

Translating Temporal Epistemic Logics to Monodic First-Order Temporal Logic *

M. C. Fernández-Gago, C. Dixon, M. Fisher, U. Hustadt and B. Konev
Department of Computer Science, University of Liverpool L69 3BX
(`{m.c.gago,c.dixon,m.fisher,u.hustadt,b.konev}@csc.liv.ac.uk`)

Abstract. Temporal logics of knowledge are useful for dealing with certain situations where the knowledge of some agents in a system is involved. There exist proof methods for these systems, however due to the complexity of the systems they are not easy to implement. In this paper we present a translation from temporal logics of knowledge into the monodic fragment of first-order temporal logic in order to use an existent implementation of a theorem-prover for monodic first-order temporal logic called TeMP. Thus problems specified in temporal logics of knowledge could be tested automatically.

Keywords:

Abbreviations: FOTL – First-Order Temporal Logic; $KL_{(n)}$ – Temporal Logic of Knowledge; SNF_K – Separated Normal Form for temporal logics of knowledge

1. Introduction

Temporal logics have been shown useful in computer science and artificial intelligence in order to specify how a system changes over time. Logics of knowledge are useful in order to specify systems where the knowledge of an agent is important.

In this paper we use the logic $KL_{(n)}$. This logic is the fusion of linear time temporal logic with finite past and infinite future combined with the multi-modal logic S5. This logic have been used for the specification and verification of multi-agent systems (Fisher and Wooldridge, 1997; Halpern, 1987; Meyer and van der Hoek, 1995), security protocols (Dixon et al., 2003; Syverson, 1993) and some other games involving knowledge such as Cluedo (Dixon, 2004) or case studies as the muddy children (Dixon et al., 1998; Fagin et al., 1995). Some of these systems have been verified by hand using a resolution calculus for $KL_{(n)}$ (Dixon et al., 1998). However, we are interested in the mechanisation of this proofs. The implementation of such a resolution calculus is not easy, for this reason we intend to use an existing implementation of another system in order to verify systems specified in $KL_{(n)}$ syntax. We will use the theorem-prover TeMP (Hustadt et al., 2004), an implementation of

* Partially supported by EPSRC project



a resolution-based calculus for monodic first-order temporal logic over expanding domains (Konev et al., 2003). A FOTL formula ϕ is *monodic* if any subformulae of the form $T\psi$ or $\psi T\phi$, where T is a temporal operator contains at most one free variable.

Our intention is to use TeMP in order to verify problems specified in $KL_{(n)}$ syntax. Thus, in this paper, we present a translation from $KL_{(n)}$ to the monodic fragment of first-order temporal logic in order to use TeMP.

2. Temporal Logic of Knowledge

The logic, $KL_{(n)}$, a *temporal logic of knowledge* 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 (Halpern and Vardi, 1989).

2.1. SYNTAX

Formulae of $KL_{(n)}$ are constructed from a set of propositional symbols $\mathcal{P} = \{p, q, r, \dots\}$; the standard propositional connectives \neg (not), \vee (or), \wedge (and), \Rightarrow (implies). The future-time temporal connectives that we use include ‘ \diamond ’ (*sometime in the future*), ‘ \circ ’ (*at the next moment in time*), ‘ \square ’ (*always*) ‘ \mathcal{U} ’ (*until*), and ‘ \mathcal{W} ’ (*unless, or weak until*). We interpret these connectives over a discrete, linear temporal model of time with finite past and infinite future. Thus, the model of time is isomorphic to the set of natural numbers, \mathbb{N} with the usual order relation $<$, ‘less than’. For knowledge we assume a set of agents $A_g = \{1, \dots, n\}$ and we introduce a set of unary modal connectives K_i for $i \in A_g$, where a formula $K_i\phi$ is read as “agent i knows ϕ ”.

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

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

$$\begin{array}{cccccc} \neg A & A \vee B & A \wedge B & A \Rightarrow B & K_i A & \\ \diamond A & \square A & A \mathcal{U} B & A \mathcal{W} B & \circ A & \end{array}$$

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 $i \in Ag$.*

2.2. SEMANTICS

We first 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. Now, 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 $M = \langle TL, R_1, \dots, R_n, \pi \rangle$, 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 $\langle M, q \rangle$ (where M is a model and q is a point in $TL \times \mathbb{N}$), and formulae in WFF_K . The rules defining the satisfaction relation are given below.

$\langle M, (t, u) \rangle \models \mathbf{true}$	
$\langle M, (t, u) \rangle \not\models \mathbf{false}$	
$\langle M, (t, u) \rangle \models p$	iff $\pi((t, u), p) = T$ (where $p \in \mathcal{P}$)
$\langle M, (t, u) \rangle \models \neg A$	iff $\langle M, (t, u) \rangle \not\models A$
$\langle M, (t, u) \rangle \models A \vee B$	iff $\langle M, (t, u) \rangle \models A$ or $\langle M, (t, u) \rangle \models B$
$\langle M, (t, u) \rangle \models \bigcirc A$	iff $\langle M, (t, u + 1) \rangle \models A$
$\langle M, (t, u) \rangle \models \Box A$	iff $\forall u' \in \mathbb{N}$, if $(u \leq u')$ then $\langle M, (t, u') \rangle \models A$
$\langle M, (t, u) \rangle \models \Diamond A$	iff $\exists u' \in \mathbb{N}$ such that $(u \leq u')$ and $\langle M, (t, u') \rangle \models A$
$\langle M, (t, u) \rangle \models AU B$	iff $\exists u' \in \mathbb{N}$ such that $(u' \geq u)$ and $\langle M, (t, u') \rangle \models B$, and $\forall u'' \in \mathbb{N}$, if $(u \leq u'' < u')$ then $\langle M, (t, u'') \rangle \models A$
$\langle M, (t, u) \rangle \models AW B$	iff $\langle M, (t, u) \rangle \models AU B$ or $\langle M, (t, u) \rangle \models \Box A$
$\langle M, (t, u) \rangle \models K_i A$	iff $\forall t' \in TL. \forall u' \in \mathbb{N}$. if $((t, u), (t', u'))$ $\in R_i$ then $\langle M, (t', u') \rangle \models A$

For convenience of presenting the normal form for $KL_{(n)}$ we introduce a symbol **start** such that $\langle M, (t_0, 0) \rangle \models \mathbf{start}$.

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 (Emerson, 1990)).

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. These axioms are those shown in Figure 1. 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*.

2.3. NORMAL FORM

Formulae in $KL_{(n)}$ can be transformed into a normal form called SNF_K (Dixon et al., 1998) (Separated Normal Form for temporal logics of knowledge).

$\mathbf{K}\phi \Rightarrow \phi$	Reflexivity
$\neg K\phi \Rightarrow K\neg K\phi$	Euclideaness
$K(\phi \Rightarrow \psi) \Rightarrow (K\phi \Rightarrow K\psi)$	K

Figure 1. S5 axioms

Renaming techniques (Plaisted and Greenbaum, 1986) are also used. Complex formulae are replaced by new propositions and the truth value of these propositions is linked to the formulae they replaced in all states. In order to achieve this we introduce a new operator \Box^* , which allows nesting of K_i and \Box operators. This operator is defined in terms of the C (*common knowledge*) and E (*everybody knows*) operators. The operator \Box^* is defined as the maximal fixpoint of

$$\Box^* \Leftrightarrow \Box(\phi \wedge C \Box^* \phi)$$

where the operator C is defined as

$$C\phi \Leftrightarrow E(\phi \wedge C\phi)$$

and E is defined as

$$E\phi \Leftrightarrow E(\phi \wedge C\phi)$$

Formulae in SNF_K are of the general form

$$\Box^* \bigwedge_j T_j$$

where each T_j , known as a *clause*, must be in one of the varieties described in Figure 2 where 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 ' \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 .

start	$\Rightarrow \bigvee_{b=1}^r l_b$	(an <i>initial</i> clause)
$\bigwedge_{a=1}^g k_a$	$\Rightarrow \bigcirc \bigvee_{b=1}^r l_b$	(a <i>step</i> clause)
$\bigwedge_{a=1}^g k_a$	$\Rightarrow \diamond l$	(a <i>sometime</i> clause)
true	$\Rightarrow \bigvee_{b=1}^r m_{ib}$	(a K_i -clause)
true	$\Rightarrow \bigvee_{b=1}^r l_b$	(a literal clause)

Figure 2. Clauses in SNF_K

3. Monodic First-Order Temporal Logic

3.1. SYNTAX

First-Order (discrete linear time) Temporal Logic, FOTL, is an extension of classical first-order logic with operators that deal with a linear a discrete model of time (isomorphic to \mathbb{N} with the usual order relation, $<$, ‘less than’).

Formulae in FOTL are constructed in a standard way (Fisher, 1997; Hodkinson et al., 2000) from:

- *Predicate symbols* P_0, P_1, \dots each of which is of some fixed arity (null-ary predicate symbols are called *propositions*).
- *Individual variables* x_0, x_1, \dots
- *Individual constants* c_0, c_1, \dots
- *Boolean operators* $\wedge, \neg, \vee, \Rightarrow$, **true**(‘true’), **false**(‘false’)
- *quantifiers* \forall and \exists .
- *Temporal operators* ‘ \diamond ’ (*sometime in the future*), ‘ \bigcirc ’ (*at the next moment in time*), ‘ \square ’ (*always*) ‘ \mathcal{U} ’ (*until*), and ‘ \mathcal{W} ’ (*unless, or weak until*).

Thus,

- **true** and **false** are formulae of the language of FOTL.
- Constants are formulae of the FOTL language.
- If ϕ and ψ are FOTL formulae, P_i are predicate symbols and x_i are individual variables then the following are FOTL formulae

$$\begin{array}{llll}
\neg\phi & \phi \vee \psi & \phi \wedge \psi & \phi \Rightarrow \psi \\
\forall x_0 \forall x_1, \dots (P_i(x_0, x_1, \dots)), & \exists x_0 \forall x_1, \dots (P_i(x_0, x_1, \dots)) & & \\
\Diamond\phi & \phi \mathcal{U} \psi & \Box\phi & \\
\phi \mathcal{W} \psi & & \bigcirc\phi &
\end{array}$$

3.2. SEMANTICS

Formulae in FOTL are interpreted in *first-order temporal structures* of the form $\mathfrak{M} = \langle D_n, I_n \rangle$, where D is a non-empty set such that whenever $n < m$, $D_n \subseteq D_m$, and I_n is an interpretation of predicate and constant symbols over D_n .

DEFINITION 7. A (variable) assignment \mathbf{a} is a function from the set of individual variables to $\bigcup_{n \in \mathbb{N}} D_n$. We denote the set of all assignments by \mathfrak{A} .

For every moment of time n , there is a corresponding *first-order* structure, $\mathfrak{M}_n = \langle D_n, I_n \rangle$; the corresponding set of variable assignments \mathfrak{A}_n is a subset of the sets of all assignments, $\mathfrak{A}_n = \{\mathbf{a} \mid \mathbf{a}(x) \in D_n \text{ for every variable } x\}$. Intuitively, FOTL formulae are interpreted in sequences of *worlds*, $\mathfrak{M}_0, \mathfrak{M}_1, \dots$ with truth values in different worlds being connected via temporal operators.

DEFINITION 8. The truth relation $\mathfrak{M}_n \models^{\mathbf{a}} \phi$ in a structure \mathfrak{M} , only for those assignments \mathbf{a} that satisfy the condition $\mathbf{a} \in \mathfrak{A}_n$, is defined inductively in the usual way under the following understanding of temporal operators:

$\mathfrak{M}_n \models^a \bigcirc \phi$	iff	$\mathfrak{M}_{n+1} \models^a \phi$
$\mathfrak{M}_n \models^a \square \phi$	iff	for all $m \in \mathbb{N}$, if $(m \geq n)$ then $\mathfrak{M}_m \models^a \phi$
$\mathfrak{M}_n \models^a \diamond \phi$	iff	there exists $m \in \mathbb{N}$ such that $(m \geq n)$ and $\mathfrak{M}_m \models^a \phi$
$\mathfrak{M}_n \models^a \phi \mathcal{U} \psi$	iff	there exists $m \leq n$, and $\mathfrak{M}_m \models^a \psi$, and $\forall i \in \mathbb{N}$, if $(n \leq i < m)$ then $\mathfrak{M}_i \models^a \phi$
$\mathfrak{M}_n \models^a \phi \mathcal{W} \psi$	iff	$\mathfrak{M}_n \models^a \phi \mathcal{U} \psi$ or $\mathfrak{M}_n \models^a \square \phi$

DEFINITION 9. \mathfrak{M} is a model for a formula ϕ (or ϕ is true in \mathfrak{M}) if there exists an assignment \mathbf{a} such that $\mathfrak{M}_0 \models^a \phi$.

DEFINITION 10. A formula is satisfiable if it has a model. A formula is valid if it is true in any temporal structure under any assignment.

DEFINITION 11. A FOTL formula ϕ is called monodic if any subformulae of the form $T\phi$, where T is one of \diamond , \square , \bigcirc (or $\phi_1 T \phi_2$, where T is one of \mathcal{U} or \mathcal{W}) contains at most one free variable.

4. A Translation from Temporal Logic of Knowledge to Monodic First-Order Temporal Logic

4.1. MOTIVATION

As we mentioned earlier $KL_{(n)}$ is a very useful logic for specifying some systems. Our intention is to use an existing theorem prover for first-order temporal logic in order to verify properties specified in those systems. The translation of temporal epistemic logics into the monodic fragment of first-order temporal is possible and has been shown in (Gabbay et al., 2003). This standard translation is as follows

$$\begin{aligned}
 \pi'_2(p, x) &= \forall x.P(x) \\
 \pi'_2(p * q) &= \forall X.P(x) * Q(x) && \text{for } * \in \{\vee, \wedge, \Rightarrow\} \\
 \pi'_2(T\phi) &= \forall x.T\pi'_2(x) \\
 \pi'_2(K_i p, x) &= \forall x(\forall y(R_i(x, y) \Rightarrow \pi'_2(y)))
 \end{aligned}$$

where R_i is the accessibility relation for the modal operator K_i . Since this relation is an equivalence relation we should add to the translation reflexivity, symmetry and transitivity properties, i.e.,

$$\begin{array}{ll}
\forall x R_i(x, x) & \text{Reflexivity} \\
\forall x, y R_i(x, y) \Rightarrow R_i(y, x) & \text{Symmetry} \\
\forall x, y, z R_i(x, y) \wedge R_i(y, z) \Rightarrow R_i(x, z) & \text{Transitivity}
\end{array}$$

This translation, however, it generates some troubles when using a prover due to the fact that transitivity axiom is included. For this reason, we will present a translation from $KL_{(n)}$ into the monodic fragment of first-order logic which includes reflexive and symmetry axioms as stated before and deals with transitivity in a different way. This translation is based in the results presented in (Schmidt and Hustadt, 2003).

4.2. THE TRANSLATION

We are interested in translating a formula ϕ in $KL_{(n)}$ into the monodic fragment of first-order temporal logic. Thus, we proceed as follows.

Let ϕ be a set of clauses written in the normal form SNF_K , i.e.,

$$\Box^* \bigwedge_j T_j$$

then ϕ can be translated into the first-order temporal logic syntax by applying the transformations π_0 , π_1 and π_2 as follows, where Q is a new predicate symbol and st is a constant representing the initial moment in time.

$$\pi_0[\phi] \rightarrow \exists st(Q(st)) \wedge \Box \bigwedge_j \pi_1(T_j)$$

We define π_1 as follows:

$$\pi_1(T_j) = \forall x(\pi_2(T_j, x))$$

In the following p is a literal, ϕ and ψ are formulae in $KL_{(n)}$, Q is the new predicate symbol introduced in order to define the beginning of time, $Q_{K_i p}$ is a new proposition symbol uniquely associated to $K_i p$ and R_i is the accessibility relation for the modal operator K_i . Then the translation π_2 is as follows:

$$\begin{aligned}
\pi_2(\mathbf{start}, x) &= Q(x) \\
\pi_2(\mathbf{true}, x) &= \mathbf{true} \\
\pi_2(\mathbf{false}, x) &= \mathbf{false} \\
\pi_2(p, x) &= P(x) \\
\pi_2(\neg p, X) &= \neg P(X) \\
\pi_2(\phi \vee \psi, x) &= \pi_2(\phi, x) \vee \pi_2(\psi, x) \\
\pi_2(\phi \wedge \psi, x) &= \pi_2(\phi, x) \wedge \pi_2(\psi, x) \\
\pi_2(\phi \Rightarrow \psi, x) &= \pi_2(\phi, x) \Rightarrow \pi_2(\psi, x) \\
\pi_2(\bigcirc \phi, x) &= \bigcirc \pi_2(\phi, x) \\
\pi_2(\diamond \phi, x) &= \diamond \pi_2(\phi, x) \\
\pi_2(K_i p, x) &= Q_{K_i p}(x) \\
\pi_2(\neg K_i p, x) &= Q_{\neg K_i p}(x)
\end{aligned}$$

where $Q_{\neg K_i p}(x) = \neg Q_{K_i p}(x)$

For each $K_i p$ we add the clauses

$$\square(Q_{K_i p}(x) \Rightarrow (\forall y. R_i(x, y) \Rightarrow Q_{K_i p}(y))) \quad (1)$$

and

$$\square(Q_{K_i p}(x) \Rightarrow (\forall y. R_i(x, y) \Rightarrow \pi_2(p, y))). \quad (2)$$

For each $\neg K_i p$ we add the clauses

$$\square(Q_{\neg K_i p}(x) \Rightarrow (R_i(x, f(x)))) \quad (3)$$

and

$$\square(Q_{\neg K_i p}(x) \Rightarrow \pi_2(\neg p, f(x))) \quad (4)$$

where f is a skolem function uniquely associated with $\neg K_i p$.

For every modal operator, K_i , we will also add reflexivity and symmetry axioms to the translation.

$$\begin{array}{ll}
\forall x. R_i(x, x) & \text{Reflexivity} \\
\forall x, y. (R_i(x, y) \Rightarrow R_i(y, x)) & \text{Symmetry}
\end{array}$$

For practical reasons we will also be considering clauses that are not in the normal form but, however, the transformation π_1 can be applied to them in the same way as it is applied to clauses in SNF_K . The reason is that they do not need to be translated into the normal form since TeMP accepts their syntax.

$$\begin{aligned} \pi_1\left(\bigwedge_{a=1}^g k_a \Rightarrow \square \bigvee_{b=1}^r l_b\right) &= \forall x(\pi_2\left(\bigwedge_{a=1}^g k_a, x\right) \Rightarrow \square \pi_2\left(\bigvee_{b=1}^r l_b, x\right)) \\ \pi_1\left(\bigwedge_{a=1}^g k_a \Rightarrow \bigvee_{b=1}^r l_b \mathcal{U} \bigvee_{c=1}^s n_c\right) &= \forall x(\pi_2\left(\bigwedge_{a=1}^g k_a, x\right) \Rightarrow \pi_2\left(\bigvee_{b=1}^r l_b, x\right) \mathcal{U} \pi_2\left(\bigvee_{c=1}^s n_c, x\right)) \end{aligned}$$

where k_a , l_b , and n_c are literals.

4.3. CORRECTNESS

First we need some definitions. In particular, what it means that the clauses derived after applying the translations are satisfiable.

The translated clauses are interpreted over first-order temporal structures $\mathfrak{M} = \langle D_n, I_n \rangle$, as defined in Section 3.2, where I_n is an interpretation of predicate and constant symbols over the domain D_n . The idea is to build a model for the FOTL structures based on the existing model for $KL_{(n)}$.

Let M be a model for $KL_{(n)}$ then $M = \langle TL, R_i, \dots, R_n, \pi \rangle$ then the FOTL model, $\mathfrak{M}^M = \langle D_n, I_n \rangle$ is defined as follows.

DEFINITION 12. *If Points is the set of points (t, u) in where the valuation π is defined, then $D_n = \text{Points}$ for every $n \in \mathbb{N}$.*

DEFINITION 13. *Let π be the valuation in the $KL_{(n)}$ model and p a proposition symbol such that $\langle M, (t, u) \rangle \models p$, then the interpretation I_n , for $n \in \mathbb{N}$, is defined as*

$$I_n \models P(x)[x \rightarrow (t, u)]$$

$$I_n \models Q(x)[x \rightarrow (t, u)]$$

iff $t = t_0$ and $u = 0$, where Q is the new proposition symbol that we introduce in order to represent the beginning of time.

$$I_n \models R_i(x, y)[x \rightarrow (t_1, u_1), y \rightarrow (t_2, u_2)]$$

iff $((t_1, u_1), (t_2, u_2)) \in R_i$

THEOREM 1. *Let ϕ be a set of clauses in SNF_K . ϕ is satisfiable if and, only if, $\pi_0[\phi]$ is also satisfiable.*

Proof.

First we prove that for any formula ψ that is satisfied in a $KL_{(n)}$ model, i.e., such that $\langle M, (t, u) \rangle \models \psi$ then for every $n \in \mathbb{N}$, $I_n \models \pi_2(\psi, x)[x \rightarrow (t, u)]$.

The proof proceeds by induction on the structure of ψ .

- If ψ is **true**, then for every point (t, u) $\langle M, (t, u) \rangle \models \mathbf{true}$ and obviously $I_n \models \mathbf{true}[x \rightarrow (t, u)]$.
- The same for **false**.
- If $\psi = \mathbf{start}$ then by definition $I_n \models Q(x)[x \rightarrow (t_0, 0)]$, i.e., $I_n \models \pi_2(\mathbf{start}, x)[x \rightarrow (t_0, 0)]$.
- For $p \in P$, $I_n \models \pi_2(p, x)$ by definition.
- Let ψ be $\neg p$ for p a propositional symbol. If $\langle M, (t, u) \rangle \not\models p$, following the definition of the propositional case, $I_n \not\models P(x)[x \rightarrow (t, u)]$.
- Let a formula be of the form $\phi \vee \psi$ for ϕ and ψ propositional symbols. $\langle M, (t, u) \rangle \models \phi \vee \psi$, then $\langle M, (t, u) \rangle \models \phi$ or $\langle M, (t, u) \rangle \models \psi$. By induction hypothesis and definition of I_n , $I_n \models P(x)[x \rightarrow (t, u)]$ or $I_n \models R(x)[x \rightarrow (t, u)]$
- The same for \wedge and \Rightarrow .
- Let ψ be of the form $\bigcirc\psi$. If $\langle M, (t, u) \rangle \models \bigcirc\psi$ then $\langle M, (t, u + 1) \rangle \models \psi$. Therefore by induction hypothesis and the definition of I_n , $I_n \models \psi(x)[x \rightarrow (t, u + 1)]$. Since all the interpretations are the same $I_{n+1} \models p(x)[x \rightarrow (t, u + 1)]$, i.e., $I_n \models \bigcirc\psi(x)[x \rightarrow (t, u + 1)]$.
- the same for $\diamond\psi$ and $\square\psi$.
- Let us now consider a formula of the form $K_i p$.
If $\langle M, (t, u) \rangle \models K_i p$ then $\forall t'$ and $\forall u' \in \mathbb{N}$ if $((t, u), (t', u')) \in R_i$ then $\langle M, (t', u') \rangle \models p$. By definition of I_n , $I_n \models \pi_2(p, y)[y \rightarrow (t', u')]$ if, and only if, $\langle M, (t', u') \rangle \models p$. Also by definition of I_n , but on R_i , $I_n \models R_i(x, y)[x \rightarrow (t, u), y \rightarrow (t', u')]$ since $((t, u), (t', u')) \in R_i$. Because of clause 2 $I_n \models Q_{K_i p}(x)$ and by definition of π_2 , $\pi_2(K_i p, x) = Q_{K_i p}(x)$. Thus, $I_n \models \pi_2(K_i p, x)[x \rightarrow (t, u)]$.
- Now we consider the case of a formula ψ of the form $\psi = \neg K_i p$.
If $\langle M, (t, u) \rangle \models \neg K_i p$ then there exists a point (t', u') such that $((t, u), (t', u')) \in R_i$ and $\langle M, (t', u') \rangle \not\models p$, i.e., $\langle M, (t', u') \rangle \models \neg p$. By definition of I_n , $I_n \models \pi_2(\neg p, f(x))[f(x) \rightarrow (t', u')]$, where f is a skolem function. Because of clause 3 $I_n \models Q_{\neg K_i p}[x \rightarrow (t, u)]$ and then $I_n \models R_i(x, f(x))[x \rightarrow (t, u)]$. Because of clause 4 $I_n \models \pi_2(\neg K_i p, x)$. Therefore, $I_n \models \pi_2(\neg K_i p, x)[x \rightarrow (t, u)]$.

So far, we have shown that for every clause T_j in SNF_K if $\langle M, (t, u) \rangle \models T_j$ then $I_n \models \pi_2(T_j, x)[x \rightarrow (t, u)]$. If $\phi = \Box^* \bigwedge_j T_j$ is satisfiable in M then it means that for all points $(t, u) \in \text{Points}$ $\langle M, (t, u) \rangle \models T_j$. Since we have shown that $\langle M, (t, u) \rangle \models T_j$ then $I_n \models \pi_2(T_j, x)[x \rightarrow (t, u)]$. If this holds for all the points in Points and we define $\text{Points} = D_n$ for all $n \in \mathbb{N}$, we can deduce that $\forall x(\pi_2(T_j, x))$, i.e., the result by applying π_1 to T_j .

Now we will consider a set of clauses SNF_K , ϕ such that $\pi_0(\phi)$ is satisfiable. We must show that there is a model M for $KL_{(n)}$ such that $M \models \phi$.

If \mathfrak{M} is a model for $\pi_0(\phi)$ it means that $\mathfrak{M}_o \models \pi_0(\phi)$, where $\pi_0(\phi) = Q(st) \wedge \Box \bigwedge_j \pi_1(T_j)$. Therefore $\mathfrak{M}_o \models Q(st)$ and $\mathfrak{M}_o \models \forall x \pi_2(T_j, x)$,

i.e.,

$\mathfrak{M}_o \models \pi_2(T_j, x)[x \rightarrow d]$ for all $d \in D_0$.

We construct a model $M = \langle TL, R_1, \dots, R_n, \pi \rangle$ for $KL_{(n)}$.

- We define $(t_0, 0) = st$. If $d \in D_n$ then we define a point as (d, n) .
- We construct the timelines as follows.

For every $d \in \bigcup_{n \geq 0} D_n$, let $\min(d) = n$ if, and only if, $d \in D_n$ and $n = 0$ or $n > 0$ and $d \notin D_{n-1}$. Thus, for every $n \geq 0$ and $d \in D_n$ we define a timeline as $(d, n) = t_{n-\min(d)}^d$, which assures us that the timeline has an initial moment.

Since we are assuming the case of expanding domains, if $d \in D_n$ then $d \in D_m$ for $m \geq n$. This and the fact that we define the timelines in such a way that they have an initial moment assures that we can define timelines.

- The relation R_i is defined as follows.

Let (t, u) and (t', u') be points in M . Let (t, u) corresponds to (d, n) and (t', u') to (d', n') . Then $((t, u), (t', u')) \in R'_i$ if, and only if $n = n'$ and $I_n \models R_i(x, y)[x \rightarrow d, y \rightarrow d']$. Let R_i be the transitive closure of R'_i .

- The valuation $\pi : \text{Points} \times \mathcal{P} \rightarrow \{T, F\}$ is defined in a similar way as we defined I_n .

If p is a proposition symbol then we define $\pi((t, u), p) = I_n(p(x))$.

In the following we will consider every single case of a clause T_j in the normal form such that $\pi_2(T_j, x)$ is satisfiable in \mathfrak{M}_o . In order to

show that ϕ is satisfiable in M , we must show that the clause T_j is true at every point (t, i) in the $KL_{(n)}$ model we have constructed. Let (t, i) correspond to (d, n) .

- Let T_j be a clause of the form $\mathbf{start} \Rightarrow \bigvee_{j=1}^m l_j$. We know that $\mathfrak{M}_o \models \forall x \pi_2(\mathbf{start} \Rightarrow \bigvee_{j=1}^m l_j, x)$, i.e., $\mathfrak{M}_o \models (Q(x) \Rightarrow \bigvee_{j=1}^m L_j(x))[x \rightarrow d]$ for $d \in D_0$.

Q is the new proposition symbol introduced in order to represent the beginning of time. We know that $\mathfrak{M}_o \models Q(st)$, so there is $d \in D_0$ such that $I_0 \models Q(x)[x \rightarrow d]$. Let d correspond to $(t_0, 0)$ by definition and by construction of the model M , $\langle M, (t_0, 0) \rangle \bigvee_{j=1}^m L_j$ is true.

- Let T_j a clause of the form $\bigwedge_{a=1}^g k_a \Rightarrow \bigcirc \bigvee_{j=1}^r l_j$. We know that $\mathfrak{M}_i \models \forall x \pi_2(\bigwedge_{a=1}^g k_a \Rightarrow \bigcirc \bigvee_{j=1}^m l_j, x)$, i.e., $\mathfrak{M}_i \models (\bigwedge_{a=1}^g K_a(x) \Rightarrow \bigcirc \bigvee_{j=1}^m L_j(x))[x \rightarrow d]$ for all $d \in D_i$. This means that there is an interpretation I_n such that, $I_n \models \bigwedge_{a=1}^g K_a(x) \Rightarrow \bigcirc \bigvee_{j=1}^m L_j(x)[x \rightarrow d]$. So either $I_n \not\models \bigwedge_{a=1}^g K_a(x)$ or $I_{n+1} \models \bigvee_{j=1}^m L_j(x)[x \rightarrow d]$.

By definition of the model M we have constructed for $KL_{(n)}(d, n+1)$ corresponds to $(t, i+1)$ and $\langle M, (t, i+1) \rangle \models \bigvee_{j=1}^m l_j$ is true.

- The same for clauses of the form $\bigwedge_{a=1}^g k_a \Rightarrow \diamond \bigvee_{j=1}^m l_j$ and $\mathbf{true} \Rightarrow \bigvee_{j=1}^m l_j$ for k_a and l_j literals.
- Now we consider clauses of the form $\mathbf{true} \Rightarrow \bigvee_{b=1}^r m_{ib}$ where m_{ib} are modal literals.

We have to show that

$$\langle M, (t, u) \rangle \models \mathbf{true} \Rightarrow K_{ip_1} \vee \dots \vee K_{ip_r}$$

if, and only if,

$$\langle M, (t, u) \rangle \models K_{ip_1} \vee \dots \vee K_{ip_r}$$

if, and only if

$$\langle M, (t, u) \rangle \models K_{ip_1} \text{ or } \langle M, (t, u) \rangle \models K_{ip_2} \text{ or } \dots \langle M, (t, u) \rangle \models K_{ip_r}.$$

if, and only if there exists l , $1 \leq l \leq r$ such that $\langle M, (t, u) \rangle \models K_{ip_l}$, i.e., for all t' and for all u' if $((t, u), (t', u')) \in R_i$ then $\langle M, (t', u') \rangle \models p_l$

$$\text{If } \mathfrak{M}_{n_0} \models \forall x \pi_2(\mathbf{true} \Rightarrow K_{ip_1} \vee \dots \vee K_{ip_r}, x).$$

Let $d_0 \in D_{n_0}$, where (d_0, n_0) corresponds to (t, u) , then we know that there is an interpretation I_{n_0} such that

$$I_{n_0} \models (\mathbf{true} \Rightarrow Q_{K_{ip_1}}(x) \vee \dots \vee Q_{K_{ip_r}}(x))[x \rightarrow d_0] \quad (5)$$

or

$$I_{n_0} \models (Q_{K_{ip_1}}(x) \vee \dots \vee Q_{K_{ip_r}}(x))[x \rightarrow d_0] \quad (6)$$

$$I_{n_0} \models (Q_{K_{ip_1}}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow Q_{K_{ip_1}}(y)))[x \rightarrow d_0] \quad (7.1)$$

$$I_{n_0} \models (Q_{K_{ip_2}}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow Q_{K_{ip_2}}(y)))[x \rightarrow d_0] \quad (7.2)$$

... ..

$$I_{n_0} \models (Q_{K_{ip_r}}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow Q_{K_{ip_r}}(y)))[x \rightarrow d_0] \quad (7.3)$$

$$I_{n_0} \models (Q_{K_{ip_1}}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow P_1(y)))[x \rightarrow d_0] \quad (8.1)$$

$$I_{n_0} \models (Q_{K_{ip_2}}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow P_2(y)))[x \rightarrow d_0] \quad (8.2)$$

... ..

$$I_{n_0} \models (Q_{K_{ip_r}}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow P_r(y)))[x \rightarrow d_0] \quad (8.3)$$

$$I_{n_0} \models \forall x R_i(x, x) \quad \text{Reflexivity} \quad (9)$$

$$I_{n_0} \models \forall x, y R_i(x, y) \Rightarrow R_i(y, x) \quad \text{Symmetry} \quad (10)$$

If clause 6 holds it means that for some index l $I_{n_0} \models Q_{K_i p_l}(x)[x \rightarrow d_0]$. Due to clauses 9 and 10 we know that R_i is reflexive and symmetric. Due to clause 7 whenever there is a sequence d_0, d_1, \dots, d_n such that $(d_i, d_{i+1}) \in R_i$, for all $i \leq n-1$, then $I_{n_0} \models Q_{K_i p_l}(x)[x \rightarrow d_i]$ then $I_{n_0} \models Q_{K_i p_l}(y)[y \rightarrow d_{i+1}]$. So $I_{n_0} \models Q_{K_i p_l}(y)[y \rightarrow d_1]$ for all $d_1 \in \{d \mid (d_0, d) \in (R_i \cup R_i^v)\}$, where $R_i \cup R_i^v$ is the reflexive, symmetric and transitive closure of R_i . By clause 8 if $I_{n_0} \models Q_{K_i p_l}(x)[x \rightarrow d]$ then $I_{n_0} \models p_l(x)[x \rightarrow d]$.

By construction of M , $((t, u), (t', u')) \in R'_i$ if, and only if, $(d_0, d_1) \in R_i$ for d_0 corresponding to (t, u) and d_1 corresponding to (t', u') and R_i is the transitive closure of R'_i . As we have shown that $I_{n_0} \models p_l(x)[x \rightarrow d_1]$ for every $d_1 \in \{d \mid (d_0, d) \in (R_i \cup R_i^v)\}$, by construction of M $\langle M, (t', u') \rangle \models p_l$ where (t', u') corresponds to $d_1 \in \{d \mid (d_0, d) \in (R_i \cup R_i^v)\}$. Thus, $\langle M, (t, u) \rangle \models K_i p_l$.

5. Example

Let $\phi = (\Box Kp \Rightarrow \bigcirc Kp)$ be the formula we want to prove. We translate into monodic first-order temporal logic using the translation presented in Section 4.2. First, we translate it into SNF $_K$. We negate it, obtaining

$$\neg\phi = \Box Kp \wedge \bigcirc \neg Kp$$

We anchor it to q

$$\begin{aligned} \mathbf{start} &\Rightarrow t_0 \\ t_0 &\Rightarrow \Box Kp \wedge \bigcirc \neg Kp \end{aligned}$$

Thus the normal form is as follows

$$\begin{aligned} \mathbf{start} &\Rightarrow q \\ \mathbf{true} &\Rightarrow \neg t_1 \vee Kp \\ \mathbf{true} &\Rightarrow \neg t_0 \vee t_1 \\ \mathbf{true} &\Rightarrow \neg t_0 \vee t_3 \\ t_3 &\Rightarrow \bigcirc t_1 \\ t_3 &\Rightarrow \bigcirc t_3 \\ t_0 &\Rightarrow \bigcirc t_2 \\ \mathbf{true} &\Rightarrow \neg t_2 \vee \neg Kp \end{aligned}$$

We introduce a new predicate symbol for Kp . Let this new predicate symbol be PKA and let $a1$ be $\neg pka$. Thus, the clause

According to the translation defined in Section 4.2 for every Kp we add some new clauses

$$(PKA(x) \Rightarrow (\forall y.R(x, y) \Rightarrow PKA(y)))$$

$$(PKA(X) \Rightarrow (\forall Y.R(x, y) \Rightarrow P(y)))$$

For $\neg Kp$ we add the clauses

$$(A1(x) \Rightarrow (R(x, skolem(x))))$$

$$(A1(x) \Rightarrow (R(x, y) \Rightarrow \neg P(skolem(x))))$$

After applying π_2 the resultant clauses are as follows

$$\begin{aligned} \mathbf{true} &\Rightarrow \neg Q(x) \vee T_0(x) \\ \mathbf{true} &\Rightarrow \neg T_1(x) \vee PKA(x) \\ \mathbf{true} &\Rightarrow \neg T_0(x) \vee T_1(x) \\ \mathbf{true} &\Rightarrow \neg T_0(x) \vee T_3(x) \\ T_3(x) &\Rightarrow \bigcirc T_1(x) \\ T_3(x) &\Rightarrow \bigcirc T_3(x) \\ T_0(x) &\Rightarrow \bigcirc T_2(x) \\ \mathbf{true} &\Rightarrow \neg T_2(x) \vee A1(x) \\ \mathbf{true} &\Rightarrow \neg A1(x) \vee \neg PKA(x) \\ PKA(x) &\Rightarrow (\forall y.R(x, y) \Rightarrow PKA(y)) \\ PKA(X) &\Rightarrow (\forall Y.R(x, y) \Rightarrow P(y)) \\ A1(x) &\Rightarrow (R(x, skolem(x))) \\ A1(x) &\Rightarrow (R(x, y) \Rightarrow \neg P(skolem(x))) \end{aligned}$$

Then we apply π_1 and π_0 obtaining as the final translation the following

$$\begin{aligned}
& Q(st) \\
\forall x(\mathbf{true} & \Rightarrow \neg Q(x) \vee T_0(x)) \\
\forall x(\mathbf{true} & \Rightarrow \neg T_1(x) \vee PKA(x)) \\
\forall x(\mathbf{true} & \Rightarrow \neg T_0(x) \vee T_1(x)) \\
\forall x(\mathbf{true} & \Rightarrow \neg T_0(x) \vee T_3(x)) \\
\forall x(T_3(x) & \Rightarrow \bigcirc T_1(x)) \\
\forall x(T_3(x) & \Rightarrow \bigcirc T_3(x)) \\
\forall x(T_0(x) & \Rightarrow \bigcirc T_2(x)) \\
\forall x(\mathbf{true} & \Rightarrow \neg T_2(x) \vee A1(x)) \\
\forall x(\mathbf{true} & \Rightarrow \neg A1(x) \vee \neg PKA(x)) \\
\forall x(PKA(x) & \Rightarrow (\forall y.R(x, y) \Rightarrow PKA(y))) \\
\forall x(PKA(X) & \Rightarrow (\forall Y.R(x, y) \Rightarrow P(y))) \\
\forall x(A1(x) & \Rightarrow (R(x, skolem(x)))) \\
\forall x(A1(x) & \Rightarrow (R(x, y) \Rightarrow \neg P(skolem(x))))
\end{aligned}$$

6. TeMP

The reason why we presented the translation from $KL_{(n)}$ to monodic FOTL in Section 4.2 is because we are interested in using TeMP (Hustadt et al., 2004), a theorem-prover for the monodic fragment of FOTL. TeMP is based on a resolution calculus for the monodic fragment of FOTL with expanding domains (Konev et al., 2003). The operations that simulate classical resolution have been carried out using an efficient resolution-based prover for first-order classical logic. This prover is called Vampire (Riazanov and Voronkov, 2002; Voronkov, 1995).

7. Experimental Results

In the following section we will present the results obtained by applying TeMP when the translation presented above is used to specify case studies expressed in $KL_{(n)}$.

7.1. CLUEDO

Cluedo is a board game, commercially produced by Hasbro (Clu,), where players gather information about a murder. The suspects, murderer weapons and rooms where the murder took place are represented by cards. One card of each type is removed and kept aside representing the murderer, the murder weapon and the room where the murder took place. The remaining cards are shuffled and handed to the players. The

players aim to find out who the murderer was, which murder weapon was used and where the murder took place. In order to achieve this, they use the knowledge of their own cards, the knowledge obtained from cards that other players revealed during the game or from statements that another player does not have such a card. Players take turns to make a *suggestion*: suspect, weapon and room. If the player to the left holds one of these cards it is shown to the player making the suggestion but without the other players seeing it. If the player to the left does not hold any of the suggested cards she or he declares it and the player to her or his left must try to show one of the cards to the suggesting player. This continues until a card is shown to the suggesting player or no card has been shown for any player for this suggestion. Players use the knowledge about their cards and the cards that have or have not been shown in order to eliminate suspects, weapons and rooms for their next suggestions. When a player knows the murder, weapon and room makes an *accusation* and checks the cards kept aside. If the player is right, this player wins the game. If the player is wrong cannot make suggestions and accusations, but can answer the suggestions from other players.

Cluedo game has been specified in (Dixon, 2004) using $KL_{(n)}$. The game has been reduced (in order to make the specification simpler) to four suspects (Prof. Plum, Rev. Green, Col. Mustard and Miss Scarlett), four murder weapons (lead piping, spanner, revolver and rope) and no rooms. We assume three players Catherine, Wendy and Jane.

Let $\{c, w, j\}$ be the set of players. We denote by $K_i p$ that player i knows p , where $i \in \{c, w, j\}$. We use propositions in order to specify which player holds each of the cards

- 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.

We denote if a suspect is the murderer, or a weapon is the murderer weapon as follows.

- 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.

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

After this deal is made we can prove the following statements

1. At time one Catherine knows that Miss Scarlett is not the murderer.
This is specified in $KL_{(n)}$ as

$$\bigcirc K_c \neg r_m$$

2. At time two Catherine makes the suggestion ‘Miss Scarlett and the lead piping’. After all players have their turns for showing their cards and no cards are revealed, it is expected Catherine to make an accusation. Since no accusation is made by Catherine, Jane and Wendy can deduce that Catherine holds one of these cards which is specified in $KL_{(n)}$ as follows

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

At this point Catherine should also be able to deduce that the lead piping is the murder weapon, which it would be the next statement we will try to show using TeMP. This statement in $KL_{(n)}$ is written as

$$\bigcirc \bigcirc K_c l_m$$

3. At time three Wendy makes the suggestion lead piping and Col. Mustard. Jane shows Wendy Col. Mustard. Thus at time three Catherine knows Jane holds either the lead piping or Col. Mustard, i.e., $\bigcirc\bigcirc\bigcirc K_c(l_j \vee y_j)$. Since in the previous stage Catherine deduced that the murder weapon was the lead piping she can deduce now that the murderer is Prof. Plum. That is, at stage three Catherine can deduce both the murderer and the murder weapon, i.e.,

$$\bigcirc\bigcirc\bigcirc K_c(l_m \wedge b_m)$$

4. We have also proved a statement that uses the temporal resolution rule. We have proved that from time one onwards Catherine knows that Miss Scarlett is not the murderer, i.e.

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

5. At time one we can also prove that Wendy knows the revolver is not the murder weapon.

$$\bigcirc K_w \neg v_m$$

6. If Catherine makes the suggestion ‘Col. Mustard and the lead piping’, Wendy will not show any card to Catherine but Jane will show her Col. Mustard. Thus, at time two Catherine knows that Col. Mustard is not the murderer.

$$\bigcirc\bigcirc K_c \neg y_m$$

The results are presented in table I.

Table I. Results by TeMP

Case	Number of clauses generated	Time in seconds
1	476	0.047
2	977	0.074
3	5707	1.246
4	7926	0.640
5	460	0.029
6	548	0.035

7.2. MUDDY CHILDREN

We consider the *muddy children problem*, a well known problem in reasoning about knowledge. We use the version taken from (Fagin et al., 1995)

Imagine n children playing together... Now it happens during their play that some of the children, say k of them, get mud on their foreheads. Each can see the mud on others but not on his forehead. Along comes the father, who says, “At least one of you has mud on your forehead”, thus expressing a fact known to each of them before he spoke (if $k > 1$). The father then asks the following question, over and over: “Does any of you know whether you have mud on your own forehead?” Assuming that all the children are perceptive, intelligent, truthful, and they answer simultaneously, what will happen? There is a ‘proof’ that the first $k - 1$ times he asks the question, they will say “No”, but then the k^{th} time the children with muddy foreheads will all answer “Yes”.

We consider the case of only two children, in order to make it simpler. The translation of this example into $KL_{(n)}$ can be seen in (Dixon et al., 1998). We use m_1 to denote that child one has a muddy forehead and m_2 to show that child two has a muddy forehead. If the father announces that at least one of the children’s foreheads is muddy, i.e.,

$$\square(m_1 \vee m_2)$$

then we could prove the following statements.

1. If initially both children’s foreheads are muddy, then we can prove that at time 2 both children know they are muddy, i.e.,

$$\bigcirc \bigcirc (K_1 m_1 \wedge K_2 m_2)$$

2. If we assume that only child one has a muddy forehead, then at time 1 child one will know that he is muddy.

$$\bigcirc K_1 m_1$$

The results are presented in table II.

Table II. Results by TeMP

Case	Number of clauses generated	Time in seconds
1	271	0.015
2	186	0.029

7.3. THE NEEDHAM-SCHROEDER PROTOCOL WITH PUBLIC KEYS

The Needham-Schroeder protocol with public keys (Needham and Schroeder, 1978) intends to establish authentication between an agent A who initiates the protocol and an agent B who responds to A . The complete protocol consists of seven messages, but we here focus on a simplified version consisting of only three messages. These are sufficient to illustrate the specification and verification of the protocol in $KL_{(n)}$. The messages that we omit are those whereby the agents request other agent's public keys from a server.

The protocol can then be described as the three following steps:

Message	Direction	Contents
Message 1	$A \rightarrow B :$	$\{N_A, A\}_{pub_key(B)}$
Message 2	$B \rightarrow A :$	$\{N_B, N_A\}_{pub_key(A)}$
Message 3	$A \rightarrow B :$	$\{N_B\}_{pub_key(B)}$

Note that message contents of the form $\{X, Y\}_{pub_key(Z)}$ represent messages containing both X and Y but then encrypted with Z 's public key. Elements of the form N_X are special items of data, called *nonces*. Typically, agents in the protocol will generate their own unique nonce (often encrypted) which is initially unknown to all other agents.

Message 1: A sends B an encrypted nonce together with A 's identity, all encrypted with B 's public key.

Message 2: When B receives Message 1, it decrypts it to obtain N_A . Then B returns to A the nonce N_A and generates another nonce of his own, N_B , and sends it back, this time encrypted with A 's public key.

Message 3: When A receives Message 2, it returns B 's nonce, this time encrypted with B 's public key in order to prove A 's authenticity.

It would seem that A should be sure he is talking to B , since only B 'should' be able to decrypt Message 1. In the same way, B seems to be sure that he is talking to A since only A 'should' be able to decrypt Message 2. However, this is not always the case.

We use $KL_{(n)}$ to specify the Needham-Schroeder protocol (full details of the specification and axioms can be found at (Dixon et al., 2003;

Dixon et al., 2004)). We use the following syntactic conventions. Let M_1 and M_2 be variables over messages, Key be a variable over keys and X, Y, \dots be variables over agents. Moreover, for every agent, X , we assume there are keys $pub_key(X)$ and $priv_key(X)$, while in this protocol A and B are constants representing two specific agents. We identify the following predicates:

- $send(A, Msg, Key)$ is satisfied if agent A sends message Msg encrypted by Key ;
- $rcv(A, Msg, Key)$ is satisfied if agent A receives message Msg encrypted by Key ;
- $Msg(M_1)$ is satisfied if M_1 is a message;
- $val_pub_key(X, V)$ is satisfied if the value public key of X is V
- $val_priv_key(X, V)$ is satisfied if the value private key of X is V
- $val_nonce(N_A, V)$ is satisfied if the value of nonce N_A is V
- $contains(M_1, M_2)$ is satisfied if the message M_2 is contained within M_1 .

Using this notation we can specify axioms related to the protocol such as

- Structural assumptions concerning keys and messages contents.
- Scenario assumptions.
- Basic knowledge axioms.
- Communication axioms.

Using TeMP we could prove the following statements related to this protocol.

1. B's knowledge on receipt of N_A .

Once B receives the nonce of A encoded by B's public key then B knows the nonce of A.

This is translated into $KL_{(n)}$ as

$$\square(rcv(B, m1, pub_key(B)) \Rightarrow \bigcirc \exists V. K_B val_nonce(N_A, V))$$

2. Confirmation of B 's Knowledge

Once A receives m_2 (which, in turn, contains N_A) back, then it can infer that B knows N_A , i.e.

$$rcv(A, m_2, pub_key(A)) \Rightarrow \bigcirc K_A K_B N_A$$

The results obtained by applying TeMP are stated in the following table III

Table III. Results for the Needham-Schroeder Protocol

Case	Time in seconds
1	0.733
2	0.048

8. Conclusions

We have presented a translation from temporal logics of knowledge, $KL_{(n)}$ into the monodic fragment of first-order temporal logic. This translation allows us to use an existing monodic first-order temporal logic theorem prover, called TeMP, in order to verify some case studies which are specified in $KL_{(n)}$. We have shown correctness of this translation. Using this translation we have been able to prove some properties of a game called Cluedo, the muddy children example and some properties of the Needham-Schroeder protocol with public keys. All these examples have been proven by hand, however automatic proofs for them could not be carried out due to the lack of an appropriate implemented prover for $KL_{(n)}$.

In the future we intend to use the translation in order to specify more case studies specified in $KL_{(n)}$ as well as testing TeMP.

References

- 'Cluedo'. <http://www.hasbro.com>.
- Dixon, C.: 2004, 'Specifying and Verifying the Game Cluedo Using Temporal Logics of Knowledge'. Technical Report ULCS-04-03, <http://www.csc.liv.ac.uk/research>.
- Dixon, C., M. Fernández-Gago, M. Fisher, and W. van der Hoek: 2003, 'Using Temporal Logics of Knowledge in the Formal Verification of Security Protocols'. Technical Report 03-022, ULCS, <http://www.csc.liv.ac.uk/research>.

- Dixon, C., M. Fernández-Gago, M. Fisher, and W. van der Hoek: 2004, ‘Using Temporal Logics of Knowledge in the Formal Verification of Security Protocols’. In: *Proceedings of TIME2004*. IEEE.
- Dixon, C., M. Fisher, and M. Wooldridge: 1998, ‘Resolution for Temporal Logics of Knowledge’. *Journal of Logic and Computation* **8**(3), 345–372.
- Emerson, E. A.: 1990, ‘Temporal and Modal Logic’. In: J. van Leeuwen (ed.): *Handbook of Theoretical Computer Science*. Elsevier, pp. 996–1072.
- Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi: 1995, *Reasoning About Knowledge*.
- Fisher, M.: 1997, ‘A Normal Form for Temporal Logic and its Application in Theorem-Proving and Execution’. *Journal of Logic and Computation* **7**(4), 429–456.
- Fisher, M. and M. Wooldridge: 1997, ‘On the Formal Specification and Verification of Multi-Agent Systems’. *International Journal of Cooperative Information Systems* **6**(1).
- Gabbay, D. M., A. Kurusz, F. Wolter, and M. Zakharyashev: 2003, *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier.
- Halpern, J. Y.: 1987, ‘Using Reasoning about Knowledge to Analyze Distributed Systems’. *Annual Review of Computer Science* **2**.
- Halpern, J. Y. and M. Y. Vardi: 1989, ‘The Complexity of Reasoning about Knowledge and Time. I Lower Bounds’. **38**, 195–237.
- Hodkinson, I., F. Wolter, and M. Zakharyashev: 2000, ‘Decidable fragments of First-Order Temporal Logic’. *Annals of Pure and Applied Logic* **106**, 85–134.
- Hustadt, U., B. Konev, A. Riazanov, and A. Voronkov: 2004, ‘TeMP: A Temporal Monodic Prover’. Technical Report 04-004, ULCS, <http://www.csc.liv.ac.uk/research>.
- Konev, B., A. Degtyarev, C. Dixon, M. Fisher, and U. Hustadt: 2003, ‘Towards the Implementation of First-Order Temporal Resolution: the Expanding Domain Case’. In: *Proceedings of the 10th International Symposium on Temporal Representation and Reasoning (TIME-ICTL)*.
- Meyer, J. J. C. and W. van der Hoek: 1995, *Epistemic Logic for Computer Science and Artificial Intelligence*, Vol. 41 of *Cambridge Tracts in Theoretical Computer Science*.
- Needham, R. and M. Schroeder: 1978, ‘Using Encryption for Authentication in Large Networks of Computers’. *Communications of the ACM* **21**, 993–999.
- Plaisted, D. A. and S. A. Greenbaum: 1986, ‘A Structure-Preserving Clause Form Translation’. *Journal of Symbolic Computation* **2**(3), 293–304.
- Riazanov, A. and A. Voronkov: 2002, ‘The Design and Implementation of Vampire’. *AI Communications* **15**(2-3), 91–110.
- Schmidt, R. A. and U. Hustadt: 2003, ‘A Principle for Incorporating Axioms into the First-Order Translation of Modal Formulae’. In: *Automated Deduction—CADE-19*, Vol. 2741 of *Lecture Notes in Artificial Intelligence*. pp. 412–426, Springer.
- Syverson, P.: 1993, ‘Adding Time to a Logic of Authentication’. In: *Proceedings of the First ACM Conference on Computer and Communications Security*. pp. 97–101, ACM Press.
- Voronkov, A.: 1995, ‘The Anatomy of Vampire’. *Journal of Automated Reasoning* **15**(2), 237–265.