IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 March 2024, accepted 30 April 2024, date of publication 6 May 2024, date of current version 15 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3397724

THEORY

Preferences in Temporal Relational Databases

LUCA ANSELMA[1] **AND PAOLO TERENZIANI**[2]
[1]Dipartimento di Informatica, Università degli Studi di Torino, 10149 Turin, Italy
[2]DISIT, Università degli Studi del Piemonte Orientale Amedeo Avogadro, 15121 Alessandria, Italy

Corresponding author: Luca Anselma (luca.anselma@unito.it)

This work was partially supported by the LEArning Personalization with Artificial Intelligence and of Artificial Intelligence (LEAP-AI) Project through Fondazione Compagnia San Paolo and Fondazione Cassa Depositi e Prestiti under Grant D13C23001280007.

ABSTRACT Despite the huge amount of work devoted to the treatment of time within the relational context, some relevant phenomena remain to be fully addressed. We focus on one of them, i.e., temporal indeterminacy with preferences. In several domains (e.g., workflows, guidelines) and tasks (e.g., planning, scheduling), the exact time of occurrence of facts is not known: only an interval of possible values for their starting time, and a range of possible durations is available. Additionally, preferences can be assigned to the different temporal possibilities. We propose the first relational temporal database approach coping with such issues. We introduce a new data model to cope with indeterminate time with preferences, considering a family of preference functions, and we propose new definitions of relational algebraic operators to query the new data model. We also ascertain the properties of the new model and algebra, with emphasis on reducibility, and on the correctness of the algebraic operators.

INDEX TERMS Temporal relational databases, temporal relational algebra, temporal indeterminacy, preferences, data model.

I. INTRODUCTION

In this paper we propose a relational temporal database (TDB) approach to deal with ***temporal indeterminacy with preferences***. In this Section, we introduce the problem, the context, and the motivations of our work progressively. First (Subsection A) we briefly survey the relational TDB literature to motivate the adoption of ***dedicated techniques to deal with time***. Second (Subsection B), we introduce an additional source of complexity, the treatment of ***temporal indeterminacy*** (i.e., coping with the time of occurrence of facts when it is not exactly known). Third (Subsection C), we consider the importance of dealing also with ***temporal preferences*** in the case of temporal indeterminacy. Finally (Subsection D), we sketch the main contributions and the organization of the rest of the paper.

A. TEMPORAL DATABASES

Time is an intrinsic part of the human way of perceiving and modelling reality. Thus, also DataBases (DBs) have to cope with it. However, already in the 1990s, the scientific community has recognized that time has a special status, and *dedicated* techniques need to be devised to deal with it, e.g., in the context in the contexts of Artificial Intelligence (AI) [6], [28] and of relational DBs [23] (for instance, readers can find in Section I of [23] concrete examples showing that "standard" relational DB approaches hardly manage even simple queries concerning facts holding over time intervals). Since then, the scientific community has proposed hundreds of *dedicated* approaches to cope with time in *temporal relational databases* (TDBs in the following) [31]. A landmark in the field was the definition of TSQL2, a consensus approach of the TDB scientific community, coordinated by R.T. Snodgrass (one of the "fathers" of the TDB research area) [23]. In summary, in the TSQL2 book, the community proposed a dedicated approach to TDBs, designing a temporal data model and query language, grounded on an explicit semantics (BCDM) and on a temporal algebra. An important part of the approach is the proof that, in case time is not considered, the proposed temporal algebra *reduces* to Codd's algebra [8], which is commonly used by the DB community as a reference to denote the completeness of a

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

query language. Indeed, reducibility is an essential property to also grant interoperability with non-temporal DB approaches: if time is disregarded, TSQL2 behaves like them. Nowadays, most of the major DBMSs have temporal support (http://rts.cs.arizona.edu/sql3.html). However, since the treatment of time in DB introduces a nuance of different problems, some of which have not been fully addressed yet by the scientific community, the research in the area is still quite vibrant. For example, J. Gamper, M. Ceccarello, and A. Dignös survey the main research results in TDBs in the last five years [13], and the recent *Encyclopedia of Database Systems* contains about one hundred entries dedicated to time [16].

Despite such a huge amount of work, several problems have not been fully explored yet. One of them is the treatment (*data model* plus *algebra*) of *temporally indeterminate data*.

B. TEMPORAL INDETERMINACY IN TDBs

Indeed, most TDB approaches focus on the time of occurrence of facts (*valid time* [23]) and/or on the time of their insertion/deletion in the DB (*transaction time* [23]), assuming that both times are *exactly known*. Unfortunately, in many contexts (e.g., workflows, guidelines) and tasks (e.g., planning, scheduling), the exact *valid time* of facts is not available. Valid-time indeterminacy ("*don't know exactly when*" information [12]) comes into play whenever the valid time associated with a fact is not known in an exact way.

Let us consider Ex.1.

Ex.1 John was at home from 10:00-10:10 to 14:00-14:30. ■

It is important to notice that, indeed, Ex.1 is an implicit way of denoting (if we consider the granularity of minutes) the 11·31=341 precise scenarios/instantiations sketched in Ex.1' below:

Ex.1' John was at home from 10:00 to 14:00, or John was at home from 10:00 to 14:01, or …

John was at home from 10:10 to 14:30. ■

Despite the plethora of application contexts in which the valid time of facts is only known in an approximate way, temporal indeterminacy has not been widely explored by the TDB research. Among the most relevant approaches in this context, Dekhtyar et al. [10] introduce temporal probabilistic tuples (modeling instantaneous events) to cope with data such as "data tuple d is in relation r at some point of time in the interval $[t_i,t_j]$ with probability between p and p'", and Dyreson and Snodgrass [12] associate an interval of indeterminacy with the starting and ending points of tuples (modeling durative facts), and a probability distribution over it. In [4], the authors of this paper, in cooperation with R.T. Snodgrass, propose a reference model and algebra for an explicit treatment of valid-time indeterminacy (e.g., making explicit all the scenarios of Ex.1), and then specify, analyze and compare a family of sixteen more compact implicit representations. Additionally, Anselma et al. extend the reference model and the compact representation models to cope with probabilities.

In a recent paper [3], the authors of this paper have tried to generalize upon the existent literature, proposing a general methodology to deal with temporal indeterminacy (and also with other phenomena, such as repeated and periodic facts [26], [27], in which an explicit representation of time is quite space-expensive) in TDBs, that we are going to follow also in this paper. The starting point of Anselma et al.'s methodology is the consideration that all the existent TDB representational models and algebrae for temporal indeterminacy adopt an ***implicit compact*** representation of time. Indeed, this is a well-motivated choice, since an ***explicit*** representation would be highly inefficient, both from the space and from the computational points of view (as a trivial example, consider the impact of explicitly modelling and querying the 341 alternative scenarios sketched in Ex.1').

Thus, to deal with temporal indeterminacy in TDBs, theoretically grounded approaches should not only define

(1) a suitable (implicit) ***compact representation*** (data) ***model*** and

(2) a ***relational algebra*** to query it,

but also

(3) formally define the ***data semantics*** of the model (i.e., the correspondence between the compact representation and the intended explicit one), and

(4) prove the ***correctness*** of the algebra with respect to the data semantics (informally: the fact that, though -for the sake of efficiency- the algebra operates on the compact implicit representation, the results it obtains are equivalent to the results that would be obtained by operating on the explicit representation).

C. PREFERENCES

To the best of our knowledge, all TDB approaches in the literature (including the ones mentioned above) consider all the scenarios/instantiations denoted by a temporally indeterminate fact (see again the comment to Ex.1) as *equally desirable* while in many contexts/applications, there are ***preferences*** among such scenarios/instantiations (i.e., in Dubois et al.'s words [11], a *ranking of the instantiations that are acceptable*).

Preferences have attracted a lot of attention in AI, where a whole stream of research has extended Constraint Satisfaction Problems (CSP) methodologies to consider ***soft constraints***, to explicitly deal with ***preferences*** or priorities (consider, e.g., the milestone paper [11]). Rossi et al.'s book [21] discusses a large set of approaches to explicitly manage *preferences* in AI.

In particular, the AI research shows that managing preferences is very important in connection with temporal indeterminacy: "***temporal preferences*** *are quantitative preferences that pertain to the position and duration of events in time*" [21].

As a toy example, consider the following.

Ex. 2 John starts breakfast between 7:30 and 8:20 and eats it for 10-40 minutes with (at least) low preference, and starts breakfast between 8:00 and 8:20 and eats it for 20-30 minutes with high preference. ■

Temporal preferences play a key role for the ***personalization*** of approaches, and have an important role in many areas, including, e.g., economics [9] and medicine [1], [29]. Indeed, many domains (e.g., workflows, guidelines) and tasks, including planning and scheduling, may benefit from the treatment of temporal preferences. A prototypical example can be found in the domain of clinical treatments and clinical guidelines. Usually, in the clinical guideline context, it is not possible/useful to state precisely when actions have to be executed [25]. Temporal constraints in the guidelines can be typically interpreted as general recommendations, to be respected with different levels of medical preferences [1].

Consider the purely illustrative example below that we use as a running example.

Ex. 3 Mary must undergo a physical therapy and assume two drugs, hydrocodone for treating pain and diphenhydramine for treating an allergy.

(3a) Mary's physical therapy must start between March 14 and March 22 and last for 6-8 days with high preference, start between March 13 and March 23 and last for 5-13 days with a medium preference, and start between March 10 and March 30 and last 4-20 days with a low preference.

(3b) Mary can start hydrocodone between March 10 and March 15 and take it for 3-6 days with high preference, start between March 7 and March 15 and take it for 3-9 days with medium preference, start between March 1 and March 20 and take it for 1-20 days with low preference.

(3c) Mary can start diphenhydramine on March 25 and take it for 10 days with high preference, start between March 20 and March 27 and take it for 9-11 days with a medium preference, and start it between March 11 and March 30 and take it for 7-13 days with a low preference. ■

Additionally, patients' preferences can be considered by physicians to provide patients with a ***personalized*** temporal schedule of therapies that they are comfortable with (as in the landmark MobiGuide project [19], funded by the European Community), to increase conformance of non-hospitalized patients [29].

D. MAIN CONTRIBUTIONS AND ORGANIZATION OF THE PAPER

In this paper we propose the first TDB model and relational algebra coping with ***temporal indeterminacy with preferences***. We consider qualitative (i.e., non-numeric) preferences. For generality, we do not commit to a specific scale of preferences (e.g., <high, medium, low> in Ex. 3 above): any scale of qualitative or quantitative preferences (see Def. 4 below) is supported as long as a total order can be defined on its elements.

Our approach grounds on the general methodology proposed in [3]. We provide (1) a ***compact representation*** (data) ***model*** and (2) a ***relational algebra*** to query it, we define (3) the ***data semantics*** of the model and (4) prove the ***correctness*** of the algebra. Additionally, following TSQL2 proposal, we also (5) prove the ***reducibility*** of our approach to Codd's standard relational algebra, to grant the compatibility and interoperability with standard approaches.

In Section II we propose the new data model and its semantics. In Section III we introduce our temporal algebra and ascertain its properties. In Section IV we propose final discussions, comparisons, and future work.

II. DATA MODEL

In this Section, we propose our data model, to compactly represent indeterminate temporal data with preferences. For the sake of clarity, we introduce it gradually, in three steps. First (Subsection A) we focus on (our representation of) indeterminate time "per se". Then (Subsection B) we introduce scales of preferences and preference functions, as a way of pairing indeterminate time with preferences. Then, (Subsection C) we propose a relational representation for the association of indeterminate time with preferences to data, representing facts. Finally (Subsection D), we formally define the semantics of the overall data model.

A. INDETERMINATE TIME

As in TSQL2 [23], BCDM [15] and in many approaches reviewed in [23], in our approach time is discrete, linearly ordered, and isomorphic to a subset of the integers. For the sake of simplicity, a single granularity (e.g., day) is assumed.

Definition 1 (Chronon): The chronon is the basic time unit. The chronon domain T^C, also called timeline, is the ordered set of chronons $\{\ldots, c_i, \ldots, c_j, \ldots\}$, with $c_i < c_j$ as $i < j$. ■

Events, properties, and facts (henceforth we adopt ***facts*** as a cover term for all of them) are associated with the time when they occur (i.e., ***valid time*** [23]). In case it is precisely known, the valid time of facts can be specified as below:

Definition 2: The (***precisely known***) **valid time** of a fact can be equivalently specified by:

(Representation 2.a) a pair <s,e> such that s,e∈ T^C, s≤e, or (Representation 2.b) a pair <s,d> such that s∈ T^C, d∈ N. ■

While Representation 2.a explicitly locates the starting and ending point of facts, Representation 2.b focuses on their starting point and duration. When valid time is exactly known, Representations 2.a and 2.b are equivalent.

Unfortunately, however, in many domains and applications only an *approximation* of the valid time of facts can be available: e.g., only a range of possible values for the starting point ($[s_m, s_M)$ in Def. 3) and for the duration of the fact ($[d_m, d_M)$ in Def. 3) may be known. This is typical, e.g., in several medical applications (consider the milestone Asbru temporal specification language [22] and the survey in [25]). In the following, as in several TDB approaches [23], without loss of generality, we consider all intervals as closed on the left and open on the right.

Definition 3: The ***temporally indeterminate*** **valid time** of a fact can be specified by a pair of ranges $<[s_m, s_M), [d_m, d_M)>$ such that $s_m, s_M \in T^C$, $s_m \leq s_M$, $d_m, d_M \in N, d_m \leq d_M$. . ■

In Def. 3, $<[s_m,s_M),[d_m,d_M)>$ is an ***implicit representation*** to state that the fact may start in any chronon in $[s_m,s_M)$ and have any duration in $[d_m,d_M)$.

Notably, in Def. 3 we have chosen the Representation 2.b above (i.e., starting time + duration) to represent temporal indeterminacy. It is worth point out that, in the case of temporal indeterminacy, Representations 2.a and 2.b in Definition 2 are not equivalent.[1] In this paper we deal with the Representation 2.b, which has not been considered yet in the TDB context, though it is widely used in several application domains (see, e.g., [22], [25]).

B. INDETERMINATE TIME WITH PREFERENCES

In our approach different scales of preferences can be used.

Definition 4 (Scale of Qualitative Preferences (SQP)): An SQP (or "scale", for short) S_r of ***cardinality*** r is composed by an enumerative set $\{p_1, \ldots, p_r\}$ of r labels (r>0), and a strict and total ordering relation $<$ over the set. We denote an SQP by an ordered list $\langle p_1, \ldots, p_r\rangle$, such that $\forall i,\ 1 \le i < r, p_i < p_{i+1}$. ■

Terminology. Given an SQP S_r of ***cardinality*** r, we indicate by $S_r(i)$ the i[th] value in the scale S_r ($1 \le i \le r$) and we denote by $dom(S_r)$ the domain of S_r. ■

Ex. 4. An SQP applicable to Ex. 2 is S_3^{ex}: <low, medium, high>, $S_3^{ex}(2)$ = medium, and $dom(S_r)$ = {low, medium, high}. ■

Given a scale S_r, a (temporal) preference function associates preference values in $dom(S_r)$ with pairs <s,d>, such that $s \in T^C$, $d \in N$. In this paper, we focus on a specific class of temporal preference functions, ***layered preference functions*** [2]. Indeed, in many areas and applications, preferences distribute in a "regular" way, forming a sort of "***pyramid***" of *nested* ranges, in which the top range has the highest preference and the bottom the lowest one. Ex. 3 is just one example, but this is true in all cases in which preferences are "centered" on a given set of temporal values and decrease while getting far from this center.

Definition 5 (Layered Preference Function pref$_{\mathbf{r}}$*):* Given an indeterminate valid time $<[s_m,s_M),[d_m,d_M)>$ and a scale S_r of cardinality r, a layered preference function $pref_r$ is a function $pref_r$: $T^C \times N \rightarrow dom(S_r)$ that associates a preference value with each pair <s,d> such that $s \in T^C$, $s_m \le s < s_M$, $d \in N$, $d_m \le d < d_M$ in such a way that preferences follow the pattern:

$\exists s_m^1, s_M^1, \ldots, s_m^r, s_M^r \in T^C, d_m^1, d_M^1, \ldots, d_m^r, d_M^r \in N$, such that

$s_m = s_m^1 \le s_m^2 \le \ldots \le s_m^r < s_M^r \le s_M^{r-1} \le \ldots \le s_M^1 = s_M$, $d_m = d_m^1 \le d_m^2 \le \ldots \le d_m^r < d_M^r \le d_M^{r-1} \le \ldots \le d_M^1 = d_M$

$\forall i\ 1 \le i \le r\ \forall t_1 \in T^C, \forall d_1 \in N$ such that $s_m^i \le t_1 < s_M^i \wedge d_m^i \le d_1 < d_M^i$ then $pref_r(t_1,d_1) \ge S_r(i)$ ■

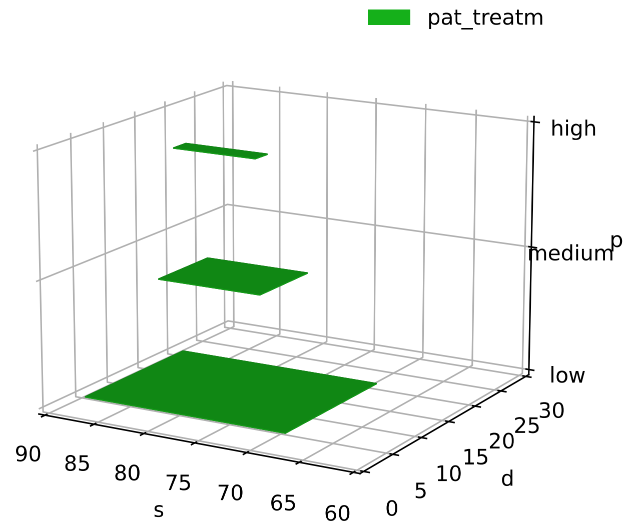

FIGURE 1. Pyramid preferences for Ex.3a.

For instance, pyramid preference functions allow to represent all the temporal preferences in Ex. 3 above. As a graphical example, Fig. 1 shows the tri-dimensional pyramid of preferences corresponding to Ex. 3.a. In the example, we assume that time starts at the beginning of the year, and to adopt the granularity of days. Thus, March 14 corresponds to day 73, and March 22 to day 81, so that the top rectangle, with coordinates (73,82,6,9), represents a starting time between day 73 and day 82 (*s* axis) and a duration between 6 and 9 days (*d* axis) when Mary's physical therapy can be performed with high preference (*p* axis). Notably, the rectangle with low preference (coordinates: (69,90,4,21)) includes the rectangle with medium preference (coordinates: (72,83,5,14)), which includes the one with high preference.

We also admit "degenerate" pyramid preferences, in which one or more of the "top layers" may be missing (but at least the bottom layer must exist). We adopt the convention that an empty layer is represented by the quadruple (0, 0, 0, 0).

C. A RELATIONAL REPRESENTATION OF DATA WITH TIME AND PREFERENCES

A ***temporal relation with layered preferences (TRwLP for short)*** associates facts with valid times with pyramid preferences.

Definition 6 (TRwLP): Given a scale S_r, which is unique for a given DB, the ***schema*** of a ***temporal relation with layered preferences*** $R = (A_1, \ldots, A_n | (T_{sm}^1, T_{sM}^1, T_{Dm}^1, T_{DM}^1), \ldots, (T_{sm}^r, T_{sM}^r, T_{Dm}^r, T_{DM}^r))$ consists of an arbitrary number of non-temporal attributes $A_1, \ldots, A_n$, encoding some fact, and of r quadruples $(T_{sm}^i, T_{sM}^i, T_{Dm}^i, T_{DM}^i)$ of temporal attributes

[1] In the case of temporal indeterminacy, the start+end representation (i.e., Representation 2.a in Definition 2) and the start+duration representation (i.e., Representation 2.b in Definition 2) are not equivalent. E.g., <[1,3),[1,3)> in the start+duration representation denotes the intervals which start on 1 or 2 and have a duration of 1 or 2, i.e. the set time intervals S={{1,2},{1,2,3},{2,3},{2,3,4}} (here we represent a time interval as a set of chronons).

No specification in the start+end representation can denote the set S. For example, <[1,3),[2,5)> in the start+end representation denotes the set S'={{1,2},{1,2,3},{1,2,3,4},{2,2},{2,3}, {2,3,4}} (where also the sets {1,2,3,4} –with duration 3– and {2,2} –with duration 0– are included). Vice versa, the set of intervals S'={{1,2},{1,2,3},{1,2,3,4},{2,2},{2,3},{2,3,4}} is represented by <[1,3),[2,5)> in the start+end formalism, but cannot be represented in the start+duration formalism.

(termed Temporal Layer(s) – TL for short) with domains T^C (T^1_{sm}, T^1_{sM},…, T^r_{sm}, T^r_{sM}) or N (T^1_{Dm}, T^1_{DM},…,T^r_{Dm}, T^r_{DM}). The non-temporal attributes and the temporal ones are separated by a symbol "|". ■

Thus, a tuple $x = (v_1,\ldots, v_n|$ (s^1_m,s^1_M,d^1_m,d^1_M), …, (s^r_m,s^r_M,d^r_m,d^r_M)) in a TRwLP relation $r^{TR}(R)$ on the schema R represents that the fact $< v_1,\ldots, v_n >$ starts between s^1_m and s^1_M and have duration between d^1_m and d^1_M with preference at least $S_r(1)$, …, starts between s^r_m and s^r_M and lasts between d^r_m and d^r_M with preference at least $S_r(r)$.

Ex. 5. Ex. 3 can be represented, using the scale in Ex. 4, by the relation *pat_treatm*^{TP}, to model treatments to be applied to patients (Ex.3a), with schema (*patient_t*, *treatment*| (T^1_{sm}, T^1_{sM}, T^1_{Dm}, T^1_{DM}), (T^2_{sm}, T^2_{sM}, T^2_{Dm}, T^2_{DM}), (T^3_{sm}, T^3_{sM}, T^3_{Dm}, T^3_{DM})), and the relation *drug_admin*^{TP}, to store drug administration (Ex. 3b, Ex. 3c), with schema (*patient_d*, *ad_drug*| (T^1_{sm}, T^1_{sM}, T^1_{Dm}, T^1_{DM}), (T^2_{sm}, T^2_{sM}, T^2_{Dm}, T^2_{DM}), (T^3_{sm}, T^3_{sM}, T^3_{Dm}, T^3_{DM})):

pat_treatm^{TP} ={(Mary, phys | (69,90,4,21), (72,83,5,14), (73,82,6,9))}

drug_admin^{TP} ={(Mary, diphenhydramine | (60,80,1,21), (66,75,3,10), (69,75,3,7)), (Mary, hydrocodone | (70,91,7,14), (79,87,9,12), (84,85,10,11))}. ■

D. DATA SEMANTICS

TRwLP proposes a compact and implicit representation of the association of valid times and preferences with facts, since four rectangle vertices are used to summarize all the bi-temporal points in each rectangle (see Fig. 1). *Following the methodology in [3]* (see Subsection I-B), we formally define the semantics for TRwLP through the definition of the functions *Ext* and *Make-Explicit*. The function ***Ext***, given a Temporal Layer (s^i_m,s^i_M,d^i_m,d^i_M) taken from the temporal part of a tuple in a TRwLP, and corresponding to the i-th level of the scale of preference, makes explicit all the pairs <start_point,duration> in the indeterminate valid time that have preference at least $S_r(i)$.

Definition 7 (Data Semantics for TRwLP): **Ext function**. Let V_r = <(s^1_m,s^1_M,d^1_m,d^1_M), …, (s^r_m,s^r_M,d^r_m,d^r_M)> be the temporal values associated with a tuple in a TRwLP with time scale S_r, and let $V_r[i]$ denote the i-th ($1 \le i \le r$) quadruple in it (i.e., <s^i_m,s^i_M,d^i_m,d^i_M >), the ith Temporal Layer.

Ext(<s^i_m,s^i_M,d^i_m,d^i_M >) ={<s,d> \ $s^i_m \le s < s^i_M \wedge d^i_m \le d < d^i_M$}. ■

For example, $V_r[1]$=<69,90,4,21> compactly represents the pairs <start,duration> having at least *low* preference for the tuple in *pat_treatm* (see Ex. 5 and also Figure 1 above). The Ext functions makes all such pairs explicit:

Ex. 6. Ext(<69,90,4,21>) ={<69,4>, <69,5>, …, <69,20>, …, <89,4>, <89,5>, …, <89,20>}. ■

The function *Make-Explicit*, applied to a TRwLP r^{TP}, provides as output a new relation r^{Expl} in which each fact $x = (v_1,\ldots, v_n)$ in r^{TP} is paired with the explicit set of all the triples <s,d,p> such that x, starting at s and lasting d, has at least preference p.

Definition 8 (Data Semantics for TRwLP): **Make-Explicit function**. Given a TRwLP r^{TP} over a schema $R = (A_1,\ldots, A_n|$ (T^1_{sm}, T^1_{sM}, T^1_{Dm}, T^1_{DM}), …, (T^r_{sm}, T^r_{sM}, T^r_{Dm}, T^r_{DM})) and a scale S_r, Make-Explicit provides as output a new relation r^{Expl} defined over the schema $R' = (A_1,\ldots, A_n|S)$, where S is a set of triples <s,d,p_i> s∈T^C, d∈N, p_i∈dom(S_r) defined as follows:

r^{Expl} = Make-Explicit(r^{TP}) = {($v_1,\ldots, v_n$|{<s,d,p_i>)\ ($v_1,\ldots, v_n$|V_r) ∈r^{TP} ∧ <s,d>∈Ext(V_r[i]), 1≤i≤r ∧ p_i =S_r(i)}) }.

For example, Make-Explicit(*pat_treatm*) makes explicit all the pieces of information denoted by (*pat_treatm*), i.e., it associates with each fact all the triples denoting its starting time, duration, and preference (for the sake of brevity, we use L, M, and H to denote low, medium, and high respectively).

Ex. 7. Make-Explicit (*pat_treatm*) = {(Mary, phys | {<69,4,L>, …, <89,20,L>, <72,5,M>, …, <82,13,M>, <73,6,H>, …, <81,8,H>})}. ■

III. TEMPORAL RELATIONAL ALGEBRA

Codd defined as complete any query language that is as expressive as his set of five relational algebraic operators: relational union (∪), relational difference (–), selection (σ_P), projection (π_A), and Cartesian product (×) [8]. We propose an extension of Codd's operators to query the data model in Section II. The TDB literature proposes several temporal extensions to Codd's operators [17], [23], [24]. To the best of our knowledge, however, only the algebrae in [4] and [10] define the $\cup^T$, $-^T$, and $\times^T$ algebraic temporal operators manging temporal indeterminacy, and no relational TDB approach manages temporal preferences. A large majority of TDB approaches follows the convention that extended operators behave like standard Codd's operators on the non-temporal attributes, and perform union (for $\cup^T$), difference (for $-^T$), and intersection (for $\times^T$) on the temporal attributes [16], [23], [24].

A. RELATIONAL ALGEBRA FOR PREFERENTIAL TIME

We base our approach on such a "consensus" background, extending Codd's operators to cope with time and preferences. In the rest of the paper, we use the superscript "TP" (temporal preferences) for our operators.

Definition 9 (Temporal Algebraic Operators for TRwLP): Given a DB defined over the scale S_r, let r^{TP} and s^{TP} denote TRwLP relations having the proper schema (in this definition *v*, *v1*, *v2* stand for the values of non-temporal attributes in a tuple, and V_r, $V1_r$, $V2_r$ for the values of the temporal attributes – having the general form ((s^1_m,s^1_M,d^1_m,d^1_M), …, (s^r_m,s^r_M,d^r_m,d^r_M)).

$$\sigma_P^{\mathbf{TP}}(\mathbf{r^{TP}}) = \{(v|V_r)\backslash(v|V_r) \in r^{TP} \wedge P(v)\}$$
$$\pi_A^{\mathbf{TP}}(r^{\mathbf{TP}}) = \{(v'|V_r)\backslash\exists v, V_r((v|V_r) \in r^{TP} \wedge v' = \pi_A^{Codd}(\{v\}))\}$$

$$\mathbf{r1^{TP}} \cup^{\mathbf{TP}} \mathbf{r2^{TP}} = \{(v|V_r)\backslash(v|V_r) \in r1^{TP} \vee (v|V_r) \in r2^{TP})\}$$

$$\mathbf{r1^{TP}} \times^{\mathbf{TP}} \mathbf{r2^{TP}} = \{(v1 \cdot v2|V_r)\backslash\exists V1_r, V2_r((v1|V1_r) \in r1^{TP} \wedge (v2|V2_r) \in r2^{TP} \wedge V_r = \text{pyramid\_intersect}(V1_r, V2_r) \wedge V_r \neq \emptyset)\},$$

where
pyramid_intersect($V1_r = ((s^1_{m1},s^1_{M1},d^1_{m1},d^1_{M1}), \ldots,(s^r_{m1},s^r_{M1}, d^r_{m1}, d^r_{M1}))$, $V2_r = ((s^1_{m2},s^1_{M2},d^1_{m2},d^1_{M2}), \ldots,(s^r_{m2},s^r_{M2}, d^r_{m2},d^r_{M2}))) = ((\max(s^1_{m1},s^1_{m2}), \min(s^1_{M1},s^1_{M2}),\max(d^1_{m1},d^1_{m2}), \min(d^1_{M1},d^1_{M2})), \ldots, (\max(s^r_{m1},s^r_{m2}), \min(s^r_{M1},s^r_{M2}), \max(d^r_{m1},d^r_{m2}), \min(d^r_{M1},d^r_{M2}))$)

$$\mathbf{r}^{TP} -^{\mathbf{TP}} \mathbf{s}^{TP} = \{(v|V_r)\backslash\exists(v|V1_r) \in r1^{TP} \wedge V_r \in \text{sub\_n\_m}(\{V1_r\}, \{V2_r\backslash(v|V2_r) \in r2^{TP}\}) \wedge V_r \neq \emptyset\},$$

where
sub_n_m(S, $\emptyset$) = S
sub_n_m(S1,S2) = $\cup_{e \in S1}$sub_1_m(e,S2)
sub_1_m(e, $\{e_1,\ldots,e_k\}$) = sub_n_m(pyramid_diff(e,e_1), $\{e_2,\ldots,e_k\}$), where
pyramid_diff($((s^1_{m1},s^1_{M1},d^1_{m1},d^1_{M1}), \ldots, (s^r_{m1},s^r_{M1},d^r_{m1},d^r_{M1}))$, $((s^1_{m2},s^1_{M2},d^1_{m2},d^1_{M2}), \ldots,(s^r_{m2},s^r_{M2},d^r_{m2},d^r_{M2}))$) =
{ $((s^i_{m1},\max(s^i_{m1},s^1_{m2}),\max(d^i_{m1},d^1_{m2}),d^i_{M1})$ with $1 \leq i \leq r$),
$((\max(s^i_{m1},s^1_{m2}),s^i_{M1},\min(d^i_{M1},d^1_{M2}),d^i_{M1}))$ with $1 \leq i \leq r$),
$((\min(s^i_{M1},s^1_{M2}), s^i_{M1},d^i_{m1},\min(d^i_{M1},d^1_{M2}))$ with $1 \leq i \leq r$),
$((s^i_{m1},\min(s^i_{M1},s^1_{M2}),d^i_{m1},\max(d^i_{m1},d^1_{m2}))$ with $1 \leq i \leq r$) }.

In each operator, for each Temporal Layer Vr[i], if $s^i_m \geq s^i_M$ or $d^i_m \geq d^i_M$, then Vr[i]=(0,0,0,0). Note that $V_r = \emptyset$ iff $V_r = ((0,0,0,0), \ldots, (0,0,0,0))$. ■

Selection σ^{TP}_P and *projection* π^{TP}_A only operate on non-temporal attributes, and need no extension with respect to Codd's operators (see, e.g., [16], [23]). On the other hand, as motivated above, $\cup^{TP}$, $\times^{TP}$ and $-^{TP}$operate in the standard way on the non-temporal attributes and manipulate the temporal attributes.

Union $\cup^{TP}$ simply computes the union of the tuples of the input relations as, e.g., in TSQL2 [23].

Cartesian product $r^{TP} \times^{TP} s^{TP}$ performs the concatenation of the two non-temporal parts and performs through the *pyramid_intersect* function the intersection of the two pyramids modelling the temporal parts of the tuples. Notably, the intersection of two pyramids is either empty or a (possibly degenerate) pyramid.

Difference $r^{TP} -^{TP} s^{TP}$ gives as output all the tuples $(v|V1_r) \in r^{TP}$ that have no *value-equivalent* tuples (i.e., tuples with the same value v for the non-temporal attributes [23]) occurring in s^{TP}. Moreover, if s^{TP} contains tuples value equivalent to v, their temporal extent must be eliminated from the pyramid modelled by $V1_r$. The difference between two pyramids is performed by the *pyramid_diff* operation. For example, Fig. 2 graphically shows the definition of *pyramid_diff* at a specific level of preference (the base level), in case the section $(s^1_{m2},s^1_{M2}, d^1_{m2},d^1_{M2})$ of the subtrahend pyramid is contained into the section $(s^1_{m1},s^1_{M1},d^1_{m1},d^1_{M1})$ of the minuend pyramid.

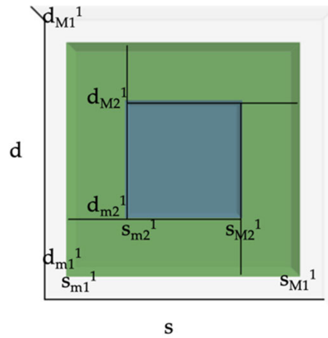

FIGURE 2. Pyramid_diff at level 1: an example.

At most four pyramids may be generated as output of *pyramid_diff*. While *pyramid_diff* operates on two pyramids, each pyramid in the output must eliminate the temporal extents of all the value-equivalent tuples in the subtrahend. The functions sub_n_m and sub_1_n are recursively used to generalize *pyramid_diff* as an operation applied to sets of pyramids (many-to-many and one-to-many applications of *pyramid_diff* respectively). Notably, the need to perform many-to-many temporal difference to cope with value-equivalent tuples is common to many TDB approaches, including TSQL2 (see, e.g. [23], [24]).[2] Finally, also notice that, given two pyramids $V1_r$ and $V2_r$, *pyramid_diff*($V1_r$,$V2_r$) eliminates the bottom level $V2_r[1]$ of the second pyramids from all the levels $V1_r[i]$ $1 \leq i \leq r$, to model the fact that, in the presence of any level of preference for a pair *<s,d>* in the subtrahend, a tuple with (non-temporal) value v is present in the subtrahend, so that v at time *<s,d>* must not be part of the output (this is similar to other treatments of temporal indeterminacy, see, e.g., [4]).

As examples, we consider two queries on Ex. 3.

Ex. 8. "What drugs can Mary take (and when, and with what preferences) during the physical therapy?"

$res^{TP} = \pi_{ad\_drug}(\sigma_{patient\_t='Mary' \wedge treatment='phys'}(pat\_treatm^{TP}) \times^{TP} \sigma_{patient\_d='Mary'}(drug\_admin^{TP})) =$
{(diphenhydramine | (69,80,4,21), (72,75,5,10), (73,75,6, 7)), (hydrocodone | (70,90,7,14), (79,83,9,12), (0,0,0,0)) }.■

In Fig. 3, we graphically show the application of the Cartesian product $\times^{TP}$ to the temporal part of a pair of tuples, i.e., (Mary, phys | (69,90,4,21), (72,83,5,14), (73,82,6,9)) ∈ *pat_treatm*^{TP} (red part of the figure), and (Mary, diphenhydramine | (60,80,1,21), (66,75,3,10), (69,75,3,7)) ∈ *drug_admin*^{TP}(yellow part of the figure).

The resulting tuple has as temporal part the intersection of the two pyramids, i.e., the orange pyramid.

Ex. 9. Query: "When, and with what preferences can Mary take diphenhydramine while avoiding the interaction with

[2]Indeed, the temporal complexity of our algebraic temporal operators $\times^{TP}$ and $-^{TP}$ is the same as TSQL2 ones, times the number of levels in the preference scale (i.e., intersection/difference between temporal rectangles has to be iterated at each preference level).

hydrocodone?" (taking the two drugs at the same time may cause drowsiness).

$res^{TP} = \pi_{patient\_d}(\sigma_{patient\_d='Mary' \wedge ad\_drug='\text{diphenhydramine}'}$ (drug_admin^{TP})) $-^{TP}\pi_{patient\_d}(\sigma_{patient\_d='Mary' \wedge ad\_drug='\text{hydrocodone}'}$ (drug_admin^{TP})) =
{(Mary | (60,70,7,21), (66,70,7,10), (0,0,0,0)),
(Mary | (70,80,14,21), (0,0,0,0), (0,0,0,0)),
(Mary | (60,80,1,7), (66,75,3,7), (69,75,3,7))}.

Notably, in Ex. 9, Mary can take diphenhydramine with high preference starting in [69,75) and with a duration in [3,7), and three pyramids (and thus three tuples) are needed to model the result.

In Fig. 4, we graphically show the result of the application of $-^{TP}$ to the temporal part of the tuples (green part of the figure). The black part of the figure shows the rectangle corresponding to the low-level preference of hydrocodone (i.e., the time to be deleted from the pyramid corresponding to (Mary, diphenhydramine)). Notably, the graphical result (green part) consists of three pyramids: the pyramid consisting of (a) rectangles (i.e., (60,70,7,21), (66,70,7,10), (0,0,0,0)), the one consisting of (b) rectangles (i.e., (70,80,14,21), (0,0,0,0), (0,0,0,0)), and the one consisting of (c) rectangles (i.e., (60,80,1,7), (66,75,3,7), (69,75,3,7)).

Other operators can be added to consider the temporal and preference aspects of the data model. As an example, we propose an operator for temporal and preference selection.

Definition 10: Given a DB defined over the scale S_r, and a TRwLP relation r^{TP},
$\sigma^{TP}_{start=s,duration=d,preference=p}(r^{TP}) = \{(v1 \mid V_r) \setminus (v1 \mid V_r) \in r^{TP} \wedge V_r[i] = (s^i_m, s^i_M, d^i_m, d^i_M) \wedge s^i_m \le s < s^i_M \wedge d^i_m \le d < d^i_M$, where $S_r(i)=p\}$. ■

As an example, we consider the following query.

Ex. 10. Query: "What drugs can Mary take starting from day 84 for 10 days with a high preference?"
$\sigma^{TP}_{start=84,duration=10,preference=high}$(drug_admin^{TP}) ={(Mary, hydrocodone | (70,91,7,14), (79,87,9,12), (84,85,10,11))}. ■

B. PROPERTIES OF THE ALGEBRA

As motivated in the introduction, since our algebraic operators perform a manipulation of an implicit representation, a proof of correctness is required. Informally speaking, for each algebraic operator Op^{TP} in our algebra we have to prove that, if we first apply Op^{TP} to relations in our representation and then we move to an explicit representation through the Make-Explicit function (see Def. 8), we get the same result that we would obtain by moving from our representation to an explicit one, and then applying the "explicit" operator Op^{Expl} corresponding to Op^{TP}. Such a proof can be graphically schematized as shown in Fig. 5.

Obviously, for the sake of the proof, the algebraic operators working on the explicit representation (i.e., on the data semantics, see Def. 8) must be defined. In the following, we define $\times^{Expl}$ as a specific example. Indeed, it simply operates in the standard way on non-temporal attributes, and performs the intersection of the sets of triples

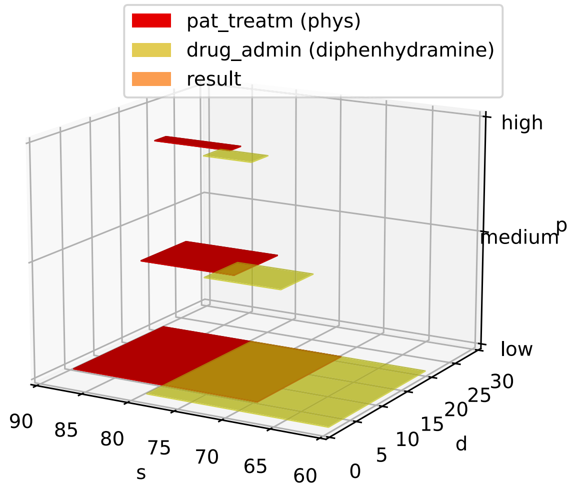

FIGURE 3. Graphical representation of the intersection (orange part of the figure) between the pyramids corresponding to (Mary, phys) ∈ pat_treatm^{TP} (red part of the figure) and (Mary, diphenhydramine) ∈ drug_admin^{TP} (yellow part of the figure).

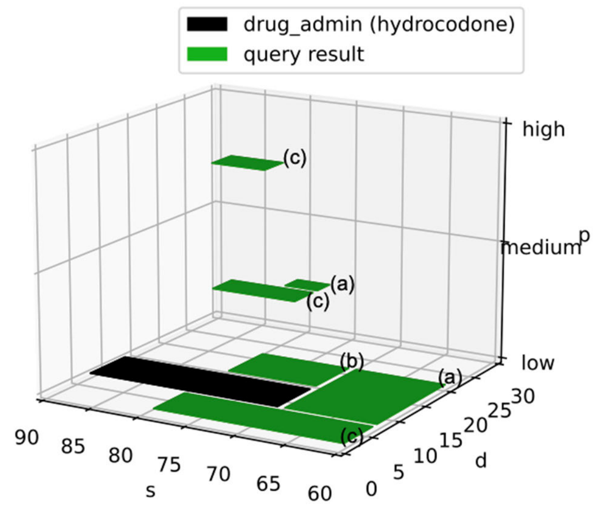

FIGURE 4. Graphical representation of the temporal part of the result (green part of the figure) of the application of $-^{TP}$ to the pyramids corresponding to (Mary, diphenhydramine) ∈ drug_admin^{TP} and (Mary, hydrocodone) ∈ drug_admin^{TP}. The black part of the figure represents the time when drug_admin^{TP} and (Mary, hydrocodone) has low preference.

<start,duration,preference> constituting the temporal parts of tuples.

Definition 11: **Cartesian product on the explicit representation.** $\mathbf{r1^{Expl}} \times^{Expl} \mathbf{r2^{Expl}} = \{(v1 \cdot v2 \mid S) \setminus \exists S1, S2($
$(v1 \mid S1) \in r1^{Expl} \wedge (v2 \mid S2) \in r2^{Expl} \wedge S = S1 \cap S2 \wedge S \neq \emptyset)\}$.

Property 1. Correctness of the manipulation of time intervals. Our extended algebraic operators, operating on the implicit temporal model, are correct: for each TP algebraic operator Op^{TP} extending Codd's operators in our approach, Make-Explicit(r^{TP} Op^{TP} s^{TP}) = Make-Explicit(r^{TP}) Op^{Ext} Make-Explicit(s^{TP}), where Op^{Ext} is the algebraic operator on the explicit model corresponding to Op^{TP}.

Proof: For the sake of brevity, we prove the property considering the Cartesian product, i.e.,
Make-Explicit(r^{TP} $\times^{TP}$ u^{TP}) =Make-Explicit(r^{TP}) $\times^{Expl}$ Make-Explicit(u^{TP}).

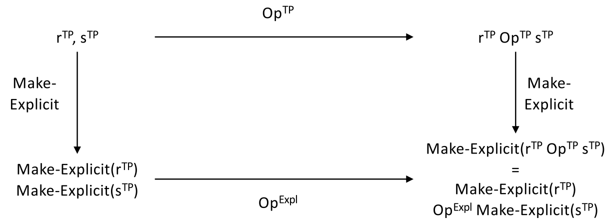

FIGURE 5. Graphical representation of the correctness property of TP operators.

The proofs for the other operators are similar.

Let r^{TP} and u^{TP} be two TRwLP relations with schemas (A|T) and (B|T) respectively, where A, B and T stand for the attributes {A1, . . ., Al}, {B1, . . ., Bm} and $\{(T^1_{sm}, T^1_{sM}, T^1_{Dm}, T^1_{DM}),\ldots,(T^r_{sm}, T^r_{sM}, T^r_{Dm}, T^r_{DM})\}$ respectively. We show the equivalence by proving the two inclusions separately, i.e., we prove that the left-hand side of the formula (henceforth lhs) implies the right-hand side (henceforth rhs) and that the rhs implies the lhs.

(x"∈lhs⇒x"∈rhs)

Let x"[A,B|S] ∈ lhs. Then, by definition of Make-Explicit and Ext, there exists a tuple x'[A,B|T]∈ $r^{TP} \times^{TP} u^{TP}$ such that x'[*A,B*]=x"[*A,B*] and, for any <s,d,p_i >∈x"[S], there exists a V_r such that, with 1≤i≤r such that x'[T]=V_r and $V_r[i]=<s^i_m,s^i_M,d^i_m,d^i_M>$, <s,d>∈Ext($V_r$[i]), i.e., $s^i_m \le s < s^i_M \wedge d^i_m \le d < d^i_M$ and pi = S_r(i).

By definition of $\times^{TP}$ and of Pyramid-Intersect, there exist tuples x1[A|T]∈ r^{TP} and x2[B|T]∈ u^{TP} such that x1[A]=x'[A], x2[B]=x'[B] and, with V_r = x'[T], $V_r[i]=<s^i_m,s^i_M, d^i_m,d^i_M>$, with $V1_r$ = x1[T], $V1_r[i]=<s1^i_m, s1^i_M, d1^i_m,d1^i_M>$ and with $V2_r$ = x2[T], $V2_r[i]=<s2^i_m,s2^i_M, d2^i_m,d2^i_M>$, $<s^i_m,s^i_M, d^i_m, d^i_M> = <\max(s^i_{m1},s^i_{m2}), \min(s^i_{M1}, s^i_{M2}), \max(d^i_{m1},d^i_{m2}), \min(d^i_{M1},d^i_{M2})>$.

Let us now reconsider <s,d,p_i >∈x"[S]. Since $s^i_m \le s < s^i_M$ and $s^i_m = \max(s^i_{m1},s^i_{m2})$ and $s^i_M = \min(s^i_{M1},s^i_{M2})$, then $\max(s^i_{m1},s^i_{m2}) \le s < \min(s^i_{M1},s^i_{M2})$. Since $d^i_m \le d < d^i_M$ and $d^i_m = \max(d^i_{m1},d^i_{m2})$ and $d^i_M = \min(d^i_{M1},d^i_{M2})$, then $\max(d^i_{m1},d^i_{m2}) \le d < \min(d^i_{M1},d^i_{M2})$.

Thus, by definition of Ext, both <s,d>∈Ext($V1_r$[i]) and <s,d>∈Ext($V2_r$[i]). Thus, by definition of Make-Explicit, there exists a tuple x1'[A|S]∈Make-Explicit(r^{TP}) such that x1'[A]=x1[A]=x'[A] and <s,d,p_i >∈x1'[S], and there exists a tuple x2'[B|S]∈Make-Explicit(u^{TP}) such that x2'[B]=x2[B]=x'[B] and <s,d,p_i>∈x2'[S].

Therefore, by definition of $\times^{Expl}$, there exists a tuple x12"[A,B|S]∈rhs such that x12"[A]=x1'[A], x12"[B]=x2'[B] and, since both <s,d,p_i>∈x1'[S] and <s,d,p_i>∈x2'[S], <s,d,p_i>∈x1'[S]∩x2'[S] and thus <s,d,p_i>∈x12"[S].

By construction, x12"=x".

(x"∈rhs⇒x"∈lhs)

Let x"[A,B|S] ∈ rhs. By definition of $\times^{Expl}$, there exist tuples x1'[A|S]∈Make-Explicit(r^{TP}) and x2'[B|S]∈Make-Explicit(u^{TP}) such that x1'[A]=x"[A], x2'[B]=x"[B] and, for any <s,d,p_i>∈x"[S], <s,d,p_i>∈x1'[S] and <s,d, p_i>∈x2'[S].

By definition of Make-Explicit and of Ext, since x1'[A|S]∈Make-Explicit(r^{TP}), there exists a tuple x1[A|T]∈r^{TP} such that x1'[A]=x1[A] and, with $V1_r$ =x1[T], <s,d>∈Ext ($V1_r$[i]). Analogously, since x2'[B|S]∈Make-Explicit(u^{TP}), there exists a tuple x2[B|T]∈u^{TP} such that x2'[B]=x2[B] and, with $V2_r$ =x2[T], <s,d> ∈ Ext($V2_r$[i]).

Since, with $V1_r[i] = <s1^i_m, s1^i_M, d1^i_m, d1^i_M>$ and $V2_r[i]=<s2^i_m,s2^i_M,d2^i_m,d2^i_M>$, by definition of Ext, $s1^i_m \le s < s1^i_M$, $s2^i_m \le s < s2^i_M$, $d1^i_m \le d < d1^i_M$, $d2^i_m \le d < d2^i_M$, and, therefore, $\max(s^i_{m1},s^i_{m2}) \le s < \min(s^i_{M1},s^i_{M2})$ and $\max(d^i_{m1},d^i_{m2}) \le d < \min(d^i_{M1},d^i_{M2})$. Then, by definition of $\times^{TP}$, there exists a tuple x'[A,B|T]∈ $r^{TP} \times^{TP} u^{TP}$ such that x'[A]=x1[A], x'[B]=x2[B], x'[T]=V_r with $V_r[i]=<\max(s^i_{m1},s^i_{m2}), \min(s^i_{M1},s^i_{M2}), \max(d^i_{m1},d^i_{m2}), \min(d^i_{M1},d^i_{M2})>$ (and $\max(s^i_{m1},s^i_{m2}) \le s < \min(s^i_{M1},s^i_{M2})$ and $\max(d^i_{m1},d^i_{m2}) \le d < \min(d^i_{M1},d^i_{M2})$).

By definition of Make-Explicit, there exists a tuple x12"[A,B|S] such that x12"[A,B]=x'[A,B] and <s,d,i>∈ x12"[S].

By construction, x12"=x". ■

Reducibility is fundamental for all TDB approaches, to grant that the new operators, which extend simpler operators to cope with new phenomena, reduce to simpler operators when the new phenomena are disregarded [15], [23].

Intuitively speaking, to prove reducibility, standard TDB approaches introduce a time-slice operator that removes time, by selecting all tuples holding at a specific time [15], [23]. In our extension, the time-slice operator considers three parameters: a time point (s), a duration (d), and a level of preference (i). Our time-slice operator $\tau^i_{<s,d>}$ is defined as follows.

Definition 12 (Time-Slice Operator$\tau^i_{<s,d>}$): Let r^{TP} a TP relation, defined on the schema $R = (A_1,\ldots, A_n|(T^1_{sm}, T^1_{sM}, T^1_{em}, T^1_{eM}),\ldots, (T^r_{sm}, T^r_{sM}, T^r_{em}, T^r_{eM}))$, S_r a scale, s∈T^C and d∈N, and i, 1≤i≤r, a level of preference in S_r: $\tau^i_{<s,d>}(r^{TP})$ ={z\∃x∈r^{TP} (z[A]=x[A] ∧ <s,d>∈Ext(V_r[i]))}. ■

Given the above definition, Property 2 holds.

Property 2. Reducibility to Codd's algebra. Our temporal extension of Codd's operators is reducible to Codd's operators, i.e., for each pair of TRwLP r^{TP} and s^{TP} (defined over a proper schema, and considering a scale S_r), for each s∈T^C, d∈N, and level of preference i (1≤i≤r) in the scale S_r: $\boldsymbol{\tau^i_{<s,d>}(r^{TP}Op^{TP}s^{TP}) = \tau^i_{<s,d>}(r^{TP})\ Op^{Codd}\tau^i_{<s,d>}(s^{TP})}$, where Op^{TP} and Op^{Codd} represent corresponding relational operators in our algebra and in Codd's algebra respectively.

Proof: For the sake of brevity, we prove the property considering the Cartesian product, i.e.,

$\tau^{\mathbf{i}}_{<\mathbf{s,d}>}(r^{TP} \times^{TP} u^{TP}) = \tau^{\mathbf{i}}_{<\mathbf{s,d}>}(r^{TP}) \times^{Codd} \tau^{\mathbf{i}}_{<\mathbf{s,d}>}(u^{TP})$.

The proofs for the other operators are similar.

Let r^{TP} and u^{TP} be two TRwLP relations with schemas (A|T) and (B|T) respectively, where A, B and T stand for the attributes {A1, . . ., Al}, {B1, . . ., Bm} and $\{(T^1_{sm}, T^1_{sM}, T^1_{Dm}, T^1_{DM}),\ldots,(T^r_{sm}, T^r_{sM}, T^r_{Dm}, T^r_{DM})\}$ respectively. We show the equivalence by proving the two inclusions separately, i.e., we prove that the left-hand side of the formula (henceforth lhs) implies the right-hand side (henceforth rhs) and that the rhs implies the lhs.

(x''∈lhs⇒x''∈rhs)

Let x''[A,B] ∈ lhs. Then, by definition of $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}$, there exists a tuple x'[A,B|T] ∈ $r^{TP} \times^{TP} u^{TP}$ such that x'[*A*,*B*]=x''[*A*,*B*] and there exists a V_r such that x'[T]=V_r and, with $V_r[i]=<s^i_m,s^i_M,d^i_m,d^i_M>$, $<s,d>\in Ext(V_r[i])$, i.e., $s^i_m \le s<s^i_M \wedge d^i_m \le d<d^i_M$.

By definition of $\times^{TP}$ and of Pyramid-Intersect, there exist tuples x1[A|T]∈r^{TP} and x2[B|T]∈ u^{TP} such that x1[A] = x'[A], x2[B] = x'[B] and $<s^i_m,s^i_M,d^i_m, d^i_M> = <\max(s^i_{m1},s^i_{m2}), \min(s^i_{M1},s^i_{M2}), \max(d^i_{m1},d^i_{m2}), \min(d^i_{M1},d^i_{M2})>$, where $V1_r$ =x1[T], $V1_r[i]=<s1^i_m,s1^i_M,d1^i_m,d1^i_M>$, $V2_r$ =x2[T], $V2_r[i]=<s2^i_m,s2^i_M,d2^i_m,d2^i_M>$.

Since $\max(s^i_{m1},s^i_{m2}) = s^i_m \le s < s^i_M = \min(s^i_{M1},s^i_{M2})$ and $\max(d^i_{m1},d^i_{m2}) = d^i_m \le d < d^i_M = \min(d^i_{M1},d^i_{M2})$, by definition of Ext, both $<s,d>\in Ext(V1_r[i])$ and $<s,d>\in Ext(V2_r[i])$. Thus, by definition of $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}$, there exists a tuple x1'∈ $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}(r^{TP})$ such that x1'[A]=x1[A]=x'[A], and there exists a tuple x2'∈ $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}(u^{TP})$ such that x2'[B]=x2[B]=x'[B].

Therefore, by definition of $\times^{Codd}$, there exists a tuple x12''∈ rhs such that x12''[A]=x1'[A] and x12''[B]=x2'[B].

By construction, x12''=x''.

(x''∈rhs⇒x''∈lhs)

Let x''[A,B] ∈ rhs. By definition of $\times^{Codd}$, there exist tuples x1'[A]∈ $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}(r^{TP})$ and x2'[B]∈ $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}(u^{TP})$ such that x1'[A]=x''[A], x2'[B]=x''[B].

By definition of $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}$, there exists a tuple x1[A|T]∈r^{TP} such that x1'[A]=x1[A] and, with $V1_r$ = x1[T], $<s,d>\in Ext(V1_r[i])$, and there exists a tuple x2[B|T]∈u^{TP} such that x2'[B]=x2[B] and, with $V2_r$ = x2[T], $<s,d>\in Ext(V2_r[i])$.

With $V1_r$ = x1[T]=$<s1^i_m,s1^i_M, d1^i_m,d1^i_M>$ and $V2_r$ = x2[T]=$<s2^i_m,s2^i_M, d2^i_m,d2^i_M>$, we thus have that $s1^i_m \le s < s1^i_M$, $s2^i_m \le s < s2^i_M$, $d1^i_m \le d < d1^i_M$, $d2^i_m \le d < d2^i_M$, and, therefore, $\max(s^i_{m1},s^i_{m2}) \le s < \min(s^i_{M1},s^i_{M2})$ and $\max(d^i_{m1},d^i_{m2}) \le d < \min(d^i_{M1},d^i_{M2})$.

Therefore, by definition of $\times^{TP}$, there exists a tuple x'[A,B|T]∈ $r^{TP} \times^{TP} u^{TP}$ such that x'[A]=x1[A], x'[B]=x2[B], x'[T]=V'_r with $V'_r[i]=<\max(s^i_{m1},s^i_{m2}), \min(s^i_{M1},s^i_{M2}), \max(d^i_{m1},d^i_{m2}), \min(d^i_{M1},d^i_{M2})> \neq \emptyset$ since $\max(s^i_{m1},s^i_{m2}) \le s < \min(s^i_{M1},s^i_{M2})$ and $\max(d^i_{m1},d^i_{m2}) \le d < \min(d^i_{M1},d^i_{M2})$.

Therefore, by definition of $\tau^{\mathbf{i}}_{<\mathbf{s},\mathbf{d}>}$, there exists a tuple x12'' such that x12''[A,B]=x'[A,B].

By construction, x12''=x''. ■

IV. CONCLUSION, RELATED AND FUTURE WORK, AND APPLICATIONS

We follow the theoretical guidelines provided by the TDB literature (consider [3], [15], [16], [23]) to propose the first relational TDB approach considering ***preferences*** for ***indeterminate valid time***, as required by several applications contexts including ***guidelines, plans*** and ***workflows***. Our approach is parametric with respect to a qualitative scale of preferences, and considers *layered* preference functions, as we defined in [2] (notably, in [2] layered preferences are used in a purely AI context). As main contributions of our approach, we provide (i) a data representation model for temporally indeterminate valid time with layered preferences, and (ii) its data semantics. We (iii) define an extension of Codd's operators to query the new model, and (iv) prove its correctness with respect to the data semantics and (vi) its reducibility to standard Codd's operators.

The most closely related approaches in the TDB literature are [4], [10], and [12] –see the discussion in Section I. They all consider *indeterminate valid time*. Reference [12] provides a probabilistic data model but no algebra, [10] deals with events (instantaneous facts) and probabilities, while in [4] we provide a family of data models for durative facts and algebrae. However, none of such approaches cope with preferences. Additionally, a minor difference regards the fact that in [4], [10], and [12] indeterminacy only regards the temporal location of points [10] or interval endpoints [4], [12], while in this paper we explicitly cope also with durations (see footnote 1). Notice, however, that an approach combining preferences with a <start,end> representation of indeterminate valid time can be devised, along the guidelines we provided in this paper. It will be a –minor– future work of us.

We are implementing our algebraic operators in the open-source DBMS PostgreSQL, to demonstrate the feasibility of our approach, and to analyze its computational complexity. In particular, we are using PL/pgSQL, and we are exploiting the "int4range" range type to model the ranges of starting times and of durations, and the "∗" (intersection) and "&&" (empty overlap) operators to efficiently manipulate them.

On top of such an implementation, in our future work we plan to consider all the steps needed to propose a full extension of PostgreSQL to manage temporal preferences. The first step will be an extension of SQL to support the creation and the querying of tables with temporal preferences. Following the methodology highlighted in Section 24 of the TSQL2 book [23], we will then consider the extensions to data dictionary, DDL and Query compilers, run-time evaluator, transaction and data manager, possibly exploiting the extensions already provided in https://github.com/scalegenius/pg_bitemporal to manage (bi)temporal tables in PostgreSQL.

Though the approach proposed in this paper is fully domain and application independent, we have developed it within a specific project, "Personalized Training of Professional Competences with AI" (PTPC-AI [7]; a part of a large national project: "*Learning Personalization with AI and of AI*" (AI-LEAP) [5]), considering medical education on the basis of Computer-Interpretable clinical Guidelines (CIGs). Clinical guidelines encode evidence-based medical practices to diagnose and treat specific diseases. There are thousands of clinical guidelines in the medical literature (consider, e.g., the collection provided by the Guideline International Network [14]) and are widely used in the medical practice, and in medical education. Several approaches in the Medical Informatics area have provided computer systems to acquire CIGs and use them for medical decision support and education (see, e.g., the surveys [18], [20], [30]). Different temporal issues have to be considered by such CIG systems, including the definition/use of TDBs to model clinical data, which are intrinsically temporal, and in which temporal indeterminacy plays a relevant role; see, e.g. the survey in [25]. The adoption of ***preferences*** in general, and of ***temporal preferences*** in particular, plays an important role in the CIG context, e.g., to *improve the quality* of medical treatments (consider, e.g., Ex. 3), and/or to improve patients' compliance to treatments, through temporal *personalization* [1], [19], [29]. Thus, the definition of a TDB approach supporting the management of preferences for temporally indeterminate (medical) data is a crucial step in our PTPC-AI project.

ACKNOWLEDGMENT

The authors would like to thank R. T. Snodgrass for many inspiring comments and suggestions on the initial drafts of the article and to Antonella Coviello for her invaluable contributions to its implementation.

REFERENCES

[1] A. Andolina, M. Guazzone, L. Piovesan, and P. Terenziani, "Temporal reasoning and query answering with preferences and probabilities for medical decision support," *Expert Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116565.
[2] L. Anselma, A. Mazzei, L. Piovesan, and P. Terenziani, "Reasoning and querying bounds on differences with layered preferences," *Int. J. Intell. Syst.*, vol. 36, no. 5, pp. 1998–2035, May 2021.
[3] L. Anselma, L. Piovesan, and P. Terenziani, "Dealing with temporal indeterminacy in relational databases: An AI methodology," *AI Commun.*, vol. 32, no. 3, pp. 207–221, Jul. 2019.
[4] L. Anselma, P. Terenziani, and R. T. Snodgrass, "Valid-time indeterminacy in temporal relational databases: Semantics and representations," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2880–2894, Dec. 2013.
[5] M. Baldoni, C. Baroglio, A. Bottrighi, M. Bucciarelli, S. Capecchi, E. Gandolfi, C. Gena, F. Iani, E. Marengo, D. Marocco, R. Micalizio, L. Piovesan, M. Ponticorvo, A. Rapp, A. Roveta, P. Terenziani, and F. Ugo, "Fostering awareness and personalization of learning artificial intelligence," in *Proc. CEUR Workshop*, 2023, pp. 449–454.
[6] R. Bartak, R. A. Morris., and K. B. Venable, "An introduction to constraint-based temporal reasoning," in *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Cham, Switzerland: Springer, 2014.
[7] A. Bottrighi, E. Girardi, A. Maconi, S. Nera, E. Raina, A. Roveta, L. Piovesan, F. Ugo, and P. Terenziani, "Clinical guidelines acquisition, representation and simulation for medical education," in *Proc. SITIS*, Nov. 2023, pp. 1–12.
[8] E. F. Codd, "Relational completeness of data base sublanguages," in *Database Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1972.
[9] J. Cohen, K. M. Ericson, D. Laibson, and J. M. White, "Measuring time preferences," *J. Econ. Literature*, vol. 58, no. 2, pp. 299–347, Jun. 2020.
[10] A. Dekhtyar, R. Ross, and V. S. Subrahmanian, "Probabilistic temporal databases, I: Algebra," *ACM Trans. Database Syst.*, vol. 26, no. 1, pp. 41–95, 2001.
[11] D. Dubois, H. Fargier, and H. Prade, "Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty," *Int. J. Speech Technol.*, vol. 6, no. 4, pp. 287–309, Oct. 1996.
[12] C. E. Dyreson and R. T. Snodgrass, "Supporting valid-time indeterminacy," *ACM Trans. Database Syst.*, vol. 23, no. 1, pp. 1–57, Mar. 1998.
[13] J. Gamper, M. Ceccarello, and A. Dignos, "What's new in temporal databases?" in *Proc. ADBIS*, 2022, pp. 45–58.
[14] *GIN. Guidelines International Network*. Accessed: May 7, 2024. [Online]. Available: https://g-i-n.net/
[15] C. S. Jensen and R. T. Snodgrass, "Semantics of time-varying information," *Inf. Syst.*, vol. 21, no. 4, pp. 311–352, Jun. 1996.
[16] L. Liu and M. T. Ozsu, *Encyclopedia of Database Systems*. Cham, Switzerland: Springer, 4614.
[17] L. E. McKenzie and R. T. Snodgrass, "Evaluation of relational algebras incorporating the time dimension in databases," *ACM Comput. Surveys*, vol. 23, no. 4, pp. 501–543, Dec. 1991.
[18] M. Peleg, "Computer-interpretable clinical guidelines: A methodological review," *J. Biomed. Inform.*, vol. 46, no. 4, pp. 744–763, 2013.
[19] M. Peleg, Y. Shahar, S. Quaglini, and A. Fux, "MobiGuide: A personalized and patient-centric decision-support system and its evaluation in the atrial fibrillation and gestational diabetes domains," *User Modeling User-Adapted Interaction*, vol. 27, no. 1, pp. 1–18, Jun. 2017.
[20] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe, and M. Stefanelli, "Comparing computer-interpretable guideline models: A case-study approach," *J. Amer. Med. Inform. Assoc.*, vol. 10, no. 1, pp. 52–68, 2003.
[21] F. Rossi, K. B. Venable, and T. Walsh, "A short introduction to preferences: Between AI and social choice," in *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 1st ed. San Rafael, CA, USA: Claypool Publishers, 2011.
[22] Y. Shahar, S. Miksch, and P. Johnson, "The asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines," *Artif. Intell. Med.*, vol. 14, nos. 1–2, pp. 29–51, Sep. 1998.
[23] R. T. Snodgrass, *The TSQL2 Temporal Query Language*. Alphen aan den Rijn, Netherlands: Kluwer, 1995.
[24] A. U. Tansel, "Temporal algebras," in *Encyclopedia of Database Systems*. Cham, Switzerland: Springer, 2018, pp. 3905–3909.
[25] P. Terenziani, E. German, and Y. Shahar, "The temporal aspects of clinical guidelines," in *Computer-Based Medical Guidelines and Protocols: A Primer and Current Trends*. IOS Press, 2008, pp. 81–100.
[26] P. Terenziani, "Irregular indeterminate repeated facts in temporal relational databases," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 1075–1079, Apr. 2016.
[27] P. Terenziani, "Nearly periodic facts in temporal relational databases," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2822–2826, Oct. 2016.
[28] (2022). *International Symposium on Temporal Representation and Reasoning*. [Online]. Available: https://time-symposium.org/t/
[29] M. van der Pol, D. Hennessy, and B. Manns, "The role of time and risk preferences in adherence to physician advice on health behavior change," *Eur. J. Health Econ.*, vol. 18, no. 3, pp. 373–386, Apr. 2017.
[30] W. Van Woensel et al., "A community-of-practice-based evaluation methodology for knowledge intensive computational methods and its application to multimorbidity decision support," *J. Biomed. Informat.*, vol. 142, Jun. 2023, Art. no. 104395.
[31] Y. Wu, S. Jajodia, and X. S. Wang, "Temporal database bibliography update," in *Temporal Databases*. Cham, Switzerland: Springer, 1997, pp. 338–366.

LUCA ANSELMA was born in Turin, Italy, in 1975. He received the M.S. and Ph.D. degrees in computer science from the University of Turin, Italy, in 2002 and 2006, respectively.

From 2006 to 2022, he was an Assistant Professor with the University of Turin, where he has been an Associate Professor with the Department of Computer Science, since 2022. He is the author of more than 80 articles published in international journals and conferences. His research interests include temporal relational databases, temporal reasoning, health informatics, and natural language processing and generation.

PAOLO TERENZIANI was born in Turin, Italy, in 1963. He received the M.S. degree in information science from the University of Turin, in 1989, and the joint Ph.D. degree in computer science from the University of Turin and University of Milan, in 1993.

From 1992 to 1998, he was a Research Assistant with the University of Turin. From 1998 to 2000, he was an Assistant Professor with the University of Eastern Piedmont, Alessandria, Italy, where he has been a Full Professor with the Science and Technological Innovation Department, since 2000. He is the author of more than 150 papers in international journals and conferences. His research interests include temporal relational databases, knowledge acquisition and representation, temporal reasoning, process mining, and their applications in medical informatics.

Open Access funding provided by 'Università degli Studi di Torino' within the CRUI CARE Agreement