Clausal Resolution for Logics of Time and Knowledge with Synchrony and Perfect Recall

Clare Dixon and Michael Fisher

Centre for Agent Research and Development, Department of Computing and Mathematics
Manchester Metropolitan University, Manchester M1 5GD, UK

Tel: (+44) 161 247 {1549,1488}
Fax: (+44) 161 247 1483
Email: {C.Dixon,M.Fisher}@doc.mmu.ac.uk
WWW: http://www.card.mmu.ac.uk

Abstract

Temporal logics of knowledge are widely used in the specification and analysis of distributed and multi-agent systems where the behaviour of a component is captured by its set of possible executions. In spite of this, proof methods for these logics are rare, particularly in cases where the temporal and epistemic dimensions can interact. In this paper, we present a clausal resolution system for a temporal logic of knowledge admitting such interactions. The particular interaction we consider captures synchronous systems with perfect recall. A system is said to have perfect recall if the set of executions an agent considers possible stays the same or decreases over time. Soundness, termination and completeness results for the resolution system are discussed.

Keywords: non-classical resolution, temporal logics of knowledge, synchrony and perfect recall, agent-based systems.

1 Introduction

Temporal logics of knowledge are widely used for analysing distributed and multi-agent systems [7, 9]. Although proof methods have been developed for both temporal logics [1, 4] and modal logics [13], on which the combined logics of knowledge and time are based, little work has been carried out on proof methods for logics comprising both modal and temporal dimensions. For certain classes of systems, interaction occurs between knowledge and time so that, for example, statements such as *"if an agent knows that in the next moment p holds then, in the next moment, the agent knows p holds"* are valid. For many of the interactions described in the literature [11, 12] this increases the complexity of the logic (sometimes dramatically) even for the single agent case. Thus, proof methods for such systems are even rarer than those for the basic logic of time and knowledge.

In this paper we present a resolution-based proof method for temporal logics of knowledge allowing certain interactions between the temporal and epistemic dimensions. The interaction we consider is that relating to synchrony and perfect recall [11, 12]. This work extends our initial development of clausal resolution for temporal logics of knowledge without interactions [2, 3]. Moreover our suggestion to achieve this adds clauses to the normal form to account for the synchrony and perfect recall axiom so that only small extensions of existing resolution rules are required and the temporal resolution theorem prover developed for temporal logics of knowledge (without interactions) can be used to prove formulae with only a small number of changes.

2 Syntax and Semantics

We first give the syntax and semantics of a logic $KL_{(n)}$, a *temporal logic of knowledge* where each modal relation K_i is restricted to be an equivalence relation [12]. The temporal component is interpreted over a discrete linear model of time with finite past and infinite future; an obvious choice for such a flow of time is $(\mathbb{N}, <)$, i.e., Natural numbers ordered by the usual 'less than' relation. This logic has been studied in detail [12] and is the most commonly used temporal logic of knowledge.

2.1 Syntax

Formulae are constructed from a set $\mathcal{P} = \{p, q, r, \ldots\}$ of *primitive propositions*. The language $KL_{(n)}$ contains the standard propositional connectives $\neg$ (not), $\vee$ (or), $\wedge$ (and), $\Rightarrow$ (implies) and $\Leftrightarrow$ (if, and only if). For knowledge we assume a set of agents $Ag = \{1, \ldots n\}$ and introduce a set of unary modal connectives K_i, for $i \in Ag$, where a formula $K_i\phi$ is read as "agent i knows ϕ". For the temporal dimension we take the usual [8] set of future-time temporal connectives $\bigcirc$ (*next*), $\diamondsuit$ (*sometime* or *eventually*), $\square$ (*always*), $\mathcal{U}$ (*until*) and $\mathcal{W}$ (*unless* or *weak until*).

The set of well-formed formulae of $KL_{(n)}$, $\textsc{wff}_K$ is defined as follows:

- **false**, **true** and any element of $\mathcal{P}$ is in $\textsc{wff}_K$;
- if A and B are in $\textsc{wff}_K$ then so are (where $i \in Ag$)

$$\begin{array}{llllll} \neg A & A \vee B & A \wedge B & A \Rightarrow B & A \Leftrightarrow B & \\ \diamondsuit A & \square A & A\,\mathcal{U}\,B & A\,\mathcal{W}\,B & \bigcirc A & K_i A \end{array}$$

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

Definition 1 *A* literal *is either p, or $\neg p$, where $p \in \mathcal{P}$.*

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

Notation: in the following, l are literals, m are either literals or modal literals and D are disjunctions of literals or modal literals.

2.2 Semantics

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

Definition 3 *A* timeline *t, is an infinitely long, linear, discrete sequence of states, indexed by the natural numbers. Let TLines be the set of all timelines. (Note that timelines correspond to the* runs *of Halpern and Vardi [12].)*

Definition 4 *A* point *q, is a pair $q = (t, u)$, where $t \in TLines$ is a timeline and $u \in \mathbb{N}$ is a temporal index into t. Let Points be the set of all points.*

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

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

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

- *π is a valuation.*

As usual, we define the semantics of the language via the satisfaction relation '$\models$'. For $KL_{(n)}$, this relation holds between pairs of the form $\langle M, p\rangle$ (where M is a model and p is a point), and formulae in WFF_K. The rules defining the satisfaction relation are given below.

$$
\begin{array}{lll}
\langle M,(t,u)\rangle \models \mathbf{true} & & \\
\langle M,(t,u)\rangle \models p & \text{iff} & \pi((t,u),p) = T \text{ (where } p \in \mathcal{P}) \\
\langle M,(t,u)\rangle \models \neg A & \text{iff} & \langle M,(t,u)\rangle \not\models A \\
\langle M,(t,u)\rangle \models A \vee B & \text{iff} & \langle M,(t,u)\rangle \models A \ \text{ or } \ \langle M,(t,u)\rangle \models B \\
\langle M,(t,u)\rangle \models \bigcirc A & \text{iff} & \langle M,(t,u+1)\rangle \models A \\
\langle M,(t,u)\rangle \models \square A & \text{iff} & \forall u' \in \mathbb{N}, \text{ if } (u \le u') \text{ then } \langle M,(t,u')\rangle \models A \\
\langle M,(t,u)\rangle \models \Diamond A & \text{iff} & \exists u' \in \mathbb{N}, \text{ if } (u \le u') \text{ then } \langle M,(t,u')\rangle \models A \\
\langle M,(t,u)\rangle \models A\,\mathcal{U}\,B & \text{iff} & \exists u' \in \mathbb{N} \text{ such that } (u' \ge u) \text{ and} \\
 & & \quad \langle M,(t,u')\rangle \models B, \text{ and } \forall u'' \in \mathbb{N}, \\
 & & \quad \text{if } (u \le u'' < u') \text{ then } \langle M,(t,u'')\rangle \models A \\
\langle M,(t,u)\rangle \models A\,\mathcal{W}\,B & \text{iff} & \langle M,(t,u)\rangle \models A\,\mathcal{U}\,B \text{ or } \langle M,(t,u)\rangle \models \square A \\
\langle M,(t,u)\rangle \models K_i A & \text{iff} & \forall t' \in \mathit{TLines}.\ \forall u' \in \mathbb{N}. \text{ if } ((t,u),(t',u')) \in R_i \\
 & & \quad \text{then } \langle M,(t',u')\rangle \models A
\end{array}
$$

Satisfiability and validity in $KL_{(n)}$ are defined in the usual way [12].

As agent accessibility relations in $KL_{(n)}$ models are equivalence relations, the axioms of the normal modal system S5 are valid in $KL_{(n)}$ models. The system S5 is widely recognised as the logic of idealised *knowledge*, and for this reason $KL_{(n)}$ is often termed a *temporal logic of knowledge*.

3 Resolution for Temporal Logics of Knowledge

Resolution based proof methods for temporal logics of knowledge without interactions are given in [3, 2].

3.1 Normal Form

Formulae in $KL_{(n)}$ can be transformed into a normal form SNF_K (Separated Normal Form for temporal logics of knowledge). SNF was introduced first in [4] for linear-time temporal logics and was extended for temporal logics of knowledge in [3]. For the purposes of the normal form we introduce a symbol $\mathbf{start}$ such that $\langle M,(t_0,0)\rangle \models \mathbf{start}$. The translation to SNF_K uses the renaming technique [15] where complex subformulae are replaced by new propositions and the truth value of these propositions is linked to the formulae they replaced in all states. In linear time temporal logic this is achieved by ensuring that such formulae are in the scope of a $\square$ operator, i.e. they hold at all reachable states. For temporal logics of knowledge we introduce the $\square^*$ operator, which allows nesting of K and $\square$ operators to achieve the same result. The $\square^*$ operator is defined in terms of the C (or *common knowledge*) and E operators. We define E by $E\phi \Leftrightarrow \bigwedge_{i \in Ag} K_i\phi$. The common knowledge operator, C, is then defined as the maximal fixpoint of the formula $C\phi \Leftrightarrow E(\phi \wedge C\phi)$. Finally, the $\square^*$ operator is defined as the maximal fixpoint of $\square^*\phi \Leftrightarrow \square(\phi \wedge C\square^*\phi)$. Formulae in SNF_K are of the general form

$$\square^* \bigwedge_j T_j$$

$$
\begin{array}{lcll}
\mathbf{start} & \Rightarrow & \bigvee_{b=1}^{r} l_b & \text{(an } \textit{initial} \text{ clause)} \\
\bigwedge_{a=1}^{g} k_a & \Rightarrow & \bigcirc l & \text{(a } \textit{step} \text{ clause)} \\
\bigwedge_{a=1}^{g} k_a & \Rightarrow & \Diamond l & \text{(a } \textit{sometime} \text{ clause)} \\
\mathbf{true} & \Rightarrow & \bigvee_{b=1}^{r} m_b & \text{(a } K_i\text{–clause)} \\
\mathbf{true} & \Rightarrow & \bigvee_{b=1}^{r} l_b & \text{(a literal clause)}
\end{array}
$$

Figure 1: Clauses in SNF_K

where each T_j, known as a *clause*, must be in one of the varieties described in Figure 1 where k_a, l_b, and l are literals and m_b are either literals or modal literals. Each K_i clause involves literals, or modal literals including the K_i operator where at least one of the disjuncts is a modal literal. The outer '$\Box^*$' 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 T_j.

Here we consider the resolution rules for the temporal logic of knowledge $KL_{(1)}$. To simplify notation we shall write the single modal operator K_1 as K.

3.2 Resolution Rules

The resolution rules presented are split into four groups: those concerned with initial resolution, modal resolution, step resolution and temporal resolution. As well as the resolution rules presented, simplification and subsumption also takes place.

Initial Resolution An initial clause may be resolved with either a literal clause or an initial clause as follows

$$
\text{[IRES1]} \quad \begin{array}{lcl}
\mathbf{true} & \Rightarrow & (A \lor l) \\
\mathbf{start} & \Rightarrow & (B \lor \neg l) \\
\hline
\mathbf{start} & \Rightarrow & (A \lor B)
\end{array}
\qquad
\text{[IRES2]} \quad \begin{array}{lcl}
\mathbf{start} & \Rightarrow & (A \lor l) \\
\mathbf{start} & \Rightarrow & (B \lor \neg l) \\
\hline
\mathbf{start} & \Rightarrow & (A \lor B)
\end{array}
$$

Modal Resolution During modal resolution we apply the following rules which are based on the modal resolution system introduced by Mints [13]. We may resolve a literal or modal literal and its negation or the formulae Kl and $K\neg l$ as we cannot both know something and know its negation.

$$
\text{[MRES1]} \quad \begin{array}{lcl}
\mathbf{true} & \Rightarrow & D \lor m \\
\mathbf{true} & \Rightarrow & D' \lor \neg m \\
\hline
\mathbf{true} & \Rightarrow & D \lor D'
\end{array}
\qquad
\text{[MRES2]} \quad \begin{array}{lcl}
\mathbf{true} & \Rightarrow & D \lor Kl \\
\mathbf{true} & \Rightarrow & D' \lor K\neg l \\
\hline
\mathbf{true} & \Rightarrow & D \lor D'
\end{array}
$$

Next, as we have the T axiom, $\vdash Kp \Rightarrow p$, we can resolve formulae such as Kl with $\neg l$ (giving MRES3). The rule MRES4 requires the function $\mathrm{mod}(D')$, defined below, and is justified due to the external K operator surrounding each clause, i.e, we push K into the second clause and resolve $\neg Kl$ with Kl. The

"mod" function ensures that during this distribution at most one K or $\neg K$ operator applies to each literal due to the equivalences $\neg K \neg K\varphi \Leftrightarrow K\varphi$ and $\neg KK\varphi \Leftrightarrow \neg K\varphi$ in S5.

$$
[\text{MRES3}] \quad \begin{array}{rcl} \mathbf{true} & \Rightarrow & D \vee Kl \\ \mathbf{true} & \Rightarrow & D' \vee \neg l \\ \hline \mathbf{true} & \Rightarrow & D \vee D' \end{array}
\qquad
[\text{MRES4}] \quad \begin{array}{rcl} \mathbf{true} & \Rightarrow & D \vee \neg Kl \\ \mathbf{true} & \Rightarrow & D' \vee l \\ \hline \mathbf{true} & \Rightarrow & D \vee \mathrm{mod}\,(D') \end{array}
$$

Definition 7 *The function* $\mathrm{mod}\,(D)$*, defined on disjunctions of literals or modal literals D, is defined as follows.*

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

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

$$
[\text{MRES5}] \qquad \begin{array}{rcl} \mathbf{true} & \Rightarrow & L \vee Kl_1 \vee Kl_2 \vee \ldots \\ \hline \mathbf{true} & \Rightarrow & L \vee l_1 \vee l_2 \vee \ldots \end{array}
$$

Here, L is a disjunction of literals. The set of modal rules can, if necessary, be extended to the multi-modal case [2].

Step Resolution 'Step' resolution consists of the application of standard classical resolution to formulae representing constraints at a particular moment in time, together with simplification rules for transferring contradictions within states to constraints on previous states (standard simplification and subsumption rules are also applied). The following resolution rules may be applied by resolving two step clauses or a step clause with a literal clause.

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

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

Once a contradiction within a state is found, the following rule can be used to generate additional literal clauses.

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

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

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

Temporal Resolution During temporal resolution the aim is to resolve a sometime clause, $Q \Rightarrow \Diamond l$, with a set of step clauses that together imply $\square\neg l$. However, the interaction between the '$\bigcirc$' and '$\square$' operators in $KL_{(n)}$ makes the definition of such a rule non-trivial and further the translation from $KL_{(n)}$ to SNF_K will have removed all but the outer level of $\square$–operators. So, resolution will be between a sometime clause and a *set* of clauses that together imply an $\square$–formula which will contradict the sometime clause. Thus, given a set of clauses in SNF_K, then for every clause of the form $Q \Rightarrow \Diamond l$ temporal resolution may be applied between this sometime clause and a set of conjunctions of one or more step clauses, which taken together force $\neg l$ to always be satisfied in the future. The temporal resolution rule (TRES) is:

$$\begin{array}{rcl} A_0 & \Rightarrow & \bigcirc F_0 \\ \dots & & \dots \\ A_n & \Rightarrow & \bigcirc F_n \\ Q & \Rightarrow & \Diamond l \\ \hline Q & \Rightarrow & (\bigwedge_{i=0}^{n} \neg A_i)\, \mathcal{W}\, l \end{array} \quad \begin{array}{c}\text{with}\\ \text{side}\\ \text{conditions}\end{array} \quad \left\{ \begin{array}{rcl} \text{for all } 0 \le i \le n & \vdash & F_i \Rightarrow \neg l \\ \text{and} & \vdash & F_i \Rightarrow \bigvee_{j=0}^{n} A_j \end{array} \right\}$$

These side conditions ensure that the set of clauses $A_i \Rightarrow \bigcirc F_i$ together imply $\bigcirc \square \neg l$. In particular the first side condition ensures that each clause, $A_i \Rightarrow \bigcirc F_i$, makes $\neg l$ true in the next moment if A_i is satisfied. The second side condition ensures that the right-hand side of each clause satisfies the left-hand side of one of the clauses in the set.

So once the left hand side of one of these rules is satisfied, i.e. if A_i is satisfied for some i in the last moment in time, then $\neg l$ will hold now and the left hand side of another rule will also be satisfied. Thus at the next moment in time again $\neg l$ holds and the left hand side of another rule is satisfied and so on.

So if any of the A_i are satisfied then $\neg l$ will be *always* be satisfied in the future, i.e.,

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

Such a set of rules are known as a *loop* in $\neg l$ (for further details, see [4, 6]).

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

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

3.3 The temporal resolution algorithm

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

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

4 Interactions

Here we define the interaction between knowledge and time we consider, and how to adapt the resolution system without interactions for this specific, but useful, interaction.

4.1 Definitions

The temporal logics of knowledge described above provide no account of how knowledge evolves over time, i.e. how the knowledge and time components interact. For example we may wish to model agents whose knowledge grows or increases over time. This has been termed no forgetting or *perfect recall* in the literature [11, 12]. Additionally, an assumption often made of systems is that of synchrony, i.e. access to a global clock. In the rest of the paper we consider systems with synchrony and perfect recall.

We begin with some definitions; firstly, the notion of *indistinguishability*.

Definition 8 *Two points* (l, u) *and* (l', u') *are said to be* indistinguishable *by agent i if* $((l, u), (l', u')) \in R_i$.

Given this, we define the particular interaction we are concerned with (from [12]).

Definition 9 *Systems with* perfect recall *are those where the number of possible timelines indistinguishable to an agent either decreases or stays the same over time.*

In this case, if an agent considers two points on two timelines indistinguishable now, the two timelines are indistinguishable (by the agent) at all points in the past.

Definition 10 *The notion of* synchrony *characterises those systems where each agent has access to a global clock that ticks at every moment of time.*

Effectively, each agent can distinguish between each distinct point in a run with the clock reading being part of the state.

When we have systems with the properties of synchrony and perfect recall together, given point $(r, n+1)$, if we have the point $((s, m+1), (r, n+1)) \in R_i$ then $((s, m), (r, n)) \in R_i$ and $m = n$. It has been shown that the usual axiomatisations of propositional reasoning, knowledge, linear-time temporal logic plus the axiom $\vdash K_i \bigcirc \varphi \Rightarrow \bigcirc K_i \varphi$ give a sound and complete axiomatisation for systems of synchrony and perfect recall [11].

The complexity of the validity problem for the logics derived from allowing different combinations are given in [11, 12]. In the single agent case the complexity of the validity problem for systems of synchrony and perfect recall is double-exponential time, while in the multi-agent case, the complexity is nonelementary time.

In the single agent case, we denote the logics that capture the class of systems with synchrony and perfect recall as $KL^{sync,pr}$.

4.2 Translation to Normal Form

We use a similar normal form, SNF$_K$, to before but add new propositions and clauses once the translation has been carried out. For every literal, l, occurring in the SNF form (or disjunctive literals introduced over the page), add new propositions $new(l)$ in the form of the following new SNF clauses:

$$N1(l): \ \mathbf{true} \Rightarrow \neg new(l) \lor Kl \qquad N2(l): \ \mathbf{true} \Rightarrow new(l) \lor \neg Kl.$$

The important point to note is that these clauses ensure $new(l) \Leftrightarrow Kl$ which will be useful in both §4.4 and §5. In the rest of this document we call literals of the form $new(l)$ or $\neg new(l)$ *SPR-literals* and any other literals derived in the transformation to the normal form *basic literals*.

4.3 New Step Resolution Rules

In addition to SRES1 and SRES2 we add the following addition step resolution rules in order to generate the most comprehensive set of of step clauses.

$$\text{[SRES3]} \quad \begin{array}{rcl} P \wedge l & \Rightarrow & \bigcirc A \\ Q \wedge \neg l & \Rightarrow & \bigcirc B \\ \hline (P \wedge Q) & \Rightarrow & \bigcirc (A \vee B) \end{array}$$

$$\text{[SRES4]} \quad \begin{array}{rcl} \mathbf{true} & \Rightarrow & (A \vee l) \\ Q \wedge l & \Rightarrow & \bigcirc B \\ \hline \neg A \wedge Q & \Rightarrow & \bigcirc B \end{array}$$

It is easy to show soundness of these additional rules by rewriting implications as disjunctions and applying standard classical resolution. Indeed SRES3 and SRES4 could also be added as rules in the proof method for temporal logics of knowledge without interactions but are not necessary for completeness.

4.4 New Step Clauses

We now consider the extension of the resolution system given in §3 to handle the interaction between knowledge and time characterised by synchrony and perfect recall. For every step clause generated during the proof we add a new step clause from applying the interaction axiom and renaming the modal literals. In particular we push an external K operator through every step clause in SNF, apply the synchrony and perfect recall axiom to the result and replace the modal literals by the relevant new proposition above. For example for the step clause $x \Rightarrow \bigcirc y$ where x and y are both literals we push through a K to obtain $Kx \Rightarrow K\bigcirc y$. Next we apply the synchrony and perfect recall axiom to swap the K and the $\bigcirc$ operators to obtain $Kx \Rightarrow \bigcirc Ky$. and finally Kx and Ky are replaced by $new(x)$ and $new(y)$ to give $new(x) \Rightarrow \bigcirc new(y)$. A slight complication arises with step rules containing disjunctions (of basic literals) on the right hand side these are replaced by a new proposition symbol and a clause is added giving the meaning of the new symbol.

More formally for any resolvent

$$X \Rightarrow \bigcirc l$$

where l is a basic literal we add the clause

$$NEW(X) \Rightarrow \bigcirc new(l)$$

where the function NEW is defined below. Given a disjunction of basic literals on the right hand side, i.e.

$$X \Rightarrow \bigcirc \bigvee_{j=1}^{m} l_j$$

we rename the right hand side using a new name $\neg p_{\bigvee_j l_j}$ and add the clauses

$$\begin{array}{rcl} NEW(X) & \Rightarrow & \bigcirc new(\neg p_{\bigvee_j l_j}) \\ \mathbf{true} & \Rightarrow & \neg p_{\bigvee_j l_j} \vee \bigvee_j l_j \\ \mathbf{true} & \Rightarrow & p_{\bigvee_j l_j} \vee \neg l_j \end{array}$$

Let the literals of the form $p_{\bigvee_j l_j}$ be known as *disjunctive literals*. More generally on the right hand side we may have a disjunction of basic literals, SPR literals or disjunctive literals. The disjunctive literals may be resolved away using their definitions, creating one or more step clauses containing only SPR literals or basic literals. Hence for any step rule of the form

$$X \Rightarrow \bigcirc \left(\bigvee_j new(l_j) \vee \bigvee_i l_i \right)$$

we add

$$\begin{array}{rcl} NEW(X) & \Rightarrow & \bigcirc(\bigvee new(l_a) \vee new(p_{\bigvee_i l_i}) \\ \mathbf{true} & \Rightarrow & \neg p_{\bigvee_i l_i} \vee \bigvee_i l_i \\ \mathbf{true} & \Rightarrow & p_{\bigvee_i l_i} \vee \neg l_i \end{array}$$

where *NEW* is defined below.

$$\begin{array}{rcll} NEW(\bigwedge_j l_j) & = & \bigwedge_j NEW(l_j) & \\ NEW(l) & = & new(l) & \text{where } l \text{ is a basic literal} \\ NEW(new(l)) & = & new(l) & \\ NEW(\neg new(l)) & = & \neg new(l) & \end{array}$$

Finally as $new(l)$ is just a new name for Kl we add $\mathrm{mod}\,(new(l)) = new(l)$ to the definition of the mod function given in §3.2, essentially to match the definition for Kl.

4.5 Examples

Example 1

We begin by simply proving the validity of the the synchrony and perfect recall axiom, i.e. $\vdash K\bigcirc\varphi \Rightarrow \bigcirc K\varphi$. We negate the axiom, and translate into the normal form giving

$$\begin{array}{rrcl} 1. & \mathbf{start} & \Rightarrow & x \\ 2. & \mathbf{true} & \Rightarrow & \neg x \vee Ky \\ 3. & y & \Rightarrow & \bigcirc\varphi \\ 4. & x & \Rightarrow & \bigcirc z \\ 5. & \mathbf{true} & \Rightarrow & \neg z \vee \neg K\varphi \end{array}$$

Next we add the *new* versions of clauses 3 and 4 to obtain the following.

$$\begin{array}{rrcll} 6. & new(y) & \Rightarrow & \bigcirc new(\varphi) & [3,\ SPR] \\ 7. & new(x) & \Rightarrow & \bigcirc new(z) & [4,\ SPR] \end{array}$$

Then the refutation proceeds as follows.

$$\begin{array}{rrcll} 8. & \mathbf{true} & \Rightarrow & \neg z \vee \neg new(\varphi) & [5, N1(\varphi),\ MRES1] \\ 9. & new(y) & \Rightarrow & \bigcirc\neg z & [6, 8,\ SRES2] \\ 10. & (x \wedge new(y)) & \Rightarrow & \mathbf{false} & [4, 9,\ SRES1] \\ 11. & \mathbf{true} & \Rightarrow & \neg x \vee \neg new(y) & [10,\ SRES3] \\ 12. & \mathbf{true} & \Rightarrow & \neg x \vee \neg Ky & [11, N2(y),\ MRES1] \\ 13. & \mathbf{true} & \Rightarrow & \neg x & [2, 12,\ MRES1] \\ 14. & \mathbf{start} & \Rightarrow & \mathbf{false} & [1, 13,\ IRES1] \end{array}$$

Example 2

As a more complex example we show that $K\Box\varphi \Rightarrow \Box K\varphi$ is valid in $KL^{sync,pr}$. First we negate, giving $K\Box\varphi \wedge \neg\Box K\varphi$, and translate into the normal form.

$$\begin{array}{rrcl} 1. & \mathbf{start} & \Rightarrow & a \\ 2. & a & \Rightarrow & \Diamond b \\ 3. & \mathbf{true} & \Rightarrow & \neg b \vee \neg K\varphi \\ 4. & \mathbf{true} & \Rightarrow & \neg a \vee Kc \\ 5. & \mathbf{true} & \Rightarrow & \neg c \vee \varphi \\ 6. & \mathbf{true} & \Rightarrow & \neg c \vee t \\ 7. & t & \Rightarrow & \bigcirc t \\ 8. & t & \Rightarrow & \bigcirc\varphi \end{array}$$

Next we add the new step clauses 9 and 10 (due to 7 and 8).

$$\begin{array}{rrcl} 9. & new(t) & \Rightarrow & \bigcirc new(t) \\ 10. & new(t) & \Rightarrow & \bigcirc new(\varphi) \end{array}$$

The refutation proceeds as follows.

$$
\begin{array}{llcll}
11. & \mathbf{true} & \Rightarrow & \neg b \vee \neg Kc & [3,5,\ \mathit{MRES}4] \\
12. & \mathbf{true} & \Rightarrow & \neg b \vee \neg new(\varphi) & [3, N1(\varphi),\ \mathit{MRES}1] \\
13. & new(t) & \Rightarrow & \bigcirc \neg b & [10,15,\ \mathit{SRES}2]
\end{array}
$$

We note that by merging clauses 9 and 16 we obtain $new(t) \Rightarrow \bigcirc(new(t) \wedge \neg b)$ which can be (temporally) resolved with clause 2, as follows.

$$
\begin{array}{lrcll}
14. & a & \Rightarrow & \neg new(t)\, \mathcal{W}\, b & [2,9,13,\ \mathit{TRES}] \\
15. & \mathbf{true} & \Rightarrow & \neg a \vee b \vee \neg new(t) & [13] \\
16. & \mathbf{true} & \Rightarrow & \neg a \vee b \vee \neg Kt & [15, N2(t),\ \mathit{MRES}1] \\
17. & \mathbf{true} & \Rightarrow & \neg a \vee b \vee \neg Kc & [6,16,\ \mathit{MRES}4] \\
18. & \mathbf{true} & \Rightarrow & \neg a \vee \neg Kc & [11,17,\ \mathit{MRES}1] \\
19. & \mathbf{true} & \Rightarrow & \neg a & [4,18,\ \mathit{MRES}1] \\
20. & \mathbf{start} & \Rightarrow & \mathbf{false} & [1,19,\ \mathit{IRES}1]
\end{array}
$$

Example 3

Next we show that $(K\Diamond p \wedge K\neg p) \Rightarrow \bigcirc K \Diamond p$. First translating into the normal form we obtain the following.

$$
\begin{array}{lrcl}
1. & \mathbf{start} & \Rightarrow & x \\
2. & \mathbf{true} & \Rightarrow & \neg x \vee Ky \\
3. & y & \Rightarrow & \Diamond p \\
4. & \mathbf{true} & \Rightarrow & \neg x \vee K\neg p \\
5. & x & \Rightarrow & \bigcirc z \\
6. & \mathbf{true} & \Rightarrow & \neg z \vee \neg K \neg w \\
7. & \mathbf{true} & \Rightarrow & \neg w \vee \neg p \\
8. & \mathbf{true} & \Rightarrow & \neg w \vee t \\
9. & t & \Rightarrow & \bigcirc t \\
10. & t & \Rightarrow & \bigcirc \neg p
\end{array}
$$

Next we add the new formulae from the step rules.

$$
\begin{array}{lrcl}
11. & new(x) & \Rightarrow & \bigcirc new(z) \\
12. & new(t) & \Rightarrow & \bigcirc new(t) \\
13. & new(t) & \Rightarrow & \bigcirc new(\neg p)
\end{array}
$$

Next we perform temporal resolution between rules 9, 10 and 3 giving resolvent

$$y \Rightarrow \neg t\, \mathcal{W}\, p.$$

Translating into normal form we obtain

$$
\begin{array}{lrcl}
14. & w_p & \Rightarrow & \bigcirc(p \vee \neg t) \\
15. & w_p & \Rightarrow & \bigcirc(p \vee w_p) \\
16. & \mathbf{true} & \Rightarrow & \neg y \vee p \vee w_p \\
17. & \mathbf{true} & \Rightarrow & \neg y \vee \neg t \vee p
\end{array}
$$

and their “new” counterparts. Translating into normal form we obtain

18.	w_p	$\Rightarrow$	$\bigcirc(\neg w \vee \neg t)$	$[7, 14, SRES2]$
19.	w_p	$\Rightarrow$	$\bigcirc(\neg w)$	$[8, 18, SRES2]$
20.	$new(w_p)$	$\Rightarrow$	$\bigcirc new(\neg w)$	$[SPR(19)]$
21.	$\mathbf{true}$	$\Rightarrow$	$\neg z \vee \neg new(\neg w)$	$[6, N2(\neg w)]$
22.	$new(w_p)$	$\Rightarrow$	$\bigcirc \neg z$	$[20, 21, SRES2]$
23.	$new(w_p) \wedge x$	$\Rightarrow$	$\bigcirc \mathbf{false}$	$[5, 22, SRES1]$
24.	$\mathbf{true}$	$\Rightarrow$	$\neg new(w_p) \vee \neg x$	$[23, SRES3]$
25.	$\mathbf{true}$	$\Rightarrow$	$\neg Ky \vee \neg K\neg p \vee new(w_p)$	$[16, N2(w_p), MRES4]$
26.	$\mathbf{true}$	$\Rightarrow$	$\neg Ky \vee \neg K\neg p \vee \neg x$	$[24, 25, MRES1]$
27.	$\mathbf{true}$	$\Rightarrow$	$\neg Ky \vee \neg x$	$[4, 26, MRES1]$
28.	$\mathbf{true}$	$\Rightarrow$	$\neg x$	$[2, 27, MRES1]$
29.	$\mathbf{start}$	$\Rightarrow$	$\mathbf{false}$	$[1, 28, IRES1]$

5 Correctness

We give outlines of the soundness, termination and completeness arguments below[1]. Note that, correctness of the resolution system for the single agent case without interactions is given in [3], while characterisation of synchrony and prefect recall by the axiom $\vdash K\bigcirc\varphi \Rightarrow \bigcirc K\varphi$ is given in [11].

Soundness It is easy to show that the application of any of the resolution rules preserves satisfiability. The introduction of the new variables $new(x)$ does not affect soundness as they are just new names for modal literals with corresponding clauses defining their meaning. The addition of the new step clauses is sound due to the K axiom and the synchrony and perfect recall axiom.

Termination The core of the termination argument is to show that none of the resolution rules, generate resolvents that require new proposition symbols to be defined. Firstly, although new propositions are required to translate the resolvent from performing temporal resolution into SNF we have shown that we can add these, for each eventuality, at the start of the proof [6]. Here we add the corresponding N1 and N2 clauses for each of these new propositions. None of the other resolution rules require new proposition symbols in their conclusion. Further we don't have to define new names for formulae such as $Knew(l)$ (which may occur following the addition of step clauses after temporal resolution) because $Knew(l) \Leftrightarrow new(l)$ as follows. As $new(l) \Leftrightarrow Kl$, $Knew(l) \Leftrightarrow KKl$ and from the axioms T and 5, $Kl \Leftrightarrow KKl$ so we have $new(l) \Leftrightarrow Knew(l)$. Finally during step resolution we generate new propositions of the form $p_{a \vee b}$ where a and b are from the set of basic literals. Given the set of basic literals after translation to SNF and the careful renaming of disjunctions on the right side of the step clauses there are a finite number that we need.

Thus as we have a finite set of symbols there are a finite number left and right hand sides of clauses that can be defined (modulo ordering of literals or modal literals) so at some point either we derive a contradiction or no new clauses can be derived.

Completeness We construct a graph for a set of SNF rules S where nodes are pairs (V, E), where V is a consistent set of literals and modal literals and E is a subset of the eventualities occurring in S. Temporal edges are drawn between two nodes n and n' if and only if $S' \subseteq S$ is the largest set of step rules whose left hand sides are satisfied by n and n' satisfies the right hand side of the clauses in S' (without the next operator). Modal edges are drawn between nodes with the same sets of modal literals. From the definition of the new propositions $new(l)$ holds in nodes exactly where Kl does. We show that an empty graph corresponds to an unsatisfiable set of rules S. Part of this proof involves showing that when the graph is non-empty a model can be extracted, i.e. by constructing a timeline from an initial node by

[1]Full details are given in the full paper

unwinding through the temporal edges, constructing the equivalence class at each moment in time and the corresponding temporal edges between nodes in each equivalence class. Then we show that if two nodes are in the same equivalence class their predecessors are in the same equivalence class. Finally we show that the graph is empty if and only if there is a refutation by the resolution method given in this paper.

6 Conclusions and Future Work

We have given resolution method for proof in systems of synchrony and perfect recall. Such interactions are necessary if we want to allow knowledge to evolve over time. In particular, if we consider the representation of an agent that manipulates information of some form, then the agent's knowledge is likely to change over time. Thus, we can use the resolution system described in this paper in order to verify properties of distributed and agent-based systems. No new resolution rules are required. The main features of the method are the similarity of the step clauses in the normal form to the structure of the synchrony and perfect recall axiom and the use of a new name for modal literals which can be used in the step and temporal resolution rules. The method described in this paper can be implemented with a small number of changes to a theorem prover for temporal logics of knowledge. Indeed, examples have been proved in this way using a prototype system encoded in prolog.

Allowing interaction between knowledge and time makes the complexity of the logic high. Thus, we need to devise strategies to efficiently apply the above resolution rules and so avoid the worst case complexities where possible. Future work involves both the investigation of such strategies, and the use of case studies to see which interactions occur commonly in real world problems. Indeed an interaction corresponding to the synchrony and perfect recall axiom has been highlighted as desirable during work with the developers of the multi-agent specification language KARO [17]. We intend to extend the ideas developed in this paper on synchrony and perfect recall to other axioms suggestion in [10], for example the converse of this axiom known as synchrony and no learning, and perfect recall (without synchrony) [16]. Finally, we intend to consider how to devise appropriate resolution rules given arbitrary interaction axioms of the form $\vdash K^a\bigcirc^b\varphi \Rightarrow \bigcirc^c K^d\varphi$ or $\vdash \bigcirc^a K^b\varphi \Rightarrow K^c\bigcirc^d\varphi$.

Acknowledgements This work was partially supported by EPSRC Research Grants GR/K57282 and GR/M44859. The authors would like to thank Claudia Nalon for her comments on this work.

References

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

[2] C. Dixon and M. Fisher. Resolution-Based Proof for Multi-Modal Temporal Logics of Knowledge. In S. Goodwin and A. Trudel, editors, *Proceedings of TIME-00 the Seventh International Workshop on Temporal Representation and Reasoning*, Cape Breton, Nova Scotia, Canada, July 2000. IEEE Press.

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

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

[5] M. Fisher. A Normal Form for Temporal Logic and its Application in Theorem-Proving and Execution. *Journal of Logic and Computation*, 7(4):429–456, August 1997.

[6] M. Fisher, C. Dixon, and M. Peim. Clausal Temporal Resolution. *ACM Transactions on Computational Logic*, 1(4), April 2001.

[7] M. Fisher and M. Wooldridge. On the Formal Specification and Verification of Multi-Agent Systems. *International Journal of Cooperative Information Systems*, 6(1), January 1997.

[8] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. The Temporal Analysis of Fairness. In *Proceedings of the Seventh ACM Symposium on the Principles of Programming Languages*, pages 163–173, Las Vegas, Nevada, January 1980.

[9] J. Y. Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2, 1987.

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

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

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

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

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

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

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

[17] B. van Linder, W. van der Hoek, and J.J. Ch. Meyer. Formalising abilities and opportunities of agents. Technical Report UU-CS-1998-08, Department of Computer Science,Utrecht University, Centrumgebouw Noord, Padualaan 14, De Uithof, 3584 CH Utrecht, 1998.