Irregular Indeterminate Repeated Facts in Temporal Relational Databases

Paolo Terenziani

Abstract—Time is pervasive of reality, and many relational database approaches have been developed to cope with it. In practical applications, facts can repeat several times, and only the overall period of time containing all the repetitions may be known (consider, e.g., *On January, John attended five meetings of the Bioinformatics project*). While some temporal relational databases have faced facts repeated at (known) periodic time, or single facts occurred at temporally indeterminate time, the conjunction of *non-periodic repetitions* and *temporal indeterminacy* has not been faced yet. Coping with this problem requires an in-depth extension of current techniques. In this paper, we have introduced a new data model, and new definitions of relational algebraic operators coping with the above issues. We have studied the properties of the new model and algebra (with emphasis on the reducibility property), and how it can be integrated with other models in the literature.

Index Terms—Temporal databases, database design, modeling and management

1 Introduction

TIME is pervasive of our way of dealing with reality. As a consequence, time is often modelled in databases. The scientific community agrees that time has a special status with respect to the other data, so that its treatment within a *relational database* context requires dedicated techniques [1], [2]. A plethora of dedicated approaches have been developed in the area of *temporal relational databases* (TDB in the following; see, e.g., [3], [4]), and different approaches have been proposed for supporting efficient access to temporal data [5] and improving the effectiveness of information retrieval [6]. Most TDB approaches focus on individual occurrences of facts, whose time of occurrence (*valid time* [2]) is exactly known. However, in real world applications, it is often the case that the same fact/event is repeated several times. Tuzhilin and Clifford [7] distinguished among (1) "strongly periodic" events, occurring at equally distant intervals of time (e.g., Mondays, weeks); (2) "nearly periodic" events, occurring at regular intervals of time, but not necessarily at equally distant intervals (e.g., a person going to the cinema once each week—and thus at regular intervals—, but not necessarily in the same day—thus, not at equally distant intervals); (3) "intermittent" events, which occur repeatedly in time, but without any regularity (e.g., a man visiting a pub "periodically", meaning that the visits can be quite irregular). While several approaches, including Tuzhilin and Clifford's one, considered the first two classes above, the third class has been, to the best of our knowledge, neglected by the TDB literature. Of course, if a fact/event repeats at irregular times, but each time is exactly known, it can be easily managed as a set of timestamped tuples. However, in many cases, only the span of time (*frame time* henceforth; "January 2015" in Example 1) containing all the repetitions, and the number (henceforth *cardinality*; "five" in Example 1) of such repetitions is available, such as, e.g., in Example 1

Example 1. *In January 2015*, John attended *five times* the meetings of the Bioinformatics project.

In many cases, as in the above example, the *exact* time of facts at the given TDB granularity is not known, and can only be *approximated*, so that *temporal indeterminacy* (Dyreson, 2009) has to be faced. Temporal indeterminacy is so important that "support for temporal indeterminacy" was already one of the eight explicit goals of the data types in TSQL2 consensus approach [2]. In effect, temporal indeterminacy has various possible sources, including scale, dating techniques, future planning, unknown or imprecise event times, clock measurements (this list is not exhaustive, and is taken from TSQL2 book [2]). Additionally, many facts and human activities are *repeated* in time, and, given the above causes, repetitions are expressed in a temporally indeterminate way. For instance, examples like Example 1 arise in many tasks (e.g., scheduling, planning, office automation) and domains, ranging from the recording of employee activities (see Example 1 above) to the elicitation patient symptoms (e.g., *On 9/9/2011 between 9am and 12am John had three stool evacuations),* from manufacturing (e.g., *125 machines were produced between 8am and 6pm*) to auditing (e.g., *there were 23 phone calls from Rome on 9/9/2011 between 9am and 12am*) and monitoring (e.g., *John had 81 heart-beats at 10:15*). In all such domains, it is quite unrealistic to pretend that the exact time of each episode (each one of the repetitions) is known. For instance, a patient usually cannot record the exact starting and ending time of her/his stool evacuations, but just their number and the frame of time when they occurred. Similar considerations hold for all the other domains elicited above.

Despite the diffusion of the phenomenon, dealing with *irregularly repeated (i.e., intermittent [7]) facts/events in a frame time* is, to the best of our knowledge, an entirely new goal in TDBs. Coping with it requires the joint treatment of two phenomena: the *cardinality* of *irregular repetitions* (*intermittent events* in [7]) and *temporal indeterminacy* (see, e.g., [8]), since only the *frame time* (and not the exact time of each occurrence) is known. Up to now, only few TDB approaches have faced temporal indeterminacy, and none of them copes with cardinalities. Indeed, in Section 2 we show that coping with cardinalities is a challenging problem: they must be an explicit part of the data model and algebra, their treatment cannot be delegated to users, and involves an in-depth extension of current algebrae (as shown in Section 4, where Property 3 clarifies the differences between current algebrae and our extended one).

For generality, our approach considers also the possibility that (i) the frame time is a *non-convex* span of time (in Example 2, "working days of January 2015" excludes Saturdays, Sundays, and holidays), and/or (ii) the *exact cardinality of repetitions is unknown* (but it is bounded by a minimum and maximum value—"five" and "six" in Example 2).

Example 2. In the working days of January 2015, Ann attended five or six meetings of the Bioinformatics project.

In this paper, we extend the current TDB literature to cope with such phenomena. In Section 2 we discuss the key problems and challenges, and sketch our solutions. Then, we provide (i) a ***data model*** to represent intermittent indeterminate repetitions (Section 3), and (ii) a ***temporal relational algebra*** to query it, investigating its properties (Section 4). Section 5 presents related works, and Section 6 contains conclusions.

2 Main Problems and Solutions

We aim at identifying a *data model* and *relational algebraic operators* to cope with irregular indeterminate repetitions (like in Example 1 and Example 2 above) in TDBs. The *data model* must be *closed* with respect to the algebraic operators to query it, so that the *results* of the application of such operators must still be *expressible* in our data model. Such a goal leads to important implications about our data model

We proceed incrementally. First, we focus on exact cardinality and convex frame time (as in Example 1), and then we generalize. Two different problems have to be faced to cope with cases like

• *The author is with the DISIT, Università del Piemonte Orientale "Amedeo Avogadro," Alessandria, Italy. E-mail: paolo.terenziani@uniupo.it.*

Manuscript received 13 June 2015; revised 4 Dec. 2015, 9 Oct. 2015; accepted 10 Dec. 2015. Date of publication 17 Dec. 2015; date of current version 3 Mar. 2016.
Recommended for acceptance by R. Cheng.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2015.2509976

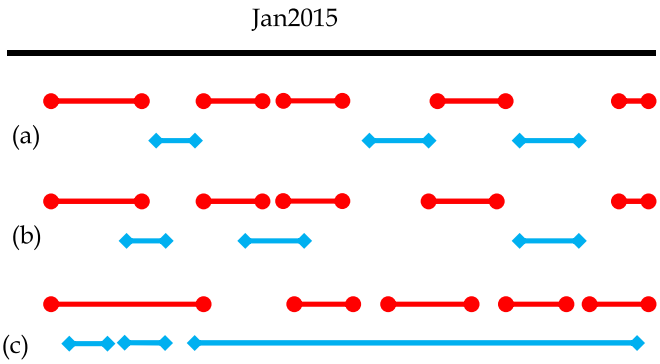

Fig. 1. Different scenarios for Q1.

Example 1: *repetitions* of facts, and *temporal indeterminacy*. As regard repetitions, we could allow for *duplicates*: several tuples representing occurrences of value-equivalent facts, holding at the same frame time, could be inserted in the same relation. However, the adoption of duplicates has several disadvantages, e.g., as regard space, and cannot be generalized to the case in which the exact number of repetitions is unknown (as in Example 2). We thus propose a compact solution, where numeric attributes are used to represent the *number of repetitions* of a fact in a frame time. For instance, as a first approximation, Example 1 could be modelled by the tuple <*John, Bioinformatics* | *5, January 2015*>. It is worth emphasizing that such a number of repetitions (i.e., the *cardinality*) cannot be coped with as a "standard" numeric attribute, to be managed directly by users/developers. We will show in Section 4 that relational algebraic operators have to be re-defined to correctly deal with such numbers. Such a definition must be provided once-and-for-all, and cannot be demanded to users/developers.

The second problem regards the treatment of *temporal indeterminacy*. Indeed, in Example 1, "January 2015" is not the exact time when any of the five meetings occurred: it is a span of time containing (the exact valid time of) the repetitions. Such a form of temporal indeterminacy makes the definition of *relational algebraic operators* quite challenging, as exemplified in Example 3 below.

Example 3. Let us consider a temporal relation SYMPT = {<John, headache | 5,Jan2015>, <Mary,headache | 3,Jan2015>} representing patient symptoms (defined on the schema <Patient, Symptom | Repetitions, Frame-time>) and suppose we want to know

(Q1) when both John and Mary had headache?

Since we only know that the symptoms are during January 2015, many different scenarios are possible. Three of them are shown in Fig. 1 (where John's headache episodes are in red, with round endpoints, and Mary's ones in blue, with diamond endpoints).

Scenarios (a) and (c) show two extreme situations: in (a) the intersection is empty (its cardinality is zero), while in (c) the cardinality of the intersection is the maximum possible between five and three intervals (i.e., seven).

Abstracting from the specific example, since we don't know the exact temporal location of the input facts (but just the frame of time containing them), we cannot know (i) the exact location of the intersections (but just their frame time), and (ii) the exact number of the intersections, but just a minimum (zero in the example) and maximum (seven in the example) bound for it.

However, we want that our data model is *expressive* enough to model the results of relational algebraic operations. The generalization to model not just a fixed cardinality, but a *minimum* and *maximum bound* of it, is a natural way to achieve such a goal.

A second generalization, required to cope with complex cases such as Example 2, regards the fact that, in our approach, the frame time may be a non-convex period of time. For instance, "the working days of January 2015" cover (in our University) the time intervals [Jan 7, Jan 9], [Jan 12, Jan 16], [Jan 19, Jan 23], [Jan 26, Jan 30].

It is worth noticing that the cases in which

(i) there is just one repetition and\or
(ii) the exact number of repetitions of events is known

can be easily modelled, and constitute specific cases of our general model, in which the minimum and maximum cardinalities are (i) set to the value 'one' or (ii) set to be equal respectively. Indeed, the example in Fig. 1 demonstrates that, even in case the exact input cardinalities are known, the cardinalities obtained after the application of relational operators may only be bounded by a minimum and a maximum value. In particular, temporal intersection (and, thus, Cartesian Product) always produce a lower bound of 0 regardless of the initial values. As a result, we may end up with query results that are quite imprecise. This is especially disadvantageous when the query is heavily nested as the result imprecision can be accumulated from one level to the next level. However, we stress that such a behaviour is not due to our choice of the data model and algebraic operators, but is an intrinsic feature of the phenomena we cope with.

In the rest of the paper, the above ideas are detailed.

3 Data Model

Tuples are associated with *valid time* (for the sake of brevity, *transaction time* is not considered in this paper). The timeline is partitioned into granules of a chosen *basic granularity*. As is BCDM [2; Chap.X] (which is the semantic model underlying many TDB approaches, including the "consensus" TSQL2 [2]), the time domain is totally ordered and is isomorphic to the subsets of the domain of natural numbers. The domain of valid times D_{VT} is given as a set $D_{VT} = \{t_1, t_2, \ldots, t_k\}$ of granules. The number of repetitions of a fact is encoded by two cardinality attributes N and M, defined on the domains of natural numbers and of positive natural numbers respectively, with the constraint that the minimal cardinality is less or equal that the maximum cardinality. The schema of a "IR" (Irregular Repeated) *temporal relation* R = (A_1, ... , A_n | N,M,FT) consists of an arbitrary number of non-temporal attributes A_1, ... , A_n, encoding some fact, of a minimal cardinality attribute N, of a maximal cardinality attribute M, and of an attribute FT representing a (possibly non-convex) frame time as a *temporal element* [9] (i.e., a set of *non-overlapping time intervals*), with domain $2^{D_{VT}}$. Thus, a tuple $x = (a_1, \ldots, a_n | n_1, n_2, t)$ (where $n_1 \leq n_2$, and $n_2 > 0$) in a temporal relation r(R) on the schema R consists of a n-tuple of values for the non-temporal attributes associated with a minimum cardinality n_1, a maximum cardinality n_2, and a frame time $t \in D_{VT}$, and represents the fact that *there are between* n_1 *and* n_2 *occurrences of the fact* $a_1, \ldots, a_n$ *in the frame time* t. As an example, consider a temporal relation MEET modelling meetings at the granularity of days. The schema of MEET is <Employee, Meeting | N, M, FT>. The first two tuples represent Example 1 and Example 2 respectively, while the third tuple represents Example 4

Example 4. Sue attended *one* Math meeting in January 2015.

Notation 1. Given a tuple x defined on the schema R = (A_1, ... , A_n | N,M,FT), we denote by A the set of attributes A_1, ..., A_n. x [A] denotes the values of the A attributes in x, x[FT] denotes the frame time, x[N] and x[M] denote the minimum and maximum cardinality respectively.

Note. It is worth pointing out that the frame time may easily contain timestamps $t_i \in D_{VT}$ (instead of durative time intervals), represented by "degenerate" intervals $[t_i,t_i]$.

TABLE 1
Relation MEET: Representation of Ex. 1, 2, and 4

Employee	Meeting	N	M	FT
John	BioInf	5	5	{[Jan1, Jan31]}
Ann	BioInf	5	6	{[Jan7, Jan9], [Jan12, Jan16], [Jan19, Jan23], [Jan26, Jan30]}
Sue	Math	1	1	{[Jan1, Jan31]}

3.1 Consistent Extension Properties

Recently, Anselma et al. [10] have proposed a family of data models and algebrae to cope with different forms of temporal indeterminacy (but not with repetitions). Our data model is a consistent extension of one of such models, called "ITE" (Property 1), as well as of valid-time TSQL2 relations (Property 2). In particular, in the ITE approach, a (possibly non-convex) set of granules G_S is used to represent the valid time of a tuple, meaning that the tuple may hold at any possible subset of the granules in G_S. E.g., an ITE tuple <John, Bioinf | {1,2,3}> represents the fact that John attended a Bioinf. meeting at granules {1}, or {2}, or {3}, or {1,2}, or {1,3}, or {2,3}, or {1,2,3}, or Ø –so that he might also not have attended the meeting).

Properties 1 and 2 grant that we can cope with the same content than "ITE" and TSQL2 valid-time relations. They are, indeed, a restriction of our relations, in which both minimum and maximum cardinalities of tuples must be set to the value 'one', and (in the case of TSQL2) the frame time is a single (convex) time interval.

Property 1. *"ITE" relations can be modelled by temporal relations in our approach.*

Property 2. *TSQL2 valid-time relations can be modelled by temporal relations in our approach.*

For example, at the granularity of days, the third tuple in Table 1 may represent the "ITE" tuple <Sue, Math | {Jan1, Jan2, …, Jan31}> and the TSQL2 tuple <Sue, Math | [Jan1, Jan31]>. However, it is important to stress that in TSQL2 a tuple <Sue, Math | [Jan1, Jan31]> is interpreted (and treated by algebraic operators) as meaning that Sue attended the Math meeting at all granules between January 1st and January 31st. On the other hand, in our approach (and in the ITE model) the tuple has a quite different interpretation: Sue attended a Math meeting in a time *contained in* January. Such a semantics is supported by the algebraic operators defined in Section 4.

4 Temporal Relational Algebra

Codd defined as complete any query language that is as expressive as his set of five relational algebraic operators: relational union ($\cup$), relational difference ($-$), selection (σ_P), projection (π_A), and Cartesian Product ($\times$) [11]. Here we propose a temporal extension of Codd's operators to query the data model in Section 3. Several temporal extensions to Codd's operators have been provided in the TDB literature [9]. In most cases, such extensions behave like standard non-temporal operators on the non-temporal attributes, and involve the application of set operators on the temporal attributes. For instance, in TSQL2 "consensus" approach, (i) Cartesian Product involves pairwise concatenation of the values of non-temporal attributes and pairwise intersection of their temporal values, (ii) difference r-s operates in the standard way on non-temporal attributes, and make the difference of valid times (by subtracting from each tuple f∈r the valid times of all the tuples f'∈s *value equivalent* [2] to it), and (iii) relational union, non-temporal selection, and projection operate in the standard way on the non-temporal part, and do not operate on the temporal part.

4.1 Relational Algebra for Irregular Repetitions

We ground our approach on such a "consensus" background, extending the algebraic operators to cope with the new attributes.

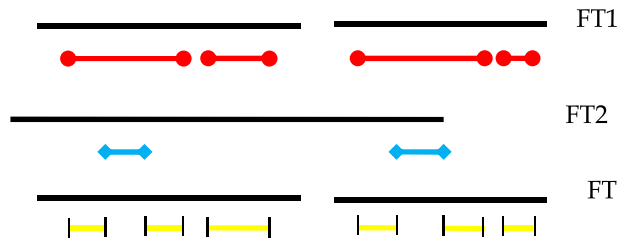

Fig. 2. Difference (shown in yellow) between red intervals and blue intervals. The figure shows an example of maximum cardinality.

In the definition Definition 2 we use the superscript "IR" (Irregular Repeated) for our operators, while ∪, ∩, -, denote standard set operators. For the sake of brevity, we preliminarily define the set ***VE_INT(f,s)*** of all the tuples in the relation s that are *value-equivalent* to a tuple f (i.e., that are equal to f as regards the values of the non-temporal attributes [2]) and whose frame time intersects with f 's frame time (Definition 1). In the following, "{x\ …}" stands for "all x's such that".

Definition 1 (Sets of value equivalent tuples). *Given two relations r and s defined over the schema $R = (A_1, \ldots, A_n \mid N,M,FT)$, and a tuple $f = <v \mid n_1, m_1, t_1> \in r$, we define* **VE_INT(f,s)** $= \{f' \setminus f' \in s \wedge f'[A] = f[A] \wedge f[FT] \cap f'[FT] \neq \emptyset\}$ *as the set of all and only the tuples f' in s that are value equivalent to f and that temporally intersect f.*

Definition 2 (Temporal algebraic operators). *Let r and s denote* temporal *relations in our model having the proper schema.*

$$r \cup^{IR} s = \{<v \mid n,m,t> \setminus <v \mid n,m,t> \in r \vee <v \mid n,m,t> \in s\}$$

$$r -^{IR} s = \{<v \mid n,m,t> \setminus \exists n_1,m_1,t_1\ (<v \mid n_1, m_1,t_1> \in r \wedge VE\_INT(<v \mid n_1,m_1,t_1>,s) = \emptyset \wedge n = n_1 \wedge m = m_1 \wedge t = t_1) \vee \exists n_1,m_1,t_1\ (<v \mid n_1, m_1,t_1> \in r \wedge VE\_INT(<v \mid n_1,m_1,t_1>,s) \neq \emptyset \wedge n = 0 \wedge m = m_1 + \sum_{f_i \in VE\_INT(<v \mid n1,m1,t1>,s)} f_i[M] \wedge t = t_1)$$

$$r \times^{IR} s = \{<v_1 \cdot v_2 \mid n,m,t> \setminus \exists n_1,m_1,t_1\ (<v_1 \mid n_1, m_1, t_1> \in r \wedge \exists n_2,m_2,t_2\ (<v_2 \mid n_2,m_2,t_2> \in s) \wedge n = 0 \wedge m = m_1 + m_2 - 1 \wedge t = t_1 \cap t_2 \wedge t_1 \cap t_2 \neq \emptyset)\}$$

$$\pi^{IR}_A(r) = \{<v \mid n,m,t> \setminus \exists\ v_1,n_1,m_1,t_1\ (<v_1 \mid n_1, m_1,t_1> \in r \wedge v = \pi_A(v_1) \wedge n = n_1 \wedge m = m_1 \wedge t = t_1\}$$

$$\sigma^{IR}_P(r) = \{(<v \mid n_1,m_1,t> \setminus (<v \mid n_1, m_1,t> \in r \wedge P(v)\}$$

As motivated above, all our algebraic relational operators operate in the standard way on the non-temporal attributes. As in TSQL2, our union, projection and non-temporal selection do not modify the temporal attributes. Considering difference (**r-^{IR}s**), any tuple f∈r that has no value-equivalent tuple in s that intersects its frame time (i.e., such that VE_INT(f,s) = Ø) is reported in output, unchanged. Otherwise, all the tuples in VE_INT(f,s) must be considered. The output tuple is value equivalent to f. Its minimum cardinality is 0, since the valid times of all repetitions in f can be "covered" by the valid times of tuples in VE_INT(f,s). Its maximum cardinaliy is the sum of all the maximum cardinalities (i.e., the maximum cardinality of f plus the maximum cardinalities of all the tuples in VE_INT(f,s)). This is due to the fact that the difference between each pair of time intervals may generate a maximum of two time intervals. Notably, the frame time of the result is the one of f.[1] As an example, Fig. 2 shows the difference between the red

1. This is due to the fact that the frame time represents just a span of time in which the valid time of each repetition is contained. Thus, the valid times in the result may be placed within the frame time of the subtrahend. For instance, the first four valid times of the result (yellow intervals with segment endpoints) in Fig. 2 are contained also in FT2. The figure shows clearly that the output frame time must be the frame time of the minuend, and may include (part of) the frame time of the subtrahend.

TABLE 2
Relation REP (Reports)

Employee	N	M	FT
John	8	8	{[Jan5, Jan15],[Jan 20,Feb10]}
Ann	10	12	{[Jan1, Jan30]}
Sue	1	1	{[Jan1, Jan31]}

intervals with round endpoints (contained in the frame time FT1) and the blue intervals with diamond endpoints (contained in FT2). The resulting frame time is FT (FT = FT_1), and the resulting repetitions are shown in yellow (segment endpoint).

Our Cartesian Product operates the intersection between frame times. The minimum cardinality is 0, since it may always be the case that the valid times of the repetitions (contained in the intersection of frame times) do not intersect with each other (see, e.g., Fig. 1a). The maximum cardinality is the sum of the two input maximum cardinalities, minus 1 (see, e.g., Fig. 1c).

As a first example of algebraic query, (Q1) in Section 2 can be asked as follows:

(Q1) $\pi^{IR}_{symptom}$ (($\sigma^{IR}_{patient\ =\ John\ \wedge\ symptom\ =\ headache}$ (SYMT)) $\times^{IR}$ ($\sigma^{IR}_{patient\ =\ Mary\ \wedge\ symptom\ =\ headache}$ (SYMT)))

As an additional example, let us consider a database containing the relation MEET in Table 1, and the relation REP in Table 2, representing the episodes when employees have written reports. Queries Q2 and Q3 ask about reports written during meetings, and not written during meetings respectively.

(Q2) $\pi^{IR}_{Employee}$ ($\sigma^{IR}_{MEET.Employee\ =\ REP.Employee}$(MEET$\times^{IR}$REP)).

(Q3) REP $-^{IR}$ ($\Pi^{IR}_{Employee}$(MEET)).

4.2 Reducibility Properties of the Algebra

Reducibility is fundamental for TDB approaches, to grant that the semantics of new operators, which extend simpler operators to cope with new phenomena, reduces to that of simpler operators when the new phenomena are disregarded [2], [9]. In general (see, e.g., [9]), given any two relational models and algebrae X and Y, the reducibility of X to Y is proved by introducing a reduction operator R and by proving that, indicating by Op^X and Op^Y two corresponding operators in X and Y, and r a relation in X, the following holds (the analogous holds for binary operators): $R(Op^X(r)) = Op^Y(R(r))$.

The specific new phenomenon dealt with by our approach is the treatment of irregular repetitions. Thus, natural candidates for reducibility include, first of all, approaches not considering repetitions. Specifically, since we cope with repetitions for which the exact valid time is unknown, we chose to consider a TDB approach coping with temporal indeterminacy, namely the "ITE" approach [10] (see also Section 3.1). To prove the reducibility of our algebra to the ITE one, we have to define a reduction operator R^{ITE}. In the following, we use three auxiliary functions: (1) *Gran(I)*, that takes in input a (convex) time interval I, and returns the granules it contains (e.g., Gran([1,3]) = {1,2,3}), (2) *Gran*({I_1, ..., I_n})* that iterates *Gran* on a set of time intervals (e.g., Gran*({[3,5],[7,8]}) = {3,4,5,7,8}), and (3) *Max_Cover(g_1, ..., g_k)* that takes in input a set of granules, and returns the minimum set of disjoint time intervals exactly covering them (e.g., Max_Cover({3,4,5,7,8}) = {[3,5],[7,8]}).

The reduction operator R^{ITE} can be defined as follows.

Definition 3 (R^{ITE}). *Let r a temporal relation in our approach, defined on the schema $R = (A_1, \ldots, A_n \mid N,M,FT)$, and let $R' = (A_1, \ldots, A_n \mid T)$ the corresponding schema in the ITE model, where $T \in 2^{D_{VT}}$ (i.e., T is a set of granules in D_{VT}). $R^{ITE}(r) = \{z \setminus \exists x \in r\ z[A] = x[A] \wedge x[M] > 0 \wedge z[T] = Gran^*(x[FT])\}$*

Given the above definition, Property 3 holds.

Property 3 (Reducibility to ITE). *Our algebra is reducible to ITE's algebra through R^{ITE}, i.e., $R^{ITE}(Op^{IR}(r)) = Op^{ITE}(R^{ITE}(r))$, where Op^{IR} and Op^{ITE} represent corresponding relational operators in our algebra and in the ITE algebra respectively.*

The proof of this and of the following property are reported in the digital library as supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2015.2509976.

As in classical proofs of reducibility to Codd operators [9], we use as reduction operator a *time-slicing* operator (that we call R_t^{CODD}), selecting all and only those tuples x holding at a specific time t (i.e., such that $x[FT] \cap t \neq \emptyset$), and *removing the temporal dimension* from them.

Defintion 4 (R_t^{CODD}). $R_t^{CODD}(r) = \{z \setminus \exists x \in r\ z[A] = x[A] \wedge x[M] > 0 \wedge x[FT] \cap t \neq \emptyset\}$.

Reducibility through R_t^{CODD} does not hold for difference.

Property 4 (Reducibility to Codd). *Our union, projection non-temporal selection and Cartesian Product (but not difference) relational operators are reducible to Codd's corresponding operators through R_t^{CODD}.*

Property 5 (Additional Overhead). *It is worth comparing our data model and algebra with the ones of other TDB approaches to see the additional overhead that we add with respect to the TDB models that do not consider repetitions. Concerning the data model, two additional attributes are required, to cope with minimum and maximum cardinality. On the other hand,* temporal elements *are used by many TDB approaches, including TSQL2, to model valid time. Concerning algebraic operators, as discussed at the beginning of Section 4, we follow the TSQL2 "consensus" approach. Thus, our definitions of temporal operators are similar to TSQL2's ones, except for the fact that our operators also work on minimum and maximum cardinality, performing simple operations (sum). Thus, the I/O operations are the same with respect to TSQL2 (and TDB approaches based on BCDM), and only a limited constant overhead to CPU time is added.*

4.3 Algebraic Operators on Time and Cardinality

New operators can be introduced to cope with the temporal and the cardinality components of our data model. For instance, we show the *cardinality selection operator* $\sigma^{IR}_{\varphi}(r)$, which selects all those tuples whose minimum and maximum cardinality satisfy a selection predicate φ

$\sigma^{IR}_{\varphi}(r) = \{z \setminus \exists x \in r\ z[A] = x[A] \wedge z[N] = x[N] \wedge z[M] = x[M] \wedge z[FT] = x[FT] \wedge \varphi(x[N],x[M])\}$.

For instance, with φ:(x[N] = 3 $\wedge$ x[M] = 3) one can select all the facts which occurred exactly three times.

4.4 Interplay with Other Temporal Algebrae

Besides relations storing irregular repetitions ("IR" relations), a database can consist also of other types of (temporal) relations. Thus, it is important to study whether a (algebraic) query can operate on such different types of relations. Properties 1 and 3 above are very important to this respect. Given Property 1, an ITE relation can be extended to become an IR one, so that IR operators can be applied on it (and on primitive IR relations). Given Property 3, the R^{ITE} operator can be applied to reduce an IR relation to an ITE one, so that ITE operators can be applied. Thus, with the simple addition of R^{ITE} and of an operator extending ITE relations into IR ones, ITE and IR relations can be combined in the queries. Another interesting result regards the joint treatment of IR relations and relations dealing with *periodic* events [7]. Several different approaches have been developed to deal with periodic events (see, e.g., [12], [13], [14], [15], [16], [17] and the surveys in [7], [18]). However,

independently of the representation model, if periodic data are temporally bounded (i.e., when there are no infinite repetitions), they can be converted into an explicit set of temporal non-periodic data, by making the *extensions* of the periodicity explicit (see, e.g., [17]). After such a transformation, periodic data can be easily represented within the ITE data model, whose interplay with the IR model has been discussed above. In such a way, a comprehensive temporal database consisting of ITE, IR and periodic relations can be managed in an integrated way by temporal algebraic operators.

5 Related Works

In TDBs, several approaches have focused their attention on periodic events [7], coping in an *intensional* way (i.e., without making all occurrences explicit) with periodicity. Roughly speaking, such approaches can be divided into three mainstreams (terminology derived from [12], [13]): (i) *Deductive rule-based* approaches, using deductive rules. For instance, Chomicki and Imielinsky [14] dealt with periodicity via the introduction of the successor function in Datalog; (ii) *Constraint-based* approaches, using mathematical formulae and constraints (e.g., [15]); *Symbolic* approaches (e.g., [16], [17]), providing symbolic languages to cope with temporal periodicity in a compositional way. (See also the surveys in [7], [18]).

On the other hand, in this paper we cope with *irregular repetitions* ("intermittent periodic events" in [7]) whose valid time is *indeterminate*, since it is only approximated by a *frame time* containing all its occurrences. Despite its practical relevance, such a phenomenon has not been faced in the TDB area yet. Our work also involves the treatment of *temporal indeterminacy*, which is intrinsically involved in the notion of "frame time". A survey of TDB approaches to temporal indeterminacy has recently been provided in [8]. In one of the earliest TDB work on temporal indeterminacy, an indeterminate instant was modeled with a set of possible chronons [19]. Dyreson and Snodgrass [20] and Dekhtyar et al. [21] have proposed probabilistic approaches. Recently, Anselma et al. [10] have introduced a family of algebraic approaches coping with different forms of temporal indeterminacy. Our model is a consistent extension of the ITE approach in [10] (see Property 1), and, if we disregard repetitions, our algebra can be reduced to ITE's one (Property 3).

6 Discussion and Conclusions

Despite the importance of the phenomenon, our approach is the first TDB approach coping with *irregular indeterminate repetitions* of facts *in a frame time*. We have introduced a new data model, new relational algebraic operators, and we have studied their *reducibility* properties. In Section 4.4, we have also started to analyse the integration of our approach with the ITE one and with current approaches coping with periodic events. In our future work, we aim at further exploring such a promising research direction, by (i) devising a temporal relational approach (data model and algebra) coping with *nearly-periodic* events [7] (supporting also the cardinality of repetitions), and (ii) proposing the first comprehensive approach coping in an integrated way with the different types of repeated data (*periodic*, *nearly periodic*, and *intermittent* [7]), as well as with temporally indeterminate and "standard" temporal data. We are developing a prototypical implementation of our approach. Experimental evaluations will follow, to experimentally show the overhead we add with respect to TDBs approaches not managing repetitions.

References

[1] R. T. Snodgrass, *Developing Time-Oriented Database Applications in SQL*. San Mateo, CA, USA: Morgan Kaufmann, 1999.

[2] R. T. Snodgrass Ed., "The TSQL2 temporal query language," Norwell, MA: Kluwer, 1995.

[3] Y. Wu, S. Jajodia, and X. S. Wang, "Temporal database bibliography update," in *Temporal Databases, Research and Practice*, in O. Etzion, S. Jajodia, S. Sripada Eds. New York, NY, USA: Springer Science & Business Media, 1998, pp. 338–366.

[4] L. Liu and M. T. Özsu, *Encyclopedia of Database Systems*. New York, NY, USA: Springer, 2009.

[5] B. Salzberg and V. J. Tsotras, "Comparison of access methods for time-evolving data," *ACM Comput. Surv.*, vol. 31, no. 2, pp. 158–221, 1999.

[6] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt, "Survey of temporal information retrieval and related applications," *ACM Comput. Surv*, vol. 47, no. 2, pp. 1–41, 2014.

[7] A. Tuzhilin and J. Clifford, "On periodicity in temporal databases, information systems," *J. Inf. Syst.*, vol. 20, no. 8, pp. 619–639, 1995.

[8] C. Dyreson. Temporal Indeterminacy, in L. Liu and M. Ozsu, Encyclopedia of Database Systems. New York, NY, Springer, 2009.

[9] L. E. McKenzie and R. T. Snodgrass, "Evaluation of relational algebras incorporating the time dimension in databases," *ACM Comput. Surv.*, vol. 23, no. 4, pp. 501–543, 1991.

[10] 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.

[11] E. F. Codd, "Relational completeness of data base sublanguages," in *Database Systems*, R. Rustin Ed. Englewood Cliffs, N.J., USA: Prentice-Hall, 1972, pp. 65–98, (and IBM Res. Rep. RJ 987, San Jose, CA, USA).

[12] M. Baudinet, J. Chomicki, and P. Wolper, "Temporal deductive databases," in *Temporal Databases*, A. Tansel et al.et al., Eds. Redwood City, CA, USA: Benjamin Cummings, 1993, pp. 294–320.

[13] M. Niezette and J.-M. Stevenne, "An efficient symbolic representation of periodic time," in *Proc. 1st Int. Conf. Inf. Knowl. Manag.*, Baltimore, MD, USA, Nov. 1992, pp. 161–168.

[14] J. Chomicki and T. Imielinsky, "Temporal deductive databases and infinite objects," in *Proc. 7th ACM Symp. Principles Database Syst.*, Austin, TX, USA, Mar. 1988, pp. 61–73.

[15] F. Kabanza, J.-M. Stevenne, and P. Wolper, "Handling infinite temporal data," in *Proc. ACM SIGACT-SIGMOD-SIGART Symp. Principles Database Syst.*, Nashville, TN, USA, 1990, pp. 392–403.

[16] B. Leban, D. D. McDonald, and D. R. Forster, "A representation for collections of temporal intervals," in *Proc. 5th Nat. Conf. Artif. Intell.*, Philadelphia, PA, USA, 1986, pp. 367–371.

[17] P. Terenziani, "Symbolic user-defined periodicity in temporal relational databases," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 489–509, Mar./Apr. 2003.

[18] P. Terenziani, Temporal Periodicity, in L. Liu and M. Ozsu, Encyclopedia of Database Systems. New York, NY, Springer, 2009.

[19] R. T. Snodgrass, "Monitoring distributed systems: A relational approach," Ph.D. dissertation, CS Dept., Carnegie Mellon, Pittsburgh PA, usa, 1982.

[20] C. E. Dyreson and R. T. Snodgrass, "Supporting valid-time indeterminacy," *ACM Trans. Database Syst.*, vol. 23, no. 1, pp. 1–57, 1998.

[21] 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.