# On Incompleteness of Multi-dimensional First-order Temporal Logics

David Toman

Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
E-mail: david@uwaterloo.ca

## Abstract

*In this paper we show that first-order temporal logics form a proper expressiveness hierarchy with respect to dimensionality and quantifier depth of temporal connectives. This result resolves (negatively) the open question concerning the existence of an expressively complete first-order temporal logic, even when allowing multi-dimensional temporal connectives.*

## 1 Introduction

We study the expressive power of multi-dimensional first-order temporal logics and their relationship to two-sorted first-order logic. In particular, we are interested in the following result:

> Two-sorted first-order logic is strictly more expressive than *any* fixed-dimensional first-order temporal logic with a finite set of temporal connectives.

The paper proves this claim even when only finite temporal structures are considered. To obtain the result we combine results of Bidoit et al. [3, 4] on order independent properties definable using standard temporal logic with results of Toman and Niwinski [16] on multi-dimensional temporal logics over dense linear order of time.

Interest in expressively complete temporal logics was, in the temporal database community, originally motivated by (unsuccessful) attempts to define expressively complete temporal relational algebras closed over the timestamp or bi-temporal data models [8, 5] in order to implement expressively complete temporal query languages, such as SQL/Temporal [13] or SQL/TP [14, 15]. The result presented in this paper, however, is a general result on temporal logics and equally applies to problems in the area of knowledge representation.

The paper is organized as follows: Section 2 provides the necessary background and definitions. Section 3 introduces (appropriate extensions of) results in [3, 4, 16] needed to prove the claims in this paper. Section 4 gives the main result and Section 5 concludes and discusses directions for future research.

## 2 Definitions

In this section we give basic definitions of temporal structures and temporal query languages. The notation and definitions are based on the development in the chapter *Temporal Logic in Information Systems* [7].

**Temporal Structures.** Temporal structures (databases) are built from the following three basic building blocks:

1. a structure $\langle T; \rho\rangle$ to serve as the *temporal domain*, where $\rho$ usually stands for a binary predicate denoting linear order ($\leq$) on $T$. We also consider (cf. Section 4) temporal domains equipped with equality ($=$) only.
2. a structure $\langle D; =\rangle$ to serve as the *data domain* of the structure.
3. a set of single-sorted predicate symbols $\langle r_1, \ldots, r_m\rangle$; the arity of the symbol $r_i$ is $v_i$. This choice defines the *database schema* for our temporal structure.

In the rest of this section we use $\sigma$ for the signature $\langle r_1, \ldots, r_m\rangle$.

**Definition 2.1 (Temporal Structure)** *Let $T$ be a temporal domain, $D$ a data domain, and $\sigma = \langle r_1, \ldots, r_m\rangle$ a database schema. We define $R_i$ to be a two-sorted predicate symbol of the sort $T \times D^{v_i}$ for each $r_i$ in $\sigma$. We call $R_i$ the* temporal extension of $r_i$.

*A* temporal extension of $\sigma$ *defined as*

$$\tau = \langle \rho, =, R_1, \dots, R_m \rangle$$

*is a two-sorted signature composed of the signature of the temporal domain, $\rho$, the signature of the data domain, $=$, and the temporal extensions $R_i$ of the predicate symbols $r_i$ in $\sigma$.*

*We define* a temporal structure $A$ *to be a two-sorted $\tau$-structure*

$$A = \langle\, T, \rho^T; D, =^D; R_1^A, \dots, R_m^A \,\rangle.$$

The instances $R_i^A$ of $R_i$ in $A$ define the interpretation of the symbols $r_i$ in the database schema for every element of the time domain, formally:

$$r_i(c_1, \dots, c_{v_i}) \text{ holds at time } t \text{ iff } A \models R_i(t, c_1, \dots, c_{v_i})$$

for $r_i \in \sigma$, $t \in T$, and $c_j \in D, 0 < j \leq v_i$. This observation links the above definition with a Kripke-style definition of temporal structures commonly used to define semantics for temporal logics; [7] shows that these two approaches are equivalent. Note that the interpretation of the predicate symbols connected solely with the temporal ($\rho$) and data domains ($=$) is fixed, while the interpretation of the symbols $R_i$ depends on the database instance. In practice we often require the instances of the relational symbols $r_i$ to be *finite* or finite at every time instant. The later is equivalent to requiring the sets $\{(x_1, \dots, x_{v_i}) : A \models R_i(t, x_1, \dots, x_{v_i})\}$ are finite for every $t \in T$ and $0 < i \leq m$.

**Temporal Queries.** First-order properties of temporal structures can be captured by sentences (closed formulas) in an appropriate *temporal query language*. We introduce two principal ways of temporalizing a first-order query language (first order logic) over $\sigma$-structures. The first approach (often referred to as the *timestamp* language) introduces explicit *temporal variables*, *relationships between these variables* (e.g., order), and *quantifiers* to the language. The result is a two-sorted variant of first-order logic over $\tau$ (the temporal extension of $\sigma$).

**Definition 2.2** (2-FOL) *Let $M$ be the set of all formulas defined by the following BNF rule:*

$$\begin{array}{rcl} M & ::= & R(t_i, x_1, \dots, x_k) \\ & | & M \wedge M \\ & | & \neg M \\ & | & x_i = x_j \\ & | & \exists x_i.M \\ & | & t_i \rho t_j \\ & | & \exists t_i.M \end{array}$$

*We call $t_i$ a temporal variable and $x_i$ a data variable. A* 2-FOL($\rho$) query *is a formula in the language defined by the productions for $M$ above. A temporal* 2-FOL($\rho$) *property is a closed* 2-FOL($\rho$) *query.*

The semantics of formulas in this language is the standard first-order (Tarskian) semantics with respect to $\tau$-structures. Note that the database schema is *monadic* with respect to the sort $T$, i.e., the predicate symbols in the database schema have always exactly one distinguished argument of sort $T$. In the technical development we use the following syntactic property of 2-FOL formulas.

**Definition 2.3 (Quantifier Depth)** *We define function* $\mathsf{qd}$ : 2-FOL $\to \mathbf{N}$

1. *If $\varphi$ is atomic then* $\mathsf{qd}(\varphi) = 0$.
2. *If $\varphi$ is $\neg\phi$ then* $\mathsf{qd}(\varphi) = \mathsf{qd}(\phi)$.
3. *If $\varphi$ is $\phi_1 \wedge \phi_2$ then* $\mathsf{qd}(\varphi) = \max\{\mathsf{qd}(\phi_1), \mathsf{qd}(\phi_2)\}$.
4. *If $\varphi$ is $\exists x_i.\phi$ or $\exists t_i.\phi$ then* $\mathsf{qd}(\varphi) = \mathsf{qd}(\phi) + 1$.

The second approach to developing a temporal query language on top of first-order logic uses *implicit* temporal connectives to link truth of formulas with an *evaluation point*. In the case of *multi-dimensional first-order temporal logics* (FOTL) the generalized evaluation point is a vector in the multiple time dimensions. Temporal connectives can be defined by formulas in the language of the temporal domain extended with additional *place-holders* standing for FOTL formulas (to be "connected" by the connective).

**Definition 2.4 (Temporal Connective)** *Let $\rho$ be a relation symbol in the signature of the temporal domain. We define a set of formulas*

$$\begin{array}{rcl} C & ::= & X_j \\ & | & C \wedge C \\ & | & \neg C \\ & | & t_i \rho t_j \\ & | & \exists t_i.C. \end{array}$$

*A $k$*-dimensional *$l$*-ary temporal connective $\omega(X_1, \dots, X_l)$ *defined with respect to the signature of the temporal sort $\langle\, \rho \,\rangle$ is a formula defined by the productions for $C$ with free variables $t_0, \dots, t_{k-1}$ and place-holders $X_1, \dots, X_l$.*

The restriction placed on free variables of a $k$-dimensional temporal connective yields the intuitive behavior: the free variables stand for the generalized evaluation point of the whole connective; the formula that defines the connective "expects" the (sub-)formulas substituted for the place-holders $X_i$ to have the same set of (implicit) free variables.

Note that there is no restriction on the number of the place-holders $X_i$.

**Example 2.5** We can express the standard linear-time temporal connectives [11] in the temporal signature of linear order as follows:

$$\begin{aligned}
X_1 \ \mathbf{until}\ X_2 &\triangleq \exists t_2.t_0 < t_2 \wedge (\exists t_0.t_0 = t_2 \wedge X_2) \wedge \\
&\quad \forall t_1(t_0 < t_1 < t_2 \to (\exists t_0.t_0 = t_1 \wedge X_1) \\
\Diamond X_1 &\triangleq \exists t_1.t_0 < t_1 \wedge (\exists t_0.t_0 = t_1 \wedge X_1) \\
\Box X_1 &\triangleq \forall t_1.t_0 < t_1 \to (\exists t_0.t_0 = t_1 \wedge X_1)
\end{aligned}$$

Similarly, we can express the past temporal connectives **since**, ◆, and ■. Note that, in order to simplify further notation, we have added subformulas that yield a proper *renaming* of variables,

$$\exists t_0.t_1 = t_0 \wedge X,$$

to the otherwise standard definitions. Thus, the FOTL formulas rooted by such connectives are expected to have a single (implicit) temporal variable named $t_0$ free. The *renaming* subformulas guarantee that names of temporal variables match correctly when FOTL formulas are embedded into 2-FOL (cf. Definition 2.8).

**Example 2.6** In Section 4 we use the following connectives to show separation between layers of multi-dimensional temporal logics.

$$\begin{aligned}
&\Diamond_k X \equiv \exists t_0, \ldots, t_k.X \\
&\downarrow_0 X \equiv X \\
&\downarrow_1 X \equiv \exists t_0.t_0 = t_1 \wedge X \\
&\qquad\vdots \\
&\downarrow_k X \equiv \exists t_0.t_0 = t_k \wedge X
\end{aligned}$$

**Definition 2.7 (Temporal Logic)** *Let $\Omega_k^\rho$ be a finite set of $k$-dimensional connectives defined over the signature of the temporal sort $\langle\ \rho\ \rangle$. We define a set of formulas*

$$\begin{array}{rcl}
T & ::= & r(x_1, \ldots, x_k) \\
& | & T \wedge T \\
& | & \neg T \\
& | & x_i = x_j \\
& | & \exists x_i.T \\
& | & \omega(T_1, \ldots, T_k)
\end{array}$$

*for $\omega \in \Omega_k^\rho$. A* $\mathrm{FOTL}(\Omega_k^\rho)$ query *is a formula in the language defined by the productions for $T$. A* $\mathrm{FOTL}(\Omega_k^\rho)$ property *is a closed* $\mathrm{FOTL}(\Omega_k^\rho)$ *query.*

It is easy to see that all the FOTL formulas can be naturally embedded into the language of 2-FOL formulas. This embedding also defines the semantics of the FOTL-formulas relatively to the semantics of 2-FOL:

**Definition 2.8 (Embedding of FOTL in 2-FOL)** *Let* $\mathsf{Em}$ *be a mapping of formulas in the language* $\mathrm{FOTL}(\Omega_k^\rho)$ *to the language* 2-FOL($\rho$) *defined as follows:*

$$\begin{array}{lcl}
\mathsf{Em}(r_i(x_1, \ldots, x_{v_i})) & = & R_i(t_0, x_1, \ldots, x_{v_i}) \\
\mathsf{Em}(x_i = x_j) & = & x_i = x_j \\
\mathsf{Em}(F_1 \wedge F_2) & = & \mathsf{Em}(F_1) \wedge \mathsf{Em}(F_2) \\
\mathsf{Em}(\neg F) & = & \neg\, \mathsf{Em}(F) \\
\mathsf{Em}(\exists x.F) & = & \exists x.\, \mathsf{Em}(F) \\
\mathsf{Em}(\omega(F_1, \ldots, F_k)) & = & \omega^*(\mathsf{Em}(F_1), \ldots, \mathsf{Em}(F_k))
\end{array}$$

*where $\omega^*$ is the (C-)formula denoted by $\omega$ in $\Omega_k^\rho$.*

Note that embeddings of closed $\mathrm{FOTL}(\Omega_k^\rho)$ formulas yield 2-FOL formulas with free variables $t_0, \ldots, t_{k-1}$. Thus, the meaning of such formulas in a $\tau$- structure is defined with respect to an *evaluation point* $\vec{0} = (0, \ldots, 0)$. However, since there is no restriction on the content of the set $\Omega_k^\rho$, we can simulate various alternative approaches, e.g., requiring the formula $\varphi$ to be true with respect to an arbitrary evaluation point can be achieved using the $\Diamond_k$ operator introduced in Example 2.6 and writing the formula $\Diamond_k \varphi$. Similarly, requiring the formula to be true with respect to all evaluation points is equivalent to writing the formula $\neg\Diamond_k\neg\varphi$.

Thus the choice of an fixed evaluation point does not affect temporal properties expressible in $\mathrm{FOTL}(\Omega_k^\rho)$. This observation also reconciles the difference between closed formulas in FOTL and 2-FOL: sentences in 2-FOL can also be evaluated with respect to the $\vec{0}$ evaluation point, since an assignment to the $t_0, \ldots, t_{k-1}$ variables cannot change the truth value of a closed formula.

The quantifier depth of FOTL formulas is defined as the quantifier depth of their embeddings to 2-FOL. Similarly, the quantifier depth of temporal connectives is defined as the quantifier depth of their embeddings into 2-FOL, assuming the place-holders $X_i$ stand for atomic formulas. Given a finite set of temporal connectives $\Omega$ we define $\mathsf{qd}(\Omega) = \max\{\mathsf{qd}(\omega) : \omega \in \Omega\}$. It is also easy to see that all formulas in $\mathrm{FOTL}(\Omega_k)$ can be considered to be formulas of $\mathrm{FOTL}(\Omega_{k'})$ for any $k' \geq k$.

## 3 Background Results

The results in this paper depend crucially on the following two results.

### 3.1 Games for $\mathrm{FOTL}(\Omega_k^\rho)$

The first result relates to a modified version of *Ehrenfeucht-Fraïssé Games* [10] that precisely captures properties definable by First-order Temporal Logic $\mathrm{FOTL}(\Omega_k^\rho)$ with an arbitrary set of temporal connectives $\Omega_k$ such that $\mathsf{qd}(\Omega_k^\rho) \leq m$. The game is a natural extension of games for single-dimensional temporal logic used by Toman and Niwinski [16] to show separation of $\mathrm{FOTL}(\Omega_1^{\leq})$ from 2-FOL($\leq$) in the case of dense ordering of the temporal domain.

**Definition 3.1 (Game States)** *Let $A$ and $B$ be two $\tau$-structures.*

*A* state *$S$ of size $k$ is a triple consisting of three vectors of size $k$:*

1. *a vector $\vec{v} = (v_0, \ldots, v_k)$ of temporal and data variable names,*
2. *a vector $\vec{a} = (a_0, \ldots, a_k)$ of values from the domains of the structure $A$, and*
3. *a vector $\vec{b} = (b_0, \ldots, b_k)$ from domains of $B$.*

*We require that the elements $a_i$ and $b_i$ are elements drawn from the temporal domain of the appropriate structure whenever $v_i$ is a name of a temporal variable and data elements whenever $v_i$ is a name of a data variable.*

*Contiguous sub-sequences of temporal variable names in $S$ (or more precisely, in the first component of $S$) are grouped into disjoint* blocks *of size up to $m$.*

*Variable names $v_i$ and $v_j$ are* compatible in *$S$ if either of the following condition holds.*

- *both $v_i$ and $v_j$ are both (names of) data variables;*
- *$v_i$ is a data variable, $v_j$ is a temporal variable (named) $t_0$, and $v_j$ that occurs after $v_i$ in $S$;*
- *$v_i$ is a data variable, $v_j$ is a temporal variable $t_0$, such that $v_j$ is the last occurrence of $t_0$ before $v_i$ in $S$;*
- *both $v_i$ and $v_j$ are temporal variables in the same block of temporal variables in $S$; or*
- *both $v_i$ and $v_j$ are temporal variables, and $v_i$ is the last occurrence of one of $t_0, \ldots, t_{k-1}$ before $v_j$ in $S$.*

The intuition behind the definition of moves *compatible* in a state derives from the syntactical structure of the 2-FOL images of FOTL formulas.

**Example 3.2** Consider the $\mathrm{FOTL}(\Omega_2^{=})$ formula

$$\Diamond_2(\exists x.p(x)) \wedge \downarrow_1 (\exists y.q(y) \wedge \downarrow_2 (\exists z.r(z) \vee r(y)))$$

where $\Diamond_2$, $\downarrow_1$, and $\downarrow_2$ are temporal connectives introduced in Example 2.6. The embedding of this formula into 2-FOL is as follows:

$$\begin{array}{l}\exists t_0, t_1, t_2.(\exists x.P(t_0, x)) \wedge \\ (\exists t_0.t_0 = t_1 \wedge \exists y.Q(t_0, y) \wedge \\ (\exists t_0.t_0 = t_2 \wedge \exists x.R(t_0, z) \vee R(t_0, y)))\end{array}$$

It is easy to see that the variable $t_0$ introduced by the $\Diamond_2$ connective is only *visible* outside the scopes of the $\downarrow_1$ and $\downarrow_2$ connectives. Thus only data variables quantified outside these connectives can appear together with this instance of the variable $t_0$ in an atomic predicate, e.g., $P(t_0, x)$. The remaining data variables, namely, $y$ and $z$ are not *compatible* with this instance of $t_0$, since they appear within scope of another temporal connective that defines *another instance of* $t_0$.

Similar observation can be made for the variable $z$ and the instance of $t_0$ introduced by $\downarrow_1$. Moreover, for the same reason, within the (embedding of the) $\downarrow_2$ connective, the *original* variable $t_0$ is no longer visible since another $t_0$ was introduced by $\downarrow_1$. Thus, the variable $t_0$ introduced by $\Diamond_2$ is not compatible with any of the temporal variables introduced by, e.g., $\downarrow_2$.

Since moves in a game correspond to nesting of quantifiers in a formula, the above observations tell us that we only have to consider *compatible* moves in the game when defining a winning condition for duplicator

**Definition 3.3 (Winning Game State)** *A state $S$ is a* winning state *for* duplicator *if,*

*(1)* $R^A(a_{i_0}, a_{i_1}, \ldots, a_{i_{v_k}}) \iff R^B(b_{i_0}, b_{i_1}, \ldots, b_{i_{v_k}})$;

*(2)* $a_i \rho a_j \iff b_i \rho b_j$; *and*

*(3)* $a_i = a_j \iff b_i = b_j$;

*whenever the variable names corresponding to the values used as arguments of predicate symbols in (1)-(3) are pairwise compatible in $S$, and assuming the values used as arguments of predicates belong to the appropriate sorts.*

**Definition 3.4 (Game)** *A* move *is an extension of a given state $S$ of size $k$ as follows:*

1. *player I* (spoiler) *chooses a variable name $v_{k+1}$ and an element of the appropriate sort from the domains of $A$ or $B$.* [1]
2. *player II* (duplicator) *then chooses an element of the same sort from the domains of the other structure.*

*The values $v_{k+1}$, $a_{k+1}$, and $b_{k+1}$) are used to extend the three components of $S$ yielding a new state of size $k+1$. A* game of $n$ moves starting from a given state $S$ *consists of extending $S$ by $n$ moves.*

*Duplicator* wins a game of $n$ moves starting from state $S$ *if, after $n$ moves the game ends in a winning state. Spoiler wins otherwise.*

*Duplicator has* a winning strategy for games of $n$ moves starting from a state $S$ *if he wins every game of $n$ moves that starts in $S$.*

Note that whenever the duplicator wins a game of $n$ moves, starting from a state $S$, he also wins all games of length up to $n$. In other words, when following a winning strategy for the game, the duplicator always moves from a winning state to another winning state.

**Lemma 3.5** *A winning strategy for duplicator for games of length $n$ starting from a state $S = (\vec{v}, \vec{a}, \vec{b})$ and with blocks of temporal variables bounded by $m$ defines an equivalence relation $A, \vec{a} \sim^{\vec{v}}_{m,n} B, \vec{b}$ over the class of $\tau$-structures..*

This equivalence relation captures classes of $\tau$-structures indistinguishable by formulas in $\mathrm{FOTL}(\Omega^{\rho}_{k})$ of limited quantifier depth.

**Proposition 3.6** *Let $A$ and $B$ be two $\tau$-structures, $\Omega^{\rho}_{k}$ a finite set of $k$-dimensional temporal connectives such that $\mathsf{qd}(\Omega^{\rho}_{k}) \leq m$, $\vec{v}$ a vector of variable names $(t_0, \ldots, t_{k-1})$, and $\vec{0} = (0, \ldots, 0)$. Then*

$$A, \vec{0} \sim^{\vec{v}}_{m,n} B, \vec{0} \text{ if and only if } A, \vec{0} \models \varphi \iff B, \vec{0} \models \varphi$$

*for all temporal properties $\varphi \in \mathrm{FOTL}(\Omega^{\rho}_{k})$ such that $\mathsf{qd}(\varphi) \leq n$.*

P r o o f. (sketch) The proof is similar to the proof in the first-order case: games won by the spoiler correspond to formulas separating $A$ from $B$. The weakening of the winning condition for duplicator is equivalent to observing which variable names can appear together in atomic (sub-)formulas of embeddings of $\mathrm{FOTL}(\Omega^{\rho}_{k})$ formulas into 2-FOL. Thus, formulas constructed from games won by the spoiler yield $\mathrm{FOTL}(\Omega^{\rho}_{k})$ formulas (or, more precisely, their embeddings to 2-FOL) such that $\mathsf{qd}(\Omega^{\rho}_{k}) \leq m$, that distinguish $A$ from $B$. On the other hand, given a formula $\varphi \in \mathrm{FOTL}(\Omega^{\rho}_{k})$ that separates $A$ from $B$, we can construct a win for the spoiler of length at most $n$ that obeys the variable compatibility conditions. □

[1] Spoiler can also decide whether a block of temporal variables should end at this point. However, there is no advantage to the spoiler to choose blocks of size less than $m$. Thus we can assume that the blocks only end after they reach the maximal size allowed in the game.

**Corollary 3.7** *A temporal property $\varphi$ cannot be expressed by $\mathrm{FOTL}(\Omega^{\rho}_{k})$, $\mathsf{qd}(\Omega^{\rho}_{k}) \leq m$, if and only if for all $n \in N$ we can find $A_n$ and $B_n$ such that*

1. *$A_n, \vec{0} \sim^{\vec{v}}_{m,k+n} B_n, \vec{0}$,*
2. *$A_n, \vec{0} \models \varphi$, and $B_n, \vec{0} \not\models \varphi$.*

*where $\vec{v} = (t_0, \ldots, t_{k-1})$.*

### 3.2 Order-independent Temporal Properties

The game-based techniques are very general. However, their application to discrete linearly ordered structures, such as the finite temporal structures considered in this paper, is rather difficult. The main difficulty is in finding a winning strategy for duplicator in the presence of discrete order. Indeed, the separation result [16] crucially depends on the use of *dense order* of the time domain (and the use of infinite temporal structures). Abiteboul et al. [1] avoided the difficulty by using a different technique based on communication protocols. However, their technique doesn't seem to generalize to multi-dimensional temporal logics. Recently, however, Bidoit et al. [3, 4] made a crucial observation that sidesteps the difficulties associated with discrete order of the temporal domain.

**Definition 3.8 (Order-independent Property)** *Let $\varphi$ be a closed formula in* 2-FOL. *We say that $\varphi$ defines an* order-independent *property if for all temporal structures* $\mathrm{T}$ *and* $\mathrm{T}'$ *that differ only in the linear ordering of time instants we have* $\mathrm{T} \models \varphi \iff \mathrm{T}' \models \varphi$.

Then, using a variant of Craig's Interpolation Theorem [6, 9], the expressive power of order independent temporal properties is related to Ehrenfeucht-Fraïssé Games *that consider only equality* on the temporal structure. We extend the technique to $k$-dimensional temporal logics. First an extension of the interpolation theorem:

**Proposition 3.9** *Let $\Omega^{\leq}_{k}$ be a finite set of temporal connectives over a linear order $\langle \leq \rangle$ and $\varphi \in \mathrm{FOTL}(\Omega^{\leq}_{k})$ a formula expressing an order-independent property. Then there is a set of connectives defined only using equality $\langle = \rangle$ on* *the temporal structure,* $\Omega_k^=$ *and a formula* $\psi \in \mathrm{FOTL}(\Omega_k^=)$ *such that* $\psi \equiv \varphi$.

P r o o f. (sketch) Let $\varepsilon(R)$ be a sentence expressing that a binary relation $R$ is linear order and $\varphi'(R)$ the formula $\mathsf{Em}(\varphi)$ in which all occurrences of the symbol $\leq$ have been replaced by the binary symbol $R$.

The rest follows from a direct extension of Craig's Interpolation Theorem [9], since, for order-independent properties, we have

$$\varepsilon(R) \wedge \varphi'(R) \models \varepsilon(S) \to \varphi'(S)$$

for $R, S$ two binary symbols not in $\varphi$. The rest is similar to the proof used by Chang and Keisler [6] pages 87–89; we only have to observe that the interpolant is indeed an $\mathrm{FOTL}(\Omega_k^=)$ formula. □

## 4 Inexpressibility Results

We use the following property to separate $\mathrm{FOTL}(\Omega_k^\leq)$ from 2-FOL($\leq$).

**Definition 4.1 (Cover Property)** *We define*

$$\mathsf{cover}_k \equiv \exists t_0, \ldots, t_k . \forall x . R(t_0, x) \vee \ldots \vee R(t_k, x);$$

*we call* $\mathsf{cover}_k$ *the* cover property of level $k$.

The $\mathsf{cover}_k$ property asks if there are $k+1$ time instants such that the data values related by the relation $R$ to these time instants (*snapshots*) "cover" the data domain of the underlying temporal structure. Note that the cover properties can also be expressed by the following range-restricted [2] formula:

$$\exists t_0, \ldots, t_k . \forall x . (\exists t . R(t, x)) \to (R(t_0, x) \vee \ldots \vee R(t_k, x)).$$

First we show that the cover property $\mathsf{cover}_k$ can also be defined in $k+1$-dimensional temporal logic.

**Lemma 4.2** *There is a temporal property* $\mathsf{cover}_k^{\mathrm{TL}} \in \mathrm{FOTL}(\Omega_{k+1}^=)$ *such that*

$$A, \vec{0} \models \mathsf{cover}_k^{\mathrm{TL}} \iff A \models \mathsf{cover}_k$$

*for any* $\tau$*-structure* $A$*. Moreover, the quantifier depth of the connectives* $\mathsf{qd}(\Omega_{k+1}^=) = k+1$.

P r o o f. We define

$$\mathsf{cover}_k^{\mathrm{TL}} \equiv \Diamond_k \forall x . (\downarrow_0 r(x)) \vee \ldots \vee (\downarrow_k r(x))$$

where $\downarrow_0, \ldots, \downarrow_k$, and $\Diamond_k$ are $k+1$-dimensional connectives introduced in Example 2.6. □

In the rest of this section we show that $\mathsf{cover}_{k+m}^{\mathrm{TL}} \notin \mathrm{FOTL}(\Omega_k^\leq)$ for any set of temporal connectives $\Omega_k^\leq$ such that $\mathsf{qd}(\Omega_k^\leq) \leq m$. We use the following pairs of temporal structures containing a single unary relational symbol $r$ to show our separation results.

$$A_{k,n} = \langle\, \mathbf{Z}, \leq; \{1, \ldots, (k+1)n\}, =; R \,\rangle$$
$$B_{k,n} = \langle\, \mathbf{Z}, \leq; \{0, \ldots, (k+1)n\}, =; R \,\rangle$$

where $\langle\, \mathbf{Z}, \leq \,\rangle$ is the linear order on integers and where the instances of the relation $r$ are defined by

$$R = \{(i + mj, v) : v \in S_i, 1 \leq i \leq m, 0 \leq j < n\}$$

for $S_1, \ldots, S_m$ the enumeration of all $n$ element subsets of the data domain of the underlying structure.

Note that in both $A_{k,n}$ and $B_{k,n}$ the instances of the predicate symbol $R$ are *finite*. Thus, the temporal domain of the structures can be restricted to a finite sets of time instants corresponding to non-empty snapshots[2] of $R$ (and the state 0), yielding finite $\tau$-structures. Such a restriction does not affect the following results.

**Lemma 4.3** *Let* $\Omega_k^=$ *be a set of temporal connectives such that that* $\mathsf{qd}(\Omega_k^=) \leq m$*. Then* $\mathsf{cover}_{m+k}^{\mathrm{TL}} \notin \mathrm{FOTL}(\Omega_k^=)$.

P r o o f. It is easy to see that for any $m, n, k$ we have

$$A_{m+k,n}, \vec{0} \models \mathsf{cover}_{m+k}^{\mathrm{TL}} \quad \text{while} \quad B_{m+k,n}, \vec{0} \not\models \mathsf{cover}_{m+k}^{\mathrm{TL}} .$$

We now show that temporal properties in $\mathrm{FOTL}(\Omega_k^=)$ with quantifier depth at most $n$ cannot separate the above temporal structures $A_{m+k,n}$ and $B_{m+k,n}$. We show this by defining a winning strategy for duplicator in the

$$A_{m+k,n}, \vec{0} \sim_{m,n}^{\vec{v}} B_{m+k,n}, \vec{0} \;\text{ where }\; v = (t_0, \ldots, t_{k-1})$$

game and then appealing to Corollary 3.7.

The game starts in an initial state $(\vec{v}, \vec{0}, \vec{0})$. The following strategy shows that duplicator can play $n$ moves always ending in a winning state (clearly, the initial state is a winning state for the duplicator). During the course of the game the duplicator maintains an invariant guaranteeing a winning strategy.

Let $v_{i_1}, \ldots, v_{i_l}$ be all temporal variables in $S$ compatible with the *next move*[3]. For each of these variables we define a snapshot in the corresponding structure as follows:

$$P_{i_j}^A = \{x : A \models R(a_{i_j}, x)\}$$
$$P_{i_j}^B = \{x : B \models R(b_{i_j}, x)\}$$

[2] A snapshot of $P$ at time $t$ is the set $\{x : P(t, x)\}$.

[3] Note that we can consider pairs of variables only, since all relations in the signatures of the $\tau$-structures used in this proof are binary.

Note that the elements $a_{i_j}$ and $b_{i_j}$ associated with $v_{i_j}$ in $S$ are from the appropriate temporal sort of $A$ and $B$, respectively. Also, the cardinality of each of the $P^A_{i_j}$ and $P^B_{i_j}$ is $n$. The invariant for the game is defined by the following two rules:

1. $a_i \in P^A_{i_j} \iff b_i \in P^B_{i_j}$ for all $i$ such that $v_i$ is a data variable in $S$ and $i_j \in \{i_1, \ldots, i_l\}$;
2. $|\bigcap_{i_j \in I} P^A_{i_j}| = |\bigcap_{i_j \in I} P^B_{i_j}|$ for all $I \subseteq \{i_1, \ldots, i_l\}$.

In other words, we require that, in both the structures, all values associated with the data variables in the current state belong to the same snapshots associated with the compatible temporal variables and that the cardinalities of intersections of the snapshots associated with an arbitrary subset of the compatible temporal variables are the same.

This invariant is used to guarantee that all states of the game are winning states for duplicator. Since the initial state $(\vec{v}, \vec{0}, \vec{0})$ is a winning state for the duplicator that satisfies the above conditions (trivially), the following strategy preserves the invariant through $n$ game moves. There are two cases to consider.

1. A temporal move $t_i$: There can be at most $m + k - 1$ temporal variables compatible with the current move and at most $n - 1$ data variables in the current state. Assume, without loss of generality, that the spoiler has chosen a value $a_i \in T^A$. The duplicator then replies as follows:
   - If the value $a_i$ is equal to a value associated with any of the temporal variables compatible with this move, the duplicator chooses the corresponding value in the other structure.
   - Otherwise, the new move defines a distinct snapshot $P^A_i$. The duplicator must choose $b_i \in T^B$ such that the snapshot $P^B_i$ preserves the above defined invariant. This, however, is always possible since all $n$-element subsets of the respective data domains are present as snapshots in the structures and there can be at most $m+k-1$ compatible temporal moves (thus the data domains must contain at least $n$ elements outside of the union of all the snapshots).

   The case in which the spoiler chooses $b_i \in T^B$ is symmetric to the above case.

2. A data move $x_i$: There are $k$ temporal variables compatible with this move (defining $k$ snapshots) and at most $n - 1$ data variables in the current state. Again, we consider the case in which the spoiler chooses $a_i \in D^A$. The duplicator replies as follows:
   - If the value $a_i$ is associated with any of the data variables in the current state, duplicator picks the corresponding value from the current state.
   - Otherwise, if the value belongs to
   $$\bigcap_{i_j \in I} P^A_{i_j} \text{ for a maximal } I \subseteq \{i_1, \ldots, i_l\}$$
   the duplicator chooses an arbitrary value that belongs to
   $$\bigcap_{i_j \in I} P^B_{i_j} \text{ for the same } I \subseteq \{i_1, \ldots, i_l\};$$
   this is always possible since the intersections have the same cardinality in both structures.
   - Otherwise the value does not belong to any of the snapshots $P^A_{i_j}$ and the duplicator thus picks a value not belonging to any of the snapshots $P^B_{i_j}$ in the other structure. Again, this is always possible, since the set of such data values must contain at least $n$ values in both of the structures.

   The case in which the duplicator chooses $b_i \in D^B$ is again symmetric.

In all the cases, the resulting state is a winning state for the duplicator for games of up to $n$ moves starting from the state $(\vec{v}, \vec{0}, \vec{0})$. □

Now we are ready to apply the results of Bidoit et al. [3, 4] to extend this result to ordered temporal domains.

**Lemma 4.4** $\mathsf{cover}_k$ *is order-independent.*

P r o o f. Immediate from definition of order-independence (Def. 3.8). □

Combining the results of Corollary 3.7, Proposition 3.9, and Lemmas 4.3 and 4.4 we obtain the desired result.

**Theorem 4.5** *Let $\Omega^{\le}_k$ be a finite set of temporal connectives defined over a discrete linear order $\le$. Then there is a finite set $\Omega^{\le}_{k'}$, $\mathsf{qd}(\Omega^{\le}_{k'}) = k'$, such that $\mathrm{FOTL}(\Omega^{\le}_k)$ is strictly weaker than $\mathrm{FOTL}(\Omega^{\le}_{k'})$ for $k' \ge \max\{k, \mathsf{qd}(\Omega^{\le}_k)\} + 1$.*

This result holds for finite and discrete (integer-like) flows of time. Dense flows of time (with time instants modeled by rational numbers) were considered by Toman and Niwinski [16]. However, the construction of the structures $A_{k,n}$ and $B_{k,n}$ and the associated game for FOTL gives an alternative proof for dense linear order as well.

**Corollary 4.6** $\mathrm{FOTL}(\Omega_k^{\leq})$ *is strictly weaker than* 2-FOL$(\leq)$ *for any* $k > 0$ *and any finite set of temporal connectives* $\Omega_k^{\leq}$.

## 5 Conclusion

We have shown that there cannot be a fixed-dimensional and expressively complete temporal logic. This fact also precludes the use of fixed-dimensional temporal relational algebras [8], e.g., algebras based on the *bi-temporal data model* [12], to implement expressively complete temporal query languages based on 2-FOL, e.g., SQL/TP [14, 15].

### 5.1 Future Work

While the results in this paper achieve our main goal—showing that fixed-dimensional temporal logics are strictly weaker than two-sorted first order logic, no matter what finite set of $k$-dimensional connectives is used—the results are not quite satisfactory (tight enough): using the approach in this paper, the standard one-dimensional FOTL is separated from 2-FOL by the $\mathsf{cover}_3$ property. The results in [1, 3, 4, 16] can be easily modified to show the separation using $\mathsf{cover}_1$. We conjecture the following:

**Conjecture 5.1** $\mathsf{cover}_k$ *is not expressible in* $\mathrm{FOTL}(\Omega_k^{\leq})$ *independently of the quantifier depth of connectives in* $\Omega$.

Another direction of research considers more complex structures of time (the language and theory of the temporal domain), e.g., temporal domains equipped with distance measure, periodic sets, etc. For the separation results to hold, we have to extend Theorem 3.9 to reduce temporal connectives in the extended language of the temporal domain to connectives defined only using equality for temporal properties preserved under permutations of the temporal domain. The separation results, however, can only apply to temporal domains whose theories cannot encode pairs of time instants as a single instant (i.e., theories that cannot define pairs and projections).

## Acknowledgments

The author gratefully acknowledge the Natural Sciences and Engineering Research Council of Canada, the Canadian Foundation for Innovation, the Communications and Information Technology of Ontario, and Nortel Networks Ltd. for their support of this research.

## References

[1] S. Abiteboul, L. Herr, and J. Van den Bussche. Temporal Versus First-Order Logic to Query Temporal Databases. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 49–57, 1996.

[2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[3] N. Bidoit, S. de Amo, and L. Segoufin. Propriétés Temporelles Indpéndantes de l'ordre. In *17émes Journées de Bases de Données Avancées*, pages 219–225, 2001.

[4] N. Bidoit, S. de Amo, and L. Segoufin. Order Independent Temporal Properties. Technical report, `http://www.deamo.prof.ufu.br/arquivos/journalLC.ps`, 2002. (to appear in *Journal of Logic and Computation*).

[5] M. H. Böhlen, C. S. Jensen, and R. T. Snodgrass. Temporal Statement Modifiers. *ACM Transactions on Database Systems*, 25(4):407–456, 2000.

[6] C. C. Chang and H. J. Keisler. *Model Theory, 3rd ed.* Studies in Logic and Foundations of Mathematics, vol. 73. Elsevier Science Publishers, 1985.

[7] J. Chomicki and D. Toman. Temporal Logic in Information Systems. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 31–70. Kluwer, 1998.

[8] J. Clifford, A. Croker, and A. Tuzhilin. On Completeness of Historical Relational Query Languages. *ACM Transactions on Database Systems*, 19(1):64–116, 1994.

[9] W. Craig. Three uses of the Herbrand-Genzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, 22:269–285, 1957.

[10] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[11] D. M. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.

[12] C. S. Jensen, R. T. Snodgrass, and M. D. Soo. The TSQL2 Data Model. In *The TSQL2 Temporal Query Language*, pages 153–238. Kluwer Academic Publishers, 1995.

[13] R. T. Snodgrass, M. H. Böhlen, C. S. Jensen, and A. Steiner. Adding valid time to sql/temporal. ISO/IEC JTC1/SC21/WG3 DBL MAD-146r2 21/11/96, (change proposal), International Organization for Standardization, 1996.

[14] D. Toman. Point-based Temporal Extensions of SQL. In *International Conference on Deductive and Object-Oriented Databases*, pages 103–121, 1997.

[15] D. Toman. SQL/TP: A Temporal Extension of SQL. In Kuper, Libkin, and Paredaens, editors, *Constraint Databases*, chapter 19, pages 391–399. Springer Verlag, 2000.

[16] D. Toman and D. Niwinski. First-Order Queries over Temporal Databases Inexpressible in Temporal Logic. In *Advances in Database Technology, EDBT'96*, volume 1057, pages 307–324, 1996.