**Mariusz Giero**
University of Białystok

# QUERYING TEMPORAL DATABASE WITH THE LANGUAGE OF FIRST-ORDER TEMPORAL LOGIC[1]

## 1. Introduction

A temporal database [Etz, Ste, Tan] is defined as a database maintaining object histories, i.e., past, present, and possibly future data. There are numerous application domains dealing with temporal data: Medical Systems (e.g. patient's records), Computer Applications (e.g. history of file back ups), Archive Management Systems (e.g. sporting events, publications and journals), Reservation Systems (e.g. when was a flight booked) and many others [Sno]. Support for time-varying data within a traditional relational database is not straightforward. There have been more than two dozens extended relational data models proposed [JenSno]. Time-varying data is commonly represented by timestamping values [JenSno, Jen]. Timestamps can be time points, intervals or a set of intervals and can be added to tuples or attributes. There are also different considerations of what time stamps represent: valid time, i.e., time when data (tuple) is true in the universe of discourse, transaction time, i.e., time when data is stored in a database or both time references together.

In this paper we consider a temporal database model with tuple timestamping. Tuples are timestamped by a set of intervals which represent valid time.

[1] The research reported in this paper is part of the project entitled *Temporal Representation of Knowledge and Its Implementation in the Computer Systems of Medical Conducts* supported by the Polish Ministry of Science and Higher Education, grant no. 3 T11F 011 30.

## 2. Structure of Time

In this paper, we assume that the flow of time $(T, <)$ is a linear, discrete and ordered structure with no end points. $T$ is a set of time points and $<$ is a binary order relation defined on $T$ which satisfies the following conditions:

- transitivity $\forall x, y(x < y \wedge y < z \rightarrow x < z)$
- irreflexivity $\forall x \neg(x < x)$
- totality $x = y$ or $x < y$ or $y < x$, where $x, y, z \in T$

## 3. Relational Database

The relational data model was introduced in the 1970s by E. F. Codd [Cod, Dat]. Currently, it is the most widespread data model used for database applications. Formally, it can be defined as follows:

**Definition 1.**
A *relational database schema* is a quintuple $S = (R, A, D, \mathrm{attr}, \mathrm{dom})$, where:

- $R = \{R_1, \ldots, R_k\}$ is a set of relation names,
- $A = \{A_1, \ldots, A_n\}$ is a set of attribute names,
- $D = \{D_1, \ldots, D_m\}$ is a set of domains,
- $\mathrm{attr} : R \rightarrow \mathrm{TUP}(A)$, where $\mathrm{TUP}(A)$ denotes a set of finite tuples of different elements of $A$, is a mapping that assigns to each relation name a tuple of attribute names,
- $\mathrm{dom} : A \rightarrow D$ is a mapping that assigns to each attribute name a domain.

**Definition 2.**
An *instance of relational database* (or just a *relational database*) for schema $S = (R, A, D, \mathrm{attr}, \mathrm{dom})$ is a set $DB = \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_k\}$ where $\boldsymbol{R}_i$ is a relation instance (or just a relation) over the relation name $R_i \in R$, i.e.,

$$\boldsymbol{R}_i \subseteq \mathrm{dom}(A_1) \times \ldots \times \mathrm{dom}(A_l),$$

where $\mathrm{attr}(R_i) = (A_1, \ldots, A_l)$, $l \leq n$ and $\times$ is the Cartesian product operator.

**Example 1.**
Let us consider a database storing data about the patients at a hospital. For simplicity, we will use only two attributes and one relation name.

$R = \{\text{PATIENTS}\}$, $A = \{\text{ID}, \text{NAME}\}$, $D = \{\text{N}, \text{CHAR}\}$[2],
$\text{attr}(\text{PATIENTS}) = (\text{ID}, \text{NAME})$,
$\text{dom}(\text{ID}) = \text{N}$, $\text{dom}(\text{NAME}) = \text{CHAR}$.

$DB = \{\textbf{PATIENTS}\}$
$\textbf{PATIENTS} = \{(1, \text{Kowalski}), (2, \text{Kozłowski}), (3, \text{Piasecka})\}$

The fact $(x, y) \in \textbf{PATIENTS}$ means that a person named $y$ with an identifier $x$ is a patient at a specific hospital. A database can also be represented (not formally, for the sake of readability) as a set of tables. A table represents a relation. In this example:

**PATIENTS**

| ID | NAME |
|---|---|
| 1 | Kowalski |
| 2 | Kozłowski |
| 3 | Piasecka |

## 4. Temporal Database

**Definition 3.**

An *instance of temporal database* (or just a *temporal database*) for schema $S = (R, A, D, \text{attr}, \text{dom})$ over the flow of time $(T, <)$ is a set $TDB = \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_k\}$ where $\boldsymbol{R}_i$ is a temporal relation instance (or just a temporal relation) over the relation name $R_i \in R$, i.e.,

$$\boldsymbol{R}_i(\text{dom}(A_1) \times \ldots \times \text{dom}(A_l)) \times 2^T,$$

where $\text{attr}(R_i) = (A_1, \ldots, A_l)$.

**Example 2.**

Let $S$ be the same schema as in Example 1. We take the flow of time to be that of days $T = \{\ldots, 2007\text{–}03\text{–}01, 2007\text{–}03\text{–}02, 2007\text{–}03\text{–}03, \ldots\}$.

$TDB = \{\textbf{PATIENTS}\}$
$\textbf{PATIENTS} =$

$\{((1, \text{Kowalski}), [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25]$[3] $\cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]),$
$((2, \text{Kozłowski}), [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01]),$
$((5, \text{Piasecka}), [2007\text{–}02\text{–}20, 2007\text{–}03\text{–}05] \cup [2007\text{–}04\text{–}01, 2007\text{–}04\text{–}16])\}$

[2] N denotes a set of natural numbers, CHAR a set of character sequences.
[3] $[a, b] = \{x : x \in T, a \leq x \leq b\}$.

The set of time points (stamps) associated to a tuple describes when data represented by the tuple are true in modelled reality, i.e., in this example, when a person is (or was or is going to be) a patient at the hospital. The temporal database can also be represented as a set of tables:

**PATIENTS**

| ID | NAME | |
|---|---|---|
| 1 | Kowalski | [2007–02–01, 2007–02–25] ∪ [2007–03–15, 2007–03–16] |
| 2 | Kozłowski | [2007–02–25, 2007–03–01] |
| 3 | Piasecka | [2007–02–20, 2007–03–05] ∪ [2007–04–01, 2007–04–16] |

## 5. The Query Language for Temporal Database

Let $S = (R, A, D, \mathrm{attr}, \mathrm{dom})$ be a relational database scheme, $(T, <)$ be the flow of time and $TDB = \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_k\}$ be a temporal database for $S$ over $(T, <)$. The *query language* $(QL)$ for $TDB$ is based on the language of first-order temporal logic [Gab, ChoTom]. It has the following categories of basic symbols:

- Domain variables: $x_1, x_2, \ldots$;
- Domain constants: $c_1, c_2, \ldots$;
- time variables: $t_1, t_2, \ldots$;
- time constants: $e_1, e_2, \ldots$;
- elements of $R$ as predicate symbols: $R_1, R_2, \ldots, R_k$ and a predicate symbol $\mathsf{time}$;
- equality symbol: $=$;
- logical connectives: $\neg, \wedge$;
- existential quantifier: $\exists$;
- temporal connectives: $\mathsf{U}, \mathsf{S}$;
- punctuation symbols: (, ).

**Syntax**

A term is either a constant or a variable. The atomic formulas of the language are of the form:

- $a_i = a_j$, where $a_i$ and $a_j$ are terms of the same sort, i.e., either domain terms or time terms,
- $R_i(a_1, a_2, \ldots, a_n)$, where $n$ is the length of the sequence $\mathrm{attr}(R_i)$ and $a_j$ is a domain variable (constant) that ranges over (is element of) the domain $\mathrm{dom}([\mathrm{attr}(R_i)]_j)$,
- $\mathsf{time}(a)$, where $a$ is a time term.

Formulas of $QL$ are finite strings of basic symbols defined in the following recursive manner:
(1) Any atomic formula is a formula,
(2) if $\varphi, \psi$ are formulas, so also are $\neg\varphi$, $\varphi \wedge \psi$, $\exists a\, \varphi$, $\mathsf{U}(\varphi, \psi)$, $\mathsf{S}(\varphi, \psi)$, where $a$ is any variable $x_i$.

**Semantics**

We define interpretation $\Theta$ as follows: $\Theta(R_i) = \boldsymbol{R}_i$, $\Theta(c_i) \in \bigcup D$, $\Theta(e_i) = T$, for every $i$. An assignment $v$ is a mapping that associates every domain variable $x_i$ with a domain value $v(x_i) \in \bigcup D$ and every time variable $t_i$ with a time point $v(t_i) \in T$. It is convenient to extend an assignment over constants by making $v(c_i) = \Theta(c_i)$ and $v(e_i) = \Theta(e_i)$, for every $i$. We define a formula $\varphi$ to be true in $TDB$ at time $t$ under assignment $v$ (denoted by $TBD, v, t \models \varphi$) by induction on the structure of the formula:

**Definition 3a.**
(1) $TBD, v, t \models R_i(a_1, \ldots, a_s)$ iff $((v(a_1), \ldots, v(a_s)), \tau) \in \Theta(R_i)$, where $\tau \subseteq T$ and $t \in \tau$,
(2) $TBD, v, t \models a_i = a_j$ iff $v(a_i) = v(a_j)$,
(3) $TBD, v, t \models \mathsf{time}(a_i)$ iff $v(a_i) = t$,
(4) $TBD, v, t \models \neg\varphi$ iff not $TBD, v, t \models \varphi$,
(5) $TBD, v, t \models \varphi \wedge \psi$ iff $TBD, v, t \models \varphi$ and $TBD, v, t \models \psi$,
(6) $TBD, v, t \models \exists x_i\, \varphi$ iff $TBD, v^*, t \models \varphi$,
where $v^*$ is an assignment which agrees with the assignment $v$ on the values of all variables except, possibly, on the values of $x_i$,
(7) $TBD, v, t \models \mathsf{U}(\varphi, \psi)$ iff
there exists a $t_1 \in T$ with $t < t_1$ and $TBD, v, t_1 \models \varphi$
and for every $t_2 \in T$ such that $t < t_2 < t_1$ holds $TBD, v, t_2 \models \psi$,
(8) $TBD, v, t \models \mathsf{S}(\varphi, \psi)$ iff
there exists a $t_1 \in T$ with $t_1 < t$ and $TBD, v, t_1 \models \varphi$
and for every $t_2 \in T$ such that $t_1 < t_2 < t$ holds $TBD, v, t_2 \models \psi$.

For convenience, we will introduce additional symbols: $\vee, \rightarrow, \leftrightarrow$ (other logical connectives), $\forall$ (universal quantifier) and $\mathsf{F}, \mathsf{P}, \mathsf{G}, \mathsf{H}, \mathsf{X}, \mathsf{Y}$ (other temporal connectives, (see Fig. 1)) defined as:

**Definition 3b.**
(1) $\varphi \vee \psi \quad \equiv_{\text{def}} \quad \neg(\neg\varphi \wedge \neg\psi)$
(2) $\varphi \rightarrow \psi \quad \equiv_{\text{def}} \quad \neg\varphi \vee \psi$
(3) $\varphi \leftrightarrow \psi \quad \equiv_{\text{def}} \quad (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$
(4) $\forall x\, \varphi \quad \equiv_{\text{def}} \quad \neg\exists x\, \neg\varphi$
(5) $\mathsf{F}\varphi \quad \equiv_{\text{def}} \quad U(\varphi, \top)$

(6) $G\varphi \equiv_{\mathrm{def}} \neg F\neg\varphi$
(7) $P\varphi \equiv_{\mathrm{def}} S(\varphi, \top)$
(8) $H\varphi \equiv_{\mathrm{def}} \neg P\neg\varphi$
(9) $X\varphi \equiv_{\mathrm{def}} U(\varphi, \bot)$ (if the flow of time is discrete)
(10) $Y\varphi \equiv_{\mathrm{def}} S(\varphi, \bot)$ (if the flow of time is discrete)

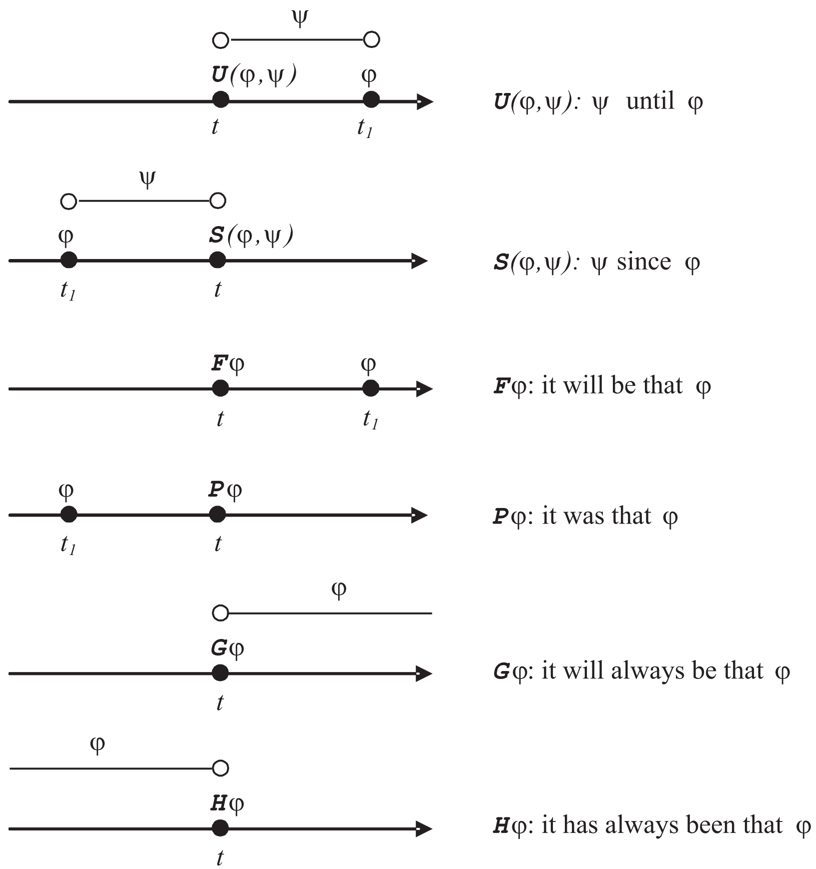


**Fig. 1. Graphical Representation of Temporal Connectives**

**Definition 4.**
A *temporal database query* is a formula of $QL$ with at least one free variable. The *answer of the query* $\varphi$ (denoted by $\varphi(TDB)$) is the temporal relation it generates in the database:

$$\varphi(TDB) = \{((v(x_1), \ldots, v(x_s)), \tau) : TBD, v, t \models \varphi \text{ and } t \in \tau\},$$

where $x_1, \ldots, x_s$ are all free variables of the formulae $\varphi$.

## 6. Queries

We will formulate four queries over the temporal database presented in Example 2 (we will assume that today is 2007–04–03).

**Query 1.**

*Find those who were (but are no longer) patients at the hospital*

$$\varphi = (\mathsf{P}\,\text{PATIENTS}(x_1, x_2)) \wedge \neg\text{PATIENTS}(x_1, x_2) \wedge \mathit{time}(2007\text{–}04\text{–}03)$$

According to definition 4 the answer of the query is the set:

$$\begin{aligned}\varphi(TDB) = \{((v(x_1), v(x_2)), \tau) : TBD, v, t \models (\mathsf{P}\,\text{PATIENTS}(x_1, x_2))\\ \wedge \neg\text{PATIENTS}(x_1, x_2)\\ \wedge\, \mathit{time}(2007\text{–}04\text{–}03) \text{ and } t \in \tau\}.\end{aligned}$$

From definition 3a and 3b, we have:

$TBD, v, t \models (\mathsf{P}\,\text{PATIENTS}(x_1, x_2)) \wedge \neg\text{PATIENTS}(x_1, x_2) \wedge \mathit{time}(2007\text{–}04\text{–}03)$

$\Updownarrow$ (def. 3a, p. 5)

(a) $TBD, v, t \models \mathsf{P}\,\text{PATIENTS}(x_1, x_2)$
(b) and $TBD, v, t \models \neg\text{PATIENTS}(x_1, x_2)$
(c) and $TBD, v, t \models \mathit{time}(2007\text{–}04\text{–}03)$

(a) $TBD, v, t \models \mathsf{P}\,\text{PATIENTS}(x_1, x_2)$

$\Updownarrow$ (def. 3b, p. 7)

$TBD, v, t \models \mathsf{S}(\text{PATIENTS}(x_1, x_2), \top)$

$\Updownarrow$ (def. 3a, p. 8)

there exists $t_1 \in T$ with $t_1 < t$
and $TBD, v, t_1 \models \text{PATIENTS}(x_1, x_2)$
and for every $t_2 \in T$ such that $t_1 < t_2 < t$ holds $TBD, v, t_2 \models \top$

$\Updownarrow$

there exists $t_1 \in T$ with $t_1 < t$ and $TBD, v, t_1 \models \text{PATIENTS}(x_1, x_2)$

$\Updownarrow$ (def. 3a, p. 1)

there exists $t_1 \in T$ with $t_1 < t$ and $(v(x_1), v(x_2), \tau) \in$ **PATIENTS**, where $\tau \subseteq T$ and $t_1 \in \tau$

(b) $TBD, v, t \models \neg\text{PATIENTS}(x_1, x_2)$

$\Updownarrow$ (def. 3a, p. 4)

$$\text{not } TBD, v, t \models \text{PATIENTS}(x_1, x_2)$$

$$\Updownarrow \text{ (def. 3a, p. 1)}$$

$$(v(x_1), v(x_2), \tau) \notin \textbf{PATIENTS}, \text{ where } \tau \subseteq T \text{ and } t \in \tau$$

(c) $TBD, v, t \models \mathit{time}(2007\text{–}04\text{–}03)$

$$\Updownarrow \text{ (def. 3a, p. 3)}$$

$$t = 2007\text{–}04\text{–}03$$

(a), (b) and (c) are satisfied by:

$t_1 = 2007\text{–}03\text{–}01$,
$\tau = [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01]$,
$v(x_1) = 2$, $v(x_2) = \text{Kozłowski}$

and

$t_1 = 2007\text{–}03\text{–}16$,
$\tau = [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]$,
$v(x_1) = 1$, $v(x_2) = \text{Kowalski}$, therefore,

$$\begin{aligned}\varphi(TDB) = \{&((1, \text{Kowalski}), [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup \\ &\qquad [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]), \\ &((2, \text{Kozłowski}), [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01])\}\end{aligned}$$

**Query 2.**

*Find those who stayed at the hospital more than once*

$$\begin{aligned}\varphi = \mathsf{P}\,(&\text{PATIENTS}(x_1, x_2) \wedge \\ &\mathsf{P}\,(\neg\text{PATIENTS}(x_1, x_2) \wedge \mathsf{P}\,\text{PATIENTS}(x_1, x_2))) \wedge \mathit{time}(2007\text{–}04\text{–}03)\end{aligned}$$

$$\begin{aligned}\varphi(TDB) = \{&((1, \text{Kowalski}), \\ &[2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16])\}\end{aligned}$$

(It can be shown in an analogous way to the previous query).

**Query 3.**

*When did Kowalski (id $= 1$) stay at the hospital?* (in other words: show the past history of the tuple $(1, \text{Kowalski})$)

$$\varphi = \text{PATIENTS}(1, x)$$

$$\begin{aligned}\varphi(TBD) = \{&((1, \text{Kowalski}), \\ &[2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16])\}\end{aligned}$$

**Query 4.**

*Find those who were admitted to hospital between 2007–01–01 and 2007–04–03*

$$\varphi = \mathsf{time}(2006\text{–}12\text{–}31) \wedge \neg \mathrm{PATIENTS}(x_1, x_2) \wedge \\ \mathsf{F}(\mathrm{PATIENTS}(x_1, x_2) \wedge \mathsf{F}(\mathsf{time}(2007\text{–}04\text{–}04)))$$

$$\varphi(TBD) = \{((1, \mathrm{Kowalski}), \\ [2007\text{–}02\text{–}01, 2007\text{–}02\text{–}25] \cup [2007\text{–}03\text{–}15, 2007\text{–}03\text{–}16]), \\ ((2, \mathrm{Kozłowski}), [2007\text{–}02\text{–}25, 2007\text{–}03\text{–}01]), \\ ((5, \mathrm{Piasecka}), \\ [2007\text{–}02\text{–}20, 2007\text{–}03\text{–}05] \cup [2007\text{–}04\text{–}01, 2007\text{–}04\text{–}16])\}$$

## References

[ChoTom] Chomicki J, Toman D.: *Temporal Logic in Information Systems.* In *Logics for Databases and Information Systems*, Kluwer Academic Publishers, 1998, pp. 31–70.

[Cod] Codd E. F.: *A Relational Model of Data for Large Shared Data Banks.* Communications of the ACM, 1970, 13(6), pages 377–387.

[Dat] Date C. J.: *Introduction to Database System*, Addison-Wesley, 2003.

[Etz] Etzion, Jajodia, Sripada. *Temporal Databases: Research and Practice.* Springer 1998.

[Gab] Gabbay D. M., Reynolds M. A., Finger M.: *Temporal Logic, Mathematical Foundations and Computational Aspects*, *Vol 2.* Clarendon Press. Oxford 2000.

[Jen] Jensen, C. S.: *Temporal Database Management*, PhD Thesis, Department of Computer Science, Aalborg University, 2000 [http://www.cs.aau.dk/∼csj/Thesis/].

[JenSno] Jensen C. S., Snodgrass R. T.: *Semantics of Time-Varying Information*, Information Systems, 21(4), 1996, pp. 311–352.

[Sno] Snodgrass R., editor. *The TSQL2 Temporal Query Language.* Kluwer Academic Publishers, 1995.

[Ste] Steiner A.: *A Generalisation Approach to Temporal Data Models and Their Implementations*, PhD Thesis, Departement Informatik, ETH Zurich, Switzerland, 1997.

[Tan] Tansel A. U., Clifford J., Gadia S., Jajodia S., Segev A., Snodgrass R.: *Temporal Databases: Theory, Design, and Implementation*, Benjamin-Cummings Publishing Co., Inc., 1993.