STUDIES IN LOGIC, GRAMMAR AND RHETORIC 17 (30) 2009

Mariusz Giero
Institute of Computer Science, University of Bialystok

Robert Milewski
Institute of Computer Science, University of Bialystok

STORING AND RETRIEVING INFORMATION ON THE TREATMENT OF INFERTILITY WITH THE USE OF THE BITEMPORAL DATABASE AND TEMPORAL LOGIC

Abstract: In the paper, we present basic information on bitemporal databases and temporal logic. We show how the language of temporal logic can be used as a query language of a database. We use data of the treatment of infertility to present practical aspects of our research.

1. Introduction

The bitemporal database is a database which apart from "proper" data stores valid time and transaction time of data [7, 9, 11, 13]. Valid time is the time when an event represented by data takes place. Transaction time is the time when data is held in the database (the period from the insertion of data to possible deletion of data). Storing both kinds of time information enables more accurate description of events. We can retrieve not only when an event occured but also when the information about the event is inserted (deleted) into the database. Valid time enables to query the history of events and transaction time in the history of the database states.

The bitemporal database needs, among others, a redefinition of standard operations: modification and deletion. Modification does not cause overwriting of a record. It sets, with an additional attribute, that a given record is no longer in the database and inserts a new record with updated values. Deletion does not remove a record physically but logically. It sets, with an additional attribute, that a given record is no longer in the database [10].

The paper presents the application of the bitemporal database in the management of information connected with the course of treatment using

ISBN 978–83–7431–231–8 ISSN 0860-150X

the IVF method ICSI/ET (see next chapter). In this particular case, valid time is the time when medicine is given to a patient. As far as the transaction time is concerned, there is little modification. It is the time when the decision of giving medicine is taken (and holds). The moment of taking the decision is determined not through consultation with doctors but by the insertion of the decision into database. We assume that there are no delays. It is very important that the transaction (decision) time is recorded automatically by the database system (the moment when data is inserted). The user can not establish that time personally. The user can not insert the decision with a backdate. Such functionality is aimed at ensuring the integrity of data and preventing errors and fraud. Chapter 3 provides basic information on how the bitemporal database functions.

2. The treatment of infertility using the IVF ICSI/ET procedure

The problem of infertility concerns about 10–15% couples at reproductive age [1]. The number of couples requiring treatment due to problems with getting pregnant has been constantly increasing during the last decades. Social and economic conditions cause postponing maternity by women until subsequent years of life. A constant trend in the rise of the age of women giving birth to their first child has been observed [2]. The postponing of reproductive decisions is going hand in hand with the decrease of natural reproductive potential of women [3, 4]. It also causes delayed diagnosis of possible problems with reproduction, and next, it leads to the decrease of the effectiveness of infertility treatment [5]. The most effective method of infertility treatment is the application of the in vitro fertilization (IVF) procedure with intracytoplasmic sperm injection (ICSI) and embryo transfer (ET). The average effectiveness of the infertility treatment with IVF methods oscillates at about 30% value, and in the group of women over 40 is significantly lower, and oscillates between 8% and 18% [6].

The process of infertility treatment using the IVF ICSI/ET procedure proceeds according to the following stages:

- gathering of information concerning personal data of the woman and the man, who reported to the clinic,
- gathering information about previous treatment,
- general examination:
  - medical interview of the woman,
  - medical interview of the man,

    - laboratory tests,
    - ultrasonography imaging,
- treatment during subsequent days and observation of the developing ovarian follicles,
- embryology
    - ovarian follicles aspiration,
    - sperm preparation,
    - ART procedure,
- embryo transfer
    - evaluation of pronuclei,
    - evaluation of embryos,
    - evaluation of blastocysts,
    - transfer of embryos,
- registration of treatment results
    - pregnancy,
    - childbirth,
    - information about children.

In the paper we have only concentrated on one of the stages presented above, i.e. the treatment and ovarian hyperstimulation. At this stage, serial registration of the considerable quantity of information about the stimulation process is required. Initially, the protocol of stimulation is established, and next, all information concerning medicine to be served in the following days is noted. Also the response of the woman's organism is recorded. Various information is collected, e.g., the level of estradiol, the thickness of endometrium, the quantity and the size of developing ovarian follicles in both ovaries.

3. Basic operations of the bitemporal database

Let us examine a database consisting of one relation *Treatment* with 6 six attributes: *id*, *sort*, *name*, *dose*, VT, TT where the course of treatment is recorded. Values of the listed attributes are as follows: the identifier of the patient, the group medicine belongs to, the dose of medicine, the time when medicine is given (valid time), the time when the decision of giving medicine is undertaken (transaction time). The values of VT, TT are intervals. The starting point of the interval is always a specific date and the end point of the interval is either a specific date or infinity ∞. The attributes *id*, *sort*, *name*, *dose* are defined by the user while creating table (relation). Attributes VT, TT are added to each created relation by the database system.

Insertion of data

On the 10-th of March, the user inserts the following information: GonalF (FSH group) with a dose of 150 per day is to be given to patient 1 starting on the 11th of March (this decision is taken on the 10th). The following record is inserted to the database:

Id	*Sort*	*Name*	*Dose*	*VT*	*TT*
1	FSH	GonalF	150	$[11.03 - \infty)$	$[10.03 - \infty)$

If the user does not give the end date of VT the database system sets the value to ∞. This means that medicine is to be given until the decision is changed. The start date of VT can not be earlier than the current date. The user is not allowed to perform such action. The value of TT is set by the database system. The start date is equal to the date of the insertion (i.e. current date). The end date is equal to ∞. This means that the taken decision is in force. When the user inserts a record the value of the end date of TT is always set to ∞.

Modification of data

There is a difference between modification of attribute VT and other attributes (apart from TT attribute which is not allowed to be set by the user).

Modification of VT attribute

On 15th the user modifies the end date of VT in the inserted record. He sets the value of the end date of TT to 20.03 (in other words, an event is recorded that on the 15th a new decision has been taken: giving medicine is to be finished on the 20th). This operation results in the modification of the record inserted previously (this record is to inform that the decision is no longer in force) and in inserting a new record with information on the new decision:

Id	*Sort*	*Name*	*Dose*	*VT*	*TT*
1	FSH	GonalF	150	$[11.03 - \infty)$	[10.03 – **15.03**)
1	**FSH**	**GonalF**	**150**	**[15.03 – 21.03)**	**[15.03** – ∞**)**

New values are in bold. The change of the decision is recorded by setting the end date of TT to a specific date. The date is set by the system and is equal to the current date (the date of the execution of modification). Next, the system inserts a new record. The values of non temporal attributes (i.e. *id*, *sort*, *name*, *dose*) are the same as in the previous record.

The value of *VT* is the time that remains for giving the medicine and is computed by the system. As intervals representing *VT* and *TT* are right-open, the end date of *VT* is 21.03 which means that 20.03 is the last day when the medicine is given. Analogically, the value of *VT* in the first record is 15.03. This means that 14.03 is the last day when the decision is in force.

The value of *TT* is set as in insertion operation. The start date is the current date, the end date is ∞.

Modification of non temporal attributes

On the 17th, the user changes the dosage: from 19.03, the patient is to be given 75 of GonalF (in other words, on the 17th the decision of the change of the dosage is taken). Modification is conducted on the second record in the above database (the modification does not refer to the first record, besides, the first record is no longer allowed to be modified).

The modification results in the change of the end date of *TT* of the record (as in previous modification) and inserting 2 new records: the first one informs about the dosage between the 17th and the 19th (the period between the moment when the last decision is no longer in force and the moment a new decision i.e. new dosage starts to be in force), the second one informs about the new dosage:

Id	*Sort*	*Name*	*Dose*	*VT*	*TT*
1	FSH	GonalF	150	[11.03 – ∞)	[10.03 – 15.03)
1	FSH	GonalF	150	[15.03 – 21.03)	[15.03 – **17.03**)
1	**FSH**	**GonalF**	**150**	**[17.03 – 19.03)**	**[17.03 – ∞)**
1	**FSH**	**GonalF**	**75**	**[19.03 – 21.03)**	**[17.03 – ∞)**

New values are supplied in bold. The change of the last decision (the second record) is recorded in an analogous way as in the previous modification. The third record informs on how many days the medicine is to be given with the old dosage. The attribute values of the record are computed by the system (the record represents a new decision, therefore the end date of *TT* is ∞). The fourth record informs on when the new dosage starts (the record also represents the new decision and the end date of *TT* is ∞). The end date of VT remains not changed (the user wanted only to change the start date of VT).

Deletion of data

A record is not physically removed from database but logically. It results in changing the end date of TT from ∞ to a specific date (date of executing of deletion).

In order to retrieve the course of treatment one needs to take intersection of VT and TT. With the above database we have:

Id	*Sort*	*Name*	*Dose*	$VT \cap TT$
1	FSH	GonalF	150	[11.03 – 15.03)
1	FSH	GonalF	150	[15.03 – 17.03)
1	FSH	GonalF	150	[17.03 – 19.03)
1	FSH	GonalF	75	[19.03 – 21.03)

The course of treatment is as follows: GonalF was given to patient 1 with dosage 150 from the 11th to the 18th (included) (the three records), then from the 19th to the 20th (included) with dosage 75 (the fourth record). The treatment was finished on the 20th.

4. Querying data

The alphabet and syntax of the query language

Queries can be formed with the language of first-order temporal logic [8, 9, 12]. The language, compared to classical logic, is extended with temporal operators which enable to form queries in a convenient and intuitive way. The alphabet 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: $e_1, e_2, \ldots$;
- predicate symbols: $P_1, P_2, \ldots, P_k$;
- equality symbol: $=$;
- classical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$;
- quantifiers: $\exists, \forall$;
- temporal connectives: $\mathtt{F}, \mathtt{P}, \mathtt{G}, \mathtt{H}, \mathtt{X}, \mathtt{X}^{-}\mathtt{U}, \mathtt{S}$;

- punctuation symbols:), (.
- additional predicate symbol: vt;

A *term* is either a domain term or time term. A *domain (time) term* is either a domain (time) constant or a domain (time) variable. The atomic formulas of the language are of the form:

- $P_i(a_1, a_2, \ldots, a_n)$, where P_i is n-ary predicate symbol, $a_1, a_2, \ldots, a_n$ are domain terms
- $t = u$, where t, u are both either domain terms or time terms
- $vt(t)$, where t is time term

Formulas are finite strings of basic symbols defined in the following recursive manner:

(1) any atomic formula is a formula

(2) if φ, ψ are formulas, so also are $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, $\varphi \leftrightarrow \psi$, $\exists x\varphi$, $\forall x\varphi$, $\mathtt{F}\varphi$, $\mathtt{P}\varphi$, $\mathtt{G}\varphi$, $\mathtt{H}\varphi$, $\mathtt{X}\varphi$, $X^{-}\varphi$, $\varphi\,\mathtt{U}\,\psi$, $\varphi\,\mathtt{S}\,\psi$ where x is any domain variable.

(3) there are not any other formulas than finite strings satisfying conditions (1) and (2)

Semantics of the query language

The model of the presented language is a bitemporal database defined as a structure $DB = (T, <, D, R_1, R_2, \ldots, R_n)$, where T is a set of dates, $<$ a relation ordering dates, D is a non empty set (database domain) and $R_1, R_2, \ldots, R_n$ are relations on $D \cup T$ such that

$$R_i \subseteq \{(d_1, \ldots, d_m, \tau_v, \tau_t) : d_1, \ldots, d_m \in D; \tau_v, \tau_t \text{ are integrals of the form } [t, u) \text{ or } [t, \infty), t, u \in T\}.$$

The valuation v associates every domain term to an element of D, every time term to an element of T. If c is a constant term, we make $v(c) = c$.

The interpretation I associates every predicate symbol P_i to a relation R_i of the database. We define formula A to be true in DB at time point $t \in T$ under valuation θ, writing $DB, \theta, t \models \varphi$, by induction on the structure of formula:

$DB, v, t \models P(a_1, \ldots, a_m)$ **iff** there exist τ_v, τ_t such that $(v(a_1), \ldots, v(a_m), \tau_v, \tau_t) \in I(P)$ and $t \in \tau_v \cap \tau_t$,

$DB, v, t \models x_i = x_j$ **iff** $\theta(x_i) = \theta(x_j)$

$DB, v, t \models \neg\varphi$ **iff** it is not the case that $DB, v, t \models \varphi$

$DB, v, t \models \varphi \wedge \psi$ **iff** $DB, v, t \models \varphi$ and $DB, v, t \models \psi$

$DB, v, t \models \varphi \vee \psi$ **iff** $DB, v, t \models \varphi$ or $DB, v, t \models \psi$

$DB, v, t \models \varphi \rightarrow \psi$ **iff** $DB, v, t \models \neg\varphi$ or $DB, v, t \models \psi$

$DB, v, t \models \varphi \leftrightarrow \psi$	**iff**	$DB, v, t \models \varphi \rightarrow \psi$ and $DB, v, t \models \psi \rightarrow \varphi$
$DB, v, t \models \exists x_i \varphi$	**iff**	there exists an $a \in D$ such that $DB, \theta^*, t \models \varphi$,
$DB, v, t \models \forall x_i \varphi$	**iff**	for every $a \in D$ holds $DB, \theta^*, t \models \varphi$, where θ^* is a valuation which agrees with the valuation v on the values of all variables except, possibly, on the values of x_i
$DB, v, t \models \mathtt{P}\varphi$	**iff**	there exists $t_1 \in T$ with $t_1 < t$ and $DB, v, t_1 \models \varphi$
$DB, v, t \models \mathtt{F}\varphi$	**iff**	there exists $t_1 \in T$ with $t_1 > t$ and $DB, v, t_1 \models \varphi$
$DB, v, t \models \mathtt{H}\varphi$	**iff**	for every $t_1 \in T$ whenever $t_1 < t$ then $DB, v, t_1 \models \varphi$
$DB, v, t \models \mathtt{G}\varphi$	**iff**	for every $t_1 \in T$ whenever $t_1 > t$ then $DB, v, t_1 \models \varphi$
$DB, v, t \models \mathtt{X}\varphi$	**iff**	$DB, v, t+1 \models \varphi$
$DB, v, t \models \mathtt{X}^-\varphi$	**iff**	$DB, v, t-1 \models \varphi$
$DB, v, t \models \varphi\, \mathtt{S}\, \psi$	**iff**	there exists $t_1 \in T$ with $t_1 < t$ and $DB, v, t_1 \models \psi$ and for every $t_2 \in T$ whenever $t_1 < t_2 < t$ then $DB, v, t_2 \models \varphi$
$DB, v, t \models \varphi\, \mathtt{U}\, \psi$	**iff**	there exists $t_1 \in T$ with $t_1 > t$ and $DB, v, t_1 \models \psi$ and for every $t_2 \in T$ whenever $t_1 > t_2 > t$ then $DB, v, t_2 \models \varphi$
$DB, v, t \models vt(t)$	**iff**	$vt(t) = t$

The temporal operators $\mathtt{F}, \mathtt{P}, \mathtt{G}, \mathtt{H}, \mathtt{X}, \mathtt{X}^-, \mathtt{U}, \mathtt{S}$ define the following expressions: "it will be that", "it was that", "it will always be that", "it has always been that", "next, it will be that", "previously, it was that", "... holds until ... holds", "... holds since ... holds", respectively. The predicate symbol vt is to set what time one poses the query at.

The query is every formula of the presented language with at least one free variable. The result of the query is defined as follows:

$$Q(DB, Q) = \{(v(x_1), \ldots, v(x_n)) \colon DB, v, t \models \varphi(x_1, \ldots, x_n)\},$$

where DB is a database, φ is a query and $x_1, \ldots, x_n$ are the only free variables of φ.

Examples of queries

Let us consider a database consisting one relation *Treatment* with 6 six attributes: *id*, *sort*, *name*, *dose*, *VT*, *TT* where the course of treatment is recorded. Values of the listed attributes are as the following: the identifier of the patient, the group the medicine belongs to, the dose of medicine, the time when medicine is given (valid time), the time when the decision of giving medicine is held (transaction time).

id	*sort*	*name*	*dose*	*VT*	*TT*
1	aGnRH	Dipherelina	3,75	[24.02 – 25.02)	[20.02 – ∞)
1	FSH	Gonal F	150	[11.03 – ∞)	[10.03 – 15.03)
1	FSH	Gonal F	150	[15.03 – 21.03)	[15.03 – 17.03)
1	FSH	Gonal F	150	[17.03 – 19.03)	[17.03 – ∞)
1	FSH	Gonal F	75	[19.03 – ∞)	[17.03 – 22.03)
1	FSH	Gonal F	75	[22.03 – 23.03)	[22.03 – ∞)
1	HCG	Pregnyl	10000	[23.03 – 24.03)	[22.03 – ∞)
2	FSH	Gonal F	150	[16.03 – 22.03)	[15.03 – ∞)

One can form the following queries (the predicate symbol L is interpreted as the relation *Treatment*):

1. Who is given GonalF from the group of FSH with the dosage of 150 at the present time (we assume that today is the 11th of March)?

   $\varphi = L(x_1, \mathrm{FSH}, \mathrm{Gonal\ F}, 150) \wedge vt(11.03)$

   $Q(DB, \varphi) = \{1\}$

2. Who was treated with GonalF before the 18th of March. What was the dosage then?

   $\varphi = \mathtt{P}L(x_1, \mathrm{FSH}, \mathrm{Gonal\ F}, x_2) \wedge vt(18.03)$

   $Q(DB, Q) = \{(1, 150), (2, 150)\}$

3. Who took GonalF and after that Pregnyl? We refer the query to the 31st of March

   $$\varphi = \exists x_2, x_3\ \mathtt{P}(L(x_1, \mathrm{FSH}, \mathrm{Gonal\ F}, x_2) \wedge \mathtt{F}(L(x_1, \mathrm{HCG}, \mathrm{Pregnyl}, x_3) \wedge \mathtt{F}vt(31.03))) \wedge vt(31.03)$$

   $Q(DB, Q) = \{1\}$

4. Who was treated at least 5 days? We refer the query to the 23th of March.

   $$\varphi = \exists x_2, x_3, x_4 (L(x_1, x_2, x_3, x_4)\mathtt{S}(L(x_1, x_2, x_3, x_4) \wedge vt(17.03))) \wedge vt(23.03)$$

   $Q(DB, \varphi) = \{1\}$

5. Future work

Larger amount of data should be used to examine the efficiency of the bitemporal database. Complicated queries should be formed to examine the utility of the presented query language. The implementation of the bitemporal database is also taken into consideration.

6. References

[1] Adashi E.Y., Rock J.A., Rosenwaks Z. Human infertility: an introduction. Reproductive endocrinology, surgary and technology. Lippincott-Raven Publishers, Philadelphia 1996.

[2] te Velde E.R., Pearson P.L. The variability of female reproductive ageing. Hum. Reprod. 2002; 8: 141–154.

[3] Evers J.L. Female subfertility. Lancet 2002; 360: 151–159.

[4] Menken J. et al. Age and infertility. Science 1986; 233: 1389–1394.

[5] Broekmans F.J. et al. Antral follicle counts are related to age at natural fertility loss and age at menopause. Menopause 2004; 11: 607–614.

[6] Milewski R., Milewska A.J., Domitrz J., Wołczyński S. Zapłodnienie pozaustrojowe IVF ICSI/ET u kobiet po 40. roku życia. Przegląd Menopauzalny 2008; 2: 85–90.

[7] Etzion, Jajodia, Sripada: Temporal Databases: Research and practice. Springer, 1998.

[8] Gabbay D. M., Reynolds M. A., Finger M.: Temporal Logic, Mathematical Foundations and Computational Aspects. Vol1, Clarendon Press, Oxford, 1994.

[9] Gabbay D. M., Reynolds M. A., Finger M.: Temporal Logic, Mathematical Foundations and Computational Aspects. Vol2, Clarendon Press, Oxford, 2000.

[10] Giero M.: Application of bitemporal databases containing medical data. Studies in Logic, Grammar and Rhetoric, vol. 17(30), 2009, pp.

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

[12] Trzęsicki K.: Logika temporalna. Wybrane zagadnienia. Wydawnictwo Uniwersytetu w Białymstoku, Białystok, 2008.

[13] Tansel A. U. i in.: Temporal Databases: Theory, Design, and Implementation. Benjamin-Cummings Publishing Co., Inc., 1993.

Mariusz Giero
Chair of Logic, Informatics and Philisophy of Science
University of Białystok
ul. Sosonowa 64
15–887 Białystok, Poland
giero@uwb.edu.pl

Robert Milewski
Medical University of Bialystok
Department of Statistics and Medical Informatics
Mickiewicza 2c
15–089 Białystok
milewski@umwb.edu.pl