Specifying and Verifying the Game Cluedo using Temporal Logics of Knowledge

Clare Dixon

Department of Computer Science
University of Liverpool
Liverpool L69 7ZF
United Kingdom

Tel: (+44) 151 794 3674
Fax: (+44) 161 247 3715
Email: {C.Dixon}@csc.liv.ac.uk

Abstract

Temporal logics of knowledge are useful for reasoning about situations where the knowledge of an agent or component is important, and where change in this knowledge may occur over time. Here we use temporal logics of knowledge to reason about the game Cluedo. We show how to specify Cluedo using temporal logics of knowledge and prove statements about the knowledge of the players using a clausal resolution calculus for this logic.

1 Introduction

Temporal logics are useful for specifying dynamic systems that change over time. Logics of knowledge are useful for specifying systems where statements such as *if I know that I hold card A then I know that you don't hold card A* are required. Together, temporal logics of knowledge have been used where we require both dynamic aspects and informational aspects relating to knowledge. Temporal and/or knowledge logics have been used for the specification and verification of distributed and multi-agent systems [12, 15, 19], analysing security protocols [25, 14], knowledge games such as the muddy children [9, 6] etc.

Cluedo, commercially produced by Hasbro [17], is a board game where players gather information about a murder. The suspects, murder weapons and room where the murder took place are represented by playing cards. Three cards are secretly removed from the pack to represent the murderer, the murder weapon and the location of the murder. The remaining cards are shuffled and dealt to the players. Players attempt to find out the identity of the murderer, murder weapon and location of the murder by the knowledge of their own cards, and using knowledge obtained from cards revealed by other players during the game or from statements that another player has no such card.

We show how the game can be specified using a temporal logic of knowledge and how moves in the game correspond to additional knowledge for one or more of the players. Using a simplified version of the game, we show how to prove certain inferences using a resolution based approach. The contribution of the paper is a case study using temporal logics of knowledge to represent and reason about the game Cluedo. In particular we demonstrate the suitability of this logic for specifying Cluedo and how to verify the derived knowledge of players using a resolution calculus for this logic.

The logic, KL_n, we consider is the fusion of linear time temporal logic with finite past and infinite future combined with the multi-modal logic S5 (see for example [9] for more details about this logic). To prove that a particular property φ follows from a problem specification ψ, where both φ and ψ are formulae of KL_n we must show $\vdash \psi \Rightarrow \varphi$ Here ψ represents the current situation in the game relating to the cards owned by a particular player and the knowledge gained from previous turns. As we use a refutation method we prove that $\psi \wedge \neg\varphi$ are unsatisfiable. We carry out proofs using resolution for temporal logics of knowledge [5, 6]. This calculus uses, a translation to 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 clauses containing only literals.

The paper is organised as follows. In Section 2 we describe the game Cluedo in more detail and in Section 3 we give the syntax and semantics of KL_n the temporal logic of knowledge we use to specify and verify Cluedo. In Section 4 we show how moves in the game can be specified in this logic. In Section 5 we give the resolution calculus for KL_n which we later use to prove properties that can be derived at different stages of the game. In Section 6 we show how to specify and verify Cluedo games using an example of a particular game. Player's assistants that have been developed to keep track of knowledge in Cluedo are described in Section 7 and conclusions and related work are mentioned in Section 8.

2 The Game Cluedo

Cluedo is a board game where players gather information about a murder. The suspects, murder weapons and room where the murder took place are represented by playing cards. One from each of these sets is removed and placed, without any of the players seeing, in an envelope to represent the actual murderer, murder weapon and location of the murder. The remaining cards are shuffled and dealt out to the players. Players take it in turns to make *suggestions*, a triple: suspect, weapon and room. If the player to their left has one of these cards it is shown secretly to the player making the suggestion. The other players can see a card has been shown to the player making the suggestion but do not know its identity. If the player to the left does not hold one of the three cards in the suggestion, she declares this and the player to her left must try and show the suggesting player one of the cards. This continues until a card has been shown to the suggesting player or no card has been shown by any player for this suggestion. Players use the knowledge about the cards in their hand and knowledge about cards other players may or may not hold to eliminate suspects, weapons and rooms from their enquiries. When a player knows the murderer, murder weapon, and room she makes an *accusation* and checks the hidden murder cards. If she is correct she wins the game. Otherwise she takes no further part in the game (i.e. cannot make suggestions and accusations) but does answer the suggestions of other players. Note, whilst each player makes many suggestions during a game, at most one accusation is made by each player during a game so the player should be certain about the murderer, weapon and location before making an accusation.

The commercial version of the game is produced by Hasbro [17] and involves a board which represents the rooms in the house and access between them. The players are represented on the board using coloured pieces, one for each of the suspects. Movement about the board is done by rolling a dice and the player moves their piece that number of squares towards a room. A turn consists of rolling the dice and moving (or staying in the current room), making a suggestion (if the player's piece is in or has reached a room) and making an accusation (if the player feels she knows who the murderer, weapon and location of the murder is). The coloured piece of a player must be in a particular room before they can make a suggestion involving that room. Also the coloured piece representing the relevant suspect is moved from its current position on the board into the room involved in the suggestion.

There are six suspects:-

- Col. Mustard (Yellow);
- Prof. Plum (Purple);
- Rev. Green (Green);
- Mrs. Peacock (Blue);
- Miss Scarlett (Red);
- Mrs. White (White);

six weapons :-

- Dagger;
- Candlestick;
- Revolver;
- Rope;
- Lead Piping;
- Spanner;

and nine rooms:-

- lounge;
- conservatory;
- dining room;
- study;
- ballroom;
- kitchen;
- hall;
- billiard room;
- library.

However there are different versions of the game with slightly different weapons and rooms and with small differences to the rules.

As we are interested primarily in the changes in knowledge we simplify the game and do not consider the aspects relating to the board, dice and movement around the board. Hence when it is their turn a player can make a suggestion involving any room.

3 Syntax and Semantics

The logic, KL_n, a *temporal logic of knowledge* we consider is the fusion of linear-time temporal logic with multi-modal S5. We first give the syntax and semantics of KL_n, where each modal relation is restricted to be an equivalence relation [16]. 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 [16] and is the most commonly used temporal logic of knowledge.

3.1 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) and $\Rightarrow$ (implies). 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 [13] set of future-time temporal connectives $\bigcirc$ (*next*), $\Diamond$ (*sometime* or *eventually*), $\Box$ (*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 $i \in Ag$)

$$\begin{array}{lllll} \neg A & A \vee B & A \wedge B & A \Rightarrow B & K_i A \\ \Diamond A & \Box A & A \,\mathcal{U}\, B & A \,\mathcal{W}\, B & \bigcirc A \end{array}$$

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.

3.2 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.*

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 in $TL \times \mathbb{N}$), and formulae in $\textsc{wff}_K$. The rules defining the satisfaction relation are given below.

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

For any formula A, if there is some model M and timeline t such that $\langle M,(t,0)\rangle \models A$, then A is said to be satisfiable. If for any formula A, for all models M there exists a timeline t such that $\langle M,(t,0)\rangle \models A$ then A is said to be valid. Note, this is the anchored version of the (temporal) logic, i.e. validity and satisfiability are evaluated at the beginning of time (see for example [8]).

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*.

Note, the KL_n logic does not include operators for *everyone knows* (E) or *common knowledge* (C). However these are used in the definition of the normal form (see Section 5.1) so we mention them here. 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 $C\phi \ \Leftrightarrow \ E(\phi \wedge C\phi)$.

4 Specifying a Cluedo Game

First we will reduce the game to illustrate how we may specify actions in the game. The changes in knowledge relating to moves is Cluedo has been described in [27] but using a different logic. Let us assume that we have simply four suspects (Prof. Plum, Rev. Green, Col. Mustard and Miss Scarlett), four weapons (lead piping, spanner, revolver and rope) and no rooms. It is easy to scale this up to the full number of suspects, weapons and rooms. In our example we assume three players Catherine, Wendy and Jane.

Let the set of players $Ag = \{c, w, j\}$. First we use propositions to show who holds each of the cards, where $i \in \{c, w, j\}$.

- r_i is true if player i holds Miss Scarlett
- g_i is true if player i holds Rev Green
- y_i is true if player i holds Col. Mustard
- b_i is true if player i holds Prof. Plum
- l_i is true if player i holds lead piping

- s_i is true if player i holds spanner
- v_i is true if player i holds revolver
- p_i is true if player i holds rope

A suspect is denoted as the murderer, or a weapon as the murder weapon by having m as a suffix.

- r_m is true if Miss Scarlett is the murderer
- g_m is true if Rev Green is the murderer
- y_m is true if Col. Mustard is the murderer
- b_m is true if Prof. Plum is the murderer
- l_m is true if lead piping is the murderer weapon
- s_m is true if spanner is the murderer weapon
- v_m is true if revolver is the murderer weapon
- p_m is true if rope is the murderer weapon

At The Start of the Game Initially (and throughout the game) one of the suspects must be the murderer

$$(r_m \vee g_m \vee y_m \vee b_m)$$

and one of the weapons is the murder weapon

$$(l_m \vee s_m \vee v_m \vee p_m).$$

Each player knows this information $i \in \{c, w, j\}$

$$\begin{array}{l} K_i(r_m \vee g_m \vee y_m \vee b_m) \\ K_i(l_m \vee s_m \vee v_m \vee p_m) \end{array}$$

and they know themselves that other players know this etc (i.e. it is common knowledge). Initially (and throughout the game) each card must be held by one of the players or it must be the murderer or murder weapon. For example for Miss Scarlett

$$r_c \vee r_w \vee r_j \vee r_m$$

and similarly for the other suspects and weapons. As before each player knows this and knows that other players know this etc (i.e. it is common knowledge).

If a player holds a card then the other players don't hold it and it can't be the murderer or murder weapon. If a card is the murder suspect or weapon then none of the players can hold that card.

$$\begin{array}{l} r_c \Rightarrow (\neg r_w \wedge \neg r_j \wedge \neg r_m) \\ r_w \Rightarrow (\neg r_c \wedge \neg r_j \wedge \neg r_m) \\ r_j \Rightarrow (\neg r_w \wedge \neg r_c \wedge \neg r_m) \\ r_m \Rightarrow (\neg r_w \wedge \neg r_j \wedge \neg r_c) \end{array}$$

Again this is known by each player and that other players know this etc (i.e. it is common knowledge).

After the Deal After the deal, each player knows they hold the cards that they have been dealt and knows that they don't hold the cards they haven't been dealt. For example if Catherine is dealt Miss Scarlett and Rev. Green then

$$K_c r_c \wedge K_c g_c \wedge K_c \neg y_c \wedge K_c \neg b_c \wedge K_c \neg l_c \wedge K_c \neg s_c \wedge K_c \neg v_c \wedge K_c \neg p_c$$

After a Suggestion If Catherine makes the suggestion Miss Scarlett and the lead piping, then there are two options, either the next player has one of these cards and shows her or the next player does not have one of these cards. Let us assume that Wendy first tries to answer the suggestion.

For the former, i.e. Wendy does not hold Miss Scarlett or the lead piping, Wendy states that she does not hold them for all to hear. From this statements $\neg r_w$ and $\neg l_w$ becomes known to everyone (and is common knowledge). In particular Catherine and Jane have both learnt this information.

$$K_c \neg r_w \wedge K_c \neg l_w$$
$$K_j \neg r_w \wedge K_j \neg l_w$$

For the latter i.e. Wendy holds one of Miss Scarlett or the lead piping, several inferences can be made. Firstly that that Wendy holds Miss Scarlett or the lead piping is known to each player i.e. $i \in \{c, w, j\}$

$$K_i(r_w \vee l_w)$$

in particular Jane knows this, i.e.

$$K_j(r_w \vee l_w)$$

and this again is common knowledge. Also Catherine learns the identity of one of the cards Wendy holds (say it is the lead piping) i.e.

$$K_c l_w.$$

Further Wendy knows Catherine know this etc, but we do not record further nesting of knowledge.

Lastly if a player (say Catherine) makes a suggestion (for example Miss Scarlett and the lead piping), all the other players state they do not hold one of these cards and the suggesting player, Catherine, does not make an accusation each player then knows that Catherine must hold either Miss Scarlett or the lead piping, i.e. $i \in \{c, w, j\}$

$$K_i(r_c \vee l_c).$$

This is again common knowledge.

The End of the Game The game ends when one of the players know the murderer and the murder weapon i.e.

$$K_i(x_m \wedge z_m)$$

where $i \in \{c, w, j\}$ and x is one of r, g, y, b and z is one of l, s, v, p.

Dealing with Time We assume that each element of the game where the knowledge changes occurs at the next time point. For example we assume that the deal occurs at time one, the first time an suggestion is answered is at time two etc.

If a particular player knows who holds (or does not hold) a card then we assume they do not forget this information, i.e.

$$K_i x_k \Rightarrow \bigcirc K_i x_k \text{ or } K_i \neg x_k \Rightarrow \bigcirc K_i \neg x_k$$

where $i, k \in \{c, w, j\}$ and $x \in \{r, g, y, b, l, s, v, p\}$. That is *if a player knows that someone (doesn't) holds (hold) a card then in the next moment they know that person (doesn't) hold that card*, i.e. players don't forget knowledge relating to holding cards.

Common Knowledge As noted in Section 3 the KL_n logic does not include operators for common knowledge (everyone knows that everyone knows that ...etc). We can write axioms such as *one of the suspects must be the murderer* (see "At The start of the Game") directly into the required normal form (see Section 5.1) as

$$\textbf{true} \Rightarrow (r_m \vee g_m \vee y_m \vee b_m).$$

This captures the fact that $(r_m \vee g_m \vee y_m \vee b_m)$ holds at each moment in time, it is known by each agent, each agent know this etc (i.e. it is common knowledge).

Other statements for example "at time two Wendy answers she doesn't hold Miss Scarlett or the lead piping". We would like to write something like $t_2 \Rightarrow C(\neg r_w \wedge l_w)$ where t_2 is a new proposition which holds at time two and C is the common knowledge operator. Thus instead of C we must explicitly state (one or more) knowledge operators obtaining clauses 44 and 45 in Section 6. That is we must explicitly state the depth of modal operators we require.

5 Resolution for Temporal Logics of Knowledge

The resolution calculus is clausal requiring a translation to a normal form to separate modal and temporal components and to put formulae in a particular form. A set of temporal resolution rules applied to the temporal part and modal resolution rules are applied to the modal part. Information is carried between the two components using clauses containing literals. Full details of resolution based proof methods for temporal logics of knowledge are given in [5, 6].

5.1 Normal Form

Formulae in KL_n can be transformed into a normal form SNF_K (Separated Normal Form for temporal logics of knowledge). For the purposes of the normal form we introduce a symbol **start** such that $\langle M, (t_0, 0)\rangle \models \textbf{start}$. This is not necessary but allows the normal form to be implications. An alternative would be to let initial clauses (see Figure 1) be a disjunction of literals. The translation to SNF_K removes many of the temporal operators that do not appear in the normal form by rewriting using their fixpoint definitions. Also the translation uses the renaming technique [22] 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. To achieve this we introduce the $\square^*$ operator, which allows nesting of K_i and $\square$ operators. The $\square^*$ operator is defined in terms of the C (or *common knowledge*) and E (or *everybody knows*) operators. Finally, the $\square^*$ operator is defined as the maximal fixpoint of $\square^*\phi \Leftrightarrow \square(\phi \wedge C\, \square^*\phi)$.

Thus we reason about reachable points from the initial point in the distinguished timeline t_0 (where **start** is satisfiable), i.e. the points we require in the proof.

5.1.1 Definition of the Normal Form

Formulae in SNF_K are of the general form

$$\square^* \bigwedge_j T_j$$

where each T_j, known as a *clause*, must be in one of the varieties described in Figure 1 where

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

Figure 1: Clauses in SNF_K

k_a, l_b, and l are literals and m_{ib} are either literals, or modal literals involving the K_i operator. Thus a K_i clause (also known as a modal clause) may not contain modal literals $K_i l_1$ and $K_j l_2$ (or $K_i l_1$ and $\neg K_j l_2$) where $i \neq j$. Each K_i clause involves literals, or modal literals involving the K_i operator where at least one of the disjuncts is a modal literal. The outer '$\square^*$' 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.

To apply the temporal resolution rule (see Section 5.2), one or more step clauses may need to be combined. Consequently, a variant on SNF_K called *merged-SNF$_K$)* [10], is also defined. Given a set of step clauses in SNF_K, any step clause in SNF_K is also a clause in SNF_K. Any literal clause of the form **true** $\Rightarrow F$ is written into a merged-SNF_K clause as **true** $\Rightarrow \bigcirc F$. Any two merged-SNF_K clauses may be combined to produce a merged-SNF_K clause as follows

$$
\begin{array}{rcl}
A & \Rightarrow & \bigcirc C \\
B & \Rightarrow & \bigcirc D \\
\hline
(A \wedge B) & \Rightarrow & \bigcirc (C \wedge D)
\end{array}
$$

where A and B are conjunctions of literals and C and D are conjunctions of disjunctions of literals.

5.1.2 Translation to Normal Form

The translation to SNF_K is carried out by renaming complex subformulae with new propositional variables and linking the truth of the subformula to that of the proposition at all moments. Temporal operators are removed using their fixpoint definitions. Classical and temporal equivalences (see for example [8]) are also used to get formulae into the correct format. See [6, 11] for more details.

5.2 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. So for example the step clause $a \Rightarrow \bigcirc (b \vee b \vee c)$ is automatically rewritten as $a \Rightarrow \bigcirc (b \vee c)$.

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

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

Modal Resolution During modal resolution we apply the following rules which are based on the modal resolution system in [20]. In the following we may only resolve two K_i clauses together if they relate to the same i, i.e. we may not resolve a clause containing K_1 with a clause containing K_2. We may resolve a literal or modal literal and its negation or the formulae $K_i l$ and $K_i \neg l$ as we cannot both know something and know its negation.

$$[\text{MRES1}]\quad \frac{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor m\\ \textbf{true} & \Rightarrow & D' \lor \neg m\end{array}}{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor D'\end{array}} \qquad [\text{MRES2}]\quad \frac{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor K_i l\\ \textbf{true} & \Rightarrow & D' \lor K_i \neg l\end{array}}{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor D'\end{array}}$$

Next, as we have the T axiom, $\vdash K_i p \Rightarrow p$, we can resolve formulae such as $K_i l$ with $\neg l$ (giving MRES3). The rule MRES4 requires the function $\text{mod}_i(D')$, defined below, and is justified due to the external K_i operator surrounding each clause (due to the $\square^*$ operator, i.e, we distribute K_i into the second clause and resolve $\neg K_i l$ with $K_i l$. The "mod_i" function ensures that during this distribution at most one K_i or $\neg K_i$ operator applies to each literal due to the equivalences $\neg K_i \neg K_i \varphi \Leftrightarrow K_i \varphi$ and $\neg K_i K_i \varphi \Leftrightarrow \neg K_i \varphi$ in S5.

$$[\text{MRES3}]\quad \frac{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor K_i l\\ \textbf{true} & \Rightarrow & D' \lor \neg l\end{array}}{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor D'\end{array}} \qquad [\text{MRES4}]\quad \frac{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor \neg K_i l\\ \textbf{true} & \Rightarrow & D' \lor l\end{array}}{\begin{array}{rcl}\textbf{true} & \Rightarrow & D \lor \text{mod}_i(D')\end{array}}$$

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

$$\begin{array}{rcl}\text{mod}_i(A \lor B) & = & \text{mod}_i(A) \lor \text{mod}_i(B)\\ \text{mod}_i(l) & = & \neg K_i \neg l\end{array} \qquad \begin{array}{rcl}\text{mod}_i(K_i l) & = & K_i l\\ \text{mod}_i(\neg K_i l) & = & \neg K_i l\end{array}$$

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

$$[\text{MRES5}]\quad \frac{\begin{array}{rcl}\textbf{true} & \Rightarrow & L \lor K_i l_1 \lor K_i l_2 \lor \ldots\end{array}}{\begin{array}{rcl}\textbf{true} & \Rightarrow & L \lor l_1 \lor l_2 \lor \ldots\end{array}}$$

Here, L is a disjunction of literals.

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.

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

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

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

$$\text{[SRES3]} \qquad \frac{P \;\Rightarrow\; \bigcirc\,\mathbf{false}}{\mathbf{true} \;\Rightarrow\; \neg P}$$

This rule states that if, by satisfying P, a contradiction is produced, then P must never be satisfied in *any* moment. 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 **false** is derived in the form of either **start** $\Rightarrow$ **false** or **true** $\Rightarrow$ **false**.

Temporal Resolution The temporal resolution rule is as follows, where we resolve a sometime clause, $Q \Rightarrow \Diamond l$, with a condition, $\bigvee_{k=0}^{n} A_k$, that implies $\square\neg l$ in the next moment (known as a *loop formula for* $\neg l$).

$$\text{[TRES]} \qquad \frac{\begin{array}{rcl} \bigvee_{k=0}^{n} A_k & \Rightarrow & \bigcirc\,\square\neg l \\ Q & \Rightarrow & \Diamond l \end{array}}{Q \;\Rightarrow\; (\bigwedge_{i=0}^{n} \neg A_i)\,\mathcal{W}\,l}$$

This resolvent states that once Q is satisfied then none of the A_i should be satisfied unless l is satisfied. A systematic way of deriving $\bigvee_{k=0}^{n} A_k$ from the set of step clauses such that

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

is described in [3] but is beyond the scope of this paper.

Translating the resolvent into SNF_K we obtain the following clauses for each i where w_l is a new proposition.

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

5.3 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 clauses ψ_s.
2. Perform modal and step resolution (including simplification and subsumption) until either
   (a) **true** $\Rightarrow$ **false** is derived - terminate noting ψ unsatisfiable; or
   (b) no new resolvents are generated - continue to step 3.
3. Select an eventuality from the right hand side of a sometime clause within ψ_s, for example $\Diamond l$. Search for loop formulae in $\neg l$ and generate the appropriate resolvents. If no new formulae have been generated try the next sometime clause otherwise (if new formulae have been generated) go to step 4. If there are no eventualities for which new resolvents can be derived, go to step 5.
4. Add the new resolvents to the clause-set and perform initial resolution until either
   (a) **start** $\Rightarrow$ **false** is derived - terminate noting ψ unsatisfiable; or

(b) no new resolvents are generated - continue at step 2.

5. Perform initial resolution until either

   (a) **start** $\Rightarrow$ **false** is derived - terminate noting ψ unsatisfiable; or

   (b) no new resolvents are generated -terminate declaring ψ satisfiable.

5.4 Correctness

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

Theorem 1 *A KL_n formula A is satisfiable if, and only if, $\tau_0[A]$ is satisfiable (where τ_0 is the translation into SNF_K).*

Proofs analogous to those in [6, 11] will suffice.

Theorem 2 *(Soundness) Let S be a satisfiable set of SNF_K rules and T be the set of rules obtained from S by an application of one of the resolution rules. Then T is also satisfiable.*

This can be shown by showing an application of each resolution rule preserves satisfiability. (see [6])

Theorem 3 *(Completeness) If a set of SNF_K rules is unsatisfiable then it has a refutation by the temporal resolution procedure given in this paper.*

This is carried out by constructing a graph to represent all possible models for the set of rules. Deletions in the graph represent the application of of the resolution rules. An empty graph corresponds with the generation of false (see [6]).

6 Verification Using Clausal Resolution

First we write the information held prior to the deal as clauses. As they are axioms we write directly in the normal form (and hold at all moments in time).

$$
\begin{array}{rlcl}
1. & \textbf{true} & \Rightarrow & (r_m \lor g_m \lor y_m \lor b_m) \\
2. & \textbf{true} & \Rightarrow & (l_m \lor s_m \lor v_m \lor p_m) \\
3. & \textbf{true} & \Rightarrow & (r_c \lor r_w \lor r_j \lor r_m) \\
4. & \textbf{true} & \Rightarrow & (g_c \lor g_w \lor g_j \lor g_m) \\
5. & \textbf{true} & \Rightarrow & (y_c \lor y_w \lor y_j \lor y_m) \\
6. & \textbf{true} & \Rightarrow & (b_c \lor b_w \lor b_j \lor b_m) \\
7. & \textbf{true} & \Rightarrow & (l_c \lor l_w \lor l_j \lor l_m) \\
8. & \textbf{true} & \Rightarrow & (s_c \lor s_w \lor s_j \lor s_m) \\
9. & \textbf{true} & \Rightarrow & (v_c \lor v_w \lor v_j \lor v_m) \\
10. & \textbf{true} & \Rightarrow & (p_c \lor p_w \lor p_j \lor p_m) \\
11. & \textbf{true} & \Rightarrow & (\neg r_c \lor \neg r_w) \\
12. & \textbf{true} & \Rightarrow & (\neg r_c \lor \neg r_j) \\
13. & \textbf{true} & \Rightarrow & (\neg r_c \lor \neg r_m) \\
14. & \textbf{true} & \Rightarrow & (\neg r_w \lor \neg r_j) \\
15. & \textbf{true} & \Rightarrow & (\neg r_w \lor \neg r_m) \\
16. & \textbf{true} & \Rightarrow & (\neg r_j \lor \neg r_m)
\end{array}
$$

where there are additional versions of 11–16 for each card. In the following proofs if we use a version of one of these clauses for another card eg the g version of 13, i.e. **true** $\Rightarrow (\neg g_c \lor \neg g_m)$

we denote this as $13(g)$. These come from translating the axioms from the *before the deal* section into SNF_K.

We assume that at time one the following deal has been made.

Player	Catherine	Wendy	Jane	MurderHand
Cards	Miss Scarlett Rev Green	Revolver Rope	Col. Mustard Spanner	Lead Piping Prof. Plum

Thus from the above the following holds

$$\begin{array}{lll}\textbf{start} & \Rightarrow & \bigcirc(K_c r_c \wedge K_c g_c \wedge K_c\neg y_c \wedge K_c\neg b_c \wedge K_c\neg l_c \wedge K_c\neg s_c \wedge K_c\neg v_c \wedge K_c\neg p_c \\ & & \wedge K_w\neg r_w \wedge K_w\neg g_w \wedge K_w\neg y_w \wedge K_w\neg b_w \wedge K_w\neg l_w \wedge K_w\neg s_w \wedge K_w v_w \wedge K_w p_w \\ & & \wedge K_j\neg r_j \wedge K_j\neg g_j \wedge K_j y_j \wedge K_j\neg b_j \wedge K_j\neg l_j \wedge K_j s_j \wedge K_j\neg v_j \wedge K_j\neg p_j)\end{array}$$

Writing this into the normal form we obtain,

$$\begin{array}{llll@{\qquad}llll}
17. & \textbf{start} & \Rightarrow & t_0 & 30. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w\neg b_w) \\
18. & t_0 & \Rightarrow & \bigcirc t_1 & 31. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w\neg l_w) \\
19. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c r_c) & 32. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w\neg s_w) \\
20. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c g_c) & 33. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w v_w) \\
21. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c\neg y_c) & 34. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w p_w) \\
22. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c\neg b_c) & 35. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j\neg r_j) \\
23. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c\neg l_c) & 36. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j\neg g_j) \\
24. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c\neg s_c) & 37. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j y_j) \\
25. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c\neg v_c) & 38. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j\neg b_j) \\
26. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_c\neg p_c) & 39. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j\neg l_j) \\
27. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w\neg r_w) & 40. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j s_j) \\
28. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w\neg g_w) & 41. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j\neg v_j) \\
29. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_w\neg y_w) & 42. & \textbf{true} & \Rightarrow & (\neg t_1 \vee K_j\neg p_j)
\end{array}$$

where t_0 and t_1 are new propositional variables that hold at time 0 and time 1 respectively. At this point we should be able to prove, for example, that at time one Catherine knows that Miss Scarlett is not the murderer. This is symbolised as

$$\bigcirc K_c\neg r_m$$

Thus the above clauses in addition to the clausal form of

$$\neg\bigcirc K_c\neg r_m$$

should give us a contradiction. To make the problem smaller we reuse the new propositions t_0 and t_1 (and clauses 17,18). Alternatively we could add two additional new propositions and two clauses identical to 17 and 18 with these new propositions. We obtain

$$C1.\quad \neg t_1 \vee \neg K_c\neg r_m$$

and using resolution obtain a contradiction as follows.

$$\begin{array}{lrcll}
C2. & \textbf{true} & \Rightarrow & \neg t_1 \vee \neg K_c r_c & [C1, 13\ MRES4] \\
C3. & \textbf{true} & \Rightarrow & \neg t_1 & [C2, 19\ MRES1] \\
C4. & t_0 & \Rightarrow & \bigcirc\textbf{false} & [C3, 18\ SRES2] \\
C5. & \textbf{true} & \Rightarrow & \neg t_0 & [C4\ SRES3] \\
C6. & \textbf{start} & \Rightarrow & \textbf{false} & [C5, 17\ IRES1]
\end{array}$$

That is if Φ is the conjunction of clauses 1–42 then $\vdash \Phi \Rightarrow \bigcirc K_c\neg r_m$.

Let us assume (at time two) that Catherine makes the suggestion "Miss Scarlett and the lead piping". Therefore after consulting her cards Wendy answers no, i.e. we add the formulae

$$\bigcirc\bigcirc(K_i\neg r_w \wedge K_i\neg l_w)$$

for $i \in \{c,j,w\}$ Trying to answer the suggestion then passes to Jane. She also answers no. Thus we may add the formulae

$$\bigcirc\bigcirc(K_i\neg r_j \wedge K_i\neg l_j)$$

for $i \in \{c,j,w\}$. Again reusing t_0, t_1 and clauses 17, 18 instead of adding new propositions to denote times zero and one, we obtain the following where t_2 is a new proposition.

$$\begin{array}{rlcl}
43. & t_1 & \Rightarrow & \bigcirc t_2 \\
44. & \mathbf{true} & \Rightarrow & \neg t_2 \vee K_i\neg r_w \\
45. & \mathbf{true} & \Rightarrow & \neg t_2 \vee K_i\neg l_w \\
46. & \mathbf{true} & \Rightarrow & \neg t_2 \vee K_i\neg r_j \\
47. & \mathbf{true} & \Rightarrow & \neg t_2 \vee K_i\neg l_j
\end{array}$$

As both Wendy and Jane have neither of these cards and Catherine doesn't make an accusation they (and Catherine) know Catherine must hold one of these cards i.e.

$$\bigcirc\bigcirc(K_i(r_c \vee l_c))$$

for $i \in \{c,j,w\}$. This is written into normal form as

$$\begin{array}{rlcl}
48. & \mathbf{true} & \Rightarrow & \neg t_2 \vee K_i n_0 \\
49. & \mathbf{true} & \Rightarrow & \neg n_0 \vee r_c \vee l_c
\end{array}$$

where n_0 is a new proposition and we reuse t_2 as before.

At this point Catherine should also be able to deduce that the lead piping is the murder weapon, i.e. $\bigcirc\bigcirc K_c l_m$. We negate and prove this via resolution (reusing propositions t_0, t_1, t_2).

$$\begin{array}{rlcll}
C7. & \mathbf{true} & \Rightarrow & \neg t_2 \vee \neg K_c l_m & \\
C8. & \mathbf{true} & \Rightarrow & \neg t_2 \vee \neg K_c\neg l_c \vee \neg K_c\neg l_w \vee \neg K_c\neg l_j & [C7, 7\ MRES4] \\
C9. & \mathbf{true} & \Rightarrow & \neg t_2 \vee \neg K_c\neg l_c \vee \neg K_c\neg l_j & [C8, 45(c)\ MRES1] \\
C10. & \mathbf{true} & \Rightarrow & \neg t_2 \vee \neg K_c\neg l_c & [C9, 47(c)\ MRES1]
\end{array}$$

To complete the proof we need to use that axiom (see dealing with time)

$$K_c\neg l_c \Rightarrow \bigcirc K_c\neg l_c$$

We rewrite into the normal form by first writing as a disjunction and then renaming $\neg K_c\neg l_c$ by $\neg a$ and then $K_c\neg l_c$ by b.

$$\begin{array}{c}
\neg K_c\neg l_c \vee \bigcirc K_c\neg l_c \\
\neg K_c\neg l_c \vee a \\
a \Rightarrow \bigcirc d \\
\neg d \vee K_c\neg l_c
\end{array}$$

resulting in the following clauses

$$\begin{array}{rlcl}
50. & \mathbf{true} & \Rightarrow & \neg K_c\neg l_c \vee a \\
51. & a & \Rightarrow & \bigcirc d \\
52. & \mathbf{true} & \Rightarrow & \neg d \vee K_c\neg l_c
\end{array}$$

and the remainder of the proof being as follows.

$$\begin{array}{lrcll}
C11. & \mathbf{true} & \Rightarrow & \neg t_2 \lor \neg d & [C10, 52\ MRES1] \\
C12. & a & \Rightarrow & \bigcirc \neg t_2 & [51, C11\ SRES2] \\
C13. & a \land t_1 & \Rightarrow & \bigcirc \mathbf{false} & [43, C12\ SRES1] \\
C14. & \mathbf{true} & \Rightarrow & \neg a \lor \neg t_1 & [C13\ SRES3] \\
C15. & \mathbf{true} & \Rightarrow & \neg K_c \neg l_c \lor \neg t_1 & [50, C14\ MRES1] \\
C16. & \mathbf{true} & \Rightarrow & \neg t_1 & [23, C15\ MRES1] \\
C17. & t_0 & \Rightarrow & \bigcirc \mathbf{false} & [18, C16\ SRES2] \\
C18. & \mathbf{true} & \Rightarrow & \neg t_0 & [C17\ SRES3] \\
C19. & \mathbf{start} & \Rightarrow & \mathbf{false} & [17, C18\ IRES1]
\end{array}$$

Next Wendy makes an suggestion. At time three Wendy suggests lead piping and Col. Mustard. Jane shows Wendy Col. Mustard. Thus at time three Wendy knows Jane holds Col. Mustard $\bigcirc\bigcirc\bigcirc K_w y_j$ and Catherine knows that Jane holds either the lead piping or Col. Mustard $\bigcirc\bigcirc\bigcirc K_c(l_j \lor y_j)$. We write these statements into normal form (reusing the propositions t_0, t_1, t_2) where n_1 is a new proposition.

$$\begin{array}{lrcl}
53. & t_2 & \Rightarrow & \bigcirc t_3 \\
54. & \mathbf{true} & \Rightarrow & \neg t_3 \lor K_w y_j \\
55. & \mathbf{true} & \Rightarrow & \neg t_3 \lor K_c n_1 \\
56. & \mathbf{true} & \Rightarrow & \neg n_1 \lor l_j \lor y_j
\end{array}$$

At this point Catherine can deduce both the murderer and the murder weapon i.e. $\bigcirc\bigcirc\bigcirc K_c(l_m \land b_m)$. We shall again prove this with resolution. We negate (obtaining clauses C20 and C21) and prove this via resolution (reusing propositions t_0, t_1, t_2 and t_3) where n_2 is a new proposition.

$$\begin{array}{lrcll}
C20. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg n_2 & \\
C21. & \mathbf{true} & \Rightarrow & \neg n_2 \lor \neg l_m \lor \neg b_m & \\
C22. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c l_m \lor \neg K_c b_m & [C20, C21\ MRES4] \\
C23. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j \lor \neg K_c b_m & [C22, 7\ MRES4] \\
C24. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg K_c \neg r_m \lor \neg K_c \neg g_m \lor \neg K_c \neg y_m & [C23, 1\ MRES4] \\
C25. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg K_c r_c \lor \neg K_c \neg g_m \lor \neg K_c \neg y_m & [C24, 13\ MRES4] \\
C26. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg K_c r_c \lor \neg K_c g_c \lor \neg K_c \neg y_m & [C25, 13(g)\ MRES4] \\
C27. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg K_c r_c \lor \neg K_c g_c \lor \neg K_c y_j & [C26, 16(y)\ MRES4] \\
C28. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg K_c r_c \lor \neg K_c g_c \lor \neg K_c n_1 & [C27, 56\ MRES4] \\
C29. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg K_c r_c \lor \neg K_c g_c & [C28, 55\ MRES4]
\end{array}$$

In the proof we need clauses (see dealing with time) from the normal form of the following.

$$\begin{array}{c}
K_c \neg l_w \Rightarrow \bigcirc K_c \neg l_w \\
K_c \neg l_j \Rightarrow \bigcirc K_c \neg l_j \\
K_c r_c \Rightarrow \bigcirc K_c r_c \\
K_c g_c \Rightarrow \bigcirc K_c g_c
\end{array}$$

Following how we obtained clauses 50–52 we obtain the following.

$$
\begin{array}{rrcl}
57. & \mathbf{true} & \Rightarrow & \neg K_c \neg l_w \lor a_1 \\
58. & a_1 & \Rightarrow & \bigcirc d_1 \\
59. & \mathbf{true} & \Rightarrow & \neg d_1 \lor K_c \neg l_w \\
60. & \mathbf{true} & \Rightarrow & \neg K_c \neg l_j \lor a_2 \\
61. & a_2 & \Rightarrow & \bigcirc d_2 \\
62. & \mathbf{true} & \Rightarrow & \neg d_2 \lor K_c \neg l_j \\
63. & \mathbf{true} & \Rightarrow & \neg K_c r_c \lor a_3 \\
64. & a_3 & \Rightarrow & \bigcirc d_3 \\
65. & \mathbf{true} & \Rightarrow & \neg d_3 \lor K_c r_c \\
66. & \mathbf{true} & \Rightarrow & \neg K_c g_c \lor a_4 \\
67. & a_4 & \Rightarrow & \bigcirc d_4 \\
68. & \mathbf{true} & \Rightarrow & \neg d_4 \lor K_c g_c
\end{array}
$$

The proof continues as follows.

$$
\begin{array}{lrcll}
C30. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg K_c r_c \lor \neg d_4 & [C29, 68\ MRES1] \\
C31. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j & \\
 & & & \quad \lor \neg d_3 \lor \neg d_4 & [C30, 65\ MRES1] \\
C32. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg K_c \neg l_w \lor \neg d_2 & \\
 & & & \quad \lor \neg d_3 \lor \neg d_4 & [C31, 62\ MRES1] \\
C33. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg K_c \neg l_c \lor \neg d_1 \lor \neg d_2 & \\
 & & & \quad \lor \neg d_3 \lor \neg d_4 & [C32, 59\ MRES1] \\
C34. & \mathbf{true} & \Rightarrow & \neg t_3 \lor \neg d \lor \neg d_1 \lor \neg d_2 & \\
 & & & \quad \lor \neg d_3 \lor \neg d_4 & [C33, 52\ MRES1] \\
C35. & t_2 & \Rightarrow & \bigcirc(\neg d \lor \neg d_1 \lor \neg d_2 \lor \neg d_3 \lor \neg d_4) & [C34, 53\ SRES2] \\
C36. & (t_2 \land a) & \Rightarrow & \bigcirc(\neg d_1 \lor \neg d_2 \lor \neg d_3 \lor \neg d_4) & [C35, 51\ SRES1] \\
C37. & (t_2 \land a \land a_1) & \Rightarrow & \bigcirc(\neg d_2 \lor \neg d_3 \lor \neg d_4) & [C36, 58\ SRES1] \\
C38. & (t_2 \land a \land a_1 \land a_2) & \Rightarrow & \bigcirc(\neg d_3 \lor \neg d_4) & [C37, 61\ SRES1] \\
C39. & (t_2 \land a \land a_1 \land a_2 \land a_3) & \Rightarrow & \bigcirc \neg d_4 & [C38, 64\ SRES1] \\
C40. & (t_2 \land a \land a_1 \land a_2 \land a_3 \land a_4) & \Rightarrow & \bigcirc \mathbf{false} & [C39, 67\ SRES1] \\
C41. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg a \lor \neg a_1 \lor \neg a_2 \lor \neg a_3 \lor & \\
 & & & \neg a_4) & [C40\ SRES3] \\
C42. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg a \lor \neg K_c \neg l_w \lor \neg a_2 & \\
 & & & \lor \neg a_3 \lor \neg a_4) & [C41, 57\ MRES1] \\
C43. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg a \lor \neg K_c \neg l_w \lor \neg K_c \neg l_j \lor & \\
 & & & \neg a_3 \lor \neg a_4) & [C42, 60\ MRES1] \\
C44. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg a \lor \neg K_c \neg l_w \lor \neg a_3 \lor \neg a_4) & [C43, 47\ MRES1] \\
C45. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg a \lor \neg a_3 \lor \neg a_4) & [C44, 45\ MRES1] \\
C46. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg K_c \neg l_c \lor \neg a_3 \lor \neg a_4) & [C45, 50\ MRES1] \\
C47. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg K_c \neg l_c \lor \neg K_c r_c \lor \neg a_4) & [C46, 63\ MRES1] \\
C48. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg K_c \neg l_c \lor \neg K_c r_c \lor \neg K_c g_c) & [C47, 66\ MRES1] \\
C49. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg K_c \neg l_c \lor \neg K_c r_c \lor \neg d_4) & [C48, 68\ MRES1] \\
C50. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg K_c \neg l_c \lor \neg d_3 \lor \neg d_4) & [C49, 65\ MRES1] \\
C51. & \mathbf{true} & \Rightarrow & (\neg t_2 \lor \neg d \lor \neg d_3 \lor \neg d_4) & [C50, 52\ MRES1] \\
C52. & t_1 & \Rightarrow & \bigcirc(\neg d \lor \neg d_3 \lor \neg d_4) & [C51, 43\ SRES2] \\
C53. & (t_1 \land a) & \Rightarrow & \bigcirc(\neg d_3 \lor \neg d_4) & [C52, 51\ SRES1] \\
C54. & (t_1 \land a \land a_3) & \Rightarrow & \bigcirc \neg d_4 & [C53, 64\ SRES1] \\
C55. & (t_1 \land a \land a_3 \land a_4) & \Rightarrow & \bigcirc \mathbf{false} & [C54, 67\ SRES1] \\
C56. & \mathbf{true} & \Rightarrow & (\neg t_1 \lor \neg a \lor \neg a_3 \lor \neg a_4) & [C55\ SRES3]
\end{array}
$$

$C57.$	**true**	$\Rightarrow$	$(\neg t_1 \vee \neg K_c \neg l_c \vee \neg a_3 \vee \neg a_4)$	$[C56, 50\ MRES1]$
$C58.$	**true**	$\Rightarrow$	$(\neg t_1 \vee \neg a_3 \vee \neg a_4)$	$[C57, 23\ MRES1]$
$C59.$	**true**	$\Rightarrow$	$(\neg t_1 \vee \neg K_c r_c \vee \neg a_4)$	$[C58, 63\ MRES1]$
$C60.$	**true**	$\Rightarrow$	$(\neg t_1 \vee \neg a_4)$	$[C59, 19\ MRES1]$
$C61.$	**true**	$\Rightarrow$	$(\neg t_1 \vee \neg K_c g_c)$	$[C60, 66\ MRES1]$
$C62.$	**true**	$\Rightarrow$	$\neg t_1$	$[C61, 20\ MRES1]$
$C63.$	t_0	$\Rightarrow$	$\bigcirc$**false**	$[C62, 18\ SRES2]$
$C64.$	**true**	$\Rightarrow$	$\neg t_0$	$[C63\ SRES3]$
$C65.$	**start**	$\Rightarrow$	**false**	$[C64, 17\ IRES1]$

It is Jane's turn next. Jane can only stop Catherine winning if she makes the (correct) suggestion "lead piping and Professor Plum" to which both Catherine and Wendy say no. In this case we can deduce that $\bigcirc\bigcirc\bigcirc\bigcirc K_j(l_m \wedge b_m)$ similarly to the above, i.e. Jane wins the game and can make the correct accusation "lead piping and Professor Plum". If Jane does not use this suggestion whilst she may gain additional information about the whereabouts of one or more cards she cannot correctly deduce the murderer and murder weapon.

The turn then passes to Catherine. As at time three Catherine knew the murderer and murder weapon Catherine will still know this at time five and can make the relevant accusation, i.e. we could also prove that $\bigcirc\bigcirc\bigcirc\bigcirc\bigcirc K_c(l_m \wedge b_m)$.

Finally, to give a simple illustration of the use of the temporal resolution rule (TRES) we prove that from time one onwards Catherine knows that Miss Scarlett is not the murderer, i.e.

$$\bigcirc \Box K_C \neg r_m$$

Again we negate and write

$$\bigcirc \Diamond \neg K_C \neg r_m$$

into normal form. As previously we reuse the propositions t_0 and t_1 to obtain the following where n_3 is a new proposition and apply resolution to these clauses.

$C66.$	t_1	$\Rightarrow$	$\Diamond n_3$	
$C67.$	**true**	$\Rightarrow$	$\neg n_3 \vee \neg K_c \neg r_m$	
$C68.$	**true**	$\Rightarrow$	$\neg n_3 \vee \neg K_c r_c$	$[C67, 13\ MRES4]$
$C69.$	**true**	$\Rightarrow$	$\neg n_3 \vee \neg d_3$	$[C68, 65\ MRES1]$
$C70.$	**true**	$\Rightarrow$	$\neg d_3 \vee a_3$	$[63, 65\ MRES1]$
$C71.$	a_3	$\Rightarrow$	$\bigcirc \neg n_3$	$[C69, 64\ SRES2]$
$C72.$	a_3	$\Rightarrow$	$\bigcirc a_3$	$[C70, 64\ SRES2]$

Clauses $C71$ and $C72$ can be used to apply the temporal resolution rule as they give the loop formula

$$a_3 \Rightarrow \bigcirc \Box \neg n_3.$$

When resolved with clause C66 we obtain the resolvent

$$t_1 \Rightarrow \neg a_3 \mathcal{W} n_3$$

(not in normal form) which is translated into normal form as follows where w_{n_3} is a new proposition.

$C73.$	**true**	$\Rightarrow$	$\neg t_1 \vee n_3 \vee \neg a_3$
$C74.$	**true**	$\Rightarrow$	$\neg t_1 \vee n_3 \vee w_{n_3}$
$C75.$	w_{n_3}	$\Rightarrow$	$\bigcirc(n_3 \vee \neg a_3)$
$C76.$	w_{n_3}	$\Rightarrow$	$\bigcirc(n_3 \vee w_{n_3})$

The proof continues as follows

$$
\begin{array}{llcll}
C77. & \textbf{true} & \Rightarrow & \neg t_1 \lor \neg K_c r_c \lor \neg a_3 & [C73, C68\ MRES1] \\
C78. & \textbf{true} & \Rightarrow & \neg t_1 \lor \neg K_c r_c & [C77, 63\ MRES1] \\
C79. & \textbf{true} & \Rightarrow & \neg t_1 & [C78, 19\ MRES1] \\
C80. & t_0 & \Rightarrow & \bigcirc \textbf{false} & [C79, 18\ SRES2] \\
C81. & \textbf{true} & \Rightarrow & \neg t_0 & [C80\ SRES3] \\
C82. & \textbf{start} & \Rightarrow & \textbf{false} & [C81, 17\ IRES1]
\end{array}
$$

7 Implementation

7.1 Cluedo Players Assistants

A system showing the knowledge of each players cards from the perspective of a particular player (the user) has been developed in [2]. As described in this paper the dice and position on the board are not implemented. The system is programmed in Java, has a graphical user interface and allows the input of suggestions, cards shown etc via choice boxes and buttons. The user's knowledge relating to the cards of other players are displayed in windows relating to that player. For example if the user knows that player two holds Miss Scarlett or the rope or the ballroom from player two showing a card in response to the suggestion "Scarlett with the rope in the ballroom" from another player, "scarlet or rope or ballroom" is added to the knowledge of player two's cards. Similarly if the card hall is shown to the user by player three then "hall" is added to the window for the user's knowledge about player three's cards. Players declaring they don't know any of the three cards in a suggestion will result in the negation of those cards being added to that player's window. New knowledge is inferred by resolution between statements within a particular window (representing the user's knowledge about a player). Further, inferences between players knowledge is carried out by inferring if (the user knows that) a player holds a particular card then other players do not.

A window displays the remaining triples that could be the murderer, murder weapon and room. This is updated as suggestions are made and answered. For example in with the suggestion above that player two has shown a card to another player the triple "scarlet and rope and ballroom" can be removed from the remaining murder triple possibilities.

Similarly a player's assistant has also been developed in [1]. This system uses slightly different rules to the standard rule set. Information is held and updated relating to the remaining murderer, murder weapon, and room triples; cards known to be held by other players; and cards known by players not to be the murderer, murder weapon and location. Information about knowledge about knowledge is also held. The system has a text-based user interface.

7.2 Resolution Theorem Prover for Temporal Logics of Knowledge

A prototype resolution based theorem prover for the single agent temporal logic of knowledge has been developed in Liverpool. Implemented in Prolog this prototype prover is based on the resolution calculus described in Section 5. This needs expanding to the multi-agent case before it can be used to automatically prove the examples in Section 6.

8 Conclusions and Related Work

8.1 Related Work

The properties of, complexity and axiomatisations for temporal logics of knowledge have been studied, for example, in [9, 16]. As well as dealing with fusions of the logics of knowledge and time, i.e. they do not allow (non-trivial) axioms referring to operators of both knowledge and time these papers discuss systems with interactions. For example systems with synchrony and perfect recall are considered which allow an axiom of the form $K \bigcirc \varphi \Rightarrow \bigcirc K\varphi$. Such axioms allow the change in knowledge over time. In this paper, we specify axioms relating to the knowledge of holding cards, stating that *if an agent knows a player does (doesn't) hold a card then in the next moment the agent knows this.* As agents don't forget knowledge about cards we are really talking about a system of synchrony and perfect recall. Thus if we allow the synchrony and perfect recall axiom we could re-state the axioms about not forgetting knowledge of cards as *if a player knows that someone (doesn't) holds (hold) a card then they know that in the next moment that person (doesn't) holds (hold) that card.* Applying the synchrony and perfect recall axiom to the right hand side would give us the original axiom. Resolution and tableau based proof methods for systems allowing such interactions are given in [4, 21, 7]. Note this greatly increases the complexity of the validity problem for the logic which is PSPACE for KL_n but non-elementary time for the multi-agent case of synchrony and perfect recall [9, 16].

Combinations of modal and temporal logics have been used to specify complex situations. For example BDI logics [23, 24], the fusion of either linear or branching-time (CTL or CTL*) with the modal logics KD45 for belief, and KD for desire and intention, are used to specify properties multi-agent systems. Tableau based proof methods are also given for these logics. Also in the area of multi-agent systems proof methods for a core of the agent specification logic KARO (knowledge, actions, results opportunities) [26] a complex combination of dynamic logic, and the modal logics of knowledge and wishes (KD) has been outlined in [18].

The muddy children problem is a well known problem relating to reasoning about knowledge. The problem is specified using epistemic logics in [9] and is specified and verified using temporal logics of knowledge in [6].

The specification of Cluedo game actions has been carried out in [27] in a dynamic epistemic logic (the combination of dynamic logic and a logic of knowledge allowing common knowledge). Unsurprisingly the knowledge gained from moves in the game is the same as described in this paper except the common knowledge resulting from some moves can be explicitly stated. The focus of [27] is the specification of the knowledge actions rather than verification. The paper has no axiomatisation for the logic and decidability is not discussed.

8.2 Conclusions

This paper explores a case study relating to reasoning about knowledge over time. We have applied temporal logics of knowledge to specify and verify the game Cluedo where the knowledge of players forms an integral part of the game. We have shown how the various moves in Cluedo can be specified using temporal logics of knowledge. We have proved properties of this specification by applying a resolution based calculus for temporal logics of knowledge.

The knowledge we infer from showing a card has been limited to the depth of one modal operator, i.e. in the example above (at time two) when Catherine makes the suggestion "Miss Scarlett and the lead piping" and Wendy says no we add the clausal form of

$$\bigcirc \bigcirc (K_i \neg r_w \wedge K_i \neg l_w)$$

for $i \in \{c, j, w\}$ but do not consider for example

$$\bigcirc \bigcirc (K_k K_i \neg r_w)$$

for $k, j \in \{c, j, w\}$. If we want to make deductions about knowledge about knowledge these clauses will also be needed. An alternative would be to use a temporal logic of knowledge that allowed a common knowledge operator. The corresponding resolution calculus outlined in Section 5 would have to be extended to this logic. The complexity of validity is PSPACE for

KL_n whereas it is EXPTIME if we add common knowledge [9, 16]. Note if we tried to add common knowledge to the systems of synchrony and perfect recall, mentioned above, the logic becomes undecidable [9, 16].

The systems in Section 7 for example concentrate keeping track of the knowledge of who holds which cards. Enhancing these systems to suggest which card to reveal to an opponent or what to guess as your next suggestion would be a next step. That is to move towards a system that plays Cluedo rather than an player's assistant. A simple way to show what card to reveal to an opponent (so as to help her the least) might be to show a card that the user knows the opponent knows the user holds already. This requires reasoning about knowledge about knowledge and would require additional nesting of knowledge operators to be recorded.

References

[1] J. Bailiss. Knowledge and Cluedo, 2003. Final year project, Department of Computer Science, University of Liverpool.

[2] T. Demetriou. Knowledge and Cluedo, 2003. Final year project, Department of Computer Science, University of Liverpool.

[3] C. Dixon. Temporal Resolution using a Breadth-First Search Algorithm. *Annals of Mathematics and Artificial Intelligence*, 22:87–115, 1998.

[4] C. Dixon and M. Fisher. Clausal Resolution for Logics of Time and Knowledge with Synchrony and Perfect Recall. In H. Wansing and F. Wolter, editors, *Proceedings of ICTL-00 the Third International Conference on Temporal Logic*, pages 43–52, Leipzig, Germany, October 2000.

[5] 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*, pages 163–172, Cape Breton, Nova Scotia, Canada, July 2000. IEEE Press.

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

[7] C. Dixon, C. Nalon, and M. Fisher. Tableaux for Temporal Logics of Knowledge: Synchronous Systems of Perfect Recall or No Learning. In M. Reynolds and A. Sattar, editors, *Proceedings of the Tenth International Symposium on Temporal Representation and Reasoning and the Fourth International Conference on Temporal Logic (TIME-ICTL 2003)*, pages 62–71, Cairns, Australia, July 2003. IEEE Computer Society Press.

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

[9] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[10] 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.

[11] M. Fisher, C. Dixon, and M. Peim. Clausal Temporal Resolution. *ACM Transactions on Computational Logic*, 2(1):12–56, January 2001.

[12] 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.

[13] 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.

[14] J. Glasgow, G. MacEwen, and P.Panangaden. A Logic to Reason About Security. *ACM Transactions on Computer Systems*, 10(3):226–264, August 1992.

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

[16] 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.

[17] Cluedo. http://www.hasbro.com/.

[18] U. Hustadt, C. Dixon, R. A. Schmidt, M. Fisher, J.-J. Meyer, and W. van der Hoek. Reasoning about agents in the KARO framework. In C. Bettini and A. Montanari, editors, *Proceedings of the Eighth International Symposium on Temporal Representation and Reasoning (TIME-01)*, pages 206–213, Cividale del Friuli, Italy, June 2001. IEEE Press.

[19] J.-J. C. Meyer and W. van der Hoek. *Epistemic Logic for Computer Science and Artificial Intelligence*, volume 41 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1995.

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

[21] C. Nalon, C. Dixon, and M. Fisher. Resolution for synchrony and no learning: Preliminary report. In *Proceedings of the 8th International Workshop on Logic Language, Information and Computation - WoLLIC'2001*, 2001.

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

[23] A. S. Rao. Decision Procedures for Propositional Linear-Time Belief-Desire-Intention Logics. In M. Wooldridge, K. Fischer, P. Gmytrasiewicz, N. R. Jennings, J. P. Müller, and M. Tambe, editors, *IJCAI-95 Workshop on Agent Theories, Architectures, and Languages*, pages 102–118, Montréal, Canada, August 1995.

[24] A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–342, 1998.

[25] P. Syverson. Adding Time to a Logic of Authentication. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 97–101. ACM Press, 1993.

[26] W. van der Hoek, B. van Linder, and J.J. Ch. Meyer. An integrated modal approach to rational agents. Technical Report UU-CS-1997-06, Department of Computer Science,Utrecht University, Centrumgebouw Noord, Padualaan 14, De Uithof, 3584 CH Utrecht, 1997.

[27] H.P. van Ditmarsch. The description of game actions in Cluedo. *Game theory and Applications*, 8:1–28, 2002.