Modeling Temporal Primitives: Back to Basics

Iqbal A. Goralwalla, Yuri Leontiev, M. Tamer Özsu and Duane Szafron
Laboratory for Database Systems Research
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
{iqbal,yuri,ozsu,duane}@cs.ualberta.ca

Abstract

The fundamental question about a temporal model is "what is its underlying temporal structure?" More specifically, what are the temporal primitives supported in the model, what temporal domains are available over these primitives, and whether the primitives are determinate or indeterminate? In this paper a simple, general framework for supporting temporal primitives (instants, intervals, sets of intervals) is presented. The framework allows seamless integration of dense and discrete temporal domains of time over a linearly ordered, unbounded point structure. The framework also provides a set-theoretic basis that allows uniform treatment of determinate and indeterminate temporal primitives.

1 Introduction

The primary component of a temporal model is its underlying temporal structure. Supporting a temporal structure involves making choices between alternative temporal features. This includes the temporal primitives supported (points or intervals), the temporal domain available for these primitives (dense or discrete), the temporal determinacy of the primitives, the ordering imposed on the primitives (linear or branching), and whether time is bounded or unbounded. Indeed, the temporal structure, with its various constituents, forms the basic building block of the design space of any temporal model since it is comprised of the basic temporal features that underlie any temporal model. In this paper we concentrate on temporal primitives, the temporal domains available over these primitives, and the temporal determinacy of the primitives.

The definition of temporal primitives requires prior knowledge of the underlying temporal domain. A temporal domain can be *dense* or *discrete*. Between any two temporal primitives in a dense time domain, another temporal primitive exists. For any temporal primitive in a discrete time domain, there is a unique successor and predecessor. Similarly, in handling temporal indeterminacy, some researchers assume a dense temporal domain [12], while others assume a discrete temporal domain [8, 6]. Therefore, selecting the appropriate temporal domain is an integral part of defining a temporal model.

Most of the research in the context of temporal databases has assumed that the temporal domain is discrete. Several arguments in favor of using a discrete temporal domain are made by Snodgrass [16] including the imprecision of clocking instruments, compatibility with natural language references, possibility of modeling events which have duration, and practicality of implementing the temporal data model.

However, in an excellent survey by Chomicki on temporal query languages [3], it is argued that the dense temporal domain is very useful in mathematics and physics. Furthermore, dense time provides a useful abstraction if time is thought of as discrete but with instants that are very close. In this case, the set of time instants may be very large which in turn may be difficult to implement efficiently. Chomicki further argues that query evaluation in the context of constraint databases [11, 14] has been shown to be easier in dense domains than in discrete domains. Dense temporal domains have also been used to facilitate full abstract semantics in reasoning about concurrent programs [1].

In our opinion, both views have valid arguments. While the discrete domain of time helps in promoting the practical side of temporal research, the dense domain of time provides a useful underlying abstraction. Our contention is that a temporal model should be general enough to support both the dense and the discrete temporal domains. In this paper, we propose a simple general framework for defining temporal primitives, without making any assumptions about the underlying temporal domain. We do not advocate one domain over the other; rather our framework enables leveraging the advantages of both by allowing seamless integration of dense and discrete domains of time. This allows a temporal model to provide support not only for applications which usually need a discrete temporal domain, but also for applications that need dense time as an abstraction. This is in contrast to recent proposals that handle multiple granularities [4, 13, 5, 18, 2, 17, 15]. These proposals assume a single underlying temporal domain which is usually discrete.

The contributions of this paper can be summarized as follows: (1) We present a simple, general framework for supporting temporal primitives which allows seamless integration of dense and discrete domains of time over a linearly ordered, unbounded point structure. To the best of our knowledge, this feature is novel to our work. (2) We define calendric support over the point structure which is independent of any particular temporal domain. This allows physical time to be interpreted in a meaningful manner and allows us to define various calendric systems in our framework. (3) We provide a uniform and consistent mapping of calendric primitives to the point structure. (4) A set-theoretic framework that allows uniform treatment of determinate and indeterminate temporal primitives.

The rest of the paper is organized as follows: Section 2 gives the definition of our underlying point structure, and describes how intervals and temporal elements can be defined over it. In Section 3 the foundation for calendric support is set by defining granularities over the point structure. Section 4 defines calendars and their primitives, and describes how the calendric primitives are mapped to the point structure. Section 5 provides a treatment of indeterminate temporal information. Section 6 summarizes the work presented in this paper and discusses avenues for future research.

2 Preliminaries

In this section we define our underlying domain (called the *global timeline*) which will be used as the basis of all subsequent definitions. The set of desired properties of the *global timeline* postulated here is quite weak: for example, we do not postulate either density or discreteness of the global timeline. This will allow us to use almost all subsequent constructs independently of the density of the underlying domain. The only construct that depends on the density of the underlying domain is the *infinitely divisible granularity* that will allow us to bridge between a dense domain and its discrete approximation.

Definition 2.1 *Global timeline:* The *global timeline* (denoted $\mathcal{T}$) is a point structure with precedence relation $<_{\mathcal{T}}$ (abbreviated to $<$), which is a total (linear) order without endpoints (i.e., $<$ is irreflexive, transitive and linear relation unbounded from both left and right).

Example 2.1 Examples of point structures satisfying Definition 2.1 include integers $\mathbf{Z}$, rationals $\mathbf{Q}$, and reals $\mathbf{R}$. More exotic examples are $\mathbf{Z} \odot \mathbf{Q}$ (lexicographically ordered pairs of integer and rational numbers) and $\mathbf{R} \odot \mathbf{Z} \odot \mathbf{R}$. Note that both dense and discrete point structures are allowed. The examples in this paper use $\mathcal{T} = \mathbf{Z}$ (denoted $\mathcal{T}_{\mathbf{Z}}$) and $\mathcal{T} = \mathbf{Q}$ (denoted $\mathcal{T}_{\mathbf{Q}}$).

The global timeline with $-\infty$ (minimal element) and $+\infty$ (maximal element) added to it is called the *extended global timeline* and is denoted $\widehat{\mathcal{T}}$. The power set over $\mathcal{T}$ (denoted $\mathcal{P}(\mathcal{T})$) has the usual set-theoretic operations (union, complement, intersection, and difference) defined over it. In addition, it has two distinct partial orderings namely, strong precedence $<_{\mathcal{P}(\mathcal{T})}$ and subset inclusion $\subseteq_{\mathcal{P}(\mathcal{T})}$. We will now define *intervals* over $\mathcal{T}$.

Definition 2.2 *Open time interval:* An *open time interval* (a, b), where $a \in \mathcal{T}$, $b \in \mathcal{T}$, and $a < b$ is a subset of $\mathcal{T}$ such that $(x \in (a, b)) \overset{\text{def}}{\Longleftrightarrow} (x \in \mathcal{T} \wedge a < x < b)$

Left-closed, right-closed, and closed time intervals (denoted respectively $[a, b)$, $(a, b]$, and $[a, b]$) are defined analogously. Closed intervals allow $a = b$; thus, intervals of the form $[a, a]$ are allowed. All these kinds of intervals form $INT(\mathcal{T})$, the set of all intervals over $\mathcal{T}$. Note that $\mathcal{T}$ is isomorphic to a subset of $INT(\mathcal{T})$ formed by intervals of the form $[a, a]$ with precedence relation $<_{\mathcal{P}(\mathcal{T})}$. There is one additional partial ordering, *weak precedence* $\preceq$, defined for intervals. It is defined as follows: $(I_1 \preceq I_2) \overset{\text{def}}{\Longleftrightarrow} ((\forall a \in I_1\ \exists b \in I_2\colon\ \ a \leq b) \wedge (\nexists b \in I_2\ \forall a \in I_1\colon\ \ b < a))$. The strong precedence is a suborder of the weak one. For example, in $\mathcal{T}_{\mathbf{Z}}$, $(0, 3] \preceq (2, 4)$, while $(0, 3] \not< (2, 4)$. Since time intervals are sets, usual set-theoretic operations (union, intersection, complement, difference) can be defined for them. However, the set of all intervals is not closed with respect to the above operations. For example, in $\mathcal{T}_{\mathbf{Z}}$, $(0, 4] \setminus (2, 3) \notin \mathrm{INT}(\mathcal{T}_{\mathbf{Z}})$. The notion of *temporal element* (union of a finite number of time intervals) [7] is therefore used. We will denote the set of all temporal elements over $\mathcal{T}$ as $TE(\mathcal{T})$. $TE(\mathcal{T})$ is treated as a subset of $\mathcal{P}(\mathcal{T})$, with the same set-theoretic operations and ordering. $INT(\widehat{\mathcal{T}})$ and $TE(\widehat{\mathcal{T}})$ are defined the same way as $INT(\mathcal{T})$ and $TE(\mathcal{T})$.

Figure 1 shows the relationships between the global timeline and the temporal primitives defined over it. The figure shows that $\mathcal{T}$ is isomorphic to a set of intervals $[a, a]$, the lower and upper bounds of which are points that are members of $\mathcal{T}$. The set of intervals $[a, a]$ is a subset of the set of intervals over $\mathcal{T}$, which in turn is a subset of the set of temporal elements over $\mathcal{T}$. Finally, the set of temporal elements is a subset of the power set over $\mathcal{T}$.

$$\begin{tikzcd}
\mathcal{T} \arrow[r, leftrightarrow, "\text{isomorphic}"] & \{[a,a]\}_{a \in \mathcal{T}} \arrow[r, "\subseteq"] & \mathrm{INT}(\mathcal{T}) \arrow[r, "\subseteq"] & \mathrm{TE}(\mathcal{T}) \arrow[r, "\subseteq"] & P(\mathcal{T})
\end{tikzcd}$$

Figure 1: Relationships between temporal primitives defined over the global timeline

3 Granularities

The global timeline ($\mathcal{T}$) defined in Section 2 is "flat". In this section, we define granularities over $\mathcal{T}$ in order to overcome its "flatness". Generally speaking, granularities allow $\mathcal{T}$ to be perceived at several resolution levels. For example, defining the Gregorian calendar over $\mathcal{T}$ would allow $\mathcal{T}$ to be partitioned in several levels, e.g., years, months, days. Granularities are subsequently used to define calendars in Section 4. Granularities can also be perceived as generalized forward shifts along the timeline. For example, the granularity of second (G_{second}) moves any point on the timeline one second forward.

In the following, we use $f^z,\ z \in \mathbf{Z}$ to denote integer powers of functions. Positive powers are defined as $f^n = ff \dots f(n$ times), zero power (of any function) is defined as $id_{\mathcal{T}}$ (identity over $\mathcal{T}$), and negative powers are defined as $f^{-n} = (f^{-1})^n = (f^n)^{-1}$ where f^{-1} is an inverse function. Negative powers of a function only make sense if the inverse function exists.

Definition 3.1 *Granularity:* A (partial) function $G \in \mathcal{T} \to \mathcal{T}$ with domain $\mathrm{dom}(G)$ and codomain $\mathrm{codom}(G)$ is called *a granularity* if it satisfies the following:

1. Monotonicity (MON): $\forall t_1, t_2 \in \mathrm{dom}(G)$: $(t_1 < t_2) \Rightarrow (G(t_1) < G(t_2))$
2. Forward directedness (FORW): $\forall t \in \mathrm{dom}(G)$: $t < G(t)$
3. Origin (ORIG): $\exists o \in \mathrm{dom}(G)$: $o \in ORIG(G) o \in ORIG(G) \overset{\text{def}}{\iff} \forall z \in \mathbf{Z} : o \in \mathrm{dom}(G^z) \wedge \forall t \in \mathcal{T}$: $\exists z', z'' \in \mathbf{Z}$: $G^{z'}(o) \le t \le G^{z''}(o)$

An origin o is a point on the timeline such that $CHAIN(G, o) \overset{\text{def}}{=} \{t \mid \ t = G^z(o),\ z \in \mathbf{Z}\}$ can be viewed as a partition of the timeline.

Thus, a granularity is a generalized forward shift that defines at least one partition of the timeline. In other words, a granularity allows us to "step through" back and forth the entire timeline in a countable number of steps, starting from any of its origins. This will be used to define calendars in the Section 4.

A granularity can define several partitions. A single calendar only makes use of one of them; however, different calendars can use different partitions defined by the same granularity. Note that if G a granularity, then G^{-1} can be treated as a generalized *backward* shift that defines the same partitions of the timeline as G. Note also that for any granularity G and any $n \in \mathrm{N}, G^n$ is also a granularity. Theorem 3.1 ensures that for any given domain satisfying Definition 2.1, there always exists at least one granularity.

Theorem 3.1 If $\mathcal{T}$ is a point structure satisfying Definition 2.1, then $\mathcal{T}$ has a granularity.

The proof follows from Definitions 2.1 and 3.1, and Zorn's Lemma [10].

Example 3.1 These are examples of granularities:

$$f(x) = \begin{cases} x+2 & x \text{ is even,} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$g(x) = \begin{cases} x+3 & 3k < x \le 3k+1 (k \in \mathbf{Z}), \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$i(x) = x+2$$

A granularity G is called *a total granularity* iff G is a total function (i.e. $\mathrm{dom}(G) = \mathcal{T}$). Total granularities are very useful in establishing relationships between different calendars since every point on the timeline belongs to a chain defined by a total granularity. An example of a total granularity is $i(x)$ from the above example. We now define *divisible granularities* that provide successively finer partitions of the timeline.

Definition 3.2 *Divisible granularity:* A granularity G is *divisible by* $n \in \mathrm{N},\ n > 1$ iff there exists a granularity $G^{1/n}$ such that $\forall t \in \mathrm{dom}(G)$: $(G^{1/n})^n(t) = G(t)$.

The divisibility of a granularity G by n essentially means that for every partition defined by G, there is a partition that is n times finer. It is easy to see that if G is a granularity then G^n is divisible by n. The ability to divide a granularity by an arbitrary number (thus giving "infinitely finer" partitions) is an essential property when the underlying domain is dense. A granularity that is divisible by all $n \in \mathrm{N}$ is called an *infinitely divisible granularity*.

Example 3.2 The granularity f from Example 3.1 is not divisible by any $n \in \mathrm{N}, n > 1$. The granularity g from the same example is infinitely divisible. For example, we can choose

$$g^{1/n}(x) = \begin{cases} x + 1/n & 3k < x \le 3k+1-1/n \\ x+3 & 3k+1-1/n < x \le 3k \end{cases}$$

The granularity i from the same example is divisible by 2 over $\mathcal{T}_{\mathrm{Z}}$ and is infinitely divisible over $\mathcal{T}_{\mathrm{Q}}$. In $\mathcal{T}_{\mathrm{Z}}$ we can choose $i^{1/2}(x) = x+1$ while in $\mathcal{T}_{\mathrm{Q}}$ we can choose $i^{1/n}(x) = x + 2/n$ for all $n \in \mathrm{N}, n > 1$.

Theorem 3.2 If $\mathcal{T}$ is dense then it has at least one infinitely divisible granularity.

The proof follows from Theorem 3.1 and the definition of a dense domain. The following theorem states that any chain of any total infinitely divisible granularity can be used to approximate any point on the timeline with an arbitrary precision.

Theorem 3.3 If G is a total infinitely divisible granularity over $\mathcal{T}$ then $\forall o \in ORIG(G)\ \forall t_1, t_2 \in \mathcal{T}$: $(t_1 < t_2) \Rightarrow (\exists t \in CHAIN(G, o)\ \exists n \in \mathrm{N}\ \exists m \in \mathrm{N}\ m < n$: $t_1 \le G^{m/n}(t) < t_2)$

Infinitely divisible granularities are useful for the interpretation of dense domains. Discrete domains do not have infinitely divisible granularities. Every granularity G over $\mathcal{T}$ can be extended to $\widehat{\mathcal{T}}$ to yield an *extended granularity* $\widehat{G}$: (1) $\forall t \in \mathcal{T}$: $\widehat{G}(t) = G(t)$ (2) $\widehat{G}(-\infty) = -\infty$ (3) $\widehat{G}(+\infty) = +\infty$ In the rest of the paper we will use extended granularities only and will omit $\widehat{}$ for brevity. In the next section we use granularities to define calendars over the global timeline.

4 Calendars

In order to interpret the global timeline, we propose to use calendars. A calendar is a means by which physical time can be represented in a meaningful way. A calendar is comprised of a finite number of partitions called *origin chains*, and a distinguished point on the timeline called the *origin* of the calendar. Calendars are used to define calendric time primitives called *calendric instants* and *calendric intervals*. They provide an abstraction mechanism that allows us to deal with calendric time primitives independently of the underlying domain.

Definition 4.1 *Calendar:* A *calendar* is a tuple $\langle 0, \mathcal{G} \rangle$, where $0 \in \mathcal{T}$ is the *origin* of the calendar and $\mathcal{G}$ is a finite set of granularities: $\mathcal{G} = \{G_1, \ldots, G_N\}$, where the *coarsest* granularity is G_N while the *finest* one is G_1. The calendar origin and the set of its granularities has the following collective properties:

1. Ordering (ORD): $\forall i, j \in 1, \ldots, N$: $(i < j) \Rightarrow (\forall t \in \mathrm{dom}(G_i) \cap \mathrm{dom}(G_j) \cap \mathcal{T}$: $G_i(t) < G_j(t))$

2. Origin chain (ORIG): $\forall i \in 1, \ldots, N$: $0 \in ORIG(G_i)$
   The chain of G_i to which 0 belongs $(CHAIN(G_i, 0))$ is called *the origin chain (of G_i)* and is denoted $C^0_{G_i}$ or simply C^0_i.

3. Chain inclusion (CHAIN): $C^0_N \subset C^0_{N-1} \subset \cdots \subset C^0_1$

Informally, the calendar defines (by its origin chains) a ruler on the timeline with bigger markers made by origin chains of coarser granularities and a zero marker placed at the calendar origin. An actual ruler always has at least one smaller marker between two bigger ones; only the smallest markers have no markers between them. The ruler property holds for calendars as well, since if $t \in C^0_i$, $i \in 2, \ldots, N$ then: (1) $G_{i-1}(t) \in C^0_{i-1}$ (2) $G_{i-1}(t) \notin C^0_i$ (3) $\exists n \in \mathbf{N}$: $G^n_{i-1}(t) = G_i(t) \in C^0_i$

Example 4.1 Let us consider $\mathcal{T}_{\mathbf{Q}}$ and $\mathcal{T}_{\mathbf{Z}}$ and define a calendar that corresponds to a simplified version of the standard Gregorian calendar with years, months, days, hours, minutes, and seconds. In this example, we will treat $1 \in \mathcal{T}$ as a second. Then we can define the following granularities: $G_{\text{second}}(x) = x + 1, G_{\text{minute}}(x) = G^{60}_{\text{second}} = x + 60, G_{\text{hour}}(x) = G^{60}_{\text{minute}} = x + 60{\cdot}60, G_{\text{day}}(x) = G^{24}_{\text{hour}} = x + 60{\cdot}60{\cdot}24$. In order to define months and years, two additional functions are introduced:

$$y(k) = 60{\cdot}60{\cdot}24{\cdot}(365k + \lfloor\frac{k}{4}\rfloor - \lfloor\frac{k}{100}\rfloor + \lfloor\frac{k}{400}\rfloor)$$

$$m(k) = 60{\cdot}60{\cdot}24{\cdot}(30k + \lceil\frac{k}{12}\rceil + \lceil\frac{k-2}{12}\rceil + \lceil\frac{k-4}{12}\rceil + \lceil\frac{k-6}{12}\rceil + \lceil\frac{k-7}{12}\rceil + \lceil\frac{k-9}{12}\rceil + \lceil\frac{k-11}{12}\rceil - 2\lceil\frac{k-1}{12}\rceil + \lfloor\frac{k+10}{12{\cdot}4}\rfloor - \lfloor\frac{k+10}{12{\cdot}100}\rfloor + \lfloor\frac{k+10}{12{\cdot}400}\rfloor)$$

$y(k)$ is the number of seconds between the origin, January 1 year 0001, encoded here as 0 day of 0 month of 0 year, and the beginning of year k. $m(k)$ is number of seconds between the origin and the beginning of month k. We can now define granularities corresponding to months and years:

$$G_{\text{month}}(x) = m(k+1) \qquad x = m(k), k \in \mathbf{Z}$$
$$G_{\text{year}}(x) = y(k+1) \qquad x = y(k), k \in \mathbf{Z}$$

It is easy to see that for those x for which G_{year} is defined, $G_{\text{year}}(x) = G^{12}_{\text{month}}(x)$. We will take 0 as the origin of our calendar so that $\mathcal{C}_{\text{Greg}} = \langle 0, \{G_{\text{second}}, G_{\text{minute}}, G_{\text{hour}}, G_{\text{day}}, G_{\text{month}}, G_{\text{year}}\}\rangle$. Note that G_{second} is infinitely divisible in $\mathcal{T}_{\mathbf{Q}}$ and indivisible in $\mathcal{T}_{\mathbf{Z}}$. The granularities corresponding to seconds, minutes, hours, and days are total ones while all the others are not. Therefore, since G_{second} in $\mathcal{T}_{\mathbf{Q}}$ is total and infinitely divisible, the calendar can approximate every point in $\mathcal{T}_{\mathbf{Q}}$. Note that we can now define another calendar (e.g. fiscal) with different origin and different definition of months and years and then we could use seconds, minutes, hours, and days of the Gregorian calendar in order to establish a correspondence between the two calendars.

4.1 Calendric instants

A *calendric instant* is a calendric denotation of a point on the timeline. Calendric instants are almost domain-independent (their only dependency on domains lies in the coefficient of the finest granularity); they are introduced to provide a calendar-based abstraction of the timeline.

Definition 4.2 *Calendric instant:* A *calendric instant* $I_{\mathcal{C}}$ in a calendar $\mathcal{C}$ is a tuple $\langle z_N, \ldots, z_1 \rangle$ where $z_1 = x/m$, $x, z_2, \ldots, z_N \in \mathbf{Z}$, and $m = 1$, or $m \in \mathbf{N}$ and G_1 is divisible by m.

We will denote the set of all calendric instants of $\mathcal{C}$ as INST($\mathcal{C}$). Note that if G_1 is infinitely divisible, then $z_1 \in \mathbf{Q}$. Every calendric instant I of a calendar $\mathcal{C}$ maps to a single element t_I of $\mathcal{T}$. The function $d_{\mathcal{C}}$ that realizes this mapping is defined as follows: $d_{\mathcal{C}}(I) = G_1^{z_1}(G_2^{z_2}(\cdots G_N^{z_N}(0)\cdots))$, where 0 is the origin of $\mathcal{C}$ and $G_1, \ldots, G_N$ are its granularities. Informally this means that a point on the timeline represented by a calendric instant $\langle z_N, \ldots, z_1 \rangle$, is computed by taking the calendar origin and shifting it forward z_N times by G_N, then shifting the result z_{N-1} times by G_{N-1}, etc until G_1. The last shift can be fractional as z_1 can be non-integer if G_1 is a divisible granularity.

Example 4.2 The calendric instant $I_1 = \langle$ 1995, 1, 15, 13, 16, 3 $\rangle$ in the calendar $\mathcal{C}_{\text{Greg}}$ corresponds to 13h 16min 3sec on February 16, 1996 a.d. (we routinely start counting days, months, and years from 1, and hours, minutes, and seconds from 0, while in our framework all of them are always counted starting from 0). I_1 is mapped to: $G^3_{\text{second}}(G^{16}_{\text{minute}}(G^{13}_{\text{hour}}(G^{15}_{\text{day}}(G_{\text{month}}(G^{1995}_{\text{year}}(0))))))$.
$I_2 = \langle 1994, 13, 15, 13, 16, 3\rangle$ is another calendric instant which is mapped to the same point as I_1.
$I_3 = \langle 1995, 1, 14, 13, 16, 3.098\rangle$ is an example of a calendric instant in $\mathcal{C}_{\text{Greg}}$ over $\mathcal{T}_{\mathbf{Q}}$ which is *not* a calendric instant of the same calendar in $\mathcal{T}_{\mathbf{Z}}$ (as G_{second} is indivisible over $\mathcal{T}_{\mathbf{Z}}$).

This mapping together with the ordering on $\mathcal{T}$ induces an ordering on equivalence classes of calendric instants (two calendric instants are equivalent iff they map to the same point of $\mathcal{T}$). This ordering is total and we will use $<$ to denote it. We will also use $=$ to denote the equivalence of two calendric time instants and use $\equiv$ for tuple equality. Thus, INST($\mathcal{C}$)$/=$ is isomorphic to a subset of $\mathcal{T}$.

It can be seen from Example 4.2 that different calendric instants from the same calendar can map to the same point on the timeline. In order to overcome this drawback, calendric instants can be normalized. For example in the Gregorian calendar, the month is between 0 and 11, the day is in the appropriate range for the given month, the hour is between 0 and 23, and the minutes and seconds are between 0 and 59. Distinct normalized calendric instants from the same calendar map to different points of $\mathcal{T}$. Without loss of generality, in the rest of the paper we will deal with normalized calendric instants. For normalized calendric instants, we can define a partial function, $d_{\mathcal{C}}^{-1}$, that takes a point in $\mathcal{T}$ and transforms it to a normalized calendric instant in $\mathcal{C}$. We will now use INST($\mathcal{C}$) to denote all *normalized* instants of a calendar $\mathcal{C}$. INST($\mathcal{C}$) is isomorphic to a subset of $\mathcal{T}$.

Dates are human-readable representations of normalized calendric instants. For example, in the Gregorian calendar defined in Example 4.2, a normalized instant $\langle z_{\text{years}}, z_{\text{months}}, z_{\text{days}}, z_{\text{hours}}, z_{\text{minutes}}, z_{\text{seconds}}\rangle$ would correspond to the date $YMDHMS$, where Y is $z_{\text{years}} + 1$ a.d. if $z_{\text{years}} \geq 0$ and z_{years} b.c. if $z_{\text{years}} < 0$, M is January if $z_{\text{months}} = 0$, February if $z_{\text{months}} = 1, \ldots,$ December if $z_{\text{months}} = 11$, D is $z_{\text{days}} - 1$, H is z_{hours}h, M is z_{minutes}min, and S is z_{seconds}sec. Thus, $I_1 = \langle 1995, 1, 15, 13, 16, 3\rangle$ corresponds to the date 1996 a.d. February 16 13h 16min 3sec, while $I_3 = \langle 1995, 1, 14, 13, 16, 3.098\rangle$ corresponds to the date 1996 a.d. February 15, 13h 16min 3.098sec. We will routinely omit hours, minutes and seconds when they are zeroes. In order to make the examples more illustrative, we will use dates to represent calendric instants in the rest of the paper.

4.2 Discrete calendric instants

The formalism described in this paper allows us to use discrete instants without knowing the actual structure of $\mathcal{T}$ and seamlessly integrate them if $\mathcal{T}$ is not discrete. *Discrete calendric instants* are discrete abstractions of the underlying (possibly dense) domain. They are completely domain-independent. Discrete calendric instants can have different granularities. They are mapped to intervals over the underlying domain whose length is determined by the granularity of the instant. This necessitates the definition of *calendric addition*.

Definition 4.3 *Calendric addition:* A calendric addition $+_{\mathcal{C}}$ is defined as: $(I_{\mathcal{C}} \equiv \langle z_N, \ldots, z_{i+1}, z_i, 0, \ldots, 0\rangle) \wedge (1 \leq j \leq N_{\mathcal{C}}) \Rightarrow \quad (I_{\mathcal{C}} +_{\mathcal{C}} G_j \overset{\text{def}}{=} d_{\mathcal{C}}^{-1}(G_j(d_{\mathcal{C}}(I_{\mathcal{C}}))))$

Example 4.3 1996 a.d. February 16 + 1 day $\equiv$ $\langle 1995, 1, 15, 0, 0, 0\rangle + G_{\text{day}} = \langle 1995, 1, 16, 0, 0, 0\rangle \equiv$ 1996 a.d. February 17.

In this section we will deal with a single calendar and will therefore routinely omit the subscript $_{\mathcal{C}}$.

Definition 4.4 *Discrete calendric instant:* A *discrete calendric instant* I_d is a tuple $\langle I \equiv \langle z_N, \ldots, z_1\rangle, i\rangle$ such that (1) $1 \leq i \leq N$ (2) $\forall j\ 1 \leq j < i$: $z_j = 0$

We map the discrete calendric instants to *intervals* over $\mathcal{T}$. The mapping function d_d is defined as follows:

$$d_d(I_d) = [d(I), d(I + G_i))$$

The inverse of this mapping, d_d^{-1}, is unambiguous when it exists. The mapping, d_d, together with precedence and inclusion relations on INT($\mathcal{T}$) induce these relationships on the set of all discrete instants DINST and its subsets for each calendar $\mathcal{C}$ (DINST($\mathcal{C}$)). DINST($\mathcal{C}$) is isomorphic to a subset of INT($\mathcal{T}$). For example, a discrete time instant of the granularity "day" can be viewed as a point in the global timeline that is discrete at the level of days. The well-known problem of dealing uniformly with instants of different granularities is solved by the uniform mapping of discrete calendric instants to intervals over the underlying domain.

Since INST($\mathcal{C}$) is isomorphic to a subset of $\mathcal{T}$, which is in turn isomorphic to a subset of INT($\mathcal{T}$), we can form a union UINST($\mathcal{C}$) = INST($\mathcal{C}$) $\cup$ DINST($\mathcal{C}$). This union is disjoint and it can be mapped to disjoint subsets of INT($\mathcal{T}$). From now on, we will use this mapping to map UINST($\mathcal{C}$) to INT($\mathcal{T}$) and vice versa.

Informally, UINST($\mathcal{C}$) is an abstraction of the underlying point structure determined by the calendar. Every element of UINST($\mathcal{C}$) can be viewed as a point, thus providing a seamless integration of calendric instants (that may be dense) and discrete calendric instants that are discrete at various granularities.

We will use dates indexed by the name of the appropriate granularity to represent discrete calendric instants (unindexed dates denote calendric instants).

Example 4.4
1996 February $16_{\text{day}} \equiv \langle\langle 1995, 1, 15, 0, 0, 0\rangle, \text{day}\rangle \mapsto [d(\langle 1995, 1, 15, 0, 0, 0\rangle), d(\langle 1995, 1, 16, 0, 0, 0\rangle)),$
while 1996 February 16 $\equiv \langle 1995, 1, 15, 0, 0, 0\rangle \mapsto [d(\langle 1995, 1, 15, 0, 0, 0\rangle), d(\langle 1995, 1, 15, 0, 0, 0\rangle)]$.
Thus, 1996 February $16_{\text{day}} \preceq$ 1996 February 16. It is also easy to show that 1996 February $16_{\text{day}} <$
1996 February $17_{\text{day}} <$ 1996 February 18 and
1996 February $16_{\text{day}} <$ 1996 $\text{March}_{\text{month}}$, while
1996 February 16_{day} and 1996 $\text{February}_{\text{month}}$ are incomparable.

Discrete calendric instants provide a discrete abstraction of the underlying domain of any structure. We have demonstrated that we can easily mix calendric instants and discrete calendric instants of different granularities, thus providing a uniform, sound, and intuitive way of comparing them to each other.

4.3 Calendric intervals

The uniformity of our framework can be strengthened by defining *calendric intervals*, which provide an abstraction of intervals defined over UINST($\mathcal{C}$). In order to define calendric intervals, we first define structures called *double intervals* over $\mathcal{T}$. Double intervals are intervals that have intervals rather than instants as their bounds (for example, if one or both of the bounds are discrete instants). We need double intervals in order to define calendric intervals over UINST. Element of UINST are mapped to intervals over the global timeline, and therefore, construction of intervals over UINST requires the ability to deal with "intervals" with interval bounds.

Definition 4.5 *Double interval:* A *double interval* is a structure formed analogously to an interval. While a standard (single) interval has instants as upper and lower bounds, a double interval has interval bounds. Double intervals are defined (and mapped to single ones) according to the following rules:

1. $(a \in (J_1, J_2))$ where $(J_1 < J_2) \wedge (\exists i\colon\ J_1 < [i,i] < J_2) \overset{\text{def}}{\iff} (\forall j_1 \in J_1\ \forall j_2 \in J_2\colon\ j_1 < a < j_2)$
2. $(a \in [J_1, J_2))$ where $(J_1 < J_2) \overset{\text{def}}{\iff} (\exists j_1 \in J_1\ \forall j_2 \in J_2\colon\ j_1 \leq a < j_2)$
3. $(a \in (J_1, J_2])$ where $(J_1 < J_2) \overset{\text{def}}{\iff} (\forall j_1 \in J_1\ \exists j_2 \in J_2\colon\ j_1 < a \leq j_2)$
4. $(a \in [J_1, J_2])$ where $(J_1 \preceq J_2) \overset{\text{def}}{\iff} (\exists j_1 \in J_1\ \exists j_2 \in J_2\colon\ j_1 \leq a \leq j_2)$

Just like an interval $[a, b)$ includes a and everything between a and b while excluding b, a double interval $[J_1, J_2)$ includes the intervals J_1 and everything between the intervals J_1 and J_2 while excluding the interval J_2 itself. For example, $[[0,1),(2,3])$ should include $[0,1)$ and everything between $[0,1)$ and $(2,3]$ (i.e. $[1,2]$). Thus, $[[0,1),(2,3]) \equiv [0,1)\cup[1,2] = [0,2]$. It is easy to see that the result ($[0,2]$) automatically excludes the upper bound ($(2,3]$). Double intervals of other shapes are interpreted in a similar manner. The result always includes "the middle" and includes or excludes the "interval bound" according to the outer level brackets.

Example 4.5 $[[0,1),[2,3)) \equiv [0,1)\cup[1,2) = [0,2)$
$[[0,1),[2,3)] \equiv [0,1)\cup[1,2)\cup[2,3) = [0,3)$
$([0,1),[2,3)) \equiv [1,2)$.

Having defined double intervals, we now define calendric intervals.

Definition 4.6 *Calendric interval:* A *calendric interval* $\mathcal{I}$ is a structure formed analogously to single intervals with upper and lower bounds taken from UINST. They are mapped to INT($\mathcal{T}$) in the following way: first, the lower and upper bounds of $\mathcal{I}$ are mapped to INT($\mathcal{T}$), then $\mathcal{I}$ is converted to a double interval which in turn is mapped to INT($\mathcal{T}$).

Example 4.6 [1996 February 16_{day}, 1996 March 1_{day}) is first mapped to $[[d(\langle 1995, 1, 15, 0, 0, 0\rangle), d(\langle 1995, 1, 16, 0, 0, 0\rangle)), [d(\langle 1995, 2, 0, 0, 0, 0\rangle), d(\langle 1995, 2, 1, 0, 0, 0\rangle)))$, which is in turn mapped to $d(\langle 1995, 1, 15, 0, 0, 0\rangle), d(\langle 1995, 2, 0, 0, 0, 0\rangle))$.
(1996 February 15_{day}, 1996 March 1_{day}) is also mapped to $[d(\langle 1995, 1, 15, 0, 0, 0\rangle), d(\langle 1995, 2, 0, 0, 0, 0\rangle))$ which is what we intuitively expect since the first interval *includes* February 16, while the second *excludes* February 15.

Calendric intervals form the set CINT. The membership relation $\in$ on UINST $\times$ CINT is defined as subset inclusion between images of its two arguments in INT. The precedence relationships are defined similarly. *Calendric temporal elements* (CTE) are defined as finite unions of calendric intervals. They are mapped to elements of TE($\mathcal{T}$) (TE($\widehat{\mathcal{T}}$)) by mapping every calendric interval to INT and then taking union of the resulting intervals. Figure 2 shows the relationships between the calendric primitives defined in this section, and their mappings to corresponding temporal primitives that were defined in Section 2 and shown in Figure 1.

5 Temporal Indeterminacy

In the real world there are many cases when we have complete knowledge of the time or the duration of a particular event. For example, the maximum time allowed for students to complete their *Introduction to Logic Programming* examination is known for certain. This is an example of determinate temporal information. However, there are cases when the knowledge of the time or the duration of a particular event is known only to a certain extent. For example, we do not know the exact moment when the Earth was formed though we may speculate on the time frame for this event. In this case, the temporal information is indeterminate.

To treat indeterminate temporal information, we introduce an isomorphic image of $\mathcal{T}$ (denoted $\mathcal{T}'$) with isomorphism $h\colon \mathcal{T} \to \mathcal{T}'$. Since $\mathcal{T}'$ is isomorphic to $\mathcal{T}$, all constructs defined earlier for $\mathcal{T}$ are automatically defined for $\mathcal{T}'$ as well.

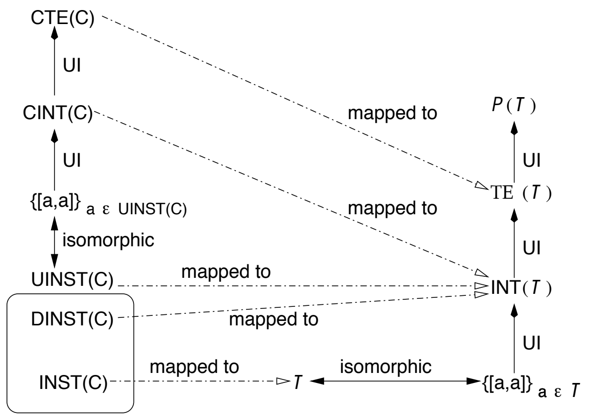

Figure 2: Relationships between calendric primitives and their mapping to primitives defined over the global timeline

We give temporal elements (and thus instants, intervals etc) the following meaning: if a temporal element comes from $\mathcal{T}$, it represents "certain" information; otherwise if it comes from $\mathcal{T}'$ it represents "possible" information. Thus if an event is known to occur during $[t_0, t_1) \cup h([t_3, t_4])$ that event is known to occur in $[t_0, t_1)$ time interval, known not to occur in $(-\infty, t_0) \cup [t_1, t_3) \cup (t_4, +\infty)$, and no information is available about this event during time interval $[t_3, t_4]$ (assuming $t_0 < t_1 < t_2 < t_3 < t_4$).

In order to be able to mix temporal elements from $\mathcal{T}'$ with those from $\mathcal{T}$, we introduce *the indeterminate time sets*.

Definition 5.1 *Indeterminate Time Set:* An *indeterminate time set* S is a member of $\mathcal{P}(\widehat{\mathcal{T}} \cup \widehat{\mathcal{T}}')$ such that $t \in \widehat{\mathcal{T}} \cap S \Rightarrow h(t) \notin S$

Example 5.1 The set $S_1 = [0, 1) \cup h([3, 4])$ is an indeterminate time set. At the same time, $S_2 = [0, 2) \cup h([1, 3))$ is not, since $1 \in \widehat{\mathcal{T}} \cap S_2 \wedge h(1) \in S_2$.

The indeterminate time sets form a family of sets $\mathcal{S}$. $\mathcal{S}$ defines set inclusion, complement, and union in a manner different from standard set-theoretic definitions for these operations.

1. $(S_1 \subseteq S_2) \overset{\text{def}}{\Longleftrightarrow} (x \in S_1 \Rightarrow (x \in \widehat{\mathcal{T}} \wedge x \in S_2) \vee (x \in \widehat{\mathcal{T}}' \wedge (x \in S_2 \vee h(x) \in S_2)))$

2. $\neg S \overset{\text{def}}{=} \{x \mid (x \in \widehat{\mathcal{T}} \wedge x \notin S \wedge h(x) \notin S) \vee (x \in \widehat{\mathcal{T}}' \wedge x \in S)\}$

3. $S_1 \cup S_2 \overset{\text{def}}{=} \{x \mid (x \in \widehat{\mathcal{T}} \wedge (x \in S_1 \vee x \in S_2)) \vee (x \in \widehat{\mathcal{T}}' \wedge (x \in S_1 \vee x \in S_2) \wedge h(x) \notin S_1 \wedge h(x) \notin S_2)\}$

Intersection and difference are defined in terms of the above operations in the usual way. The intuition behind these definitions is as follows: let us assume that we have "black" (for sure), "grey" (maybe) and "white" (certainly not) intervals. Then every indeterminate time set S partitions the whole timeline into black (defined as $S_b = \{t \mid \; t \in \widehat{\mathcal{T}} \cap S\}$), grey (defined as $S_g = \{h(t) \mid \; t \in \widehat{\mathcal{T}}' \cap S\}$), and white ($S_w = \widehat{\mathcal{T}} \setminus (S_b \cup S_g)$) areas. Thus, S carries information about time when an event did happen, might have happened, and did not happen. Intersection of white with anything is white (if one of two events is known not to occur at a particular time, then the conjunction of these events could not occur at that time), and that of grey with anything but white is grey (if we are uncertain about one event, we are uncertain about its conjunction with any other event, unless the second event is known not to happen at that time), and an intersection of black with black is black (if both events are known to occur at a particular time, their conjunction is also known to occur at that time). Likewise, a union of black with anything is black, that of grey with anything but black is grey, and the union of white and white is white (in this case we consider disjunction of two events). Analogous considerations can be used to intuitively justify our definitions of subset relationship and negation.

The operations of $\mathcal{S}$ follow the same rules as standard set-theoretic operations. $\mathcal{S}$ is closed with respect to the operations just defined. Another nice property of these operations is that both $\mathcal{P}(\widehat{\mathcal{T}})$ and $\mathcal{P}(\widehat{\mathcal{T}}')$ are sub domains of $\mathcal{S}$; set-theoretic operations and subset inclusion of $\mathcal{S}$ restricted to $\mathcal{P}(\widehat{\mathcal{T}})$ ($\mathcal{P}(\widehat{\mathcal{T}}')$) give standard set-theoretic operations and subset inclusion for these domains. The maximal element of $\mathcal{S}$ with respect to its subset inclusion is $\widehat{\mathcal{T}}$. An *indeterminate temporal element* is defined as a member of $\mathcal{S}$ that can be represented as a union of a finite number of intervals from INT($\widehat{\mathcal{T}}$) ∪ INT($\widehat{\mathcal{T}}'$).

With the introduction of $\mathcal{S}$ we can map all temporal elements from both $\widehat{\mathcal{T}}$ and $\widehat{\mathcal{T}}'$ to $\mathcal{S}$ (the mapping is identity, since $\widehat{\mathcal{T}} \subset \mathcal{S}$ and $\widehat{\mathcal{T}}' \subset \mathcal{S}$). Using the constructs described above, we are now able to represent determinate and indeterminate calendric instants and intervals and map them to indeterminate time sets that form a uniform, consistent, and sound set-theoretical basis of our framework. This is in contrast to other works on temporal indeterminacy [8, 12, 6]. In [8, 6] the underlying temporal domain is assumed to be discrete, while the work of [12] is carried out in the context of a dense temporal domain.

6 Discussion

The formalism developed in this paper gives us a framework in which granularities, calendars, calendric instants, and discrete calendric instants and intervals can be defined. They can then be mapped to a global timeline in a uniform and consistent manner, regardless of whether the underlying temporal domain is dense or discrete. If the temporal domain is dense, calendric time primitives and operations on them form a temporal structure that has both dense and discrete components of different granularities that can be mixed and operated upon uniformly and consistently. Our framework also abstracts from the underlying domain to a large extent so that when the underlying domain is enhanced (for example, is made dense) existing calendric temporal primitives do not have to be changed and operations on them still yield the same results. Lastly, our treatment of temporal indeterminacy allow us to uniformly represent determinate and indeterminate temporal primitives without making any assumptions on the underlying temporal domain.

It is important to note that our framework not only provides theoretical treatment of dense domains of time, but also provides finite approximations of dense domains which is actually what a standard computer provides. For example, if a computer provides a precision of 5 decimal places, then the finest granularity (G_1) to which a calendric instant is defined (see Definition 4.2) can be made divisible by 10^5. If another computer with a precision of 7 decimal places is now used, the old data still works since G_1 is now divisible by both 10^5 and 10^7. This gives us scalability for free, unlike the *chronon* assumption used in [15] which would involve updating every instant.

In another work [9], we investigate how a calendar provides relationships between granularities, and give procedures for converting temporal primitives from one granularity to another. We are also investigating the incorporation of the formalism developed in this paper into a temporal query language. Since the temporal primitives, set-theoretic operations and their semantics have been defined, we do not foresee this incorporation to be a major task.

References

[1] H. Barringer, R. Kuiper, and A. Pnueli. A Really Abstract Concurrent Model and its Temporal Logic. In *Proc. of the 13th ACM Symposium on Principles of Programming Languages*, pages 173–183, 1986.

[2] C. Bettini, X. Wang, E. Bertino, and S. Jajodia. Semantic Assumptions and Query Evaluation in Temporal Databases. In *Proc. ACM SIGMOD Int'l. Conf. on Management of Data*, pages 257–268, 1995.

[3] J. Chomicki. Temporal Query Languages: a Survey. In *Proceedings of the International Conference on Temporal Logic*, Bonn, Germany, July 1994.

[4] J. Clifford and A. Rao. A Simple, General Structure for Temporal Domains. In C. Rolland, F. Bodart, and M. Leonard, editors, *Temporal Aspects in Information Systems*, pages 17–30. North-Holland, 1988.

[5] E. Corsetti, A. Montanari, and E. Ratto. Dealing with Different Time Granularities in Formal Specifications of Real-Time Systems. *The Journal of Real-Time Systems*, 3(2):191–215, 1991.

[6] C.E. Dyreson and R.T. Snodgrass. Valid-time Indeterminacy. In *Proc. 9th Int'l. Conf. on Data Engineering*, pages 335–343, April 1993.

[7] S. Gadia. A Homogeneous Relational Model and Query Languages for Temporal Databases. *ACM Transactions on Database Systems*, 13(4), 1988.

[8] S.K. Gadia, S. Nair, and Y-C. Poon. Incomplete Information in Relational Temporal Databases. In *Proc. 18th Int'l Conf. on Very Large Data Bases*, pages 395–406, August 1992.

[9] I.A. Goralwalla, Y. Leontiev, M.T. Özsu, and D. Szafron. Modeling Time: Back to Basics. Technical Report TR-96-03, University of Alberta, February 1996.

[10] P.R. Halmos. *Naive Set Theory*. Springer Verlag, 1974.

[11] P.C. Kanellakis, G.M. Kuper, and P.Z. Revesz. Constraint Query Languages. In *Proc. of the 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 299–313, Nashville, Tennessee, April 1990.

[12] M. Koubarakis. Representation and Querying in Temporal Databases: the Power of Temporal Constraints. In *Proc. 9th Int'l. Conf. on Data Engineering*, pages 327–334, April 1993.

[13] A. Montanari, E. Maim, E. Ciapessoni, and E. Ratto. Dealing with Time Granularity in Event Calculus. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 702–712, June 1992.

[14] P.Z. Revesz. A Closed Form for Datalog Queries with Integer Order. In *International Conference on Database Theory*, pages 187–201, 1990.

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

[16] R.T. Snodgrass. Temporal Databases. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pages 22–64. Springer-Verlag, LNCS 639, 1992.

[17] X.S. Wang, C. Bettini, A. Brodsky, and S. Jajodia. Logical Design for Temporal Databases with Multiple Granularities. *ACM Transactions on Database Systems*, June 1997. In press.

[18] X.S. Wang, S. Jajodia, and V. Subrahmanian. Temporal Modules: An Approach Toward Temporal Databases. In *Proc. ACM SIGMOD Int'l. Conf. on Management of Data*, pages 227–236, 1993.