clauses including the pushed clauses must contain the clause $\mathbf{true} \Rightarrow Bl$ so each node in H must also contain Bl. No node can contain both Bl and $\neg Bl$ hence l cannot be in V because of a clause $\mathbf{true} \Rightarrow l$.
- Assume that l is in V from satisfying a clause $\mathbf{true} \Rightarrow D \vee l$ where D is a disjunction of modal literals and where V does not satisfy D. By pushing in the external B operator we obtain the clause $\mathbf{true} \Rightarrow D \vee Bl$ hence n must also contain Bl. As no node can contain both Bl and $\neg Bl$ the node n cannot exist because of satisfying l in a clause $\mathbf{true} \Rightarrow D \vee l$. Note if V satisfied D the case is as described above—a node $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ must exist such that V' is the same as V except it contains $\neg l$ rather than l and hence $\neg Bl$ is satisfiable.
- Assume that l is in V from satisfying a clause $\mathbf{true} \Rightarrow D \vee l$ 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 $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ such that $(n, n') \in E_B$, $\neg Bl \in V'$ and V' must satisfy $\neg M$ by construction of the graph (as M is a disjunction of modal literals). However if V' satisfies L (and therefore D) and $\neg l$ then the clause $\mathbf{true} \Rightarrow D \vee l$ is satisfied. Further $\neg Bl$ is satisfied as required. The only situation when this does not occur is when $(\neg Bl \wedge \neg M) \Rightarrow \neg L$, i.e., to satisfy the clause, $\mathbf{true} \Rightarrow L \vee M \vee l$, l must hold in each node when both $\neg M$ and $\neg Bl$ do. Consider any node n' such that $(n, n') \in E_B$. The node n' must satisfy both $\neg Bl$ and $\neg M$, by construction of the graph, and therefore

must also contain l. Hence there is no node n' such that $(n, n') \in E_B$ containing $\neg l$. However having pushed the B-operator into the clauses each node must satisfy **true** $\Rightarrow Bl \vee M \vee \neg BL$. Here no node n' such that $(n, n') \in E_B$ contains L so $\neg BL$ is unsatisfiable, M is unsatisfiable by assumption, so n must contain Bl. However we have assumed n contains $\neg Bl$ so n does not satisfy the set of pushed clauses and must be deleted. □

Lemma 33. *If $H = (N, E_B, E_T)$ is a reduced labelled behaviour graph for a set of clauses then the set of edges E_B form a relation which is transitive, serial and Euclidean.*

Proof. To show the E_B relation is transitive in the unreduced graph take any nodes $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}}) \in N$ and $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}}) \in N$ and $n'' = (V'', Y''_{\mathbf{A}}, Y''_{\mathbf{E}}) \in N$ where $(n, n') \in E_B$ and $(n', n'') \in E_B$. We must show that $(n, n'') \in E_B$. From the conditions of adding modal edges we must show that

(1) if $Bl \in V$ then $V'' \models l$ (i.e., $V'' \models B\_\text{set}(V)$); and
(2) $Bl \in V'' \Leftrightarrow Bl \in V$ and $\neg Bl \in V'' \Leftrightarrow \neg Bl \in V$ for each literal l.

From condition (2) of adding edges between nodes we know that the set of modal literals in V and V' are the same and the set of modal literals in V' and V'' are the same. Hence the set of modal literals in V and V'' are the same thus satisfying the second condition. As the set of modal literals in V and V' are the same then $B\_\text{set}(V) = B\_\text{set}(V')$. From condition (1) applied to node n', $V'' \models B\_\text{set}(V')$ thus $V'' \models B\_\text{set}(V)$ and the first condition is also fullfilled. To obtain the reduced labelled behaviour graph nodes are deleted from the unreduced labelled behaviour graph. If a node is deleted from a relation that is transitive then the relation is still transitive so the E_B relation is transitive in the reduced labelled behaviour graph.

To show Euclideaness take any nodes $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}}) \in N$ and $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}}) \in N$ and $n'' = (V'', Y''_{\mathbf{A}}, Y''_{\mathbf{E}}) \in N$ where $(n, n') \in E_B$ and $(n, n'') \in E_B$. We must show that $(n', n'') \in E_B$. From the conditions of adding modal edges we must show that

(1) if $Bl \in V'$ then $V'' \models l$ (i.e., $V'' \models B\_\text{set}(V')$); and
(2) $Bl \in V'' \Leftrightarrow Bl \in V'$ and $\neg Bl \in V'' \Leftrightarrow \neg Bl \in V'$ for each literal l.

From condition (2) of adding edges between nodes we know that the set of modal literals in V and V' are the same and the set of modal literals in V and V'' are the same, hence the set of modal literals in V'' and V' are the same. Thus the second condition follows. As the set of modal literals in V, V' and V'' are the same then $B\_\text{set}(V) = B\_\text{set}(V') = B\_\text{set}(V'')$. From the construction of the graph and as $(n, n'') \in E_B$ by assumption then $V'' \models B\_\text{set}(V)$. Thus $V'' \models B\_\text{set}(V')$ and the first condition is also fullfilled. The deletion of nodes to obtain the reduced labelled behaviour graph doesn't affect the property of Euclideaness so the E_B relation is Euclidean in the reduced labelled behaviour graph.

To show seriality we know every node in the reduced labelled behaviour graph contains a modal formula of the form $\neg Bl$. From Lemma 32 any node containing $\neg Bl$ has an edge to a node containing $\neg l$, hence the relation is serial.

Hence the sets of nodes reachable via the relations in E_B are transitive, Euclidean and serial and can be used to construct the R relation. □

Proposition 34. *A set of clauses T in* SNF_B *is unsatisfiable if and only if its reduced labelled behaviour graph is empty.*

Proof. We start by showing the 'if' part. The construction of nodes in the labelled behaviour graph generate all possible states the system may be in and any nodes reachable from the initial nodes in the reduced labelled behaviour graph can be used to construct a model for T by unwinding through the temporal edges to construct trees and using modal edges to reconstruct the transitive, Euclidean, serial relations.

We begin by justifying our choice of sets of literals and modal literals for each proposition $p \in T$ for each node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$. From Lemma 24 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 belief clause so the deletion of nodes that do not satisfy the belief clauses does not remove any models. Finally to take account of the external B operator we push B into each clause and ensure that each node satisfies this set of clauses.

Secondly, infinite trees can be constructed by starting from an initial node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$, adding to the tree a node n' for each different label occurring on edges $(n, I, n') \in E_T$ and zero or more other nodes. If all edges $(n, I, n') \in E_T$ are unlabelled, i.e., $I = \emptyset$ at least one unlabelled edge must be added. An infinite tree of propositional valuations is obtained by extracting the literals from each V. By construction of the graph, this sequence satisfies the conditions for constructing trees for T apart from the conditions concerning the satisfaction of eventualities (and none infinite paths). The reconstruction of the (belief) R relation from the reduced labelled behaviour graph, H, will construct transitive, Euclidean, serial relations from the construction of the E_B edges in H.

If the unreduced labelled behaviour graph is empty then there are no nodes that directly satisfy the set of clauses 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 clauses T then T must be unsatisfiable.

If the unreduced labelled 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. Hence we can apply deletion criterion one without removing any models. If the left hand side of a node n satisfies one or more **E** step clauses labelled by $\langle \mathsf{c_j} \rangle$ and no edge out of n is labelled by $\langle \mathsf{c_j} \rangle$ then we cannot satisfy the right hand side of the **E** step clauses so any model of T must arise from a path through the graph with n deleted. Hence we can apply deletion criterion two without removing any models.

Also if n contains an **A** 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 third deletion criterion without discarding any potential models. Similarly, if n contains an **A** eventuality l then any path

labelled by a subset of labels 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 particular labelled temporal edges. Hence we can apply the fourth deletion criterion without discarding any potential models. If $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ contains an $\mathbf{E}$ eventuality l, i.e., $l_{\langle \mathsf{c_j} \rangle} \in Y_{\mathbf{E}}$ then l must be satisfied on an $\{\langle \mathsf{c_j} \rangle\}$ reachable path. Hence if l is not satisfied on *any* path we can apply the fifth deletion criterion. Finally, given a node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ which contains eventuality l labelled by $\langle \mathsf{c_j} \rangle$, i.e., $l_{\langle \mathsf{c_j} \rangle} \in Y_{\mathbf{E}}$ any path through that node producing a model of T must satisfy l either at n or later at an $\{\langle \mathsf{c_j} \rangle\}$ reachable node. Hence we can apply the final deletion condition. The "if" part follows.

We now show the 'only if' part. Assume that the reduced labelled behaviour graph is non-empty. We know the set of initial nodes is non-empty because the reduced labelled behaviour graph is defined to be the set of nodes reachable from the initial nodes. We will construct a model of T. For any node n added to the construction, we will unwind through the temporal edges to construct trees that satisfy the set of step clauses, making sure all the eventualities triggered from satisfying the left hand side of sometime rules are satisfied. Secondly we add modal successors to any nodes constructed.

For the modal dimension we must show that for any $Bl \in n$ in the reduced labelled behaviour graph all nodes $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ such that $(n, n') \in E_B$ satisfy $V' \models l$ and for each $\neg Bl \in n$ there exists some $n' = (V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ such that $(n, n') \in E_B$ and $V' \models \neg l$ where E_B is transitive, Euclidean and serial. We ensure that for any modal literal $Bl \in n$, l is satisfied in all nodes $(n, n') \in E_B$ by construction of the belief edges E_B. Further recall that the set of belief clauses in T have an external B and every node must lead to one that satisfies these clauses. This is achieved by deleting nodes that do not immediately satisfy these clauses during the construction of the labelled behaviour graph. Finally we must check that from each node containing $\neg Bl$ there is an edge in E_B to a node containing $\neg l$. This is shown in Lemma 32. Next we check the E_B relation is transitive, Euclidean and serial. This is shown in Lemma 33.

Next we show how to construct a model from the non-empty reduced labelled behaviour graph. First we construct timelines by unwinding through the temporal dimension making sure each $\mathbf{E}$ step clause is satisfied and each eventuality is satisfied. Let node be a function from points to N which relates every point added to the construction to the node in the reduced labelled behaviour graph it was constructed from. Let lits be a function defined as $\text{lits}(n) = \{l \mid l \text{ is a literal and } l \in n\}$ where $n \in N$. We unwind through the temporal dimension starting at an initial node $n_0 = (V_0, Y_{\mathbf{A}}, Y_{\mathbf{E}})$. Let $(t, 0) = \text{lits}(n_0)$ and $\text{node}(t, 0) = n_0$. For each labelled literal $l_{\langle \mathsf{c_i} \rangle} \in Y_{\mathbf{E}}$ construct a new timeline t and unwind through an $\{\langle c_i \rangle\}$ reachable path $n_0, n_1, \ldots$, setting $(t, i) = \text{lits}(n_i)$ and $\text{node}(t, i) = n_i$, until we reach a point (t, j) such that $\text{node}(t, j) = n_j$ where l is satisfied. This must be possible because if each $l_{\langle \mathsf{c_i} \rangle}$ was not able to be satisfied in a reachable node then the node must have been deleted by the fifth or sixth deletion criterion. From (t, j) construct a path that satisfies each $\mathbf{A}$ 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 third and fourth deletion criterion. For each point $(t, k) = \text{lits}(n_k)$ added to timeline t from unwinding through a node n_k in the reduced labelled behaviour graph we add $\text{node}(t, k) = n_k$ to the function node. Repeat the process for each $l_{\langle \mathsf{c_i} \rangle} \in Y_{\mathbf{E}}$ in node n_0. Finally let I be

the set of labels of edges leading from n_0. For each label $\langle \mathsf{c_i} \rangle \in I$ such that there is no labelled literal $k_{\langle \mathsf{c_i} \rangle} \in Y_{\mathbf{E}}$ construct a new timeline t' such that node $(t', 0) = n_0$ and unwind through an $\langle \mathsf{c_i} \rangle$ labelled edge to node n'_1 where $(n, I', n'_1) \in E_T$ where $\langle c_i \rangle \in I'$, setting node $(t', 1) = n'_1$ and $(t', 1) = \text{lits}(n'_1)$. There must be such a node due to deletion criterion two. Note timelines t and t' coincide up to point $(t, 0)$. Then from n'_1 select a path that satisfies each **A** eventuality in the initial node in turn ignoring any eventualities that have been satisfied by nodes n and n'_1. 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 third and fourth deletion criterion. Repeat with each point, (t, u) where node $(t, u) = n_u$ added to the construction, i.e., extend n_u to some n_v where $(n_u, I, n_v) \in E_T$ for every label leading from n_u and make sure all outstanding eventualities are satisfied (without adding paths to satisfy eventualities already satisfied), until we have a finite tree where all eventualities in the initial node have been satisfied in every leaf.

Once all the eventualities from the initial node have been satisfied at each leaf node, let n_1 be some leaf node. The tree through n_1 is extended constructing a branch to satisfy each **E** eventuality and satisfying every **A** eventuality in n_1 as above. Again take a leaf 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 **A** eventuality is satisfied by some node later in the path and each **E** eventuality is satisfied on some path. Recall from the construction of the graph if an **A** 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_B it either must be satisfied in n_B or must appear in the set of eventualities in n_B and be satisfied in the next portion of path. For each point (t, u) such that node $(t, u) = n_u$ constructed in the tree add modal edges to any point (t, u') already constructed such that node $(t, u') = n_{u'}$ and $(n_u, n_{u'}) \in E_B$. If there is no such point construct one and construct timelines from this point and modal edges as above.

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

In the graph we have pushed the B operator into clauses to construct the graph, whereas in the resolution system we work on unpushed clauses. The next lemma shows that any resolution step we can do with the pushed clauses we can do with the unpushed clauses to obtain the same or a stronger resolvent. To prove the next theorem we require an additional resolution rule MRES4c. Later we show that this additional rule is unnecessary for completeness. Hence its omission from the presentation of resolution rules in Section 5.

$$\text{[MRES4c]}\quad \frac{\begin{array}{l}\mathbf{true} \Rightarrow D \vee B\neg l_1 \\ \mathbf{true} \Rightarrow D' \vee l_1 \vee l_2\end{array}}{\mathbf{true} \Rightarrow D \vee \text{mod}(D') \vee Bl_2}$$

Lemma 35. *Let R be a literal or belief clause and let X_R be the set of belief clauses having distributed the B operator over R. For any belief or literal clause R_1 that can be resolved with a member of X_R to give R_2 we can resolve R with R_1 to give R_3 where $R_3 \Rightarrow R_2$ using* MRES1, MRES2, MRES4a, MRES4b *or* MRES4c.

Proof. First consider the literal clause R whose right hand side contains a single literal, i.e., $\mathbf{true} \Rightarrow l$. Pushing B into this clause we obtain $\mathbf{true} \Rightarrow Bl$. This can be resolved with the clauses $\mathbf{true} \Rightarrow \neg l \vee D$, $\mathbf{true} \Rightarrow B\neg l \vee D$ or $\mathbf{true} \Rightarrow \neg Bl \vee D$ obtaining the resolvent $\mathbf{true} \Rightarrow \mathrm{mod}(D)$ by applying MRES4b and $\mathbf{true} \Rightarrow D$ by applying MRES2 or MRES1 respectively. The clause $\mathbf{true} \Rightarrow l$ can also be resolved with the same three clauses and produce $\mathbf{true} \Rightarrow D$ by applying MRES1, MRES4b and MRES4a respectively. As $(\mathbf{true} \Rightarrow D) \Rightarrow (\mathbf{true} \Rightarrow \mathrm{mod}(D))$ we are done.

Next we examine belief clauses containing a single literal, for example $\mathbf{true} \Rightarrow l \vee D'$ where D' is a disjunction of modal literals. Pushing B into this clause and obtaining $\mathbf{true} \Rightarrow Bl \vee D'$ we may resolve it with $\mathbf{true} \Rightarrow \neg l \vee D$, $\mathbf{true} \Rightarrow B\neg l \vee D$ or $\mathbf{true} \Rightarrow \neg Bl \vee D$ where D is a disjunction of literals or modal literals by applying rules MRES4b, MRES2 or MRES1 and obtaining the resolvent $\mathbf{true} \Rightarrow \mathrm{mod}(D) \vee D'$ by resolving with MRES4b and $\mathbf{true} \Rightarrow D \vee D'$ by resolving with MRES2 or MRES1. The original clause $\mathbf{true} \Rightarrow l \vee D'$ may also be resolved with each of the clauses using the resolution rules MRES1, MRES4b, MRES4a respectively to produce $\mathbf{true} \Rightarrow D \vee D'$ (recall from the definition of mod that as D' is a disjunction of modal literals $D' = \mathrm{mod}(D')$). Again this implies the previous resolvents.

Finally we consider the case where there is more than one literal in a belief clause. Without loss of generality we consider a clause 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 B operator we obtain two clauses $\mathbf{true} \Rightarrow Bl_1 \vee \neg B\neg l_2 \vee D'$ and $\mathbf{true} \Rightarrow \neg B\neg l_1 \vee Bl_2 \vee D'$. Now at least one of these clauses can be resolved with the clauses $\mathbf{true} \Rightarrow \neg l_i \vee D$, $\mathbf{true} \Rightarrow B\neg l_i \vee D$ or $\mathbf{true} \Rightarrow \neg Bl_i \vee D$ for $i = 1, 2$ where D is a disjunction of literals or modal literals by applying rules MRES4b, MRES4a, MRES2 or MRES1. We consider the resolvents for resolving with $\mathbf{true} \Rightarrow Bl_1 \vee \neg B\neg l_2 \vee D'$; the other case is similar. This clause may be resolved with $\mathbf{true} \Rightarrow \neg l_1 \vee D$, $\mathbf{true} \Rightarrow B\neg l_1 \vee D$, $\mathbf{true} \Rightarrow \neg Bl_1 \vee D$ or $\mathbf{true} \Rightarrow B\neg l_2 \vee D$. The resolvents obtained are $\mathbf{true} \Rightarrow \mathrm{mod}(D) \vee \neg B\neg l_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee \neg B\neg l_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee \neg B\neg l_2 \vee D'$ and $\mathbf{true} \Rightarrow Bl_1 \vee D \vee D'$ respectively. These resolvents, or resolvents that imply these resolvents can be generated from resolving the original clause with each of these clauses to obtain $\mathbf{true} \Rightarrow D \vee l_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee Bl_2 \vee D'$, $\mathbf{true} \Rightarrow D \vee \neg B\neg l_2 \vee D'$ and $\mathbf{true} \Rightarrow Bl_1 \vee D \vee D'$ using the resolution rules MRES1, MRES4c, MRES4a and MRES4c respectively. □

Lemma 36. *If a set of belief clauses is unsatisfiable then there is a refutation using resolution rules* IRES1, IRES2, MRES1, MRES2, MRES4a, MRES4b *or* MRES4c.

Proof. Assume that T' is the set of clauses T plus the result of pushing the external B-operator into any belief clause. Then the set of clauses T' is unsatisfiable using classical resolution between literals or modal literals and their negations, modal resolution between modal literals Bl and $B\neg l$. That is, by applying the resolution rules IRES1, IRES2, MRES1

(that resolves a formula and its negation) or MRES2 we can detect a contradiction. By conjoining the set of belief clauses and rewriting the right hand side in DNF we obtain

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

where $\bigvee_i D_i$ must be unsatisfiable, i.e., $\bigvee_i D_i \Rightarrow \mathbf{false}$. If D_i contains a literal or modal literal and its negation or $B\neg l$ and Bl by applying resolution using rules MRES1 or MRES2 we can add new clauses and exclude this disjunct. Otherwise each D_i must cause a contradiction with the right hand sides of the initial clauses and must be excluded. Thus we can use the resolution rules IRES1, IRES2, MRES1 and MRES2 to derive a contradiction.

However in the resolution system the external B-operator is not pushed into the clauses so we must make sure that any resolvents generated after B has been pushed into each belief clause can be produced by applying MRES1, MRES2, MRES4a, MRES4b or MRES4c. This is shown in Lemma 35. □

Lemma 37. *If the unreduced labelled behaviour graph for a set of* SNF_B *clauses* T *is empty then a contradiction can be obtained by applying resolution rules* IRES1, IRES2, MRES1, MRES2, MRES4a, MRES4b *or* MRES4c *to clauses in or derived from* T.

Proof. If the labelled behaviour graph is empty then by Proposition 34 the set of clauses T is unsatisfiable. Thus by Lemma 36 there is a refuatation using rules IRES1, IRES2, MRES1, MRES2, MRES4a, MRES4b or MRES4c. □

Lemma 38. *Let* M *be a satisfiable set of belief clauses such that* $M_j \in M$. *We can derive* $\mathbf{true} \Rightarrow \bigvee_i l_i$ *by applying modal resolution, if* $(\bigwedge_j M_j) \Rightarrow (\mathbf{true} \Rightarrow \bigvee_i l_i)$ *such that there is no disjunction of literals* Y *where* $(\bigwedge_j M_j) \Rightarrow (\mathbf{true} \Rightarrow Y)$ *and* $Y \Rightarrow \bigvee_i l_i$.

Proof. If $(\bigwedge_j M_j) \Rightarrow (\mathbf{true} \Rightarrow \bigvee_i l_i)$ then any model satisfying M must also satisfy $\mathbf{true} \Rightarrow \bigvee_i l_i$, i.e., every state satisfies $\bigvee_i l_i$. In particular the initial nodes must satisfy $\bigvee_i l_i$ hence $M \cup \{\mathbf{start} \Rightarrow \neg \bigvee_i l_i\}$ is unsatisfiable. By completeness of modal resolution, Lemma 36, from $M \cup \{\mathbf{start} \Rightarrow \neg \bigvee_i l_i\}$ we can derive $\mathbf{start} \Rightarrow \mathbf{false}$. Apply modal resolution rules to clauses in M to obtain M' until no new formulae are obtained. By the soundness of modal resolution M' is satisfiable. Now from above the addition of clauses $\mathbf{start} \Rightarrow \neg \bigvee_i l_i$ we can obtain a contradiction. Let the last resolution step to obtain $\mathbf{start} \Rightarrow \mathbf{false}$ be between $\mathbf{start} \Rightarrow \neg l_0$ and $\mathbf{start} \Rightarrow l_0$. Similarly let the previous step be between $\mathbf{start} \Rightarrow \neg l_1$ and $\mathbf{start} \Rightarrow l_0 \vee l_1$ etc. until all the $\mathbf{start} \Rightarrow \neg l_i$ clauses have been resolved. Note we may only resolve initial clauses with initial clauses or literal clauses, so when we resolve with the first of the added initial clauses, i.e., $\mathbf{start} \Rightarrow \neg l_n$ we must resolve it with $\mathbf{true} \Rightarrow \bigvee_i l_i$. We need to resolve with each $\mathbf{start} \Rightarrow \neg l_i$ to obtain a contradiction otherwise if we can obtain a contradiction from the clauses $\mathbf{start} \Rightarrow \neg l_i$ for $0 \leqslant i \leqslant n-1$ without using $\mathbf{start} \Rightarrow \neg l_n$ then $(\bigwedge_j M_i) \Rightarrow (\mathbf{true} \Rightarrow \bigvee_{i=0}^{n-1} l_i)$ and $\bigvee_{i=0}^{n-1} l_i \Rightarrow \bigvee_{i=0}^{n} l_i$ which is not allowed. □

Theorem 39 (Completeness). *If T a set of clauses in* SNF_B *is unsatisfiable then it has a refutation by the procedure described above using resolution rules* IRES1, IRES2, MRES1, MRES2, MRES4a, MRES4b, MRES4c, SRES1–5, *and* TRES1–4.

Let T be an unsatisfiable set of SNF_B clauses. The proof proceeds by induction on the number of nodes in the labelled behaviour graph of T. If the (unreduced) labelled behaviour graph is empty then by by Lemma 37 we can obtain a contradiction by applying resolution rules IRES1, IRES2, MRES1, MRES2, MRES4a, MRES4b or MRES4c.

Now suppose the labelled behaviour graph H is non-empty. By Proposition 34 the reduced labelled behaviour graph must be empty so there must be a node than can be deleted from H as described above.

If a terminal node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ exists, consider W, the union of the set of **A** step clauses whose left hand side satisfy the valuation V. Let J be the set of literal and belief clauses. Then, the right hand side of the clauses in W (having deleted the **A** and next operators from the **A** step clauses), L, union with any right hand sides of literal clauses, L', such that $\bigvee_j l_j \in L'$ iff $J \Rightarrow (\mathbf{true} \Rightarrow \bigvee_j l_j)$ and there is no Y where $Y \Rightarrow \bigvee_j l_j$ and $J \Rightarrow (\mathbf{true} \Rightarrow Y)$ form an unsatisfiable set of propositional clauses. Either $\mathbf{true} \Rightarrow \bigvee_j l_j$ is present in J or apply modal resolution to clauses in J until $\mathbf{true} \Rightarrow \bigvee_j l_j$ is derived. The latter is possible due to Lemma 38. In both cases by completeness of classical resolution again, there is a refutation of $L \cup L'$. Choosing an element of W or L' corresponding to each element of $L \cup L'$, we can "mimic" this classical refutation by step resolution (SRES1 and SRES4) inferences to derive an **A** step clause

$$l_1 \wedge \cdots \wedge l_k \Rightarrow \mathbf{A}\bigcirc\mathbf{false} \tag{17}$$

where each l_i is a literal which is satisfied by V. Using SRES5 the temporal resolution procedure allows us to rewrite clause (17) as

$$\mathbf{true} \Rightarrow (\neg l_1 \vee \cdots \vee \neg l_k). \tag{18}$$

By Lemma 29, adding clause (18) (and any other resolvents derived along the way) to T produces a clause set T' as every node must satisfy

$$\bigvee_{i=1}^{i=k} \neg l_i,$$

$(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ is deleted 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.

If a node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ exists where for some **E** step clause, $X \Rightarrow \mathbf{E}\bigcirc Y_{\langle \mathsf{c_j} \rangle}$, where $V \models X$ but no edge (n, I, n') leading from n exists such that $\langle \mathsf{c_j} \rangle \in I$ then this node cannot be in any model and should be deleted. For each node $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$, and label $\langle \mathsf{c_j} \rangle$, let $R^{\mathbf{E}^{\{\langle \mathsf{c_j} \rangle\}}}$ be the set of **E** step clauses of T, labelled by label $\langle \mathsf{c_j} \rangle$, which are "fired" by V—that is, the set of **E** step clauses labelled by $\langle \mathsf{c_j} \rangle$ whose left hand sides are satisfied by V. If $R^{\mathbf{E}^{\{\langle \mathsf{c_j} \rangle\}}}$ is none-empty for some $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ and there are no edges out of n labelled by $\langle \mathsf{c_j} \rangle$, then this node cannot be in any model and should be deleted. This is because the clauses from the right hand side of the clauses in $R^{\mathbf{E}^{\{\langle \mathsf{c_j} \rangle\}}}$ should be satisfied but cannot be. Thus by

the completeness of classical resolution, assuming we have carried out all modal resolution there must be a series of step resolution inferences (between elements of $R^{\mathbf{E}^{\{\langle c_j\rangle\}}}$, **A** step clauses or literal clauses) that lead to

$$l_1 \wedge \cdots \wedge l_k \Rightarrow \mathbf{E}\bigcirc\mathbf{false}_{\langle c_j\rangle}$$

where $V \models l_1 \wedge \cdots \wedge l_k$. This is rewritten as

$$\mathbf{true} \Rightarrow (\neg l_1 \vee \cdots \vee \neg l_k). \tag{19}$$

By Lemmas 29 and 30, by adding (19) and any other resolvents derived on the way to T to produce a clause set T', the labelled behaviour graph is redrawn and as $(V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ does not satisfy

$$\bigvee_{i=1}^{i=k} \neg l_i$$

it is deleted 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 a terminal subgraph of nodes reachable from n following temporal edges. We have four cases to correspond with the four resolution rules.

- **A$\Diamond l$ and A loop** If N' is the set of nodes reachable from n via E_T edges then any edges in E_T out of a node in N' lead to a node that is also in N' where for each $n' \in N'$ we have $n' \models \neg l$. For each node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}})$ in N' the set of step clauses or literal clauses whose left hand side is satisfied by V are combined to give $F_n \Rightarrow \mathbf{A}\bigcirc G_n$ 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 $\models G_n \Rightarrow \neg l$. Let V be a set of literals and modal literals from a node in H that satisfies G_n. By the construction of the labelled behaviour graph there is a temporal successor of n in H of the form $(V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$. By assumption this node is also in N' and therefore $V' \not\models l$ so for all $n \in N'$, $G_n \wedge l$ is unsatisfiable and therefore $\models G_n \Rightarrow \neg l$.
  - For each $n \in N'$ we must have $\models G_n \Rightarrow \bigvee_{n' \in N'} F_{n'}$. Let V be a set of literals and modal literals from a node in H that satisfies G_n. By the construction of H since $V \models G_n$ there is an edge from n to a node $n' \in N'$ whose valuation is V'. Since $n' \in N'$, by assumption we have $V' \models F_{n'}$. Hence $\models G_n \Rightarrow \bigvee_{n' \in N'} F_{n'}$ as required.

  We can use the set of clauses $F_n \Rightarrow \mathbf{A}\bigcirc G_n$ for resolution with each eventuality l occurring in T. Consider any node $n = (V, Y_{\mathbf{A}}, Y_{\mathbf{E}}) \in N'$ where $l \in Y_{\mathbf{A}}$, i.e., l is an unsatisfied **A**-eventuality. Let L be defined as

  $$L = \bigvee_{n \in N'} F_n.$$

  Note that $V \models L$, $V \not\models l$ and $V \models w_l$ (by Lemma 27). Through augmentation we have added the clauses

  $$w_l \Rightarrow \mathbf{A}\bigcirc(l \vee w_l)$$
  $$\mathbf{true} \Rightarrow (\neg Q \vee l \vee w_l)$$

for some $Q \Rightarrow \mathbf{A}\Diamond l \in T$. Either there is an edge $e \in E_T$ from some node $(V', Y'_{\mathbf{A}}, Y'_{\mathbf{E}})$ into n such that $l \in Y'_{\mathbf{A}}$ and $V' \not\models l$ or T contains a clause $Q \Rightarrow \mathbf{A}\Diamond l$ where $V \models Q$ and $V \not\models l$. For the former we must have $V' \models w_l$ by Lemma 27. Applying the temporal resolution rule adds the resolvent $w_l \Rightarrow \mathbf{A}\bigcirc(l \vee \neg L)$ to the set of step clauses in T. Now V' satisfies the left hand side of this clause, 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 labelled behaviour graph does not contain e. Otherwise for the latter we have $Q \Rightarrow \mathbf{A}\Diamond l$, $V \models Q$ and $V \not\models l$. Then resolving the eventuality clause with the loop we obtain $\mathbf{true} \Rightarrow \neg Q \vee l \vee \neg L$ as one of the resolvents. Now n doesn't satisfy this resolvent and so n must be deleted from H. In either case n either becomes unreachable or is deleted. So, the labelled 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.

- **$\mathbf{A}\Diamond l$ and E loop** Similar to above but we consider terminal subgraphs that cannot be exited by some set of labelled edges.
- **$\mathbf{E}\Diamond l$ and A loop** Similar to above but we consider nodes containing the **E** eventuality l and there is no path to a node which satisfies l.
- **$\mathbf{E}\Diamond l$ and E loop** Similar to above but we consider nodes containing the **E** eventuality l labelled by $\langle \mathsf{c_j} \rangle$ and there is no $\{\langle \mathsf{c_j} \rangle\}$ reachable path to a node which satisfies l.

Finally we show that the extra modal resolution rule we introduced is not necessary for completeness.

Lemma 40. *Let P_1 and P_2 be two belief clauses which can be resolved using* MRES4c *to give R_1. Then for any belief clause P_3 that can be resolved with R_1 to obtain R_2, we can resolve P_3 with P_2 and then P_1 to obtain R_3 where $R_3 \Rightarrow R_1$ using only* MRES1, MRES2, MRES4a, *or* MRES4b.

Proof. Let P_1, P_2 and R_1 be as follows and apply MRES4c.

$$\begin{array}{l} P_1 \ \mathbf{true} \Rightarrow D \vee B\neg l_1 \\ P_2 \ \mathbf{true} \Rightarrow D' \vee l_1 \vee l_2 \\ \hline R_1 \ \mathbf{true} \Rightarrow D \vee \mathrm{mod}(D') \vee Bl_2 \end{array}$$

We can resolve the modal literal Bl_2 in resolvent R_1 with the following:

$$\begin{array}{l} P_3 \ \mathbf{true} \Rightarrow D'' \vee \neg l_2 \\ P'_3 \ \mathbf{true} \Rightarrow D'' \vee \neg Bl_2 \\ P''_3 \ \mathbf{true} \Rightarrow D'' \vee B\neg l_2 \end{array}$$

to obtain

$$\begin{array}{lll} R_2 \ \mathbf{true} \Rightarrow D \vee \mathrm{mod}(D') \vee \mathrm{mod}(D'') & [R_1, P_3, \text{MRES4c}] \\ R'_2 \ \mathbf{true} \Rightarrow D \vee \mathrm{mod}(D') \vee D'' & [R_1, P'_3, \text{MRES1}] \\ R''_2 \ \mathbf{true} \Rightarrow D \vee \mathrm{mod}(D') \vee D'' & [R_1, P''_3, \text{MRES2}] \end{array}$$

Next we resolve each of P_3, P'_3 or P''_3 with one of the original parents P_2

$$\begin{array}{lll} R_3 \ \mathbf{true} \Rightarrow D' \vee l_1 \vee D'' & [P_2, P_3, \text{MRES1}] \\ R'_3 \ \mathbf{true} \Rightarrow \mathrm{mod}(D') \vee \neg B\neg l_1 \vee D'' & [P_2, P'_3, \text{MRES4a}] \\ R''_3 \ \mathbf{true} \Rightarrow \mathrm{mod}(D') \vee \neg B\neg l_1 \vee D'' & [P_2, P''_3, \text{MRES4b}] \end{array}$$

and then each of these with P_1

$$\begin{array}{ll} R_4\ \mathbf{true} \Rightarrow\ D \vee \mathrm{mod}(D') \vee \mathrm{mod}(D'') & [P_1, R_3, \mathrm{MRES4b}] \\ R'_4\ \mathbf{true} \Rightarrow\ D \vee \mathrm{mod}(D') \vee D'' & [P_1, R'_3, \mathrm{MRES1}] \\ R''_4\ \mathbf{true} \Rightarrow\ D \vee \mathrm{mod}(D') \vee D'' & [P_1, R''_3, \mathrm{MRES1}] \end{array}$$

In each case we have $R_4 \Rightarrow R_2$, $R'_4 \Rightarrow R'_2$, $R''_4 \Rightarrow R''_2$. □

Corollary 41. *If a formula, φ, in BB_n is unsatisfiable it has a refutation using the procedure given in Section* 5.

Proof. φ is unsatisfiable, therefore, by Theorem 15, $\tau_0(\varphi)$ its translation into SNF_B is unsatisfiable. By Theorem 39 we can derive a contradiction using the resolution rules in Section 5 plus MRES4c. In the proof any step that involves MRES4c can be replaced by steps using only MRES1, MRES2, MRES4a, or MRES4b. Hence we can derive a contradiction using only the resolution rules given in Section 5. □

8. Extension to the multi-modal case

Here we show how to extend the set of resolution rules given in Section 5.2 to the multi-modal case. First we note that the translation to the normal form given in Section 4 disallows clauses containing more than one different modal operator, i.e., $\mathbf{true} \Rightarrow B_1 l \vee \neg B_2 h$ would be rewritten as $\mathbf{true} \Rightarrow B_1 l \vee y$ and $\mathbf{true} \Rightarrow \neg y \vee \neg B_2 h$ where y is a new proposition. This adopts our general strategy of keeping each modal component separate. MRES1 has the additional restriction that both clauses must apply to the same modal operator (i.e., they must be both *i-belief clauses*) or at least one of them must be a literal clause and MRES2 is extended as follows.

$$[\mathrm{MRES1}]\ \frac{\mathbf{true} \Rightarrow D \vee m \qquad \mathbf{true} \Rightarrow D' \vee \neg m}{\mathbf{true} \Rightarrow D \vee D'} \qquad [\mathrm{MRES2}]\ \frac{\mathbf{true} \Rightarrow D \vee B_i l \qquad \mathbf{true} \Rightarrow D' \vee B_i \neg l}{\mathbf{true} \Rightarrow D \vee D'}$$

Similarly with MRES4a and MRES4b either both clauses must apply to the same modal operator B_i or the second must be a literal clause.

$$[\mathrm{MRES4a}]\ \frac{\mathbf{true} \Rightarrow D \vee \neg B_i l \qquad \mathbf{true} \Rightarrow D' \vee l}{\mathbf{true} \Rightarrow D \vee \mathrm{mod}_i(D')} \qquad [\mathrm{MRES4b}]\ \frac{\mathbf{true} \Rightarrow D \vee B_i l \qquad \mathbf{true} \Rightarrow D' \vee \neg l}{\mathbf{true} \Rightarrow D \vee \mathrm{mod}_i(D')}$$

where $\mathrm{mod}_i(D')$ is defined as follows

$$\begin{array}{ll} \mathrm{mod}_i(F \vee G) = \mathrm{mod}_i(F) \vee \mathrm{mod}_i(G) & \mathrm{mod}_i(\neg B_i l) = \neg B_i l \\ \mathrm{mod}_i(B_i l) = B_i l & \mathrm{mod}_i(l) = \neg B_i \neg l. \end{array}$$

We believe the proofs given in Section 7 can be extended to the multi-modal case.

9. Related work

The work we have presented is a clausal resolution method for a branching-time temporal logic of belief, i.e., the fusion of CTL and KD45.

Resolution methods have been described for a variety of modal logics in, for example, [2,3,20,35,37–39]. These fall into two main groups, those that work in the modal logic directly [2,35] or those that use a translation into first-order logic for example [37–39]. Our system follows the former route and is based on that for propositional modal logic given by Mints [35]. 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 [40]. Proof methods for the branching-time temporal logic CTL are described in, for example, [8,19]. Resolution based methods for propositional linear time temporal logics are given in [1,10,25,45]. The temporal component of the resolution mentioned here is based on that in [25].

However, few proof methods have been developed for the combination of temporal and modal logics. A resolution based proof method for temporal logics of knowledge, i.e., the fusion of linear-time temporal logic with single and multi-modal S5 are given in [15, 17]. This uses a similar approach to the above, translation to a normal form to separate modal and temporal components, a resolution method applied to the temporal part and modal resolution rules applied to the modal part. Information is carried between the two components using literal clauses. A resolution based approach to proof in the fusion of linear time temporal logic with the fusion of any multi-modal logic that is an extension of K_m by the axioms 4, 5, B, D, T is given in [31]. In this paper, the authors translate to a normal form which separates the modal and temporal components. The temporal resolution method in [25] is adopted for the temporal part, whilst the proof method for the modal part is based on the translation based approach [37], i.e., modal formulae are translated into a fragment of first-order classical logic and then resolution inference rules for classical logics applied.

Other proof methods for combined logics have been based on tableau methods. This involves the construction of a stucture, for the negation of the formula to be shown valid, from which a model can be extracted. The inability to construct such a stucture means that the negated formula is unsatisfiable and the original, therefore, valid. A tableau based proof method for the fusion of PTL plus either S5 or KD45 is given in [46] and a tableau based method for a temporal belief logic is given in [47]. The work on proof methods for BDI-logics given in [41,43] give a tableau based proof method for the fusion of either linear or branching-time (CTL or CTL*) with the modal logics KD45 for belief, and KD for desire and intention. Tableaux methods for description logics (essentially combinations of modal logics) have been described and implemented in [30]. A tableau calculus is presented in [34] for the basic description logic $\mathcal{ALC}$ combined with propositional linear-time temporal logic interpreted in models with constant domains.

Halpern, Vardi et al. consider the combination of propositional linear and branching time temporal logics with multi-modal S5 allowing a variety of interactions in [27,29]. Rather than theorem proving this series of papers concentrates on the complexity of the satisfiability problem and obtaining complete axiomatisations for various systems.

Work has also been carried out into combining arbitrary logics, see for example the work on *fibring* in [26], and issues relating to combining logics are considered in [4,5].

10. Concluding remarks

The logical representation of rational agents is currently a very active area of research. However, few of the people involved in this research have considered proof methods for these logics. The main reason for this is the complexity associated with combining multi-modal and temporal logics. In our work with KARO, we have identified a simpler logic which, while still comprising a combination of temporal and modal logics, is amenable to mechanisation. Thus, in this paper we have presented a sound, complete and terminating clausal resolution method for this particular logic.

Whilst this particular combination has been chosen due to our concentration on the development of proof methods to be applied to KARO it is part of a wider work, i.e., the development of resolution based proof methods for a range of combined modal and temporal logics. The method requires a translation into an SNF style normal form that separates the different logical components, appropriate resolution methods to be applied in each component with enough information being passed between components. Further, we are developing methods to handle interaction between the components, for example in logics that have axioms that involve both modal and temporal operators [14].

Whilst we have used the method to prove properties of small KARO specifications [32] in the future we will apply this to larger logical specifications derived from the KARO agent theory. In addition, we intend to investigate whether this simpler form of logic can be used as the basis for other agent theories. Finally we would like to extend the amount of KARO language we can cover by adding features such as interactions.

Acknowledgements

This work was partially supported by EPSRC research grants GR/L87491 and GR/M44859. This paper is an extended version of the work presented in [16].

References

[1] M. Abadi, Temporal-logic theorem proving, Ph.D. Thesis, Department of Computer Science, Stanford University, CA, 1987.
[2] M. Abadi, Z. Manna, Modal theorem proving, in: Lecture Notes in Computer Science, Vol. 230, Springer, Berlin, 1986, pp. 172–189.
[3] Y. Auffray, P. Enjalbert, Modal theorem proving: An equational viewpoint, in: Proc. IJCAI-89, Detroit, MI, Morgan Kaufmann, San Mateo, CA, 1989, pp. 441–445.
[4] B. Bennett, C. Dixon, M. Fisher, E. Franconi, I. Horrocks, U. Hustadt, M. de Rijke, Combinations of modal logics, Artificial Intelligence Rev., to appear.
[5] P. Blackburn, M. de Rijke, Why combine logics?, Studia Logica 59 (1997) 5–27.
[6] A. Bolotov, Clausal resolution for branching-time temporal logic, Ph.D. Thesis, Department of Computing and Mathematics, Manchester Metropolitan University, 2000.

[7] A. Bolotov, C. Dixon, Resolution for branching time temporal logics: Applying the temporal resolution rule, in: S. Goodwin, A. Trudel (Eds.), Proceedings of TIME-00—the 7th International Workshop on Temporal Representation and Reasoning, Cape Breton, NS, IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 163–172.
[8] A. Bolotov, M. Fisher, A resolution method for CTL branching-time temporal logic, in: Proceedings of TIME-97—the 4th International Workshop on Temporal Representation and Reasoning, Daytona Beach, FL, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 20–27.
[9] A. Bolotov, M. Fisher, A clausal resolution method for CTL branching time temporal logic, J. Experiment. Theoret. Artificial Intelligence 11 (1999) 77–93.
[10] A. Cavalli, L. Fariñas del Cerro, A decision method for linear temporal logic, in: R.E. Shostak (Ed.), Proceedings of the 7th International Conference on Automated Deduction, Lecture Notes in Computer Science, Vol. 170, Springer, Berlin, 1984, pp. 113–127.
[11] E.M. Clarke, E.A. Emerson, Design and synthesis of synchronisation skeletons using branching time temporal logic, in: D. Kozen (Ed.), Proceedings of the Workshop on the Logic of Programs, Lecture Notes in Computer Science, Vol. 131, Springer, Berlin, 1981, pp. 52–71.
[12] C. Dixon, Search strategies for resolution in temporal logics, in: M.A. McRobbie, J.K. Slaney (Eds.), Proceedings of the 13th International Conference on Automated Deduction (CADE), New Brunswick, NJ, Lecture Notes in Artificial Intelligence, Vol. 1104, Springer, Berlin, 1996, pp. 672–687.
[13] C. Dixon, Temporal resolution using a breadth-first search algorithm, Ann. Math. Artificial Intelligence 22 (1998) 87–115.
[14] C. Dixon, M. Fisher, Clausal resolution for logics of time and knowledge with synchrony and perfect recall, in: H. Wansing, F. Wolter (Eds.), Proceedings of ICTL-00—the 3rd International Conference on Temporal Logic, Leipzig, Germany, 2000.
[15] C. Dixon, M. Fisher, Resolution-based proof for multi-modal temporal logics of knowledge, in: S. Goodwin, A. Trudel (Eds.), Proceedings of TIME-00—the 7th International Workshop on Temporal Representation and Reasoning, Cape Breton, NS, IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 69–78.
[16] C. Dixon, M. Fisher, A. Bolotov, Resolution in a logic of rational agency, in: Proceedings 14th European Conference on Artificial Intelligence (ECAI 2000), Berlin, Germany, IOS Press, Amsterdam, 2000, pp. 358–362.
[17] C. Dixon, M. Fisher, M. Wooldridge, Resolution for temporal logics of knowledge, J. Logic Comput. 8 (3) (1998) 345–372.
[18] E.A. Emerson, Temporal and modal logic, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Elsevier Science, Amsterdam, 1990, pp. 996–1072.
[19] E.A. Emerson, J.Y. Halpern, Decision procedures and expressiveness in the temporal logic of branching time, J. Comput. System Sci. 30 (1) (1985) 1–24.
[20] P. Enjalbert, L. Fariñas del Cerro, Modal resolution in clausal form, Theoret. Comput. Sci. 65 (1989) 1–33.
[21] R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, Reasoning About Knowledge, MIT Press, Cambridge, MA, 1995.
[22] M. Fisher, Implementing BDI-like systems by direct execution, in: Proceedings IJCAI-97, Nagoya, Japan, Morgan Kaufmann, San Mateo, CA, 1997, pp. 316–321.
[23] M. Fisher, M. Wooldridge, On the formal specification and verification of multi-agent systems, Internat. J. Cooperat. Inform. Syst. 6 (1) (1997) 37–66.
[24] M. Fisher, M. Wooldridge, Towards formal methods for agent-based systems, in: D. Duke, A. Evans (Eds.), Proceedings of the Northern Formal Methods Workshop (NFMW), Electronic Workshops in Computing, Springer, Berlin, 1997.
[25] M. Fisher, C. Dixon, M. Peim, Clausal temporal resolution, ACM Trans. Comput. Logic 2 (1) (2001) 12–56.
[26] D.M. Gabbay, Fibred-semantics and the weaving of logics, Part 1: Modal and intuitionistic logics, J. Symbolic Logic 61 (4) (1996) 1057–1120.
[27] J.Y. Halpern, R. van der Meyden, M. Vardi, Complete axiomatizations for reasoning about knowledge and time, 1999, submitted for publication.
[28] J.Y. Halpern, M.Y. Vardi, The complexity of reasoning about knowledge and time: Extended abstract, in: Proceedings of the 18th Annual ACM Symposium on Theory of Computing, Berkeley, CA, 1986, pp. 304–315.

[29] J.Y. Halpern, M.Y. Vardi, The complexity of reasoning about knowledge and time, I: Lower bounds, J. Comput. System Sci. 38 (1989) 195–237.
[30] I. Horrocks, The FaCT system, in: Automated Reasoning with Analytic Tableaux and Related Methods: International Conference (Tableaux'98), Lecture Notes in Computer Science, Vol. 1397, Springer, Berlin, 1998, pp. 307–312.
[31] U. Hustadt, C. Dixon, R.A. Schmidt, M. Fisher, Normal forms and proofs in combined modal and temporal logics, in: Proceedings of the 3rd International Workshop on Frontiers of Combining Systems (FroCoS'2000), Lecture Notes in Artificial Intelligence, Vol. 1794, Springer, Berlin, 2000, pp. 73–87.
[32] U. Hustadt, C. Dixon, R.A. Schmidt, M. Fisher, J.J. Meyer, W. van der Hoek, Verification within the KARO agent theory, in: Proceedings of the First Goddard Workshop on Formal Approaches to Agent-Based Systems, Goddard Space, Flight Center, Greenbelt, MD, Lecture Notes in Computer Science, Springer, Berlin, 2000, pp. 33–47.
[33] U. Hustadt, C. Dixon, R.A. Schmidt, M. Fisher, J.-J. Meyer, W. van der Hoek, Reasoning about agents in the KARO framework, in: C. Bettini, A. Montanari (Eds.), Proceedings of the 8th International Symposium on Temporal Representation and Reasoning (TIME-01), Cividale del Friuli, Italy, IEEE Press, 2001, pp. 206–213.
[34] C. Lutz, H. Sturm, F. Wolter, M. Zakharyaschev, Tableaux for temporal description logic with constant domains, in: Automated Reasoning, Lecture Notes in Artificial Intelligence, Vol. 2083, Springer, Berlin, 2001, pp. 121–136.
[35] G. Mints, Gentzen-type systems and resolution rules, Part I: Propositional logic, in: Lecture Notes in Computer Science, Vol. 417, Springer, Berlin, 1990, pp. 198–231.
[36] R.C. Moore, A formal theory of knowledge and action, in: J.F. Allen, J. Hendler, A. Tate (Eds.), Readings in Planning, Morgan Kaufmann, San Mateo, CA, 1990, pp. 480–519.
[37] H. De Nivelle, R.A. Schmidt, U. Hustadt, Resolution-based methods for modal logics, Logic J. IGPL 8 (3) (2000) 265–292.
[38] A. Nonnengart, Resolution-based calculi for modal and temporal logics, in: M.A. McRobbie, J.K. Slaney (Eds.), Proceedings of the 13th International Conference on Automated Deduction (CADE), New Brunswick, NJ, Lecture Notes in Artificial Intelligence, Vol. 1104, Springer, Berlin, 1996, pp. 598–612.
[39] H.-J. Ohlbach, A resolution calculus for modal logics, in: Lecture Notes in Computer Science, Vol. 310, Springer, Berlin, 1988, pp. 500–516.
[40] D.A. Plaisted, S.A. Greenbaum, A structure-preserving clause form translation, J. Symbolic Comput. 2 (3) (1986) 293–304.
[41] 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, M. Tambe (Eds.), Proceedings IJCAI-95 Workshop on Agent Theories, Architectures, and Languages, Montréal, Quebec, 1995, pp. 102–118.
[42] A.S. Rao, M.P. Georgeff, Modeling rational agents within a BDI-architecture, in: J.F. Allen, R. Fikes, E. Sandewall (Eds.), Proceedings KR-91, Cambridge, MA, Morgan Kaufmann, San Mateo, CA, 1991, pp. 473–484.
[43] A.S. Rao, M.P. Georgeff, Decision procedures for BDI logics, J. Logic Comput. 8 (3) (1998) 293–342.
[44] B. van Linder, W. van der Hoek, J.J.Ch. Meyer, Formalising motivational attitudes of agents: On preferences, goals and commitments, in: M. Wooldridge, J.P. Müller, M. Tambe (Eds.), Intelligent Agents II, Lecture Notes in Artificial Intelligence, Vol. 1037, Springer, Berlin, 1996, pp. 17–32.
[45] G. Venkatesh, A decision method for temporal logic based on resolution, in: Lecture Notes in Computer Science, Vol. 206, Springer, Berlin, 1986, pp. 272–289.
[46] M. Wooldridge, C. Dixon, M. Fisher, A tableau-based proof method for temporal logics of knowledge and belief, J. Appl. Non-Classical Logics 8 (3) (1998) 225–258.
[47] M. Wooldridge, M. Fisher, A decision procedure for a temporal belief logic, in: D.M. Gabbay, H.J. Ohlbach (Eds.), Proceedings Temporal Logic, First International Conference, ICTL'94, Bonn, Germany, Lecture Notes in Artificial Intelligence, Vol. 827, Springer, Berlin, 1994.