C. Fernandez-Gago, et al., "First-Order Temporal Verification in Practice", *Journal of Automated Reasoning*, vol. 34, pp. 295-321, 2005.
http://doi.org/dx.doi.org/10.1007/s10817-005-7354-1
NICS Lab. Publications: https://www.nics.uma.es/publications

Journal of Automated Reasoning (2005) 34: 295–321
DOI: 10.1007/s10817-005-7354-1

First-Order Temporal Verification in Practice ⋆

M. C. FERNÁNDEZ-GAGO, U. HUSTADT, C. DIXON, M. FISHER and B. KONEV
Department of Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom.
e-mail: {m.c.gago,u.hustadt,c.dixon,m.fisher,b.konev}@csc.liv.ac.uk

(Received: February 2004; accepted: April 2005)

Abstract. First-order temporal logic, the extension of first-order logic with operators dealing with time, is a powerful and expressive formalism with many potential applications. This expressive logic can be viewed as a framework in which to investigate problems specified in other logics. The monodic fragment of first-order temporal logic is a useful fragment that possesses good computational properties such as completeness and sometimes even decidability. Temporal logics of knowledge are useful for dealing with situations where the knowledge of agents in a system is involved. In this paper we present a translation from temporal logics of knowledge into the monodic fragment of first-order temporal logic. We can then use a theorem prover for monodic first-order temporal logic to prove properties of the translated formulas. This allows problems specified in temporal logics of knowledge to be verified automatically without needing a specialized theorem prover for temporal logics of knowledge. We present the translation, its correctness, and examples of its use.

Key words: first-order temporal logic, temporal logics of knowledge, theorem proving, resolution.

1. Introduction

Temporal logics have been shown useful in computer science and artificial intelligence in order to specify how a system changes over time (Chomicki and Niwinski, 1995; Manna and Pnueli, 1992). First-order temporal logics (FOTLs) allow the use of both first-order syntax and operators relating to change over time. This powerful and expressive language has generally been avoided because of problems with completeness. However, a particular fragment of first-order temporal logic, the *monodic fragment* (Hodkinson et al., 2000), has the completeness and sometimes even the decidability property. We can use this fragment as a unifying framework for other non-classical logics such as temporal logics of knowledge or belief (Fagin et al., 1995), spatio-temporal logics (Gabelaia et al., 2003), or temporal description logics (Artale and Franconi, 1999).

TeMP (Hustadt et al., 2004) is an implementation of a resolution-based calculus for monodic first-order temporal logic over expanding domains (Konev et al., 2003). By providing satisfiability-preserving translations from the logics mentioned above into the monodic fragment of first-order temporal logic, we can

⋆ Partially supported by EPSRC project: Analysis and Mechanisation of Decidable First-Order Temporal Logics (GR/R45376/01).

then use TeMP to prove problems from a range of logics. This avoids having to develop a number of special-purpose theorem provers. In this paper we explore this idea by considering the case of temporal logic of knowledge.

Temporal logics of knowledge are useful in order to specify systems where the knowledge of an agent is important and that knowledge changes over time. The logic we use in this paper is the fusion of linear time temporal logic with finite past and infinite future combined with the multimodal logic $S5_n$. Temporal logics of knowledge have been used for the specification and verification of multiagent systems (Fisher and Wooldridge, 1997; Halpern, 1987; Meyer and van der Hoek, 1995), security protocols (Dixon et al., 2003, 2004; Syverson, 1993), games involving knowledge such as Cluedo (Dixon, 2004), and puzzles such as the muddy children puzzle (Dixon et al., 1998; Fagin et al., 1995).

Here we provide a satisfiability-preserving translation from temporal logic of knowledge into monodic first-order temporal logic. Translations of modal logics into first-order classical logics are given in, for example, (Gabbay et al., 2003). The translation used here is based on that in (Schmidt and Hustadt, 2003) and avoids a direct encoding of the transitivity axiom, thereby making it particularly suitable for mechanization. We provide experimental results from case studies expressed in a temporal logic of knowledge, $KL_{(n)}$, which have been translated into monodic first-order temporal logic. Properties of these case studies have been proved by using TeMP.

The paper is organized as follows. In Section 2 we present the syntax and semantics of $KL_{(n)}$ and describe a normal form for this logic. In Section 3 we describe monodic first-order temporal logic. In Section 4 we provide a translation from formulae in the normal form of $KL_{(n)}$ to monodic first-order temporal logic and prove its correctness. In Section 5 we give an example using the translation, and in Section 6 we discuss the experimental results from various problems expressed in $KL_{(n)}$. In Section 7 we compare the results obtained by using the translation presented in Section 4 with the results obtained by using the standard translation from $KL_{(n)}$ into the monodic fragment of FOTL. We provide concluding remarks in Section 8.

2. Temporal Logic of Knowledge

The logic $KL_{(n)}$ is the fusion of linear-time temporal logic with multi-modal S5. The description here follows that of (Halpern and Vardi, 1989). 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

Formulas of $KL_{(n)}$ are constructed from a set of propositional symbols $\mathcal{P} = \{p, q, r, \ldots\}$, the standard propositional connectives $\neg$ (not), $\vee$ (or), $\wedge$ (and),

and $\Rightarrow$ (implies), and future-time temporal connectives including $\Diamond$ (*sometime in the future*), $\bigcirc$ (*at the next moment in time*), $\Box$ (*always*), $\mathcal{U}$ (*until*), and $\mathcal{W}$ (*unless,* or *weak until*). We interpret these temporal 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 $Ag = \{1, \ldots, n\}$, and we 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."

Formally, the set of well-formed formulas 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 the following.

$$\neg A \quad A \vee B \quad A \wedge B \quad A \Rightarrow B \quad K_i A \quad (\text{for } i \in Ag)$$
$$\Diamond A \quad \Box A \quad A\,\mathcal{U}\,B \quad A\,\mathcal{W}\,B \quad \bigcirc A$$

We define some particular classes of formulas 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* and that 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 3. A model M is a structure $M = \langle TL, R_1, \ldots, R_n, \pi \rangle$, where

- TL is a set of timelines, with a distinguished initial timeline t_0. A point q is a pair $q = (t, u)$, where $t \in TL$ and $u \in \mathbb{N}$ is a temporal index into t. Let *Points* be the set of all points.
- R_i, for all $i \in Ag$ is the agent accessibility relation over *Points*, that is, $R_i \subseteq Points \times Points$, where each R_i is an equivalence relation.
- π is a function, π: $Points \times \mathcal{P} \rightarrow \{T, F\}$, called 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 formulas in WFF_K. The rules defining this satisfaction relation are given below.

$\langle M, (t, u) \rangle \models \textbf{true}$
$\langle M, (t, u) \rangle \not\models \textbf{false}$

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

Note that even if the condition for the definition of $\Diamond$ is $u \leq u'$ rather than $u < u$, the temporal resolution rule (see (Dixon et al., 1998)) for this logic is still valid; this is one of the purposes of including $\Diamond$.

For convenience in presenting the normal form for $KL_{(n)}$ we introduce a symbol **start**, such that $\langle M, (t,u)\rangle \models$ **start** if and only if $t = t_0$ and $u = 0$.

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; that is, 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. S5 axioms are K stating that $K(\phi \Rightarrow \psi) \Rightarrow (K\phi \Rightarrow K\psi)$, T (Reflexivity) stating that $K\phi \Rightarrow \phi$, and 5 (Euclideaness) stating that $\neg K\phi \Rightarrow K\neg K\phi$. However, there are other complete axiom systems for S5, for example, using reflexivity, symmetry, and transitivity axioms. This last axiom system is the approach adopted in this paper. The system S5 is widely recognized as the logic of idealized *knowledge*, and for this reason $KL_{(n)}$ is often termed a *temporal logic of knowledge*.

2.3. NORMAL FORM

Formulas in $KL_{(n)}$ can be transformed into a normal form called separated normal form for temporal logic of knowledge, abbreviated as SNF_K (Dixon et al., 1998). In this transformation, complex subformulae are replaced by new propositions, and the truth value of these propositions is linked to the formulas they replaced in all states. In addition, non-core temporal operators are simplified. 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 *common knowledge* operator C and E (*everybody knows*) operators. The operator $\Box^*$ is defined as the maximal fixpoint of

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

Thus, the semantics for the C and E operators is

$$\langle M, (t,u)\rangle \models E\phi \quad \text{iff} \quad \forall i \in Ag,\ \langle M, (t,u)\rangle \models K_i\phi,$$

and

$$\langle M, (t,u)\rangle \models C\phi \quad \text{iff} \quad \langle M, (t',u')\rangle \models \phi \text{ if } (t',u') \text{ is modally reachable from } (t,u),$$

where modal reachability is defined as follows.

DEFINITION 4. Let M be a $KL_{(n)}$-model and (t,u), (t',u') be points in M. Then (t',u') is *modally reachable* from (t,u) iff either (i) $((t,u),(t',u')) \in R_i$ for some agent $i \in Ag$; or (ii) there exists some point (t'',u'') in M such that (t'',u'') is reachable from (t,u) and (t',u') is reachable from (t'',u'').

Essentially, the common knowledge of ϕ, that is, $C\phi$, holding in a state implies that ϕ holds in every state reachable by following any R_i relation, or in other words, everybody knows that everybody knows that everybody knows ... ϕ.

Formulas 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 given in Figure 1, 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 both modal literals $K_i l_1$ and $K_j l_2$ (or $K_i l_1$ and $\neg K_j l_2$, or $\neg K_i l_1$ and $\neg K_j l_2$) if $i \neq j$. Each K_i clause contains 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 in our notation. Similarly, for convenience the conjunction is dropped, and we consider just the set of clauses T_j.

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 and discrete model of time (isomorphic to $\mathbb{N}$ with the usual order relation, $<$, less than).

start	$\Rightarrow$	$\bigvee_{b=1}^{r} l_b$	(an *initial* clause)
$\bigwedge_{a=1}^{g} k_a$	$\Rightarrow$	$\bigcirc \bigvee_{b=1}^{r} l_b$	(a *step* clause)
$\bigwedge_{a=1}^{g} k_a$	$\Rightarrow$	$\Diamond l$	(a *sometime* 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 1. Clauses in SNF_K.

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

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

Thus,

- **true** and **false** are FOTL formulas.
- If $t_1, \ldots, t_n$ are constants or variables and P is a n-ary predicate symbol, then $P(t_1, \ldots, t_n)$ is a FOTL formula.
- If ϕ and ψ are FOTL formulas and x is an individual variable, then the following are FOTL formulas.

$$\neg\phi \quad \phi \vee \psi \quad \phi \wedge \psi \quad \phi \Rightarrow \psi \quad \forall x\phi \quad \exists x\phi$$
$$\Diamond\phi \quad \Box\phi \quad \phi\,\mathcal{U}\,\psi \quad \phi\,\mathcal{W}\,\psi \quad \bigcirc\phi$$

DEFINITION 5. A FOTL formula ϕ is called monodic if any subformula of the form $T\phi$, where T is one of $\Diamond$, $\Box$, $\bigcirc$ (or $\phi_1 T \phi_2$, where T is one of $\mathcal{U}$ or $\mathcal{W}$), contains at most one free variable.

3.2. SEMANTICS

Formulas in FOTL are interpreted in first-order temporal structures of the form $\mathfrak{M} = \langle D_n, I_n \rangle$, where D_i is a nonempty set and I_n is an interpretation of predicate and constant symbols over D_n. We make the expanding domains assumption, that is, whenever $n < m$, $D_n \subseteq D_m$.

DEFINITION 6. A (variable) assignment, $\mathfrak{a}$, is a function from the set of individual variables to $\bigcup_{n\in\mathbb{N}} D_n$. We denote the set of all variable assignments by $\mathfrak{V}$.

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 $\mathfrak{V}_n$ is a subset of the sets of all assignments,

$$\mathfrak{V}_n = \{\mathfrak{a} | \mathfrak{a}(x) \in D_n \text{ for every variable } x\}.$$

Intuitively, FOTL formulas are interpreted in sequences of such moments in time, $\mathfrak{M}_0, \mathfrak{M}_1, \ldots$ with truth values in different moments being connected by means of temporal operators.

DEFINITION 7. The truth relation $\mathfrak{M}_n \models^{\mathfrak{a}} \phi$ in a structure $\mathfrak{M}$, only for those assignments $\mathfrak{a}$ that satisfy the condition $\mathfrak{a} \in \mathfrak{V}_n$, is defined inductively in the usual way under the following understanding of the temporal operators.

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

DEFINITION 8. $\mathfrak{M}$ is a model for a formula ϕ (or ϕ is true in $\mathfrak{M}$) if there exists an assignment $\mathfrak{a}$ such that $\mathfrak{M}_0 \models^{\mathfrak{a}} \phi$.

DEFINITION 9. A formula is satisfiable if it has a model. A formula is valid if it is satisfied in any temporal structure under any assignment.

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

4.1. MOTIVATION

As mentioned earlier, $KL_{(n)}$ is a very useful logic for specifying systems involving the evolution of knowledge. Our intention is to use an existing theorem prover for first-order temporal logic in order to verify properties specified in $KL_{(n)}$. The translation of temporal logic of knowledge into the monodic fragment of first-order temporal logic is possible and has been shown in (Gabbay et al., 2003). This standard translation π'_0 is as follows, where ϕ is a $KL_{(n)}$ formula.

$$\pi'_0[\phi] = \forall x.\pi'_1(\phi, x).$$

π'_1 is defined in the following way.

$$\begin{aligned}
\pi'_1(p, x) &= P(x) \\
\pi'_1(\neg p, x) &= \neg P(x) \\
\pi'_1(\phi * \psi, x) &= \pi'_1(\phi, x) * \pi'_1(\psi, x) && \text{for } * \in \{\vee, \wedge, \Rightarrow\} \\
\pi'_1(T\phi, x) &= T\pi'_1(\phi, x) && \text{for } T \in \{\Diamond, \bigcirc\} \\
\pi'_1(\phi T \psi, x) &= \pi'_1(\phi, x) T \pi'_1(\psi, x) && \text{for } T \in \{\mathcal{U}, \mathcal{W}\} \\
\pi'_1(K_i p, x) &= \forall y(R_i(x, y) \Rightarrow P(y))
\end{aligned}$$

Here P is a unary predicate, R_i is the accessibility relation for the modal operator K_i, ϕ and ψ are formulas in $KL_{(n)}$, and T is a temporal operator. Since this relation is an equivalence relation, we should add to the translation reflexivity, symmetry, and transitivity properties of the accessibility relation, as follows.

$\forall x R_i(x, x)$	Reflexivity
$\forall x, y(R_i(x, y) \Rightarrow R_i(y, x))$	Symmetry
$\forall x, y, z(R_i(x, y) \wedge R_i(y, z) \Rightarrow R_i(x, z))$	Transitivity

However, this translation is not ideal for automated theorem proving, mainly because the transitivity axiom is included. This property is hard to handle efficiently by first-order theorem provers. The problem arises in how the orderings are dealt with for the clausal form of this formula. This makes the procedure derive more inference clauses compared to other clauses. This could lead the procedure to not terminate (see Section 7 for practical results). For this reason, we present a translation from $KL_{(n)}$ into the monodic fragment of first-order temporal logic that includes reflexivity and symmetry axioms as before but deals with transitivity in a different way. This translation is based in the *axiomatic translation principle* presented in (Schmidt and Hustadt, 2003).

4.2. THE AXIOMATIC TRANSLATION

We are interested in translating $KL_{(n)}$ formulas into the monodic fragment of first-order temporal logic. Without loss of generality we can assume that formulas are already in SNF_K normal form.

Let ϕ be a set of clauses written in the normal form SNF_K, that is,

$$\phi = \Box^* \bigwedge_j T_j.$$

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

$$\pi_0[\phi] = Q(st) \wedge \Box \bigwedge_j \forall x \pi_1(T_j, x).$$

In the following p is a literal, ϕ and ψ are formulas 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 predicate uniquely associated with $K_i p$, and R_i is the accessibility relation for the modal operator K_i. The translation π_1 is as given in Figure 2.

For each $K_i p$ we add the clauses

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

and

$$\Box(Q_{K_i p}(x) \Rightarrow (\forall y.R_i(x, y) \Rightarrow P(y))). \tag{2}$$

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

$$\Box(Q_{\neg K_i p}(x) \Rightarrow (\exists y.R_i(x, y) \wedge \neg P(y))). \tag{3}$$

$$\begin{aligned}
\pi_1(\mathbf{start}, x) &= Q(x) \\
\pi_1(\mathbf{true}, x) &= \mathbf{true} \\
\pi_1(\mathbf{false}, x) &= \mathbf{false} \\
\pi_1(p, x) &= P(x) \\
\pi_1(\neg p, x) &= \neg P(x) \\
\pi_1(\phi \vee \psi, x) &= \pi_1(\phi, x) \vee \pi_1(\psi, x) \\
\pi_1(\phi \wedge \psi, x) &= \pi_1(\phi, x) \wedge \pi_1(\psi, x) \\
\pi_1(\phi \Rightarrow \psi, x) &= \pi_1(\phi, x) \Rightarrow \pi_1(\psi, x) \\
\pi_1(\bigcirc \phi, x) &= \bigcirc \pi_1(\phi, x) \\
\pi_1(\Diamond \phi, x) &= \Diamond \pi_1(\phi, x) \\
\pi_1(K_i p, x) &= Q_{K_i p}(x) \\
\pi_1(\neg K_i p, x) &= Q_{\neg K_i p}(x)
\end{aligned}$$

Figure 2. π_1 translation.

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

$\forall x.R_i(x, x)$	Reflexivity
$\forall x, y.(R_i(x, y) \Rightarrow R_i(y, x))$	Symmetry

4.3. CORRECTNESS

In this section we show that the axiomatic translation presented in Section 4.2 is correct.

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. We first show that if ϕ is a satisfiable set of clauses in SNF_K, then $\pi_0[\phi]$ is also 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 $\pi_0[\phi]$ based on an existing $KL_{(n)}$ model of ϕ.

Let $M = \langle TL, R_1, \ldots, R_n, \pi \rangle$ be a $KL_{(n)}$ model of ϕ. We define a FOTL model $\mathfrak{M}^M = \langle D_n, I_n \rangle$ as follows. If *Points* is the set of points (t, u) over timelines in TL, then $D_n = Points$ for every $n \in \mathbb{N}$.

For every $n \in \mathbb{N}$ and every proposition symbol p occurring in $\mathcal{P}$, the interpretation I_n is defined as

$$I_n \models P(x)[x \to (t, u)] \quad \text{iff} \quad \langle M, (t, u) \rangle \models p.$$

Furthermore,

$$I_n \models Q(x)[x \to (t, u)] \quad \text{iff} \quad t = t_0 \text{ and } u = 0,$$

where Q is the new proposition symbol that we introduce in order to represent the beginning of time,

$I_n \models R_i(x, y)[x \to (t_1, u_1), y \to (t_2, u_2)]$	iff	$((t_1, u_1), (t_2, u_2)) \in R_i$,
$I_n \models Q_{K_i p}(x)[x \to (t, u)]$	iff	for all $(t', u') \in D_n$ if $I_n \models R_i(x, y)[x \to (t, u), y \to (t', u')]$, then $I_n \models P(y)[y \to (t', u')]$,
$I_n \models Q_{\neg K_i p}(x)[x \to (t, u)]$	iff	there exists $(t', u') \in D_n$ and $I_n \models R_i(x, y)[x \to (t, u), y \to (t', u')]$, and $I_n \models \neg P(y)[y \to (t', u')]$.

First we prove that for any subformula ψ of ϕ that is true at some point (t, u) in M; that is, $\langle M, (t, u) \rangle \models \psi$, then for every $n \in \mathbb{N}$, $I_n \models \pi_1(\psi, x)[x \to (t, u)]$.

The proof proceeds by induction on the structure of ψ.

- If ψ is **true**, then for every point (t, u) $\langle M, (t, u)\rangle \models$ **true** and obviously $I_n \models$ **true**$[x \to (t, u)]$.
- If ψ is **false**, then for every point (t, u) $\langle M, (t, u)\rangle \not\models$ **false** and $I_n \not\models$ **false**$[x \to (t, u)]$.
- If $\psi =$ **start**, $\langle M, (t, u)\rangle \models$ **start** if and only if $t = t_0$ and $u = 0$. We have $\pi_1($**start**$, x) = Q(x)$ and by definition of I_n, $I_n \models Q(x)[x \to (t, u)]$ if and only if $t = t_0$ and $u = 0$.
- For $p \in \mathcal{P}$, by definition $I_n \models \pi_1(p, x)[x \to (t, u)]$ if and only if $\langle M, (t, u)\rangle \models p$.
- Let ψ be $\neg p$, where p is a propositional symbol. If $\langle M, (t, u)\rangle \models \neg p$, then $\langle M, (t, u)\rangle \not\models p$, following the definition of the propositional case, $I_n \not\models P(x)[x \to (t, u)]$. So, $I_n \models \neg P(x)[x \to (t, u)]$.
- Let ψ be of the form $\sigma \vee \omega$. If $\langle M, (t, u)\rangle \models \sigma \vee \omega$, then $\langle M, (t, u)\rangle \models \sigma$ or $\langle M, (t, u)\rangle \models \omega$. By induction hypothesis, $I_n \models \sigma(x)[x \to (t, u)]$ or $I_n \models \omega(x)[x \to (t, u)]$.

The same applies for $\wedge$ and $\Rightarrow$.

- Let ψ be of the form $\bigcirc\sigma$. If $\langle M, (t, u)\rangle \models \bigcirc\sigma$, then $\langle M, (t, u + 1)\rangle \models \sigma$. Therefore by induction hypothesis and the definition of I_n, $I_n \models \pi_1(\sigma, x)[x \to (t, u + 1)]$. Since all the interpretations are the same $I_{n+1} \models \pi_1(\sigma, x)[x \to (t, u + 1)]$, that is, $I_n \models \pi_1(\bigcirc\sigma, x)[x \to (t, u)]$.
- Let ψ be of the form $\Diamond\sigma$. If $\langle M, (t, u)\rangle \models \Diamond\sigma$, then there exists an index l, $l \geq u$ such that $\langle M, (t, l)\rangle \models \sigma$. Therefore by induction hypothesis and the definition of I_n, $I_n \models \pi_1(\sigma, x)[x \to (t, l)]$. Since all the interpretations are the same, $I_l \models \pi_1(\sigma, x)[x \to (t, l)]$, that is, $I_n \models \pi_1(\Diamond\sigma, x)[x \to (t, u)]$.
- Let us now consider a formula of the form $K_i p$. If $\langle M, (t, u)\rangle \models K_i p$, then for every t' and for every $u' \in \mathbb{N}$ if $((t, u), (t', u')) \in R_i$, then $\langle M, (t', u')\rangle \models p$. By definition of I_n, $I_n \models \pi_1(p, y)[y \to (t', u')]$ if and only if $\langle M, (t', u')\rangle \models p$. Also by definition of I_n, $I_n \models R_i(x, y)[x \to (t, u), y \to (t', u')]$ since $((t, u), (t', u')) \in R_i$. $I_n \models Q_{K_i p}(x)[x \to (t, u)]$ by definition of I_n. Thus, $I_n \models \pi_1(K_i p, x)[x \to (t, u)]$.
- 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 $\langle M, (t', u')\rangle \not\models p$, that is, $\langle M, (t', u')\rangle \models \neg p$. By definition of I_n, $I_n \models Q_{\neg K_i p}(x)[x \to (t, u)]$ and $I_n \models \pi_1(\neg p, y)[y \to (t', u')]$. Therefore, $I_n \models \pi_1(\neg K_i p, x)[x \to (t, u)]$.

So far, we have shown that for every clause T_j in SNF_K if $\langle M, (t, u)\rangle \models T_j$, then $I_n \models \pi_1(T_j, x)[x \to (t, u)]$. If $\phi = \Box^* \bigwedge_j T_j$ is satisfiable in M, then it means that for all points $(t, u) \in$ *Points* $\langle M, (t, u)\rangle \models T_j$. We have shown that if $\langle M, (t, u)\rangle \models T_j$, then $I_n \models \pi_1(T_j, x)[x \to (t, u)]$. Since we have defined *Points* $= D_n$ for all $n \in \mathbb{N}$, we can deduce that $\Box \bigwedge_j (\forall x(\pi_1(T_j, x)))$.

Next we consider a set of SNF_K clauses, 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 $\pi_0[\phi]$, then $\mathfrak{M}_\circ \models \pi_0[\phi]$, where $\pi_0[\phi] = Q(st) \wedge \Box \bigwedge_j \forall x \pi_1(T_j, x)$. Therefore $\mathfrak{M}_\circ \models Q(st)$ and $\mathfrak{M}_\circ \models \forall x \pi_1(T_j, x)$ for all j and for all $i \in \mathbb{N}$, that is, $\mathfrak{M}_i \models \pi_1(T_j, x)[x \to d]$ for all $d \in D_0$.

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

- We define $(t_0, 0) = st$. If $d \in D_n$, then we define a point as (d, n).
- We construct the timelines in the following way.
  For every $d \in \bigcup_{n \geq 0} D_n$, let $min(d) = n$ if and only if $d \in D_n$ and either $n = 0$ 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 $(d, n) = t^d_{n-min(d)}$, which assures us that the timeline has an initial moment.

  As set of states S we use the set $\{(d, n) \mid d \in D_n\}$. With each element d of $\bigcup_{n \geq 0} D_n$ we associate a timeline t^d.

  Since we are assuming the case of expanding domains, if $d \in D_n$ then $d \in D_m$ for $m \geq n$. This, together with the fact that we define the timelines in such a way that they have an initial moment, ensures 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 $I_n \models R_i(x, y)[x \to d, y \to d']$. Let R_i be the transitive closure of R'_i.
- The valuation π: *Points* $\times \mathcal{P} \to \{T, F\}$ is defined in a similar way as we defined I_n. Let (t, u) be a point corresponding to (d, n). If p is a proposition symbol then we define π as follows.

$$\pi((t, u), p) = \begin{cases} \textbf{true} & \text{if } I_n \models P(x)[x \to (t, u)], \\ \textbf{false} & \text{if } I_n \not\models P(x)[x \to (t, u)]. \end{cases}$$

In the following we consider every single case of a clause T_j in the normal form such that $\pi_1(T_j, x)$ is satisfiable in $\mathfrak{M}_\circ$. To show that ϕ is satisfiable in M, we must show that the clause T_j is true at every point (t, i), for $i \in \mathbb{N}$, in the $KL_{(n)}$ model we have constructed. Let (t, i) correspond to (d, n).

- Let T_j be a clause of the form $\textbf{start} \Rightarrow \bigvee^m_{j=1} l_j$. We know that $\mathfrak{M}_\circ \models \forall x \pi_1(\textbf{start} \Rightarrow \bigvee^m_{j=1} l_j, x)$, that is, $\mathfrak{M}_\circ \models (Q(x) \Rightarrow \bigvee^m_{j=1} L_j(x))[x \to d]$ for some $d \in D_0$.

  Recall that Q is the new predicate symbol introduced in order to represent the beginning of time. We know that $\mathfrak{M}_\circ \models Q(st)$, so there is a $d \in D_0$ such that $I_0 \models Q(x)[x \to d]$. Let d correspond to $(t_0, 0)$ by definition and by construction of the model M, $\langle M, (t_0, 0) \rangle \models \bigvee^m_{j=1} L_j$.
- Let T_j be a clause of the form $\bigwedge^g_{a=1} k_a \Rightarrow \bigcirc \bigvee^r_{j=1} l_j$. We know that $\mathfrak{M}_n \models \forall x \pi_1(\bigwedge^g_{a=1} k_a \Rightarrow \bigcirc \bigvee^m_{j=1} l_j, x)$, that is, $\mathfrak{M}_n \models (\bigwedge^g_{a=1} K_a(x) \Rightarrow$

$\bigcirc \bigvee_{j=1}^{m} L_j(x))[x \to d]$ for all $d \in D_i$. This means $I_n \models \bigwedge_{a=1}^{g} \pi_1(k_a, x) \Rightarrow \bigcirc \bigvee_{j=1}^{m} \pi_1(l_j, x)[x \to d]$. So either $I_n \not\models \bigwedge_{a=1}^{g} \pi_1(k_a, x)[x \to d]$ or $I_{n+1} \models \bigvee_{j=1}^{m} \pi_1(l_j, x)[x \to d]$. By definition of the $KL_{(n)}$ model M we have constructed, $(d, n+1)$ corresponds to $(t, i+1)$, and either $\langle M, (t, i)\rangle \not\models \bigwedge_{a=1}^{g} k_a$ or $\langle M, (t, i+1)\rangle \models \bigvee_{j=1}^{m} l_j$ is true.

- The same applies for clauses of the form $\bigwedge_{a=1}^{g} k_a \Rightarrow \Diamond l$ 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 all m_{ib} are modal literals.

  We know that $\mathfrak{M}_{\mathfrak{n}_\circ} \models \pi_1(\mathbf{true} \Rightarrow K_i p_1 \vee \cdots \vee K_i p_r, x)[x \to d_0]$. Let (d_0, n_0) correspond to (t, u).

  So,

$$I_{n_0} \models (\mathbf{true} \Rightarrow Q_{K_i p_1}(x) \vee \cdots \vee Q_{K_i p_r}(x))[x \to d_0], \tag{4}$$

and

$$I_{n_0} \models (Q_{K_i p_1}(x) \vee \cdots \vee Q_{K_i p_r}(x))[x \to d_0]. \tag{5}$$

In addition, we have the following.

$$I_{n_0} \models (Q_{K_i p_1}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow Q_{K_i p_1}(y)))[x \to d_0] \tag{6a}$$
$$I_{n_0} \models (Q_{K_i p_2}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow Q_{K_i p_2}(y)))[x \to d_0] \tag{6b}$$
$$\vdots$$
$$I_{n_0} \models (Q_{K_i p_r}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow Q_{K_i p_r}(y)))[x \to d_0] \tag{6c}$$
$$I_{n_0} \models (Q_{K_i p_1}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow P_1(y)))[x \to d_0] \tag{7a}$$
$$I_{n_0} \models (Q_{K_i p_2}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow P_2(y)))[x \to d_0] \tag{7b}$$
$$\vdots$$
$$I_{n_0} \models (Q_{K_i p_r}(x) \Rightarrow (\forall y R_i(x, y) \Rightarrow P_r(y)))[x \to d_0] \tag{7c}$$
$$I_{n_0} \models \forall x R_i(x, x) \tag{8}$$
$$I_{n_0} \models \forall x, y R_i(x, y) \Rightarrow R_i(y, x). \tag{9}$$

If clause 5 holds, it means that for some index l, $1 \le l \le r$, $I_{n_0} \models Q_{K_i p_l}(x)[x \to d_0]$. From clauses 8 and 9 we know that R_i is reflexive and symmetric. From clauses 6 whenever there is a sequence $d_0, d_1, \ldots, d_n$ such that $(d_i, d_{i+1}) \in R_i$, for all $i \le n-1$, then $I_{n_0} \models Q_{K_i p_l}(x)[x \to d_i]$ and $I_{n_0} \models Q_{K_i p_l}(y)[y \to d_{i+1}]$. So $I_{n_0} \models Q_{K_i p_l}(y)[y \to d_1]$ for all $d_1 \in \{d | (d_0, d) \in (R_i \cup \breve{R}_i)\}$, where $R_i \cup \breve{R}_i$ is the reflexive, symmetric, and transitive closure of R_i. By clauses 7 and 8, if $I_{n_0} \models Q_{K_i p_l}(x)[x \to d]$, then $I_{n_0} \models P_l(x)[x \to d]$.

By the 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 $d_1 \in \{d|(d_0, d) \in (R_i \cup \breve{R}_i)\}$, by construction of M $\langle M, (t', u')\rangle \models p_l$, where (t', u') corresponds to $d_1 \in \{d|(d_0, d) \in (R_i \cup \breve{R}_i)\}$. Thus, $\langle M, (t, u)\rangle \models K_i p_l$.

Since

$$\langle M, (t, u)\rangle \models \mathbf{true} \Rightarrow K_i p_1 \vee \cdots \vee K_i p_r$$

iff

$$\langle M, (t, u)\rangle \models K_i p_1 \vee \cdots \vee K_i p_r$$

iff there exists l, $1 \leq l \leq r$, such that $\langle M, (t, u)\rangle \models K_i p_l$, that is, for all t' and for all u' if $((t, u), (t', u')) \in R_i$, then $\langle M, (t', u')\rangle \models p_l$. □

4.4. GUARDED MONODIC FRAGMENT

The guarded monodic fragment of FOTL is decidable (Hodkinson, 2000). For this reason, in this section, we investigate whether the translation we presented in Section 4 falls into this FOTL fragment.

DEFINITION 10 [Guarded Monodic Fragment]. The formulas of the *guarded monodic fragment GMF* are inductively defined as follows:

(1) If A is an atom, then A is in GMF, where $t_1, \ldots, t_n, s, t$ are constants or variables.
(2) GMF is closed under Boolean combinations.
(3) If $\phi \in$ GMF and G is an atom, for which every free variable of ϕ is among the arguments of G, then $\forall \overline{x}(G \Rightarrow \phi) \in$ GMF and $\exists \overline{x}(G \wedge \phi) \in$ GMF, for every sequence $\overline{x}$ of variables. The atom G is called a *guard*.
(4) If $\phi(x) \in$ GMF and $\phi(x)$ contains at most one free variable, then $\bigcirc\phi(x) \in$ GMF, $\Box\phi(x) \in$ GMF, and $\Diamond\phi(x) \in$ GMF.
(5) If $\phi(x) \in$ GMF and $\phi(x)$ contains exactly one free variable x, then $\forall x\phi(x)$ and $\exists x\phi(x)$ are in GMF.

NOTE 1. Although the standard definition of the guarded fragment (see, for example, (Grädel, 1999)) does not contain item 5, its addition does not extend the notion of the guarded fragment; see (de Nivelle, 2000; Degtyarev et al., 2003).

Examining the axiomatic translation π_0 presented in Section 4.2 shows that, under this definition of the guarded fragment, for any formula ψ in $KL_{(n)}$we can apply a satisfiability preserving translation to ψ obtaining ϕ such that ϕ is in SNF_K,

and the result of the transformation, $\pi_0(\phi)$, is a guarded monodic formula. The class of guarded monodic formulae is decidable (Hodkinson, 2000) and, moreover, a decision procedure based on temporal resolution exists (Degtyarev et al., 2003).

5. Example

In the following we show the validity of the formula $\phi = (\Box Kp \Rightarrow \bigcirc Kp)$ using the approach described above. First, we negate ϕ and transform it into SNF_K.

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

We anchor it to t_0.

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

Thus, the normal of form $\neg\phi$ is the following set of SNF_K.

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

According to the translation defined in Section 4, for every Kp we add some new clauses.

$$(Q_{Kp}(x) \Rightarrow (\forall y.R(x, y) \Rightarrow Q_{Kp}(y)))$$
$$(Q_{Kp}(x) \Rightarrow (\forall y.R(x, y) \Rightarrow P(y)))$$

For $\neg Kp$ we add the clause

$$Q_{\neg Kp} \Rightarrow (\exists y R_i(x, y) \wedge \pi_1(\neg p, y)).$$

After applying π_1, the resulting clauses are as follows.

$$\begin{aligned} \mathbf{true} &\Rightarrow \neg Q(x) \vee T_0(x) \\ \mathbf{true} &\Rightarrow \neg T_1(x) \vee Q_{K_i} p(x) \\ \mathbf{true} &\Rightarrow \neg T_0(x) \vee T_1(x) \\ \mathbf{true} &\Rightarrow \neg T_0(x) \vee T_3(x) \\ T_3(x) &\Rightarrow \bigcirc T_1(x) \\ T_3(x) &\Rightarrow \bigcirc T_3(x) \\ T_0(x) &\Rightarrow \bigcirc T_2(x) \\ \mathbf{true} &\Rightarrow \neg T_2(x) \vee Q_{\neg K_i} p(x) \\ Q_{Kp}(x) &\Rightarrow (\forall y.R(x, y) \Rightarrow Q_{Kp}(y)) \\ Q_{Kp}(x) &\Rightarrow (\forall y.R(x, y) \Rightarrow P(y)) \\ Q_{\neg Kp}(x) &\Rightarrow (R(x, skolem(x))) \\ Q_{\neg Kp}(x) &\Rightarrow \neg P(skolem(x))) \end{aligned}$$

Then we apply π_0, obtaining, as the final translation, the following.

$$
\begin{aligned}
& Q(st) \\
\forall x(\mathbf{true} &\Rightarrow \neg Q(x) \vee T_0(x)) \\
\forall x(\mathbf{true} &\Rightarrow \neg T_1(x) \vee Q_{K_i} p(x)) \\
\forall x(\mathbf{true} &\Rightarrow \neg T_0(x) \vee T_1(x)) \\
\forall x(\mathbf{true} &\Rightarrow \neg T_0(x) \vee T_3(x)) \\
\forall x(T_3(x) &\Rightarrow \bigcirc T_1(x)) \\
\forall x(T_3(x) &\Rightarrow \bigcirc T_3(x)) \\
\forall x(T_0(x) &\Rightarrow \bigcirc T_2(x)) \\
\forall x(\mathbf{true} &\Rightarrow \neg T_2(x) \vee Q_{\neg Kp}(x)) \\
\forall x(Q_{Kp}(x) &\Rightarrow (\forall y.R(x, y) \Rightarrow Q_{Kp}(y))) \\
\forall x(Q_{Kp}(x) &\Rightarrow (\forall y.R(x, y) \Rightarrow P(y))) \\
\forall x(Q_{\neg K_p}(x) &\Rightarrow (R(x, skolem(x)))) \\
\forall x(Q_{\neg Kp}(x) &\Rightarrow \neg P(skolem(x))))
\end{aligned}
$$

Together with the latter set of clauses, as we mentioned in Section 4, we must add reflexivity and symmetry properties. There is only a K operator in this example; therefore we add the following.

$\forall x.R(x, x)$	Reflexivity
$\forall x, y.(R(x, y) \Rightarrow R(y, x))$	Symmetry

6. Experimental Results

6.1. TeMP

The main motivation for investigating variations of the standard translation of $KL_{(n)}$ to monodic FOTL (Section 4) is our interest in using TeMP (Hustadt et al., 2004), a theorem prover for the monodic fragment of FOTL. TeMP is based on a clausal resolution calculus for the monodic fragment of FOTL with expanding domains (Konev et al., 2003). TeMP implements the fine-grained temporal resolution calculus (Konev et al., 2005), which in turn makes the monodic temporal resolution work (Degtyarev et al., 2003) practical. The implementation of TeMP uses the fact that inference steps in this calculus can be simulated by inference steps in a first-order ordered resolution calculus. In particular, TeMP uses the theorem prover Vampire (Riazanov and Voronkov, 2002; Voronkov, 1995) as an efficient implementation of first-order resolution.

At present TeMP is the only automated theorem prover for the monodic fragment of FOTL.

In the following section we describe several case studies that have been formalized in $KL_{(n)}$. We apply the axiomatic translation from Section 4 and use TeMP, described in Section 6.1, to carry out the proofs.

6.2. CLUEDO

Cluedo is a board game, commercially produced by Hasbro (Cluedo, 1946), where players gather information about a murder. The suspects, murder 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. To achieve this, they use the knowledge of their own cards, the knowledge obtained from cards that other players revealed during the game, or statements that another player does not have such a card. Players take turns to make a *suggestion*: a 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, he declares it, and the player to his left must try to show one of the cards to the suggesting player. This process 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 murderer, weapon, and room, then he makes an *accusation* and checks the cards kept aside. If the player is right, this player wins the game. If the player is wrong, this player cannot make further suggestions and accusations but can answer the suggestions from other players.

The 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, abbreviated as c, w, and j, respectively.

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 the Miss Scarlett card.
- g_i is true if player i holds the Rev. Green card.
- y_i is true if player i holds the Col. Mustard card.
- b_i is true if player i holds the Prof. Plum card.
- l_i is true if player i holds the lead piping card.
- s_i is true if player i holds the spanner card.
- v_i is true if player i holds the revolver card.
- p_i is true if player i holds the rope card.

We denote that a suspect is the murderer, or a weapon is the murder 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 murder weapon.
- s_m is true if spanner is the murder weapon.
- v_m is true if revolver is the murder weapon.
- p_m is true if rope is the murder weapon.

We assume that time zero occurs before the deal is made. At time one the deal has taken place and the first player (Catherine, in our case) makes a suggestion. At time two the second player makes a suggestion and so on.

As an example, we assume that at time one the following deal has been made

Player	Catherine	Wendy	Jane	Murder Hand
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 had their turn to show a card and no cards are revealed, Catherine is expected to make an accusation. Since no accusation is made by Catherine, Jane and Wendy can deduce that Catherine holds one of these cards (i.e., Miss Scarlett or lead piping), which is specified in $KL_{(n)}$ as follows, where $i \in \{c, j, w\}$.

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

   At this point Catherine should also be able to deduce that the lead piping is the murder weapon. This statement is written in $KL_{(n)}$ as

   $$\bigcirc \bigcirc K_c l_m.$$

3. At time three Wendy makes the suggestion lead piping and Col. Mustard. Jane shows Wendy the Col. Mustard card. Thus, at time three, Catherine knows Jane holds either the lead piping or Col. Mustard, that is, $\bigcirc \bigcirc \bigcirc K_c(l_j \vee y_j)$. 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, that is,

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

Table I. Results by TeMP for various Cluedo problems

Case	SNF_K clauses	Clauses in FOTL syntax	Clauses generated	Time
1	83	143	476	0.047 s
2	101	193	977	0.074 s
3	123	236	5707	1.246 s
4	120	237	7926	0.640 s
5	84	147	460	0.029 s
6	95	172	548	0.035 s

4. We have also proved that from time one onwards Catherine knows that Miss Scarlett is not the murderer, that is,

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

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" (instead of what was suggested in part 2), 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 obtained by using TeMP in order to prove the above statements are presented in Table I.

6.3. MUDDY CHILDREN

We consider the *muddy children problem*, a well-known problem concerning reasoning about knowledge. We use a 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 own 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 and truthful, and they answer simultaneously, what will it happen?

Table II. Results by TeMP

Case	SNF_K clauses	Clauses in FOTL syntax	Clauses generated	Time
1	26	69	271	0.015 s
2	23	61	186	0.029 s

There is a "proof" that the first $k-1$ times he asks the question, they will say "No," but then the kth time the children with muddy foreheads will all answer "Yes."

We consider the case of only two children, in order to make the problem simpler. The translation of this example into $KL_{(n)}$ can be seen in (Dixon et al., 1998). This formalization of the problem makes time explicit. 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, that is,

$$\Box(m_1 \vee m_2)$$

then we could prove the following statements.

1. If initially both children's foreheads are muddy, then we can prove that at time two both children know they are muddy, that is,

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

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

$$\bigcirc K_1 m_1.$$

The results are presented in Table II.

6.4. THE NEEDHAM–SCHROEDER PROTOCOL WITH PUBLIC KEYS

The well-known Needham–Schroeder communication 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 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	$A \rightarrow B$:	$\{N_B\}_{pub\_key(B)}$

Message contents of the form $\{X, Y\}_{pub\_key(Z)}$ represent messages containing both X and Y 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 the message 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 could be sure he is talking to B because only B "should" be able to decrypt Message 1. In the same way, B has reason to be sure that he is talking to A because 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 in (Dixon et al., 2003, 2004)). We use a first-order notation, but we assume finite sets of agents, keys, and so forth, so it is essentially propositional. We use the following syntactic conventions. Let M_1 and M_2 be variables over messages, *Key* be a variable over keys, N_1 be a variable over nonces, and $X, Y, \ldots$ 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:

- $send(X, Msg, Key)$ is satisfied if agent X sends message *Msg* encrypted by *Key*;
- $rcv(X, Msg, Key)$ is satisfied if agent X 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 of the public key of X is V;
- $val\_priv\_key(X, V)$ is satisfied if the value of the private key of X is V;

- *val_nonce*(N_1, V) is satisfied if the value of nonce N_1 is V; and
- *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 the following.

- Structural assumptions concerning keys and message contents. For example,

$$\forall X, Key, M_1.send(X, M_1, Key) \Rightarrow \neg contains(M1, priv\_key(X))$$

  – agents will not reveal their private key to others.
- Scenario assumptions concerned with specifying this particular protocol. For example,

$$\begin{aligned}\forall X, Y, Z.\ & (contains(m_1, X) \Leftrightarrow ((X = A) \vee (X = N_A)))\wedge \\ & (contains(m_2, Y) \Leftrightarrow ((Y = N_A) \vee (Y = N_B)))\wedge \\ & (contains(m_3, Z) \Leftrightarrow (Z = N_B))\end{aligned}$$

  – message m_1 contains only N_A and A, message m_2 contains only N_B and N_A, and message m_3 contains only N_B.
- Basic knowledge axioms concerned with the agents knowledge of keys, nonces, and so forth.

$$\forall X, N, V.\ K_X val\_nonce(N, V) \Rightarrow \bigcirc K_X val\_nonce(N, V)$$

  – agents never forget nonces they know.
- Communication axioms concerned with sending and receiving messages.

$$\begin{aligned}\forall X, M_1, N_1\ & \bigcirc((Msg(M_1) \wedge contains(M_1, N_1)) \Rightarrow \\ & (\exists V_1 K_X val\_nonce(N_1, V_1)\ \Leftrightarrow \\ & \circ\,[K_X val\_nonce(N_1, V_1) \\ & \quad \vee(\exists Y.\exists V.rcv(X, M_1, pub\_key(Y)) \\ & \quad \wedge K_X val\_priv\_key(Y, V))])\end{aligned}$$

  – for all moments except the first moment, if M_1 is a message that contains N_1, an agent knows the content of N_1 either if it already knew the content of N_1 or if it received an encrypted version of M_1 that it could decode.

  Here, $\circ$ is a temporal operator meaning the previous moment in time.

Using TeMP, after the translation in Section 4 has been applied, we can prove the following statements related to this protocol.

1. B's knowledge on receipt of N_A
   Once B receives a nonce encoded by B's public key, then B knows the value of that nonce.

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

$$\Box(rcv(B, m1, pub\_key(B)) \Rightarrow \bigcirc\exists V.\ K_B val\_nonce(N_A, V)),$$

   where m_1 is a message.

Table III. Results for the Needham–Schroeder protocol

Case	SNF_K clauses	Clauses in FOTL syntax	Clauses generated	Time
1	105	297	7425	0.733 s
2	58	102	306	0.048 s
3	168	269	20577	3.516 s
4	167	267	76277	8.020 s

2. Confirmation of B's knowledge
   Once A receives m_2 (*which, in turn, contains* N_A)*, it can infer that B knows* N_A, that is,

$$rcv(A, m_2, pub\_key(A)) \Rightarrow \bigcirc \exists V\, K_A K_B val\_nonce(N_A, V),$$

   where m_2 is a message.
3. C's ignorance
   C will never know that the value of A's nonce is a_n.
   This statement is specified in $KL_{(n)}$ as

$$\Box \neg K_C val\_nonce(N_A, a_n).$$

4. Reception of the nonce
   C will never receive the value of the nonce a_n

$$\Box \neg rcv(C, m_1, pub\_key(C)),$$

   where the nonce of A, N_A is contained in m_1.

The results obtained by applying TeMP are stated in Table III.

Remarks. In this section we presented some examples and showed the usefulness and practicality of the translation presented in Section 4. For this purpose, all the cases we have tested by using TeMP are unsatisfiable; that is, TeMP shows the unsatisfiability of the negation of the statement we want to prove (we recall TeMP is a resolution-based theorem prover). We have not included any case where the set of clauses given to TeMP as an input were satisfiable because we consider that the unsatisfiable cases are more meaningful for the nature of the examples presented. Testing for satisfiability will take longer for TeMP because all the models need to be checked, whereas for the unsatisfiable cases, as long as one of the models is not satisfied, the proof terminates.

By observing the results in Tables I, II, and III, one can see that there are cases with a very large number of clauses generated compared to other cases from the same example. This is due, mainly, to the type of resolution operation that TeMP performs for these cases. For them *temporal resolution* (Fisher et al., 2001) needs

to be carried out. Temporal resolution takes place between clauses that occur at different moments in time, since this is the most complex part of the resolution method.

7. Comparisons with the Standard Translation

In Section 4 we presented the standard translation from temporal logic of knowledge into the monodic fragment of first-order temporal logic and we mentioned that this translation was not ideal for the purpose of automated theorem proving. Thus, we now present the results obtained by applying TeMP to the same examples we have shown in Section 6 but using the standard translation.

Cluedo. The results obtained are shown in Table IV. Comparing results in Table I and the ones in Table IV, we observe that using the standard translation, TeMP performs faster for cases 1 and 6. However, it does not produce any result for cases 2, 3, and 4, and it is slower for case 5. Thus, the usage of the axiomatic translation presented in Section 4 improves the performance of TeMP because these three cases (2, 3, and 4) can be verified, which did not happen by using the standard translation.

Muddy Children. The results are similar to those in the previous case, if we compare the results obtained in Tables V and VI, even though in cases where TeMP on the standard translation is faster. It fails on the other case.

The Needham–Schroeder Protocol with Public Keys. Again, by using the standard translation one can prove the second statement faster than by using the translation presented in this paper. However, with the standard translation we could not prove the other statements.

Overall, these experiments seem to indicate that the axiomatic translation presented in this paper improves the robustness and reliability of the prover. TeMP

Table IV. Results obtained by TeMP for the Cluedo example using the standard translation

Case	Number of clauses generated	Time in seconds
1	313	0.019
2	–	Does not terminate
3	–	Does not terminate
4	–	Does not terminate
5	309	0.034
6	351	0.023

Table V. Results obtained by TeMP for the muddy children example using the standard translation

Case	Number of clauses generated	Time in seconds
1	–	It does not terminate
2	162	0.009

Table VI. Results obtained by TeMP for the Needham–Schroeder protocol with public keys example using the standard translation

Case	Number of clauses generated	Time in seconds
1	–	It does not terminate
2	214	0.011
3	–	It does not terminate
4	–	It does not terminate

finds proofs in a reasonable time in more cases than for the standard translation. As we mentioned earlier, the inclusion of transitivity causes some problems when using theorem provers. This is, mainly, due to the fact that a large amount of clauses may be generated by resolution, which in some cases may lead the process not to terminate. However, for the examples where there are not many clauses derived from transitivity, the standard translation might be faster because, in general, this translation provides fewer input clauses than does the translation presented in this paper.

8. Conclusions

In this paper we have presented a translation from a temporal logic of knowledge, $KL_{(n)}$, into the monodic fragment of first-order temporal logic and proved its correctness. This translation allows us to use a recently developed monodic first-order temporal logic theorem prover in order to verify a variety of properties specified in $KL_{(n)}$.

We have illustrated the usefulness and practicality of this translation by formalizing a variety of examples in $KL_{(n)}$ and proving related properties by applying our translation and using the theorem prover TeMP. All these examples had previously been proven by hand.

According to the results presented in Section 7, the standard translation from $KL_{(n)}$ into monodic FOTL does not always allow us to prove certain properties

when using the theorem prover TeMP, whereas by using the translation presented in this paper significantly more properties can be proved.

By carrying out the experimental results in Section 6 we have also been able to detect some flaws and modify certain axioms stated in (Dixon et al., 2003) for the Needham–Schroeder security protocol.

As mentioned earlier, the monodic fragment of FOTL can be used as a unifying framework for other logics such as temporal logics of knowledge and belief, spatio-temporal logics or temporal description logics. Thus, we intend to develop translations from these logics into the monodic fragment of FOTL. These translations, as shown for the translation presented in this paper, will allow us to use TeMP in order to test case studies formalized in these logics.

References

Artale, A. and Franconi, E. (1999) Introducing temporal description logics, in C. Dixon and M. Fisher (eds.), *Proceedings of the 6th International Workshop on Temporal Representation and Reasoning (TIME-99)*, IEEE Computer Society Press, Orlando, FL.

Chomicki, J. and Niwinski, D. (1995) On the feasibility of checking temporal integrity constraints, *J. Comput. System Sci.* **51**(3), 523–535.

Cluedo. `http://www.hasbro.com`.

Degtyarev, A., Fisher, M. and Konev, B. (in press) Monodic temporal resolution, *ACM Trans. Computational Logic*. A preliminary version available as Technical Report ULCS-03-001, The University of Liverpool, 2003, `http://www.csc.liv.ac.uk/research`

Degtyarev, A., Fisher, M. and Konev, B. (2003) Handling equality in monodic temporal resolution, in *Proceedings of 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, Lecture Notes in Comput. Sci. 2850, Springer, Almaty, Kazakhstan, pp. 214–228.

de Nivelle, H. (2000) An overview of resolution decision procedures, in M. Faller, S. Kaufmann and M. Pauly (eds.), *Formalizing the Dynamics of Information*, CSLI Publications 91, Stanford University, Palo Alto, CA, pp. 115–130.

Dixon, C. (2005) Using temporal logics of knowledge for specification and verification – a case study, *Journal of Applied Logic*, Elsevier Science, to appear.

Dixon, C., Fernández-Gago, M. C., Fisher, M. and van der Hoek, W. (2003) Using temporal logics of knowledge in the formal verification of security protocols, Technical Report ULCS-03-022, `http://www.csc.liv.ac.uk/research/techreports`

Dixon, C., Fernández-Gago, M. C., Fisher, M. and van der Hoek, W. (2004) Using temporal logics of knowledge in the formal verification of security protocols, in *Proceedings of TIME2004*, IEEE, Computer Society Press.

Dixon, C., Fisher, M. and Wooldridge, M. (1998) Resolution for temporal logics of knowledge, *J. Logic Comput.* **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., Halpern, J. Y., Moses, Y. and Vardi, M. Y. (1995) *Reasoning about Knowledge*, MIT Press.

Fisher, M. (1997) A normal form for temporal logic and its application in theorem-proving and execution, *J. Logic Comput.* **7**(4), 429–456.

Fisher, M., Dixon, C. and Peim, P. (2001) Clausal temporal resolution, *Trans. Comput. Logic* **2**(1), 12–56.

Fisher, M. and Wooldridge, M. (1997) On the formal specification and verification of multi-agent systems, *Internat. J. Cooperative Information Systems* **6**(1), 37–65.

Gabbay, D., Kurusz, A., Wolter, F. and Zakharyaschev, M. (2003) *Many-Dimensional Modal Logics: Theory and Applications*, Elsevier.

Gabelaia, D., Kontchakov, R., Kurucz, A., Wolter, F. and Zakharyaschev, M. (2003) On the computational complexity of spatio-temporal logics, in *Proceedings of the 16th International Florida Artificial Intelligence Research Symposium Conference (FLAIRS 2003)*, AAAI press, 460–464.

Grädel, E. (1999) On the restraining power of guards, *J. Symbolic Logic* **64**, 1719–1742.

Halpern, J. Y. (1987) Using reasoning about knowledge to analyze distributed systems, *Annual Rev. Comput. Sci.* **2**, 37–68.

Halpern, J. Y. and Vardi, M. Y. (1989) The Complexity of Reasoning about Knowledge and Time. I Lower Bounds, *J. Comput. System Sci.* **38**, 195–237.

Hodkinson, I. (2000) Monodic packed fragment with equality is decidable, *Studia Logica* **72**, 185–197.

Hodkinson, I., Wolter, F. and Zakharyaschev, M. (2000) Decidable fragments of first-order temporal logic, *Ann. Pure Appl. Logic* **106**, 85–134.

Hustadt, U., Konev, B., Riazanov, A. and Voronkov, A. (2004) TeMP: A temporal monodic prover, Technical Report 04-004, ULCS. `http://www.csc.liv.ac.uk/research`

Hustadt, U. and Schmidt, R. A. (2001) Formulae which highlight differences between temporal logic and dynamic logic provers, in E. Giunchiglia and F. Massacci (eds.), *Issues in the Design and Experimental Evaluation of Systems for Modal and Temporal Logics*, Dipartimento di Ingegneria dell'Informazione, Unversitá degli Studi di Siena, Siena, Italy, pp. 68–76.

Konev, B., Degtyarev, A., Dixon, C., Fisher, M. and Hustadt, U. (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)*.

Konev, B., Degtyarev, A., Dixon, C., Fisher, M. and Hustadt, U. (2005, to appear) Mechanizing first-order temporal resolution, *Inform. and Comput.*, Elsevier Science.

Manna, Z. and Pnueli, A. (1992) *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Springer, New York.

Meyer, J. J. C. and van der Hoek, W. (1995) *Epistemic Logic for Computer Science and Artificial Intelligence*, Cambridge Tracts Theoret. Comput. Sci. 41.

Needham, R. and Schroeder, M. (1978) Using encryption for authentication in large networks of computers, *Comm. ACM* **21**, 993–999.

Plaisted, D. A. and Greenbaum, S. A. (1986) A structure-preserving clause form translation, *J. Symbolic Comput.* **2**(3), 293–304.

Riazanov, A. and Voronkov, A. (2002) The design and implementation of Vampire, *Artificial Intelligence Commun.* **15**(2–3), 91–110.

Schmidt, R. A. and Hustadt, U. (2003) A principle for incorporating axioms into the first-order translation of modal formulae, in *Automated Deduction – CADE-19*, Lecture Notes in Artificial Intelligence, 2741, Springer, pp. 412–426.

Syverson, P. (1993) Adding time to a logic of authentication, in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, ACM Press, pp. 97–101.

Voronkov, A. (1995) The anatomy of Vampire, *J. Automat. Reason.* **15**(2), 237–265.