Order Independent Temporal Properties

Nicole Bidoit, Sandra de Amo, Luc Segoufin

Order independent temporal properties

Nicole Bidoit [*] Sandra De Amo [†] Luc Segoufin [‡]

July 2, 2020

Contact : Nicole Bidoit
LRI UMR 8623 CNRS - Bat 490
Université de Paris Sud,
91405 Orsay Cedex, France
email : bidoit@lri.fr

Abstract

The paper investigates temporal properties that are invariant with respect to the temporal ordering and that are expressible by temporal query languages either explicit like FO($\leq$) or implicit like TL. In the case of an explicit time representation, these "order invariant" temporal properties are simply those expressible in the language FO($=$). In the case of an implicit time representation, we introduce a new language, TL(E^i) that captures exactly these properties. The expressive power of the language TL(E^i) is characterized via a game à la Ehrenfeucht-Fraïssé.

This provides another proof, using a more classical technique, that the implicit temporal language TL is strictly less expressive than the explicit temporal language FO($\leq$). This alternative proof is interesting by itself and opens new perspectives in the investigation of results of the same kind for more expressive implicit temporal languages than TL.
Keywords: database, expressiveness, query languages, temporal logic.

[*]LRI UMR 8623 CNRS- Bat 490, Université de Paris Sud, 91405 Orsay Cedex, France, bidoit@lri.fr

[†]Depart. of Computer Science, Federal University of Uberlândia, Uberlândia, MG, Brazil, deamo@ufu.br.
The author was supported by an individual research grant from CNPq (Brazil) and in part by the program Capes-Cofecub 301/00-1

[‡]INRIA-Rocquencourt, BP 105, 78153 Le Chesney Cedex, France, Luc.Segoufin@inria.fr

1 Introduction

It is well-known that the choice between an implicit and explicit representation of time in a temporal data model does not affect the meaning of the information stored in a temporal database [2, 5, 18]. However, one cannot state the same about temporal query languages. Following an *implicit* time representation approach, a temporal database is defined as a finite sequence of instances which can be interpreted as the evolution of the database states during its lifetime; on the other hand, following an *explicit* time representation approach, tuples in a relation are timestamped and time can represent, for instance, the valid date associated to the information in the real world (*valid time*) or the date this information has been entered into the database (*transaction time*). We notice that in both approaches, the underlying temporal structure is *linear*. Although non linear time (branching time) is potentially applicable to database problems like version management or workflows, we concentrate on temporal domains that are linearly ordered sets.

In the context of an implicit time representation, the linear temporal logic TL, with connectives *Until Since Next Previous*, and its extensions are the basic formalisms underlying query languages specification [7, 10]. When time is explicitly represented, queries are specified using the standard relational query languages [3] with built-in linear order on the timestamps. One of these languages, the relational calculus with timestamps (i.e., the first-order theory of linear order), denoted by FO($\leq$), provides a natural formalism to specify queries in the explicit perspective.

The comparison between the two approaches was done in [1, 2, 20, 12] where many fragments (restriction on the quantifier prefix as in [20]) and extensions (more temporal connectives as in [2, 1]) of the logics presented above are studied and compared. Their motivations ranged from obtaining decidability for satisfiability ([20]) or separating query language in term of expressiveness ([2, 1]).

In this paper we focus on TL and FO($\leq$). In particular the work of [1, 2] has shed new light on their relative expressiveness as temporal query languages: surprisingly, TL is strictly less expressive than FO($\leq$), in particular TL is unable to express the existence of two identical states in the temporal database, while this property is easily expressible in FO($\leq$). These results are of major interest and stand in contrast with the propositional case. In [11, 14], the notion of a *complete* temporal language is introduced via equivalence with FO($\leq$) and the authors show that propositional TL is complete.

The fact that TL is strictly less expressive than FO($\leq$) has clearly stimulated subsequent research work in proposing implicit and explicit query languages more expressible than TL. For instance, [1, 2] propose a hierarchy of implicit and explicit temporal languages which, it is worth to remark, are either strictly less expressive or strictly more expressive than FO($\leq$). [5] has introduced another hierarchy of implicit temporal languages, with the purpose of proving the following open problem: *Is there a complete implicit temporal query language ?* Investigations aiming at solving this problem are motivated, as pointed out in [8], by the simplicity and computational advantages of temporal logic which make it especially attractive as a query language for temporal databases. Indeed, because the references to time are hidden, queries are formulated in an abstract, representation-independent way. Furthermore, temporal logic and the first-order theory of linear order have very different complexity properties : for instance, the satisfiability problem for propositional TL (given a propositional formula p is there a model for p ?) is PSPACE-complete ([19]) whereas the same problem for first-order theory is non elementary ([17]).

It is important at this point, to make some brief comments about the technique employed in [1, 2] in order to prove that TL is strictly less expressive than FO($\leq$). This technique is based on communication protocols [15]. Intuitively, a communication protocol involves two partners exchanging information in order to prove or discover something. Each partner is in possession of some amount of information and is able to execute calculations. By exchanging messages, the two partners can inform each other of partial results of these calculations. Their messages take the form of finite relations. The calculations carried out by one partner can only involve the information he/she owned at the beginning of the communication process or the one he/she obtained from messages exchanged with his/her partner. Thus, the execution of a protocol is essentially characterized by a series of messages exchanged between the two partners. A protocol may be designed to test some property over the initial dataset. The number of messages exchanged during the protocol execution measures, in some extent, the *communication complexity* of the property. The proof we are interested in can be outlined as follows: (1) it can be shown that all properties expressible in TL have a constant communication complexity (does not depend on the size of the dataset); (2) the property consisting in verifying the existence of two distinct states in a temporal database has a non-constant communication complexity (roughly speaking, in order to be verified, this property requires a number of messages which depends on the number of states in the temporal instance). So, TL cannot express this property which, on the other hand, can be easily formulated in FO($\leq$).

In this paper, we propose a different proof of this same result, but using a more classical technique based on games à la Ehrenfeucht-Fraïssé. Nevertheless, the proof we present does not follow the direct proof schema used to show, for instance, that *graph connectivity* is not a first order property: (1) characterize elementary equivalence restricted to formulas of quantifier depth r by a game with r moves ; (2) find a connected graph and a non-connected one and a winning strategy over these two structures, implying that they cannot be distinguished by first order formulas. Concerning the language TL, developing a similar proof schema is not an easy task due to the fact that the time order structure is linear and discrete. We notice, for instance, that in [18], a technique of games à la Ehrenfeucht-Fraïssé is successfully used mainly because the time order considered is dense.

Our proof that TL is strictly less expressive than FO($\leq$) can be outlined as follows. First, we notice that the *witness* property used in [1, 2, 18], *"are there two identical states in the database?"*, which we call twin henceforth, is order independent, in the sense that its verification on a temporal database does not depend on the particular ordering of its states. As a matter of fact, the communication protocol technique described above takes advantage of this aspect of the twin property, since the temporal database is viewed by the two partners as a *set* of instances rather than a *sequence* of instances. We denote by FO($=$) the fragment of FO($\leq$) which includes neither the order predicate $\leq$ nor temporal constants. Next, we introduce an implicit temporal query language, denoted by TL(E^i) and its explicit counterpart in FO(=), denoted by $\text{FO}_{tl}(=)$, and show that they correspond exactly to the order independent properties expressible in TL. The proof of this result is quite complex and requires a *strong* notion of order independence involving *finite* and *infinite* instances, since a well-known counterexample given by Y. Gurevich (see exercise 17.27 in [3]) guarantees the existence of an order independent (with respect to *finite* instances) property expressible in TL which is not expressible in FO(=) (therefore, not expressible in TL(E^i)). After establishing these syntactic (implicit and explicit) characterizations of the order independent properties expressible in TL, we show, via a game à la Ehrenfeucht-Fraïssé for TL(E^i), that this restricted temporal language is unable to express the twin property. We conclude that twin is not expressible in TL since it is order independent.

The most important contribution of this paper consists in (1) the syntactic characterization of the order independent properties expressible in TL via the restricted temporal language TL(E^i), which by itself sheds new light on the expressiveness of implicit temporal query languages and (2) the alternative proof schema it provides, allowing the use of a more classical tool to show a known result on relative expressiveness of temporal query languages.

Paper Organization: In section 2, we briefly recall the definitions of the query languages TL and FO($\leq$) and present some well-known results which give the necessary background for understanding the problem we propose. Section 3 introduces the implicit temporal language TL(E^i) and its explicit counterpart $\text{FO}_{tl}(=)$, and show that TL(E^i) and $\text{FO}_{tl}(=)$ correspond exactly to the order independent properties expressible in TL. Section 4 develops a game à la Ehrenfeucht-Fraïssé in order to show that twin is not expressible in TL(E^i). Finally, we conclude the paper by discussing the perspectives of using this technique to solve some similar inexpressiveness problems.

2 Preliminaries

In the following, we consider a (static) database schema **R** and a unique domain **Dom**. The timestamp of **R**, denoted $\mathbf{R}^{ts}$ is obtained from **R** by adding a new attribute T to each relation of **R**. This temporal attribute ranges over the temporal domain which is usually $\mathbb{N}^+$ in this paper.

An implicit temporal (finite) instance $\mathcal{I}$ over **R**, is a (finite) sequence $(I_1, I_2, \ldots, I_n)$ of (finite) instances over **R** which domain is a (finite) subset of **Dom**.

I_1 is called the initial state of $\mathcal{I}$, I_n its final state, and for each i, I_i is called the state of $\mathcal{I}$ at instant i.

An explicit temporal instance is a finite instance over $\mathbf{R}^{ts}$ where, for each tuple, the temporal attribute ranges over $\mathbb{N}^+$ while the remaining attributes range over **Dom**.

To each implicit temporal instance $\mathcal{I}$ corresponds an explicit temporal instance $\mathcal{I}^{ts}$ and vice versa. Figure 2 shows the two alternative approaches for time representation.

The query language TL. The linear temporal logic TL [10] is a well-known formalism for specifying queries in an implicit time approach. The syntax of TL over a

I_1

R	A
	a
	b
	c

I_2

R	A
	b
	d

I_3

R	A
	a
	c
	d

The implicit temporal instance $\mathcal{I}$.

R^{ts}	A	T
	a	1
	b	1
	c	1
	b	2
	d	2
	a	3
	c	3
	d	3

The explicit tempral instance $\mathcal{I}^{\mathrm{ts}}$

Figure 1: Implicit and explicit time representations

database schema **R** is obtained using the formation rules for standard first order logic over **R** together with the additional formation rule:

- if φ_1 et φ_2 are formulas then φ_1 **Until** φ_2, φ_1 **Since** φ_2, **Next** φ_1 and **Previous** φ_1 are formulas.

Given a temporal instance $\mathcal{I} = (I_1,\ldots,I_n)$ over **R**, and a TL formula φ over **R**, the active domain of $\mathcal{I}$, denoted by **Adom**($\mathcal{I}$), is the set of elements of **Dom** appearing in $I_i(R)$, for some $R \in \mathbf{R}$ and some $i \in \{1, \ldots, n\}$. As for the relational calculus, different kinds of problems (*"unsafe queries"*) may arise if the conventional definitions from linear temporal logic are adapted directly to the current context (see [3] for a discussion). We opt here for restricting variables to range over the active domain of the input temporal database.

In the following, we briefly recall the (*active domain*) semantics of a TL formula φ over an implicit temporal instance $\mathcal{I}$.

Definition 2.1 (Semantics of TL) The *truth* of φ over $\mathcal{I}$ at instant $i \in \{1, \ldots, n\}$, given the valuation ν of the free variables of φ ranging over **Adom**($\mathcal{I}$), denoted $[\mathcal{I},i,\nu] \models_{tl} \varphi$, is defined as follows:

$[\mathcal{I}, i, \nu] \models_{tl} R(x_1, \ldots, x_k)$ if $(\nu(x_1), \nu(x_2), \ldots, \nu(x_k)) \in I_i(R)$.

If φ is a boolean combination of formulas or a quantification ($\exists$, $\forall$) of a formula then the definition is as usual.

$[\mathcal{I},i,\nu] \models_{tl} \varphi_1$ **Until** φ_2 iff there exists $j > i$ such that $[\mathcal{I},j,\nu] \models_{tl} \varphi_2$ and for each k such that $i \leq k < j$, $[\mathcal{I},k,\nu] \models_{tl} \varphi_1$.

$[\mathcal{I},i,\nu] \models_{tl} \varphi_1$ **Since** φ_2 iff there exists $j < i$ such that $[\mathcal{I},j,\nu] \models_{tl} \varphi_2$ and for each k such that $j < k \leq i$, $[\mathcal{I},k,\nu] \models_{tl} \varphi_1$.

$[\mathcal{I},i,\nu] \models_{tl}$ **Next** φ_1 iff $i < n$ and $[\mathcal{I},i+1,\nu] \models_{tl} \varphi_1$.

$[\mathcal{I},i,\nu] \models_{tl}$ **Previous** φ_1 iff $i > 1$ and $[\mathcal{I},i-1,\nu] \models_{tl} \varphi_1$.

It is sometimes convenient to use the following derived temporal modalities:

$F\ \varphi_1 \equiv$ true **Until** φ_1 ("*sometimes in the future* φ_1"),
$G\ \varphi_1 \equiv \neg\ F \neg\varphi_1$ ("*always in the future* φ_1")
$P\ \varphi_1 \equiv$ true **Since** φ_1 ("*sometimes in the past* φ_1"),
$H\ \varphi_1 \equiv \neg\ P \neg\varphi_1$ ("*always in the past* φ_1"),

Historically TL has been used as an implicit temporal query language by evaluating its formulas at the initial time. We will use this notion together with a more general one where the instant of evaluation is not specified and thus can be used as a parameter for the output. Recall that, in the following definition, $\vec{x}$ denotes a sequence of variables and the valuations ν always ranges over the active domain of the temporal instance.

Definition 2.2 (Query in TL) A query Q in TL over a database schema **R** is specified by an expression of the form $\{(\vec{x}) \mid \varphi(\vec{x})\}$ where $\varphi(\vec{x})$ is a formula of TL over **R** with free variables $\vec{x}$. The answer of Q over a temporal instance $\mathcal{I}$ is the relation $\mathrm{Q}(\mathcal{I})=\{\nu(\vec{x}) \mid [\mathcal{I},1,\nu] \models_{\mathrm{tl}} \varphi(\vec{x}),\ \nu$ a valuation of $\vec{x}$ ranging over $\mathbf{Adom}(\mathcal{I})\}$.

Definition 2.3 (Extended query in TL*) A query Q in TL* over a database schema **R** is specified by an expression of the form $\{(\vec{x}, t) \mid \varphi(\vec{x})\}$ where $\varphi(\vec{x})$ is a formula of TL over **R** with free variables $\vec{x}$. The answer of Q over a temporal instance $\mathcal{I}$ is the relation $\mathrm{Q}(\mathcal{I})=\{(\nu(\vec{x}), i) \mid [\mathcal{I},i,\nu] \models_{\mathrm{tl}} \varphi(\vec{x}),\ \nu$ a valuation of $\vec{x}$ ranging over $\mathbf{Adom}(\mathcal{I})\}$.

The query language FO($\leq$). A natural query language over timestamp temporal databases is obtained by considering the first order logic over $\mathbf{R}^{ts}$ with two kinds of variables: *temporal variables* (denoted by t, u, v, s) and *data variables* (denoted by x, y, z, w). The language signature contains also the binary predicate $\leq$ which is defined only for temporal variables and which is interpreted by a linear order over the temporal active domain included in $\mathbb{N}^+$.

In order to make a fair comparison with TL (respectively TL*) we will consider queries whose answers contain only data values and no free temporal variables (resp. one temporal variable). The corresponding language is denoted by FO($\leq$) (respectively FO*($\leq$)).

Definition 2.4 (Query in FO($\leq$)) A query in FO($\leq$) over **R** is an expression of the form $\{(\vec{x}) \mid \phi(\vec{x})\}$, where $\phi(\vec{x})$ is a formula of FO($\leq$) over $\mathbf{R}^{ts}$ where the free variables $\vec{x}$ are data variables. The answer of Q evaluated on a timestamp temporal instance $\mathcal{I}^{ts}$ is defined as for relational calculus, where valuations of variables range over the active domain of $\mathcal{I}^{ts}$.

Definition 2.5 (Extended query in FO*($\leq$)) A query in FO*($\leq$) over **R** is an expression of the form $\{(\vec{x}, t) \mid \phi(\vec{x}, t)\}$, where $\phi(\vec{x}, t)$ is a formula of FO*($\leq$) over $\mathbf{R}^{ts}$ where the free variables $\vec{x}$ are data variables. The answer of Q evaluated on a timestamp temporal instance $\mathcal{I}^{st}$ is defined as for relational calculus, where valuations of variables range over the active domain of $\mathcal{I}^{st}$.

In the following, FO(=) and FO*(=) denote the restriction of the previous query languages where the predicate $\leq$ and the temporal constants are not allowed. Thus, the only possible atomic formulas involving temporal variables are those of the form $t = s$, where both t, s are temporal variables.

Example 2.6 Let **R** and $\mathcal{I}$ be the database schema and temporal instance of figure 1. The TL query $\{(x) \mid R(x) \wedge F\ R(x)\}$ evaluated on $\mathcal{I}$ returns $\{a, b, c\}$. The TL* query

$\{(x,t) \mid R(x) \wedge F\ R(x)\}$ evaluated on $\mathcal{I}$ returns $\{(a,1),(b,1),(c,1),(d,2)\}$. These queries can be equivalently expressed in FO($\leq$) and FO*($\leq$) by the expressions $\{(x) \mid \exists t(\mathit{first}(t) \wedge R^{ts}(x,t) \wedge \exists s(t < s \wedge R^{ts}(x,s)))\}$ and $\{(x,t) \mid R^{ts}(x,t) \wedge \exists s(t < s \wedge R^{ts}(x,s))\}$ respectively. The formulas $t < s$ and $t > s$ are abbreviations for $t \leq s \wedge t \neq s$ and $s \leq t \wedge s \neq t$ respectively and the formula $\mathit{first}(t)$ is an abbreviation for $\neg\exists u(u < t)$.

Definition 2.7 (Global and Initial Equivalence) Two formulas are said to be *globally equivalent* if the corresponding extended queries are equivalent. They are *initially equivalent* if the corresponding queries are equivalent.

Known results concerning TL and FO($\leq$). In [11] and in [14] it is shown that, in the propositional case, TL and FO($\leq$) are *globally equivalent*, which means that the query languages TL* and FO*($\leq$) have the same expressive power. This implies *initial equivalence* between TL* and FO*($\leq$), which simply means that the query languages TL and FO($\leq$) have the same expressive power.

However, [1, 2] have proved that in the first order case, the equivalence between TL and FO($\leq$) does not hold, and so, these query languages have different expressive power.

Theorem 2.8 *[1, 2]* TL $\subsetneq$ FO($\leq$)

The property separating TL from FO($\leq$), which we call twin, checks if the input temporal instance contains two identical states. This property can be expressed in FO($\leq$) by $\exists i \exists j(i \neq j \wedge \forall x[S^{st}(x,i) \leftrightarrow S^{st}(x,j)])$ and its satisfiability does not depend on the ordering of the states $I_1, I_2, \ldots, I_n$ in $\mathcal{I}$. The technique employed in [1, 2] in order to prove that twin is not expressible in TL is based on communication protocols [15]. In section 4, we will prove the same result using a more classical technique based on Ehrenfeucht-Fraïssé's games. In order to do so, we will first give a syntactic characterization of the order independent properties expressible in TL. This is the main issue of the next section.

3 The temporal query language TL(E^i)

In this section we introduce a very simple implicit temporal language which deals with an (implicit) temporal instance $\mathcal{I}$ as a *set* of states rather than a *sequence* of states. In other words, this language completely ignores the ordering of the states in a temporal instance.

The main result of this section (Theorem 3.15) guarantees that this language, denoted by TL(E^i), expresses exactly the properties of TL which are order independent. It contains an enumerable set of operators E^i ($i \geq 1$). Intuitively, a formula $E^i\varphi$ expresses that there exists i distinct states satisfying φ.

Definition 3.1 **[Syntax]** We define the language TL*(E^i) by adding the following extra rule to the usual formation rules of first order logic:

- If φ is a formula then $E^i\varphi$ is a formula for any $i \in \mathbb{N}$.

The formulas of TL(E^i) are the formulas of the form $E^i\varphi$ where φ is in TL*(E^i).

[Semantics] Given a temporal instance $\mathcal{I}$=(I_1,..., I_n), an instant j and a valuation ν of the variables ,

- $[\mathcal{I},j,\nu] \models_{tl} E^i\varphi$ if and only if there exists i distinct instants $j_1, \ldots, j_i$, such that $[\mathcal{I},j_k,\nu] \models_{tl} \varphi$ for each $k \in \{1, \ldots, i\}$.

The notions of a TL(E^i) query, TL*(E^i) query and their answers are defined as for TL and TL* queries (definition 2.2 and 2.3).

For instance, the query $E^2\exists xR(x)$ evaluated on a temporal instance $\mathcal{I}$, expresses the existence of two states in $\mathcal{I}$ with non-empty set R.

We will need later on the following definitions:

Definition 3.2 The *quantifier temporal rank* of a formula φ of TL*(E^i), denoted by qtr(φ), is defined as: (*i*) qtr(φ) = 0 if φ is atomic; (*ii*) qtr($\varphi_1 \wedge \varphi_2$) = max{ qtr($\varphi_1$), qtr($\varphi_2$) }; (*iii*) qtr($\neg\varphi$) = qtr($\varphi$); (*iv*) qtr($\exists x\varphi$) = qtr($\varphi$)+1; (*v*) qtr($E^i\varphi_1$) = qtr($\varphi_1$) + 1. The *width* of a TL*(E^i) formula, denoted by wth(φ), is defined in the same way except for (*v*) wth($E^i\varphi_1$)= max{i, wth(φ_1)}.

For instance, consider the formula $\varphi = \exists xE^2(\forall yE^3(R(x,y)))$. In this case, we have: qtr(φ) = 4 and wth(φ) = 3.

The following proposition is an immediate consequence of definition 3.1:

Proposition 3.3 Let φ be a formula of TL(E^i). Let $\mathcal{I} = (I_1, I_2, \ldots, I_n)$ be a temporal instance, $i \in \{1, \ldots, n\}$ and ν a valuation of variables. Then:

$$[\mathcal{I}, i, \nu] \models_{tl} \varphi \Longleftrightarrow \forall k, 1 \leq k \leq n, [\mathcal{I}, k, \nu] \models_{tl} \varphi$$

The following proposition establishes, as expected, that the language TL(E^i) is a sublanguage of both TL and FO(=).

Proposition 3.4 TL*(E^i) $\subseteq$ TL* $\cap$ FO*(=), i.e., for each formula $\varphi \in$ TL*(E^i) there exists a formula $\varphi_{tl} \in$ TL* (respectively a formula $\varphi_{fo} \in$ FO*(=)) such that φ is equivalent to φ_{tl} (respectively to φ_{fo}).

Proof: The proof of the inclusion TL*(E^i) $\subseteq$ FO*(=) does not present any difficulty and relies on the natural translation of TL*(E^i) formulas into FO*(=). Next, we prove that TL*(E^i) $\subseteq$ TL*. This is achieved by induction on the structure of TL*(E^i) formulas. For the initial step and the induction steps refering to boolean combinations and quantification of TL*(E^i) formulas, the proof is straightforward. Let $\varphi = E^i\psi$, where ψ is a TL*(E^i) formula. Without loss of generality, we consider the case where $i = 2$, the other cases can be treated in a similar way. By the induction hypothesis, ψ is equivalent to a TL* formula ψ_{tl}. It is easy to check that φ is equivalent to the following TL* formula: $(\psi_{tl} \wedge F\psi_{tl}) \vee (\psi_{tl} \wedge P\psi_{tl}) \vee (F(\psi_{tl} \wedge F\psi_{tl})) \vee (F\psi_{tl} \wedge P\psi_{tl}) \vee (P(\psi_{tl} \wedge P\psi_{tl}))$. $\square$

Next, we will characterize the fragment of $FO^*(\leq)$ and $FO^*(=)$ which are equivalent to TL^* and $TL^*(E^i)$ respectively. In order to do this, let us consider the language $FO^*_{tl}(\leq)$ that is the set of $FO^*(\leq)$ formulas such that each of their subformulas in the scope of a data quantifier has at most one free temporal variable. For instance, the formula $\exists i \exists j (i \neq j \wedge \forall x [S^{st}(x,i) \leftrightarrow S^{st}(x,j)])$ does not belong to $FO^*_{tl}(\leq)$, because its subformula $[S^{st}(x,i) \leftrightarrow S^{st}(x,j)]$ lies in the scope of the data quantifier $\forall x$ and has two free temporal variables i and j. In the same way, we can define the language $FO^*_{tl}(=)$ as a sublanguage of $FO^*(=)$.

The following theorem gives the desired syntactic characterization of the fragments of $FO^*(\leq)$ and $FO^*(=)$ equivalent to the implicit temporal languages TL^* and $TL^*(E^i)$ respectively. Note that part (1) of the following theorem was independently obtained in [12] and was used in order to show decidability of the satisfiability problem for the *monadic* fragment of TL.

Theorem 3.5

(1) TL^* and $FO^*_{tl}(\leq)$ are equivalent.

(2) $TL^*(E^i)$ and $FO^*_{tl}(=)$ are equivalent.

The proof of Theorem 3.5 will make use of two intermediate results. The first one has been shown in [14, 11] and establishes that the propositional fragment of TL^* is equivalent to $FO^*(\leq)$ (without data variables). The second one (lemma 3.6) establishes a similar equivalence between the propositional fragment of $TL^*(E^i)$ and $FO^*(=)$.

Lemma 3.6 Propositional $TL^*(E^i)$ is equivalent to $FO^*(=)$.

Proof: ($\subseteq$) follows from Proposition 3.4. We thus concentrate on the proof of ($\supseteq$). Let $p_1, \ldots, p_k$ be the propositions in the signature of $TL^*(E^i)$. Because we don't have access to the order here, a temporal instance can be viewed as a *set* of states, instead of a *sequence* of states as in the general setting. For each state I, its *type* is the tuple $(u_1, \ldots, u_k)$, where u_i is the truth value of proposition p_i at state I, for $i \in \{1, \ldots, k\}$. Let Θ be the set of all possible types, thus the cardinality of Θ is 2^k. Because each state is fully characterized by its type, a temporal instance $\mathcal{I}$ (with no order) is thus characterized by a function $\alpha_{\mathcal{I}}$ from Θ to $\mathbb{N}$ which associates to each type τ the number of states of type τ in $\mathcal{I}$. For each type τ, it is naturally associated a propositional formula ϕ_τ: for instance, if $\tau = (1,0,0,1)$ then $\phi_\tau = p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge p_4$. So, if τ is the type of a state I then ϕ_τ is true at a state J if and only if $J = I$.

Now let $\varphi \in FO^*(=)$. Assume without loss of generality that φ never quantifies twice over the same variable name. Let r be the number of variables occurring in φ. From classical finite model theory we know that first order formulas with only r variables can only count up to r without the presence of an order [9]. Let A_r be the finite set of pairs $(\mathcal{I}, i)$ where $\mathcal{I} = (I_1, ..., I_n)$ is a temporal instance such that a given type τ never occurs more than r times ($\forall \tau \in \Theta \ \ \alpha_{\mathcal{I}}(\tau) \leq r$) and $i \in \{1, \ldots, n\}$. For each pair $(\mathcal{I}, i)$ of A_r, let $\psi_{\mathcal{I},i}$ be the formula of $TL^*(E^i)$: $\tau_i \wedge \bigwedge_{\tau \in \Theta} E^{\alpha_{\mathcal{I}}(\tau)} \phi_\tau$, where τ_i is the type of state I_i.

Let A_r^1 (respectively A_r^2) be the subset of A_r of pairs $(\mathcal{I}, i)$ such that φ is true (resp. false) on $\mathcal{I}$ at instant i. Consider the following formula ψ of TL$^*(E^i)$:

$$\bigvee_{(\mathcal{I},i)\in A_r^1} \psi_{\mathcal{I},i} \wedge \bigwedge_{(\mathcal{I},i)\in A_r^2} \neg\psi_{\mathcal{I},i}$$

It is easy to verify that ψ and φ agree on all pairs $(\mathcal{I}, i)$ in A_r. We have already seen that φ can count the multiplicity of an occurrence of a type only up to r. It is easy to verify that this is also the case for each $\psi_{\mathcal{I},i}$ constructed above and therefore for ψ. We will show that φ and ψ agree on all pairs $(\mathcal{I}$,$i)$ (and not only on those in A_r). Indeed, if $\mathcal{I} = (I_1, \ldots, I_n)$ is a temporal instance, consider $\mathcal{I}_r = (J_1, \ldots, J_m)$ the temporal instance which contains exactly the same types as in $\mathcal{I}$ with the same multiplicity if this one is less than r and r otherwise ($\alpha_{\mathcal{I}_r}(\tau) = min\{r, \alpha_{\mathcal{I}}(\tau)\}$). For instance, if $\mathcal{I} = (I, J, J)$ and $r = 1$ then $\mathcal{I}_1 = (I, J)$. Let $i \in \{1, \ldots, n\}$. Then, there exists $j \in \{1, \ldots, m\}$ such that (a) $[\mathcal{I}, i] \models_{tl} \psi$ if and only if $[\mathcal{I}_r, j] \models_{tl} \psi$ and (b) $[\mathcal{I}, i] \models \varphi$ if and only if $[\mathcal{I}_r, j] \models \varphi$ [1] (we take j such that $J_j = I_i$). Because $(\mathcal{I}_r, j) \in A_r$, ψ and φ have the same truth value on $(\mathcal{I}_r, j)$. Thus they have the same truth value on $(\mathcal{I}, i)$. This proves the Lemma. □

Remark 3.7 *Note that the proof of the Lemma 3.6 shows that the formula of FO$^*(=)$ with r variables are equivalent to a formula of TL$^*(E^i)$ of quantifier temporal rank 1 and width r. This fact will be needed later on.*

Proof of Theorem 3.5: The inclusion of TL* into FO$^*_{tl}(\leq)$ is easily proved by considering the natural translation of a TL* formula into a FO$^*(\leq)$ formula. The same argument can be used to show that TL$^*(E^i)$ is included in FO$^*{}_{tl}(=)$. Let us prove the inclusion FO$^*_{tl}(=) \subseteq$ TL$^*(E^i)$. This proof is by induction on the *quantifier-data depth* of a formula $f(t, \vec{x})$ in FO$^*_{tl}(=)$ which is defined as follows: (i) qdd$(f) = 0$ if f is atomic; (ii) qdd$(f \wedge g) = \max\{$qdd(f),qdd$(g)\}$; (iii) qdd$(\neg f) =$ qdd(f); (iv) qdd$(\exists t f)$ $=$ qdd(f) if t is a temporal variable; (v) qdd$(\exists x f) = 1 +$ qdd(f) if x is a data variable. *Initial step:* Let qdd$(f) = 0$ (f does not contain any data quantifier). We apply the following transformations on f: atomic formulas of the form R$(s, \vec{z})$ are replaced by unary predicates $R_{\vec{z}}(s)$ and atomic formulas of the form $(y_1 = y_2)$ or $(y = a)$ are replaced by unary predicates $EQ_{y_1=y_2}(t)$ and $EQ_{y_1=a}(t)$ respectively. Remind that t is the only free variable of f and that the interpretations of the equality predicate involving data variables or constants and the interpretation of constants are time-invariant (in the database context). After this transformation, we obtain a formula $\varphi(t)$ of FO$^*(=)$. By Lemma 3.6, there exists a propositional TL$^*(E^i)$ formula θ which is equivalent to $\varphi(t)$. Now, the "propositions" $R_{\vec{z}}$, $EQ_{y_1=y_2}$ and $EQ_{y_1=a}$ in θ are replaced by the corresponding atomic formula $R(\vec{z}), (y_1 = y_2)$ and $(y_1 = a)$. In this way, we obtain a formula θ_f in in TL$^*(E^i)$ which is equivalent to f.

[1] In order to obtain (b), it is important to recall that formulas in FO$^*(=)$ have no constant symbols. Indeed, if φ is the formula $(t = 3 \wedge p(t))$ and $\mathcal{I} = (I, I, I)$ and p is true at state I then $[\mathcal{I}, 3] \models \varphi$ but $[\mathcal{I}, i] \not\models \varphi$, for $i \neq 3$. So, if $r < 3$, there is no $i \in \{1, 2\}$ such that $[\mathcal{I}, 3] \models \varphi$ if and only if $[\mathcal{I}_r, i] \models \varphi$.

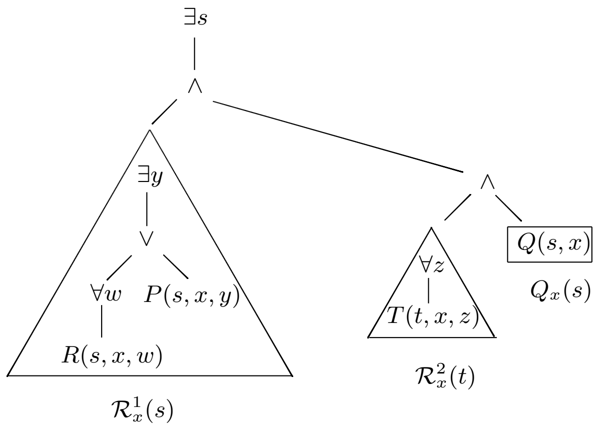

Figure 2: Transformation of the $\mathrm{FO}_{tl}^*(=)$ formula $f(t,x)$ into the $\mathrm{FO}^*(=)$ formula $\varphi(t)$

Induction step: Suppose the inclusion holds for formulas with quantifier-data depth $\leq n$ and let $f(t,\vec{x}) = Q_1 y_1 \ldots Q_m y_m g(t,\vec{x}) \in \mathrm{FO}_{tl}^*(=)$ with $\mathrm{qdd}(f) = n+1$, where $m \geq 0$, t is a temporal variable, $\vec{x}$ are the free data variables of f, Q_i are quantifiers ($\exists$ or $\forall$) and $y_1 \ldots y_m$ are data variables included in $\vec{x}$.
Case 1: m = 0 - Let us consider the *data-quantified* subformulas of f, which are formulas of the form $\exists x h(s,x,\vec{y})$, where s is a temporal variable, x is a data variable and $\vec{y}$ is a set of data variables. We say that a data-quantified subformula h_1 is *maximal* if it is not a subformula of another data-quantified subformula h_2 with $h_1 \neq h_2$. For each maximal data-quantified subformula $\exists x h(s,x,\vec{y})$ of f we associate a new relational symbol $\mathcal{R}$, with arity equal to the number of variables in $\vec{y}$. We replace the maximal data-quantified subformulas $\exists x h(s,x,\vec{y})$ of f by the corresponding atomic formulas $\mathcal{R}(s,\vec{y})$. The maximal data-quantified subformulas of f with no free temporal variable are replaced in the same way, by an atomic formula $\mathcal{R}(t,\vec{y})$. So, after this transformation we obtain a $\mathrm{FO}_{tl}^*(=)$ formula f' with $\mathrm{qdd}(f') = 0$. In this case, we have already shown that there exists a $\mathrm{TL}^*(E^i)$ formula $\theta_{f'}$ which is equivalent to f'. Figure 2 illustrates the idea: the $\mathrm{FO}_{tl}^*(=)$ formula $f(t,x) = \exists s(\exists y(\forall w R(s,w,z) \vee S(s,x,y)) \wedge \forall z T(t,x,z) \wedge Q(s,x))$ transforms into a $\mathrm{FO}^*(=)$ formula $\varphi(t) = \exists s(\mathcal{R}_x^1(s) \wedge \mathcal{R}_x^2(t) \wedge Q_x(s))$, which is equivalent to a propositional $\mathrm{TL}^*(E^i)$ formula.

Let us consider each maximal data-quantified subformula $\exists x h(s,x,\vec{y})$ of f. It is clear that $\mathrm{qdd}(h(s,x,\vec{y})) = n$. Applying the induction hypothesis on $h(s,x,\vec{y})$, we obtain an equivalent $\mathrm{TL}^*(E^i)$ formula $\theta_h(x,\vec{y})$. So, $\exists x h(s,x,\vec{y})$ is equivalent to the $\mathrm{TL}^*(E^i)$formula $\exists x \theta_h(x,\vec{y})$. Now, we replace the "propositions" $\mathcal{R}_{\vec{y}}$ in $\theta_{f'}$ (which "encapsulates" the maximal data-quantified subformulas) by the equivalent $\mathrm{TL}^*(E^i)$ formulas. In this way, we obtain a formula θ_f in $\mathrm{TL}^*(E^i)$ which is equivalent to f.
Case 2: m > 0 - From **Case 1**, we know that there exists $\theta_g \in \mathrm{TL}^*(E^i)$ which is equivalent to g. Then $Q y_1 \ldots Q y_m \theta_g$ is equivalent to f.

The proof of the other inclusion $\mathrm{FO}_{tl}^*(\leq) \subseteq \mathrm{TL}^*$ is obtained in the same way, using the Lemma 3.6 instead of Kamp's Theorem. □

Remark 3.8 *The proof of Theorem 3.5 showed the importance of* blocks *of temporal variables in the translation, a block being defined as a set of temporal variables quantified between two data quantifications. Remark 3.7 and the same induction as in the* *proof of Theorem 3.5 shows that a formula of* $FO^*_{tl}(=)$ *is translated into a formula of* $TL^*(E^i)$ *whose width corresponds to the maximal size of blocks in the formula and whose quantifier temporal rank is counted in the standard way but adding one for each block instead of the full size of the block.*

Corollary 3.9 The explicit query languages $\mathrm{FO}_{tl}(\leq)$ and $\mathrm{FO}_{tl}(=)$ are (initially) equivalent to the implicit query languages TL and $\mathrm{TL}(E^i)$ respectively.

Proof: The equivalence between $\mathrm{FO}_{tl}(\leq)$ and TL follows immediately from Theorem 3.5 (a) and the inclusion $\mathrm{TL}(E^i) \subseteq \mathrm{FO}_{tl}(=)$ relies on the natural translation of $\mathrm{TL}(E^i)$ formulas into $\mathrm{FO}_{tl}(=)$. Conversely, let f be a $\mathrm{FO}_{tl}(=)$ query. We define the maximal temporal quantified subformulas of f in a similar way as we have defined the maximal data quantified subformulas in the proof of Theorem 3.5. From Theorem 3.5(b), each such formula $\exists t g(t, \vec{x})$ is globally equivalent to $E^1\varphi$, where φ is a $\mathrm{TL}^*(E^i)$ formula. Then, it is clear that f is globally equivalent to a formula ψ in the closure of $\mathrm{TL}(E^i)$ with respect to the boolean combinations and quantification. Notice that ψ is not necessarily in $\mathrm{TL}(E^i)$. By applying Proposition 3.3 to the formula ψ, we conclude that f is initially equivalent to the $\mathrm{TL}(E^i)$ formula $E^1\psi$. □

Corollary 3.9 provides a syntactic characterization of the queries in $\mathrm{FO}(\leq)$ and $\mathrm{FO}(=)$ which are equivalent to queries in TL and $\mathrm{TL}(E^i)$ respectively. We are going to use this syntactic characterization in order to prove the main result of this section, Theorem 3.15, establishing that $\mathrm{TL}(E^i)$ corresponds exactly to the order independent properties expressible in TL.

We start by a semantic characterization of the properties expressible in TL which are *order independent*. Intuitively this means that the property is not affected by a reordering of the states of a temporal instance. We do not take the standard definition of order independence which consists of being unaffected by a reordering over a *finite* temporal instance. Instead we consider a stronger variant which requires the property to be also unaffected by a reordering over *infinite* instances. The reason for this is detailed below.

Definition 3.10 A property $\mathcal{P}$ is said to be *order independent* if, for any temporal instance (finite or infinite) $\mathcal{I} = (I_i)_{i\in V}$ and for any bijection s of V, we have: $\mathcal{I}$ satisfies $\mathcal{P}$ **iff** $\mathcal{I}^s$ satisfies $\mathcal{P}$, where $\mathcal{I}^s = (I_{s(i)})_{i\in V}$.

An important condition for being order independent is that it should be the case for both finite and infinite instances. Infinite means here that the temporal instance may contain infinitely many states, each of them being possibly over an infinite active domain. With the weaker notion of order independence which only considers finite instances, Theorem 3.15 stating that the order independent properties expressible in TL are exactly the ones expressible in $\mathrm{TL}(E^i)$ doesn't hold as we now show.

Gurevich gave an example of a query which is generic, expressible in $\mathrm{FO}(\leq)$, but not in $\mathrm{FO}(=)$ (see exercise 17.27 in [3]). In this context, the genericity of a query corresponds to invariance up to isomorphisms over finite databases but it can easily be modified in order to give an order independent TL query as follows:

Example 3.11 Consider (finite) implicit temporal instances over the database schema containing only one unary relation S. Recall that each such an instance is a finite sequence $\mathcal{I} = (I_i)_{i \in V}$ of finite states. For each non-empty subset W of states, an element x_W of the active domain of S is called a *witness* for W if $\forall i \in V \quad x_W \in I_i(S)$ iff $I_i \in W$. Consider the temporal instance depicted in Figure 3. It has a witness for each subset of $\{I_1, I_2, I_3\}$. Indeed a is a witness for $\{I_1\}$ because it is in S only in the first state, c is a witness of $\{I_3\}$, f is a witness for $\{I_1, I_3\}$, and so on.

$\mathcal{I} =$	I_1: S	I_2: S	I_3: S
	a	b	c
	d	d	e
	f	e	f
	g	g	g

Figure 3: An instance satisfying q_1 but not q_2

Consider the TL(E^i) query q_1 which says that for each non-empty subset W of V there exists a witness x_W. The query q_1 can be expressed in TL(E^i) by establishing that: (1) all singletons W have a witness and (2) the set of (finite) sets of states having a witness is closed under finite union. This is done as follows.

1. $\neg E^1(\forall x[E^2(S(x)) \vee \neg S(x)])$
2. $\forall x \; \forall y \; \exists z \;\; \neg E^1(\neg(S(z) \leftrightarrow (S(x) \vee S(y)))$

For example it is easy to verify that the instance depicted in Figure 3 verifies q_1. Consider now the query q of TL which selects the implicit temporal instances satisfying q_1 and having an even number of states. This can be done by adding to q_1 the query q_2 which checks for the existence of a witness for the subset containing exactly one state out of two in the order given by the sequence (i.e. the subset $\{I_1, I_3, \ldots, ...\}$). The query q_2 can be expressed in TL by:

$$\exists x[S(x) \;\wedge \mathbf{Next}(\neg S(x)) \wedge \; G(S(x) \leftrightarrow \mathbf{Next}(\neg S(x)))]$$

It can be verified that the instance of Figure 3 does not satisfy q_2. For instance, f cannot be a valuation for x: it is clear that $I_1 \models S(f)$, $I_2 \models \neg S(f)$ and $I_3 \models S(f)$. But, because $S(f) \leftrightarrow \mathbf{Next}(\neg S(f))$ has to be verified at I_3 then I_3 must verify $\mathbf{Next}(\neg S(f))$. However, this is not the case, since there is no state I_i with $i > 3$.

Notice first that the query $q = q_1 \wedge q_2$ is order independent (in the weak sense). Indeed q_1 insure that there is a witness for each subset which is a finite union of singleton. For finite instances it thus insure the existence of a witness for any subset. This first characteristic is trivially order independent. Therefore, if a temporal instance has an even number of states, no matter how we reorder them, there will always be a witness for the subset containing all the states of even position.

Because q checks for a parity condition, a standard Ehrenfeucht-Fraïssé's games argument shows that q cannot be expressed in FO(=) and therefore in TL(E^i).

Notice now that q is *not* order independent in the strong sense. Indeed q_1 only guarantees the existence of a witness for each *finite* subset. If the temporal instance is infinite it does not imply the existence of a witness for any subset. To see that q is indeed not order independent in the strong sense consider the following example.

Let $(V_i)_{i \in \mathbb{N}^+}$ be an enumeration of the finite subsets of $\mathbb{N}^+$ such that V_1 corresponds to the emptyset. Let W be the subset of all odd numbers.

Consider the following infinite temporal instance $\mathcal{I} = (I_i)_{i \in \mathbb{N}^+}$ over the domain $(x_{i,j})_{i \in \mathbb{N}^+, j \in \{0,1\}}$, such that $I_k \models S(x_{i,0})$ iff $k \in V_i$ and $I_k \models S(x_{i,1})$ iff $k \in V_i$ or k is odd. Intuitively $\mathcal{I}$ is constructed in such a way that $x_{i,0}$ is a witness for the finite set V_i and $x_{i,1}$ a witness for the set $W \cup V_i$.

Notice first that $\mathcal{I}$ satisfies q_2 because $x_{1,1}$ is a possible valuation for x in q_2. By construction $\mathcal{I}$ contains a witness for exactly each subset U which is either finite or the union of W with a finite set. Thus $\mathcal{I}$ satisfy q_1 and therefore q.

Consider now the infinite temporal instance $\mathcal{J} = (I_1, I_3, I_2, I_4, I_5, I_7, I_6, I_8 \ldots)$. It can be checked that $\mathcal{J}$ does not satisfy q_2 and therefore q. Thus q is not order independent in the strong sense.

If the notion of genericity is extended in order to indicate invariance up to isomorphisms over any (finite or infinite) databases, it is folklore knowledge (as a consequence of Craig Interpolation Theorem, [6]) that FO($\leq$) and FO($=$) express exactly the same extended generic queries.

As we will use exactly the same ideas we recall here how the above result can be proved.

Theorem 3.12 *[Folklore]* Let $\mathcal{P}$ be an extended generic query expressible in FO($\leq$). Then $\mathcal{P}$ is expressible in FO($=$).

Proof: The proof is by contradiction. Assume that there is a query f expressible in FO($\leq$) which is extended generic and not expressible in FO($=$). Let $\text{lin}(R)$ be a FO(R) formula expressing the fact that R is a binary predicate corresponding to a linear order. Let $f_1(R)$ be the following FO(R) formula $\text{lin}(R) \wedge f(R)$ and let $f_2(S)$ be the following FO(S) formula: $\text{lin}(S) \rightarrow f(S)$, where $f(R)$ is the formula obtained from f after replacing in it all occurrences of the symbol $\leq$ by R. Because f is extended generic, we have for any (finite or infinite) instance:

$$\mathcal{I} \models \exists R\ f_1(R) \rightarrow \forall S\ f_2(S).$$

We now need the following Lemma which is an equivalent rephrasing of Craig Interpolation Theorem [6].

Lemma 3.13 *[Craig Interpolation Theorem]* Let $f \in \text{FO}(R)$ and $g \in \text{FO}(S)$ be such that $\exists R\ f(R) \models \forall S\ g(S)$. Then there exists $h \in \text{FO}(=)$ such that:

$$\exists R\ f(R) \models h \qquad \text{and} \qquad h \models \forall S\ g(S).$$

We have seen above that f_1 and f_2 satisfy the premises of Lemma 3.13 and thus there exists h in FO($=$) such that for all temporal instances (finite or infinite) $\mathcal{I}$ we have: $(i)\quad \mathcal{I} \models \exists R\ f_1(R) \rightarrow h$ and $(ii)\quad \mathcal{I} \models h \rightarrow \forall S\ f_2(S)$. We now prove that $f(\leq)$ and h actually define the same property. Indeed, let $\mathcal{I}$ be any instance and assume that $\mathcal{I}$ satisfies $f(\leq)$, then $\mathcal{I}$ satisfies $\exists R\ f_1(R)$ (the order $\leq$ of $\mathcal{I}$ being a valid assignment for R). From (i) we conclude that $\mathcal{I}$ satisfies h. Assume now that $\mathcal{I}$ satisfies h, from (ii) we conclude that $\mathcal{I}$ satisfies $\forall S\ \ f_2(S)$ and, in particular, $f_2(\leq)$ is true in $\mathcal{I}$ and therefore $\mathcal{I}$ satisfies $f(\leq)$.

This prove the theorem. $\square$

We are looking for a variant of Theorem 3.12 for order independent queries, TL and TL(E^i). Recall from Corollary 3.9 that TL is equivalent to $\mathrm{FO}_{tl}(\leq)$ and that TL(E^i) is equivalent to $\mathrm{FO}_{tl}(=)$. Moreover the linearity of a relation R can also be expressed in $\mathrm{FO}_{tl}(=)$ thus the proof of Theorem 3.12 can be readily applied to show that the order independent queries of $\mathrm{FO}_{tl}(\leq)$ are expressible in $\mathrm{FO}_{tl}(=)$ as long as a Craig Interpolation Theorem (Lemma 3.13) holds for FO_{tl}. This Craig Interpolation Theorem could be prove using an adaptation of the standard case to our special setting. But it turns out that this result also falls into a more general picture. Indeed [4] showed that, in general, Craig interpolation holds for a logic $\mathcal{L}$ as long as there exists a precise characterization of the logic using Ehrenfeucht-Fraïssé games such that winning strategies can be expressed in the logic. We will prove such a characterization in Theorem 4.5 of Section 4. From Theorem 4.5 and [4] we thus have:

Lemma 3.14 Let $f \in \mathrm{FO}_{tl}(R)$ and $g \in \mathrm{FO}_{tl}(S)$ be such that $\exists R\ f(R) \models \forall S\ g(S)$. Then there exists $h \in \mathrm{FO}_{tl}(=)$ such that:

$$\exists R\ f(R) \models h \qquad \text{and} \qquad h \models \forall S\ g(S).$$

Lemma 3.14 and Corollary 3.9 immediately yield:

Theorem 3.15 Let $\mathcal{P}$ be an order independent property expressible in TL. Then $\mathcal{P}$ is expressible in TL(E^i).

Corollary 3.16 TL(E^i) $\equiv$ TL $\cap$ FO(=).

Proof: The inclusion from left to right follows immediately from Proposition 3.4. For the converse inclusion: Let f be a property expressible in TL and in FO(=). Because f is expressible in FO(=), f is order independent. As f is also expressible in TL, Theorem 3.15 applies and we can conclude that f is expressible in TL(E^i). $\square$

4 TL(E^i) Expressiveness

This section is devoted to investigating the expressive power of the query language TL(E^i). We will establish that this language is unable to express all order independent properties. Because of Theorem 3.15 stating that order independent properties in TL are in TL(E^i), it will moreover provide a new proof that TL is strictly less expressive than FO($\leq$).

The main result of this section states that:

Theorem 4.1

$\mathrm{TL}(E^i) \subsetneq \mathrm{FO}(=)$ and more precisely,

the query twin cannot be expressed in $\mathrm{TL}(E^i)$.

As a matter of fact, we will prove that the query twin cannot be expressed in $\mathrm{TL}^*(E^i)$. Recall that the formulas in $\mathrm{TL}^*(E^i)$ are build using the first order rules plus the E^i modalities without restriction concerning the encapsulation under E^i modalities that holds for the language $\mathrm{TL}(E^i)$. In order to prove this result, we develop a technique à la Ehrenfeucht-Fraïssé based on two-player games. Ehrenfeucht-Fraïssé's games [9] offer an elegant alternative semantics for first order logic and provide a methodology for proving that a property is not definable in classical first order logic.

Next, we proceed in a very classical manner to the presentation of two-player games for our temporal language $\mathrm{TL}^*(E^i)$. We then show that these games characterize $\mathrm{TL}^*(E^i)$ definable properties: two temporal instances can be distinguished by a $\mathrm{TL}^*(E^i)$ property if and only if these two instances can be distinguished by our games. Our games, called T-games, are in fact very similar to Ehrenfeucht-Fraïssé's games. We just need to modify and add rules in order to take into account the modalities E^i of the temporal language $\mathrm{TL}^*(E^i)$.

From now on, we use the following objects:

- $\mathcal{I} = (I_1, \ldots, I_n)$ and $\mathcal{J} = (J_1, \ldots, J_m)$ are two temporal instances over the same vocabulary (same schema), whose domains are respectively $\mathrm{dom}_{\mathcal{I}}$ and $\mathrm{dom}_{\mathcal{J}}$.
- $\vec{a} = (a_1, \ldots, a_s)$, respectively $\vec{b} = (b_1, \ldots, b_s)$ are two vectors of s elements of $\mathrm{dom}_{\mathcal{I}}$, respectively $\mathrm{dom}_{\mathcal{J}}$.
- $\vec{x} = (x_1, \ldots, x_s)$ is a vector of s variables.
- $\mathrm{int}_{\mathcal{I}}$, respectively $\mathrm{int}_{\mathcal{J}}$ are the sets of indices $\{1, \ldots, n\}$, respectively $\{1, \ldots, m\}$.
- i_0 and j_0 are indices in $\mathrm{int}_{\mathcal{I}}$, respectively $\mathrm{int}_{\mathcal{J}}$.

A T-game is played by two-players. One of the player is called the spoiler and the other one is named the duplicator. They play with two kinds of pebbles: a bunch (sufficiently many) of pairs $(p_1, q_1), \ldots, (p_k, q_k)$ of data pebbles and one pair $(t_{\mathcal{I}}, t_{\mathcal{J}})$ of temporal pebbles. One move of the game consists of either a data move (d-move) or a temporal move (t-move).

Data move: The spoiler chooses a pair of data pebbles (p_i, q_i) and either places the pebble p_i on an element of $\mathrm{dom}_{\mathcal{I}}$ or places the pebble q_i on an element of $\mathrm{dom}_{\mathcal{J}}$. Then, in the first case, the duplicator ought to answer (in $\mathcal{J}$) by placing the pebble q_i on an element of $\mathrm{dom}_{\mathcal{J}}$ and alternatively in the second case by placing the pebble p_i on an element of $\mathrm{dom}_{\mathcal{I}}$. Thus at the end of the d-move the pair of data pebbles (p_i, q_i) is instantiated with a pair (a, b) of $\mathrm{dom}_{\mathcal{I}} \times \mathrm{dom}_{\mathcal{J}}$. Note that d-moves of T-games are exactly the moves described in Ehrenfeucht-Fraïssé's games for classical first order logic.

Temporal move of width w: This move is decomposed in two steps. (Step 1) The spoiler chooses to play in $\mathcal{I}$ (respectively in $\mathcal{J}$). He picks a sequence $(i_1, \ldots, i_w)$ of

w distinct integers in $\text{int}_{\mathcal{I}}$ (respectively in $\text{int}_{\mathcal{J}}$). The duplicator is then obliged to play in the opposite instance and picks a sequence $(j_1, \ldots, j_w)$ of w distinct integers in $\text{int}_{\mathcal{J}}$ (respectively in $\text{int}_{\mathcal{I}}$). Note here that none of the sequences are required to be ordered. (Step 2) Finally, the spoiler makes the choice of a pair (i_k, j_k) of integers which becomes the instantiation of the temporal pebbles $(t_{\mathcal{I}}, t_{\mathcal{J}})$. Note that, sometimes it may be impossible to play a t-move of width w. This may happen when the length of one of the instances $\mathcal{I}$ or $\mathcal{J}$ is less than w. Such a situation is refered to as an *uncompleted t-move*.

In general, for a T-game, some data pebbles (of course pairs of pebbles) could be placed before any round is started. It is also assumed that each round of a T-game always starts with an instantiation (i_0,j_0) of the temporal pebbles. $TG_r(\mathcal{I}, i_0, \vec{a}, \mathcal{J}, j_0, \vec{b})$ denotes a T-game played with $r + s$ pairs of data pebbles, in r moves and such that the temporal pebbles are initialized by (i_0,j_0), and for $i = 1..s$, the data pebbles pairs (p_i, q_i) are initialized by (a_i, b_i).

Let us now explain how a round of such a T-game is played by the spoiler and the duplicator. A round is a sequence of at most r moves, each move is either a d-move or a t-move as described above and the sequence satisfies the following restrictions:

- **Restriction on d-moves:** If a move is the d^{th} d-move of the round then it instantiates the $(s + d)^{\text{th}}$ data pairs of pebbles. Note that the spoiler is forced to choose a new pair of data pebbles at each d-move. Because one has reserved sufficiently many pairs $(r + s)$ of data pebbles, a round may be composed of d-moves only. But in general, it can happen that at the end of a round some pairs of data pebbles have not been used.
- **Winning completed moves:** After a move is completed, assume that the instantiation of $(t_{\mathcal{I}}, t_{\mathcal{J}})$ is (i, j) and that $(a_1, b_1), \ldots, (a_{s+d}, b_{s+d})$ are all the instantiations of the played data pebbles (meaning that d d-moves have been played already). Let us denote α the mapping defined over $\{a_l \mid l = 1 \ldots (s + d)\}$ by $\alpha(a_l) = b_l$. The duplicator wins the current move if the mapping α is a partial isomorphism of the states I_i and J_j.
- **Uncompleted t-moves:** An uncompleted t-move is a winning move for the spoiler and thus a loosing move for the duplicator.
- **Next move:** If the duplicator wins the move then the round can continue if the number of moves already played is strictly less than r.

The duplicator wins a round of the T-game $TG_r(\mathcal{I}, i_0, \vec{a}, \mathcal{J}, j_0, \vec{b})$ if the round terminates after r moves and the last move is a winning move for the duplicator. It should be clear that the only case where a round of a T-game ends before r moves have been played is when the duplicator looses a move.

The duplicator has a winning strategy for the game $TG_r(\mathcal{I}, i_0, \vec{a}, \mathcal{J}, j_0, \vec{b})$ if he/she is able to win all possible rounds. A winning strategy for the duplicator is denoted by $(\mathcal{I}, i_0, \vec{a}) \sim_r (\mathcal{J}, j_0, \vec{b})$. It is quite easy to show from the definition of the T-games that the relation $\sim_r$ is an equivalence relation. Intuitively, the equivalence $(\mathcal{I}, i_0, \vec{a}) \sim_r (\mathcal{J}, j_0, \vec{b})$ indicates that the temporal instances $\mathcal{I}$ and $\mathcal{J}$ cannot be distinguished by looking at several instants and focussing only on restricted sets of constants, the observation starting at the instant i_0 with the elements $\vec{a}$ for the temporal instance $\mathcal{I}$ and at the instant j_0 with the elements $\vec{b}$ for the temporal instance $\mathcal{J}$.

In the following, we may need to restrict the width of the temporal moves of a T-game to be bounded. We write $(\mathcal{I}, i_0, \vec{a}) \sim_r^w (\mathcal{J}, j_0, \vec{b})$ to denote that the duplicator has a winning strategy for the game $TG_r^w(\mathcal{I}, i_0, \vec{a}, \mathcal{J}, j_0, \vec{b})$ whose t-moves have a width less or equal to w.

Example 4.2 Let us consider[2] the binary relational schema R and the two states I and J depicted as graphs in figure 4. If we consider the temporal instances $\mathcal{I} = (I)$ and $\mathcal{J} = (J)$, it is quite trivial to see that the duplicator has a winning startegy for the T-Game $TG_2(\mathcal{I}, 1, \_, \mathcal{J}, 1, \_)$ which is played in two moves, all data pebbles being off the board before each round. However, the duplicator has no winning strategy for the T-game $TG_3(\mathcal{I}, 1, \_, \mathcal{J}, 1, \_)$ with three moves. Indeed, let us consider the round starting by the two d-moves (1,1), (2,2). Then assume that the spoiler places the data pebble p_3 over 3 in $\mathcal{I}$. The duplicator is unable to answer appropriately because (1,2), (2,3) and (3,1) are "edges" in $\mathcal{I}$ but there is no $a \in \text{dom}_{\mathcal{J}}$ such that (1,2), (2,a) and $(a, 1)$ are in $\mathcal{J}$.

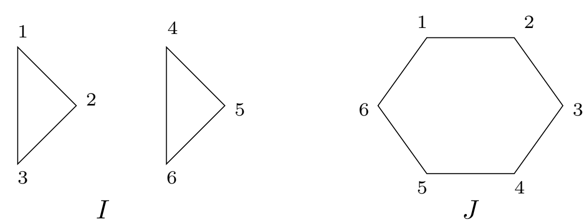

Figure 4: Two states for our running example

Let us now consider the temporal instances $\mathcal{I} = (I,I,J,I)$ and $\mathcal{J} = (I,J,I,J)$. It is not difficult to check that the duplicator has a winning strategy for the T-game $TG_3(\mathcal{I}, 1, \_, \mathcal{J}, 1, \_)$ because even in a situation where for instance the initial move is a t-move of width 4 and where the duplicator is forced by the spoiler to put a state I in correspondence with a state J, by the first part of the example, we can infer that the duplicator has a winning strategy for the following 2 moves. Let us detail the first t-move: the spoiler starts by picking the 4 instants (1, 2, 3, 4) in $\mathcal{I}$; the duplicator answers by the 4 instants (1, 3, 2, 4); finally the spoiler chooses to instantiate the time pebbles with the pair (4,4). If the two following moves are d-moves, the duplicator has a winning strategy. Of course, he/she has a wining strategy for the two following moves.

However, considering the T-game $TG_4(\mathcal{I}, 1, \_, \mathcal{J}, 1, \_)$ with 4 moves, leads to a winning strategy but this time for the spoiler. Indeed, the duplicator maintains a winning strategy for the T-game $TG_4^2(\mathcal{I}, \_, 1, \mathcal{J}, 1, \_)$ that is, when restricting the width of t-moves to be 1 or 2. But as soon as t-moves of width 3 are allowed, during an initial

[2]This example is adapted from [13].

t-move, the spoiler is able to force the duplicator to put a state I in correspondence with a state J and then, by the first part of the example, we know that the duplicator has no winning strategy for the 3 moves that remain to be played if these moves are d-moves.

We now turn to the fundamental relationship between our T-games and the temporal language $\mathrm{TL}^*(E^i)$. As a matter of fact, we establish that the equivalence relation $\sim_r^w$ characterizes the elementary equivalence $\equiv_r^w$ of instances with respect to the language $\mathrm{TL}^*(E^i)$.

Definition 4.3 The temporal instances $\mathcal{I}$ and $\mathcal{J}$ are ${}^w_r$-equivalent with respect to $\vec{a}$, $\vec{b}$ and with respect to the instants i_0, j_0, and we write $(\mathcal{I},\ i_0,\ \vec{a}) \equiv_r^w (\mathcal{J},\ j_0,\ \vec{b})$ iff for all formula φ in $\mathrm{TL}^*(E^i)$ such that $\mathrm{qtr}(\varphi) \leq r$ and $\mathrm{wth}(\varphi) \leq w$, we have:

$$[\mathcal{I}, i_0, \nu_{\vec{a}}] \models_{tl} \varphi \ \text{ iff } \ [\mathcal{J}, j_0, \nu_{\vec{b}}] \models_{tl} \varphi$$

where the valuations $\nu_{\vec{a}}$ and $\nu_{\vec{b}}$ are defined by $\nu_{\vec{a}}(x_l) = a_l$, respectively $\nu_{\vec{b}}(x_l) = b_l$.

The relationship between the duplicator having a winning strategy for a T-game TG_r^w played over two temporal instances $\mathcal{I}$ and $\mathcal{J}$ and the equivalence of these two temporal instances with respect to $\mathrm{TL}^*(E^i)$ formulas having a quantifier temporal rank bounded by r and a width bounded by w will be formally established by Theorem 4.5. The purpose of the following example is to illustrate this relationship.

Example 4.4 For instance, as we have seen in the second part of example 4.2, $(\mathcal{I}, 1, \_) \not\sim_4^3 (\mathcal{J}, 1, \_)$. Indeed, the spoiler's winning strategy summarized in this example can be viewed as playing the following formula $\mathcal{F}$:

$$E^3(\exists x_1 \exists x_2 \exists x_3 (R(x_1, x_2) \wedge R(x_2, x_3) \wedge R(x_3, x_1)))$$

Note that the quantifier temporal rank of this sentence is 4 and its width is 3. Note of course that $(\mathcal{I}, 1, \_) \not\equiv_4^3 (\mathcal{J}, 1, \_)$, because $[\mathcal{I}, 1] \models_{tl} \mathcal{F}$ and $[\mathcal{J}, 1] \not\models_{tl} \mathcal{F}$.

In order to establish the relationship between equivalence of temporal structures and winning strategies for certain T-games, we need an intermediate technical tool which is a slight variant of the Hintikka formulas [9]. The T-Hintikka formulas $\Phi_{\rho,w}^{\vec{a},i_0}[\mathcal{I}](\vec{x}, t)$, associated with the temporal instance $\mathcal{I}$ and the vector $\vec{a}$ at time points i_0 with free variables $\vec{x}, t$, are intended to capture all $\mathrm{FO}_{tl}^*(=)$ formulas which are satisfied by $\mathcal{I}$ at i_0 given the valuation $\nu_{\vec{a}}$ of the free variables $\vec{x}$. These formulas are inductively defined as follows:

- $\Phi_{0,w}^{\vec{a},i_0}[\mathcal{I}](\vec{x}, t)$ is the conjunction of all atomic formulas $\phi(\vec{x}, t)$ such that $[\mathcal{I},\ i_0,\ \nu_{\vec{a}}] \models \phi(\vec{x}, t)$.
- for $\rho \geq 1$:

$$\Phi_{\rho,w}^{\vec{a},i_0}[\mathcal{I}](\vec{x}, t) = \quad \Phi_{\rho-1,w}^{\vec{a},i_0}[\mathcal{I}](\vec{x}, t) \qquad \bigwedge_{a \in \mathrm{dom}_{\mathcal{I}}} \exists x_{s+1} \Phi_{\rho-1,w}^{\vec{a}a,i_0}[\mathcal{I}](\vec{x}x_{s+1}, t)$$

$$\wedge \ \forall x_{s+1} \bigvee_{a \in \mathrm{dom}_{\mathcal{I}}} \Phi^{\vec{a}a,i_0}_{\rho-1,w}[\mathcal{I}](\vec{x}x_{s+1}, t)$$
$$\wedge \ \Psi^{\vec{a}}_{\rho,w}[\mathcal{I}](\vec{x}, t) \quad \wedge \quad \Psi'^{\vec{a}}_{\rho,w}[\mathcal{I}](\vec{x}, t)$$

where:

$$\Psi^{\vec{a}}_{\rho,w}[\mathcal{I}](\vec{x}, t) = \bigwedge_{k \leq w} \ \bigwedge_{\not\Vert^{n}_{1}(i_1 \cdots i_k)} \exists t_1 \cdots t_k \bigwedge_{u \neq v} t_u \neq t_v \ \wedge \ \bigwedge_{j \leq k} \Phi^{\vec{a},i_j}_{\rho-1,w}[\mathcal{I}](\vec{x}, t_j), \quad \text{and}$$
$$\Psi'^{\vec{a}}_{\rho,w}[\mathcal{I}](\vec{x}, t) = \bigwedge_{k \leq w} \forall t_1 \cdots t_k \bigwedge_{u \neq v} t_u \neq t_v \ \rightarrow \ \bigwedge_{\not\Vert^{n}_{1}(i_1 \cdots i_k)} \ \bigwedge_{j \leq k} \Phi^{\vec{a},i_j}_{\rho-1,w}[\mathcal{I}](\vec{x}, t_j)$$

with $\bigwedge_{\not\Vert^{n}_{1}(i_1 \cdots i_k)}$ denoting "k distinct integers $i_1..i_k$ in $[1..n]$".

It is rather immediate to show that the T-Hintikka formulas $\Phi^{\vec{a},i_0}_{\rho,w}[\mathcal{I}](\vec{x}, t)$ is indeed in $\mathrm{FO}^*_{tl}(=)$ and that $[\mathcal{I},\ i_0,\ \nu_{\vec{a}}] \models \Phi^{\vec{a},i_0}_{\rho,w}[\mathcal{I}](\vec{x}, i_0)$. Note also that it is quite immediate to show that $[\mathcal{I},\ \mathrm{i},\ \nu_{\vec{a}}] \models \Psi^{\vec{a}}_{\rho,w}[\mathcal{I}](\vec{x}, t)$, for all i. Finally note that by Theorem 3.5 and remark 3.8 the formula $\Phi^{\vec{a},i_0}_{\rho,w}[\mathcal{I}](\vec{x}, t)$ is equivalent to a $\mathrm{TL}^*(E^i)$ formula whose quantifier temporal rank is less or equal to ρ, and whose width is less or equal to w. Therefore, in the following, we will use $\Phi^{\vec{a},i_0}_{\rho,w}$ indistinctly to denote a formula of $\mathrm{FO}^*_{tl}(=)$ or $\mathrm{TL}^*(E^i)$.

Finally, we have all necessary ingredients to state the following important result:

Theorem 4.5 The following statements are equivalent:

1. $(\mathcal{I},\ i_0,\ \vec{a}) \sim^w_r (\mathcal{J},\ j_0,\ \vec{b})$
2. $(\mathcal{I},\ i_0,\ \vec{a}) \equiv^w_r (\mathcal{J},\ j_0,\ \vec{b})$
3. $[\mathcal{J},\ j_0,\ \nu_{\vec{b}}] \models_{tl} \Phi^{\vec{a},i_0}_{r,w}[\mathcal{I}](\vec{x})$

Proof of theorem 4.5:

We are going to prove successively that [2. ⇒ 3.], [3. ⇒ 1.], and [1. ⇒ 2.].

`[2. ⇒ 3.]` It is quite immediate to derive that $[\mathcal{J},\ j_0,\ \nu_{\vec{b}}] \models_{tl} \Phi^{\vec{a},i_0}_{r,w}[\mathcal{I}](\vec{x})$ from the hypothesis $(\mathcal{I},\ i_0,\ \vec{a}) \equiv^w_r (\mathcal{J},\ j_0,\ \vec{b})$ simply, because by definition of $\Phi^{\vec{a},i_0}_{r,w}[\mathcal{I}](\vec{x})$, we have that $[\mathcal{I},\ i_0,\ \vec{a}] \models_{tl} \Phi^{\vec{a},i_0}_{r,w}[\mathcal{I}](\vec{x})$ and $\Phi^{\vec{a},i_0}_{r,w}[\mathcal{I}](\vec{x})$, as a $\mathrm{TL}^*(E^i)$ formula, has the desired temporal quantifier rank and width.

`[3. ⇒ 1.]` We proceed by induction on the number μ of moves.

Initial case: Assume that $[\mathcal{J},\ j_0,\ \nu_{\vec{b}}] \models_{tl} \Phi^{\vec{a},i_0}_{0,w}[\mathcal{I}](\vec{x})$. It is then obvious that the duplicator has a winning strategy for the 0-move game $\mathrm{TG}^w_0(\mathcal{I},\ i_0,\ \vec{a},\ \mathcal{J},\ j_0,\ \vec{b})$ because our assumption entails that the states I_{i_0} and I_{j_0} are equivalent for first order formulas without quantifiers.

Induction Step: Suppose that `[3. ⇒ 1.]` has been proved for $\mu < r$ and assume that [#] $[\mathcal{J},\ j_0,\ \nu_{\vec{b}}] \models_{tl} \Phi^{\vec{a},i_0}_{r,w}[\mathcal{I}](\vec{x})$. Let us consider the T-game $\mathrm{TG}^w_r(\mathcal{I},\ i_0,\ \vec{a},\ \mathcal{J},\ j_0,\ \vec{b})$. Two cases arise:

[case 1] The first move is a t-move. Assume that the spoiler plays in $\mathcal{I}$ and picks the sequence of distinct instants $(i_1,\ldots,i_v)$ where $v \leq w$. Because of the assumption [#], we have that $[\mathcal{J},\, j_0,\, \nu_{\vec{b}}] \models_{tl} \Psi^{\vec{a}}_{r,w}[\mathcal{I}](\vec{x})$. Thus, by definition, there exists a sequence of distinct instants $(j_1,\ldots,j_v)$ such that $\forall \ell \in \{j_1,\ldots,j_v\}$, $[\mathcal{J},\, j_\ell,\, \nu_{\vec{b}}] \models_{tl} \Phi^{\vec{a},i_\ell}_{\rho-1,w}[\mathcal{I}](\vec{x})$. Thus the duplicator's answer to the spoiler is $\{j_1,\ldots,j_v\}$. Assume that during the second phase of the t-move, the spoiler makes the choice to put the temporal pebbles $t_{\mathcal{I}}$ and $t_{\mathcal{J}}$ over the instants i_ℓ and j_ℓ. This t-move is a winning move for the duplicator because, by construction of the duplicator's answer, we have that $[\mathcal{J},\, j_\ell,\, \nu_{\vec{b}}] \models_{tl} \Phi^{\vec{a},i_\ell}_{r-1,w}[\mathcal{I}](\vec{x})$. At this point, the induction hypothesis tells us that the duplicator has a winning strategy for the T-game $\mathrm{TG}^w_{r-1}(\mathcal{I},\, i_\ell,\, \vec{a},\, \mathcal{J},\, j_\ell,\, \vec{b})$ and allows us to conclude that the duplicator has a winning strategy for the T-game $\mathrm{TG}^w_r(\mathcal{I},\, i_0,\, \vec{a},\, \mathcal{J},\, j_0,\, \vec{b})$. Assume now that the spoiler plays in the temporal instance $\mathcal{J}$ and picks the sequence of distinct instants $(j_1,\ldots,j_v)$ where $v \leq w$. Because of the assumption [#], we have that $[\mathcal{J},\, j_0,\, \nu_{\vec{b}}] \models_{tl} \Psi'^{\vec{a}}_{r,w}[\mathcal{I}](\vec{x})$. Thus, by definition, one of the disjuncts must be true and this gives a sequence of distinct instants $(i_1,\cdots,i_v)$ such that $\forall \ell \in \{j_1,\ldots,j_v\}$, $[\mathcal{J},\, j_\ell,\, \nu_{\vec{b}}] \models_{tl} \Phi^{\vec{a},i_\ell}_{\rho-1,w}[\mathcal{I}](\vec{x})$ and we conclude as above that this is indeed a winning strategy.

[case 2] Suppose now that the first move is a d-move. Assume that the spoiler plays over $\mathcal{I}$ and places the pebble p_{s+1} over a. Let us consider the T-Hintikka formula $\Phi^{\vec{a}a,i_0}_{r-1,w}[\mathcal{I}](\vec{x}x_{s+1})$ associated with $\mathcal{I}$. By definition of $\Phi^{\vec{a},i_0}_{r,w}[\mathcal{I}](\vec{x})$ and by our assumption [#], we can infer that $[\mathcal{J},\, j_0,\, \nu_{\vec{b}}] \models_{tl} \exists x_{s+1}\Phi^{\vec{a}a,i_0}_{r-1,w}[\mathcal{I}](\vec{x}x_{s+1})$. This allows the duplicator to choose a constant b such that $[\mathcal{J},\, \nu_{\vec{b}b},\, j_0] \models_{tl} \Phi^{\vec{a}a,i_0}_{r-1,w}[\mathcal{I}](\vec{x}x_{s+1})$. and to win the move. At this point, the induction hypothesis tells us that the duplicator has a winning strategy for the T-game $\mathrm{TG}^w_{r-1}(\mathcal{I},\, i_0,\, \vec{a}a,\, \mathcal{J},\, j_0,\, \vec{b}b)$ and thus he/she has a winning strategy for the T-game $\mathrm{TG}^w_r(\mathcal{I},\, i_0,\, \vec{a},\, \mathcal{J},\, j_0,\, \vec{b})$.

Assume now that the spoiler plays over $\mathcal{J}$ and places the pebble q_{s+1} over b. Because of the assumption [#], we have that $[\mathcal{J},\, j_0,\, \nu_{\vec{b}}] \models_{tl} \forall x_{s+1} \bigvee_{a\in \mathrm{dom}_{\mathcal{I}}} \Phi^{\vec{a}a,i_0}_{\rho-1,w}[\mathcal{I}](\vec{x}x_{s+1})$ and thus there exists an element a such that $[\mathcal{J},\, j_0,\, \nu_{\vec{b}b}] \models_{tl} \Phi^{\vec{a}a,i_0}_{\rho-1,w}[\mathcal{I}](\vec{x}x_{s+1})$. Let this element a be the duplicator's answer to win the move. At this point, the induction hypothesis leads to conclude that the duplicator has a winning strategy for the T-game $\mathrm{TG}^w_r(\mathcal{I},\, i_0,\, \vec{a},\, \mathcal{J},\, j_0,\, \vec{b})$.

[1. $\Rightarrow$ 2.] We obviously proceed by induction on the number ρ of moves.
Initial step: Immediate.
Induction step: We now assume that [1. $\Rightarrow$ 2.] has been proved for $\rho < r$ and suppose that $(\mathcal{I},\, i_0,\, \vec{a}) \sim^w_r (\mathcal{J},\, j_0,\, \vec{b})$. In order to prove that $(\mathcal{I},\, i_0,\, \vec{a}) \equiv^w_r (\mathcal{J},\, j_0,\, \vec{b})$, we proceed by contradiction and assume that there exists a formula Ψ with qtr(Ψ)=r such that $[\mathcal{I},\, i_0,\, \nu_{\vec{a}}] \models_{tl} \Psi$ and $[\mathcal{J},\, j_0,\, \nu_{\vec{b}}] \not\models_{tl} \Psi$. First assume that Ψ is $\exists x_{s+1}\psi$ with qtr(ψ)=$r-1$. This case is rather simple and treated as in the proof for the classical first order logic. Now assume that Ψ is $E^v\psi$. The spoiler is starting a t-move by selecting a sequence of v distinct indices $(i_1,\ldots,i_v)$ of $\mathcal{I}$ such that $[\mathcal{I},\, i_\ell,\, \nu_{\vec{a}}] \models_{tl} \psi$ for $\ell = 1..v$. Of course, the duplicator prepares an answer $(j_1,\ldots,j_v)$ of v distincts indices of $\mathcal{J}$ and because $[\mathcal{J},\, j_0,\, \nu_{\vec{b}}] \not\models_{tl} \Psi$ the spoiler is able to complete the move by a choice of (i_f, j_f) such that $[\mathcal{I},\, i_f,\, \nu_{\vec{a}}] \models_{tl} \psi$ and $[\mathcal{J},\, j_f,\, \nu_{\vec{b}}] \not\models_{tl} \psi$. However because the duplicator has a winning

strategy for the T-game $TG_r^w(\mathcal{J}, i_0, \vec{a}, \mathcal{J}, j_0, \vec{b})$, he/she has a winning strategy for the T-game $TG_{r-1}^w(\mathcal{I}, i_f, \vec{a}, \mathcal{J}, j_f, \vec{b})$. More precisely, this entails (by induction hypothesis) that we have both $(\mathcal{I}, i_f, \nu_{\vec{a}}) \sim_{r-1}^w (\mathcal{J}, j_f, \nu_{\vec{b}})$ and $(\mathcal{I}, i_f, \vec{a}) \equiv_{r-1}^w (\mathcal{J}, j_f, \vec{b})$. This is a contradiction with the fact that $[\mathcal{I}, i_f, \nu_{\vec{a}}] \models_{tl} \psi$ and $[\mathcal{J}, j_f, \nu_{\vec{b}}] \not\models_{tl} \psi$. □

The previous theorem is quite powerful because, as in the framework of classical first order logic, it provides a methodology for determining what cannot be said in the temporal language $TL^*(E^i)$. Basically, in order to show that a property $\mathcal{P}$ cannot be expressed by a formula in $TL^*(E^i)$, it is sufficient to show, for each w and for each r, that there exists two temporal instances $\mathcal{I}_r^w$ and $\mathcal{J}_r^w$ such that, on the one hand $\mathcal{I}_r^w$ satisfies the property $\mathcal{P}$ and $\mathcal{J}_r^w$ does not, and on the other hand $\mathcal{I}_r^w$ and $\mathcal{J}_r^w$ cannot be distinguished by a T-game TG_r^w (i.e. $\mathcal{I}_r^w \sim_r^w \mathcal{J}_r^w$). This methodology is developed for establishing the following:

Lemma 4.6 The twin property is not $TL^*(E^i)$ definable, and thus it is not $TL(E^i)$ definable.

Proof: Let us fix both w and r for the rest of the presentation of this proof. The temporal instances that are constructed now are closely related to the ones used by Toman and Niwinski in [18]. In the sequel, we consider a set $S = \{1, \ldots, 2k+1\}$, for k sufficiently big, together with the following pairs of temporal instances $\mathcal{I}$ and $\mathcal{J}$ (for the sake of readability, we do not insert the indices w and r but the reader should keep in mind that these indices are parameters for building our temporal instances):

- $\mathcal{I}$ is an enumeration of the (k+1)-subsets [3] of S with exactly two occurrences of each subset. Note that, by definition of $\mathcal{I}$, we have that $[\mathcal{I}, t] \models_{tl}$ twin, for any t.
- $\mathcal{J}$ is an enumeration of the k-subsets and (k+1)-subsets of S, this time with exactly one occurrence of each subset. Note that, by construction, we have that $[\mathcal{J}, t] \not\models_{tl}$ twin, for any t.

For instance, if $k = 1$, $S = \{1, 2, 3\}$, a possible temporal instance $\mathcal{I}$ is $\{1,2\}, \{1,3\}, \{2,3\}, \{1,2\}, \{1,3\}, \{2,3\}$ and a possible temporal instance $\mathcal{J}$ is $\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}$.

Although it not very important for the remaining, the reader may notice that the temporal instances $\mathcal{I}$ and $\mathcal{J}$ have the same number of states because of $\binom{2k+1}{k} = \binom{2k+1}{k+1}$.

We are now going to show that these two temporal instances cannot be distinguished by the T-game TG_r^w. For the time being, we only assume that k is big enough. The exact condition on k will be given later.

In order to show that $(\mathcal{I}, i_0, \_) \sim_r^w (\mathcal{J}, j_0, \_)$ that is, in order to exhibit a winning strategy for the duplicator playing the T-game $TG_r^w(\mathcal{I}, i_0, \_, \mathcal{J}, j_0, \_)$, we assume that g moves of a round have been successfully played by the duplicator. Consider that d moves among these g moves are d-moves and thus d pairs of data pebbles are instantiated with the pairs of elements (a_1, b_1), …, (a_d, b_d) on the one hand and the

[3] a i-subset is a subset of cardinality i

temporal pair of pebbles is instantiated with $(i,\ j)$. Let us examine the two possible cases for the next (g+1)$^{\text{th}}$ move.

[case 1]: The spoiler chooses to play a d-move in $\mathcal{I}$ (the dual move in $\mathcal{J}$ would be treated in a similar way). Consider that the spoiler places the pebble p_{d+1} on the element a_{d+1} of S. Then the duplicator is going to answer with an element b_{d+1} that he/she will carefully choose such that $a_{d+1} \in I_i$ iff $b_{d+1} \in J_j$. Because of the cardinality of S, he/she is always able to make to such a choice.

[case 1]: The spoiler chooses to play a t-move in $\mathcal{I}$ and starts by providing v distinct instants $i_1, \dots, i_v$ with $v \leq w$. The duplicator is then going to answer as follows. First, he/she considers the set of states of $\mathcal{I}$ corresponding to the choice of the spoiler. Let us assume that this set is E=$\{I_{i_1},\ \dots,\ I_{i_v}\}$. Next, he/she builds the equivalence classes of E induced by the equivalence relation ∂ defined by $I_t\ \partial\ I_{t'}$ iff $I_t \cap \{a_1, \dots, a_d\}$=$I_{t'} \cap \{a_1, \dots, a_d\}$. Intuitively, I_t and $I_{t'}$ are in the same equivalence class if they share the same "played" elements in $\mathcal{I}$. The reader can notice here that the maximal number of equivalent states (or the maximal cardinality of an equivalence class) is less or equal to v and thus to w. Let us denote $E_1, \dots, E_z$ the partition of E induced by ∂. Each class E_ℓ is characterized by the set $I_t \cap \{a_1, \dots, a_d\}$ where $I_t \in E_\ell$ and that set is enumerated as $a_{h_1} \dots a_{h_\ell}$. The cardinality of E_ℓ is denoted c_ℓ (notice that, $c_\ell \leq w$). Then, for a class E_ℓ, the duplicator is going to select c_ℓ states K in $\mathcal{J}$ such that $K \cap \{b_1, \dots, b_d\} = \{b_{h_1}, \dots, b_{h_\ell}\}$. This choice entails that each of the c_ℓ states K is partially isomorphic to any of the instance in the equivalence class E_ℓ. This choice can always be done as soon as the cardinality of the set S on which is based the construction of $\mathcal{I}$ and $\mathcal{J}$ is big enough. The cardinality 2k+1 of S is determined to satisfy the following: the number of subsets of S including u fixed elements and excluding $p - u$ other fixed elements is greater or equal to $w + 1$ for $p = 1 \cdots r$ and $u = 1 \cdots p$. This is insured as soon as k satisfies $\binom{2k+1-r}{k-r} \geq w + 1$.

Of course, because of the construction of the duplicator's answer, the second step of the t-move leads to a winning move for the duplicator. □

Corollary 4.7 The twin property cannot be expressed in TL.

Proof: Indeed, if we assume that the property twin can be expressed in TL then Theorem 3.15 entails that twin can be expressed in TL(E^i), since it is order independent. This is a contradiction with the fact that twin cannot be defined in TL*(E^i). □

5 Conclusion

The alternative proof schema provided for showing that TL is strictly less expressive than FO(<) via the syntactic characterization of the order independent properties expressible in TL, seems to open a new direction for similar results concerning other temporal languages as a matter of fact temporal languages that are more expressive than TL. For instance, in [5], the implicit query language RNTL has been introduced and proved to be more expressive than TL. This language is obtained from TL by adding two new temporal modalities $\aleph$ et $\Re$. The operator $\aleph$, which has been independently introduced in [16], restricts the scope of the usual TL past operators **Since** and **Prev**. Usually, the evaluation of the past operators in a TL formula φ is made by considering the initial instant 1 as the (unique) origin of time. However, when evaluating a RNTL formula of the form $\aleph(\varphi)$ at an instant t, a new origin of time is created: the instant t of evaluation becomes the starting point with respect to which the past operators of φ will be evaluated. The time interval going from 1 to t is temporally "forgotten". The operator $\Re$ plays a dual role by restoring the time fragments which have been "forgotten" when evaluating subformulas containing $\aleph$. We will not present here the temporal language RNTL in detail, but only remind some important results concerning its expressive power [5].

First, RNTL is able to express the twin property. By the way, this property can be expressed in NTL, a sublanguage of RNTL, obtained by adding only the operator $\aleph$ to TL. In fact, the NTL formula expressing twin is the following:

$$F\aleph F(\forall x(S(x) \leftrightarrow P(\mathsf{First} \wedge S(x))))$$

where First is the formula $\neg$**Prev** true.

Despite being conjectured in [5], the equivalence between RNTL and FO($\leq$) remains an open problem. We think that, using a proof schema similar to the one proposed in the present paper, it is possible to show that RNTL is strictly less expressive than FO($\leq$). The property which seems to be the candidate to separate these two languages is gen-twin, stating that *"each state has a twin"*. This property can easily be expressed in FO($\leq$) by the formula $\forall i \exists j(i \neq j \wedge \forall x[S^{st}(x,i) \leftrightarrow S^{st}(x,j)])$. In order to prove that this property cannot be expressed in RNTL, we would have to introduce a counterpart of the language TL(E^i) with respect to RNTL, that is, a language $\mathcal{L}$ which expresses exactly the order independent properties of RNTL. And then, we would have to develop a technique based on Ehrenfeucht-Fraïssé's games, in order to show that the order independent property gen-twin is not expressible in the language $\mathcal{L}$.

References

[1] Abiteboul, S., Herr, L. and Van den Bussche J.: Temporal Connectives Versus Explicit Timestamps in Temporal Query Languages, In Recent Advances in Temporal Databases, S. Clifford and A. Tuzhilin, Eds, Springer Verlag (1995) 43–60.

[2] Abiteboul, S., Herr, L. and Van den Bussche, J.: Temporal Connectives Versus Explicit Timestamps to Query Temporal Databases. JCSS 58(1): 54-68 (1999).

[3] Abiteboul, S., Hull, R. and Vianu, V.: Foundations of Databases, Addison-Wesley (1995).

[4] Barwise, J. and Benthem, J.: Interpolation, Preservation, and Pebble games. J. Symbolic Logic, 64:2, pp 881–903, 1999.

[5] Bidoit, N. and De Amo, S.: Implicit temporal query languages : towards completeness, FST&TCS, Chennai, India, LNCS Vol. 1738, 1999, pp 245-257.

[6] Chang, C. and Keisler, H.: Model Theory, North Holland, (1990).

[7] Chomicki, J.: Temporal Query Languages: a survey, Temporal Logic, First Int. conf., LNAI 827 (1994) 506–534.

[8] Chomicki, J. and Toman, D.: Temporal Logic in Information Systems, Logics for Databases and Information Systems (1998) 31-70.

[9] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer Verlag, 1995.

[10] Emerson, E. A.: Temporal and Modal Logic, In Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, Jan van Leeuwen, Ed., Elsevier Science Publishers (1990) 995–1072.

[11] Gabbay, D., M., Pnueli, A., Shelah, S., and Stavi, J.: On the Temporal Basis of Fairness, Symposium on Principles of Programming Languages, (1980) 163–173.

[12] Hodkinson I., Wolter F., and Zakharyaschev M.: Decidable fragments of first-order temporal logics, Annals of Pure and Applied Logic 106, 2000, 85 - 134.

[13] Immerman, N.: Descriptive Complexity. Springer Graduate Texts in Computer Science, (1999).

[14] H.W. Kamp: Tense Logic and the Theory of Linear Order, PhD thesis, University of California, Los Angeles (1968).

[15] Kushilevitz, E., and Nisan, N.: Communication Complexity, Cambridge University Press, 1996.

[16] Laroussinie, L., and Schnoebelen, Ph.: A hierarchy of temporal logics with past, TCS, 148, 2, (1995) 303–324.

[17] Meyer, A.R.: Weak monadic second order theory of successor is not elementary recursive. In Proceedings Logic Colloquium, Lecture Notes in Mathematics, Vol. 453, pp. 132–154, Springer-Verlag, 1975.

[18] Niwinski, D. and Toman, D.: First Order Queries over Temporal Databases Inexpressible in Temporal Logic, EDBT (1996) 307–324.

[19] Sistla, A.P., and Clarke, E.M.: The complexity of proposicional linear temporal logic. Journal of the ACM, 32(3) : 733–749, July 1985.

[20] Sturm H. and Wolter F.: First-order expressivity for S5-models: modals vs. two-sorted languages. Journal of Philosophical Logic 30, 2001, 571 - 591.

[21] Wolper, P.: Temporal Logic can be more expressive. Information and Control, 56, 72–99, 1983.